

FileMaker Master Book

中級編

FileMaker 19 プラットフォーム

Claris

中級編

© 2020 Claris International Inc. All rights reserved. Claris、FileMaker、ファイルメーカー、FileMaker Cloud、FileMaker Go およびファイルフォルダロゴは、Claris International Inc. の米国および／またはその他の国における登録商標です。クラリス、Claris Connect、Claris ロゴ、FileMaker Pro、FileMaker Server および FileMaker WebDirect は、Claris International Inc. の商標です。Mac および Mac ロゴは、米国およびその他の国における Apple Inc. の登録商標です。Amazon Web Services、アマゾン ウェブ サービス、および「Powered by AWS」のロゴは、米国および／または他の国における Amazon.com Inc. またはその関連会社の商標です。その他すべての第三者の商標は該当する所有者の財産です。

Apple、Apple ロゴ、macOS、iPad、iPhone、iPod touch は米国および他の国々で登録された Apple Inc. の商標です。iPhone の商標は、アイホン株式会社のライセンスにもとづき使用されています。App Store は Apple Inc. のサービスマークです。

Microsoft, Windows, Excel, Word は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。その他のすべての商標は該当する所有者の財産です。記載されている製品の仕様は予告なく変更する場合があります。

サンプルに登場する企業、ドメイン名、E メールアドレス、人物、場所、およびイベントなどのデータはすべて架空のもので、実在する企業、ドメイン名、E メールアドレス、人物、場所、およびイベントとは一切関係ありません。

この文書はあらゆる保証なく「現状通り」で提供されるものであり、CLARIS は、明示的または黙示的に関わらず、商用性または特定の目的における適合性の黙示的保証、あるいは非侵害性の保証を含み、これらに限定せず、あらゆる保証を提供しません。CLARIS または当社のサプライヤは、直接的、間接的、偶発的、または因果的な業務利益の損失、懲罰的または特定的な障害を含む障害について、事前に忠告されていた場合であっても、あらゆる場合にそれらに対する責任は問われないものとします。法域によってはこの責任の免除または制限が適用されない場合があります。常時、CLARIS はこの文書の内容を予告なしに変更する場合があります。この文書は最新版ではない場合があります。CLARIS はこの情報を更新する義務を負いません。

中級編執筆協力　株式会社スプラッシュ
www.splash.jp

目次

第 1 章　開発者になろう 1

1.1　はじめに 1
- 1.1.1　業務 2
- 1.1.2　カスタム App 2
- 1.1.3　システム 2
- 1.1.4　データ 3

1.2　開発しよう 4
- 1.2.1　考えておくこと 4
- 1.2.2　開発者 5

1.3　PDCA 6

1.4　準備 8
- 1.4.1　ソフトウェアの準備 8
- 1.4.2　演習ファイルの準備 8
- 1.4.3　高度なツール 9
- 1.4.4　開発環境 10
- 1.4.5　リソース（参考資料） 10

第 2 章　企画しよう 11

2.1　なぜ企画するのか 11

2.2　問題や課題の収集 12
- 2.2.1　整理する 12
- 2.2.2　対象と制約 12

2.3　概要の把握と決定 13
- 2.3.1　概要 13
- 2.3.2　決定 13
- 2.3.3　書き残す 14
- 2.3.4　期間 14
- 2.3.5　その他 14

第 3 章　要件を定義しよう 16

3.1　ヒアリング 16
- 3.1.1　なにを聞くか 16
- 3.1.2　誰に聞くか 16
- 3.1.3　業務の流れ 17
- 3.1.4　入出力の項目 18
- 3.1.5　できないこともある 18
- 3.1.6　期限を切る 19

3.2　要件を定義する 1 20
- 3.2.1　なぜ要件にするのか 20
- 3.2.2　規模感 20
- 3.2.3　項目 22

3.3 要件を定義する 2：作業を考える 23
3.3.1 業務の流れのどこか 24
3.4 要件を定義する 3：機能を考える 26
3.4.1 作業を細分化する 27
3.4.2 機能を定義する 28
3.4.3 定義内容の確認 28
3.5 要件を定義する 4：その他 29
3.5.1 セキュリティ 29
3.5.2 現在のデータはどうするか 30
3.5.3 他のシステムは関係あるか 31
3.5.4 環境 31
3.6 要件定義を合意する 32
3.6.1 合意 32
3.6.2 書き残す 32
3.6.3 期間 32
3.6.4 さらに知りたいなら 32

第 4 章　モデリングと正規化 37

4.1 構造設計 37
4.2 データモデリング 38
4.2.1 Entitiy Relationship モデル 38
4.2.2 エンティティを探す 38
4.2.3 エンティティの種類 40
4.2.4 関連の種類 41
4.2.5 1 対多 41
4.2.6 多対多 42
4.2.7 1 対 1 43
4.2.8 他のシステムとの関係 44
4.3 正規化 45
4.3.1 正規化とは 45
4.3.2 はじめの正規化 47
4.3.3 主キーと外部キー 48
4.3.4 2 番目の正規化 49
4.3.5 3 番目の正規化 51
4.3.6 多対多の実現 53
4.4 その他 55
4.4.1 依存 55
4.4.2 正規化しすぎない 55
4.4.3 ルールは絶対ではない 55
4.4.4 設計と変更 55

第 5 章　リレーションシップ（関連） 57

5.1 関連の定義 57
5.1.1 [データベースの管理] ダイアログ 58
5.1.2 リレーションシップグラフ 59
5.1.3 テーブルオカレンス（TO） 60
5.1.4 テーブルオカレンスグループ（TOG） 61

5.1.5 ［リレーションシップの編集］ダイアログ 62
5.1.6 関連定義のルール 64
5.1.7 ER 図とリレーションシップグラフの違い 65
5.2 設計からカスタム App へ 66
5.2.1 テーブルオカレンスグループ（TOG）のわけ方 66
5.2.2 アンカー / ブイ方式 67
5.2.3 TO の命名ルール 70
5.3 多対多とデータ 71
5.3.1 1 対多とレイアウト 74
5.3.2 多対多とレイアウト 76
5.3.3 設計とデータの表示 80
5.4 関連とデータ 81
5.4.1 照合条件 81
5.4.2 照合フィールド 81
5.4.3 参照元と参照先 84
5.4.4 関連によるレコード作成 85
5.4.5 関連の表示 106

第 6 章　データとフィールド 107

6.1 データ 107
6.2 フィールド定義 108
6.2.1 フィールドの名前 108
6.2.2 フィールドタイプ 109
6.2.3 ハウスキーピングフィールド 110
6.3 フィールドオプション 115
6.4 フィールドオプション：入力値の自動化 116
6.4.1 作成情報／修正情報 117
6.4.2 シリアル番号 118
6.4.3 ［直前に参照したレコード値］ 118
6.4.4 ［データ：］ 118
6.4.5 ［計算値］ 119
6.4.6 ［ルックアップ値］ 121
6.4.7 ［データ入力時の値変更の禁止］ 122
6.4.8 設定の保存 122
6.5 フィールドオプション：入力値の制限 130
6.5.1 ［このフィールドの入力値を制限する］ 131
6.5.2 ［データの入力時にユーザによる上書きを許可する］ 131
6.5.3 ［必要条件］ 131
6.5.4 ［制限値以外の入力時にカスタムメッセージを表示］ 132
6.5.5 設定の保存 133
6.6 その他の制限方法 146
6.7 発行日と期限日 147

第 7 章　データと索引・格納 152

7.1 「データの格納」タブについて 152
7.2 索引 153
7.2.1 索引内容 153

7.2.2　フィールドタイプと索引 154
7.2.3　索引が作れないもの 159
7.3　データの格納 160
7.3.1　グローバル格納 160
7.3.2　共有とグローバル格納 163
7.3.3　どこにつくるか 164
7.3.4　検索とグローバル格納 164
7.3.5　関連とグローバル格納 164
7.4　オブジェクトタイプフィールド 174
7.4.1　格納できるもの 174
7.4.2　オブジェクトの保存種類 176
7.4.3　バイナリデータの保存：埋め込み 177
7.4.4　バイナリデータの保存：外部格納 178
7.4.5　バイナリデータの保存：外部格納のパス 178
7.4.6　［オブジェクトの管理］ダイアログ 180
7.4.7　オブジェクトタイプフィールドのデータ書式 190
7.4.8　オブジェクトの情報 195
7.4.9　インポートとエクスポート 199
7.4.10　その他 200

第 8 章　画面を設計しよう 201

8.1　画面の設計 201
8.2　ユーザインターフェース 202
8.3　画面の役割 203
8.4　画面を考える 204
8.4.1　画面一覧をつくる 204
8.4.2　画面の大きさを想定する 207
8.4.3　画面の配置を考える 208
8.4.4　画面遷移 215
8.4.5　メニューやホーム画面 218
8.5　モバイルデバイスの画面 220
8.5.1　どのような作業をおこなうか 220
8.5.2　クライアントの種類で画面を分ける 221
8.6　画面に共通のこと 222
8.7　ナビゲーション 224

第 9 章　レイアウトツール 229

9.1　画面について 229
9.2　レイアウトモード 231
9.2.1　［レイアウトの管理］ダイアログ 232
9.3　レイアウトの作成 234
9.3.1　新規レイアウト作成のウイザード 234
9.4　フィールドの配置 236
9.4.1　「フィールド」タブ（左パネル） 236
9.4.2　ステータスツールバーのツール 237
9.5　レイアウトの設定 245
9.5.1　［レイアウト設定］ダイアログ 245

9.5.2 レイアウト設定の変更 248
9.6 レイアウトパート 250
9.6.1 パートの種類 250
9.6.2 パートの作成、変更、削除 252
9.6.3 [パート設定] ダイアログ 252
9.6.4 [パートの定義] ダイアログ 253
9.7 レイアウトエリア 254
9.7.1 レイアウトエリアのサイズ 254
9.7.2 エリア外のレイアウトオブジェクト 255
9.7.3 画面ステンシル 255
9.7.4 レイアウトオブジェクトの識別バッジ 256
9.8 ステータスツールバーとレイアウトバー 257
9.8.1 ステータスツールバー 257
9.8.2 レイアウトバー 258
9.9 インスペクタ（右パネル） 259
9.10 オブジェクトタブ（左パネル） 263
9.11 フィールドオブジェクトについて 276
9.11.1 フィールドのコントロールスタイル 276
9.11.2 値一覧 277
9.11.3 値一覧を使用するコントロールスタイル 278
9.11.4 関連を利用した値一覧 288
9.11.5 フィールド入力 294
9.11.6 レイアウトオブジェクトの表示／非表示 296
9.11.7 クイック検索対象の指定 298
9.11.8 フィールドへの移動 299
9.12 様々なレイアウトオブジェクト 301
9.12.1 テキスト／線／図形 301
9.12.2 ボタン／ボタンバー／ポップオーバーボタン 301
9.12.3 タブコントロール／スライドコントロール 306
9.12.4 ポータル 308
9.12.5 Web ビューア 309
9.13 テーマとスタイル 315
9.13.1 テーマを設定する 315
9.13.2 スタイルを設定する 316
9.13.3 スタイルとテーマの保存 316
9.13.4 コピーと貼り付け（ペースト） 317
9.13.5 レイアウトオブジェクトの表示の優先順位 318
9.14 画面の構成：マスタ／詳細レイアウト 319
9.14.1 ポータルで現在のテーブルを選択 323
9.14.2 設定によるポータルの動作の違い 324
9.14.3 アクティブな行 325
9.14.4 関連テーブルのポータルのマスタ／詳細 326

第 10 章 関数と計算式 329

10.1 計算式は論理 329
10.1.1 計算式の基本 329
10.1.2 評価 331

10.2 ［計算式の指定］ダイアログ 332
10.2.1 計算式の作成 332
10.2.2 フィールドパネル 333
10.2.3 計算式パネル 334
10.2.4 関数パネル 336
10.3 フィールド定義と計算式 337
10.3.1 関連テーブルのフィールドを使った計算 338
10.3.2 同じテーブルのフィールドを使った計算 339
10.4 関数 340
10.4.1 引数 340
10.4.2 関数の種類 342
10.4.3 UUID 343
10.4.4 数字関数 344
10.4.5 テキスト関数 1 345
10.4.6 日本語関数 348
10.5 レイアウトと計算式 352
10.5.1 ［次の場合にオブジェクトを隠す］ 352
10.5.2 条件付き書式 353
10.6 関数　その 2 365
10.6.1 取得関数 365
10.6.2 テキスト関数 2 365
10.6.3 論理式（0 か 1 か） 367
10.7 レイアウトと計算式 その 2 368
10.7.1 ポータルフィルタ 368
10.7.2 ポータルフィルタと集計タイプフィールド 380
10.7.3 ポータルフィルタ（2 つのフィールドの条件） 382
10.8 関数 その 3 384
10.8.1 日付と時刻に関する関数 384
10.8.2 論理関数 1 390
10.8.3 その他の関数 393
10.9 変数 394
10.10 関数 その 4 408
10.10.1 論理関数 2 408
10.10.2 オブジェクト関数 411
10.10.3 統計関数 412
10.10.4 JSON 関数 417
10.10.5 論理関数 3　再帰の関数 418
10.11 カスタム関数 421
10.11.1 カスタム関数に必要なもの 421
10.11.2 カスタム関数の作成 422
10.11.3 カスタム関数の利用 427
10.11.4 まとめてファイルに置いておく 428

第 11 章 スクリプトと自動化 429

11.1 スクリプトも論理 429
11.1.1 スクリプトの基本 429
11.1.2 スクリプトの実行 431

11.1.3　スクリプトの流れ 431
11.1.4　スクリプトとエラー 432
11.1.5　スクリプトの終了 433
11.2　[スクリプトワークスペース] 433
11.2.1　スクリプトパネル 434
11.2.2　スクリプト編集パネル 436
11.2.3　スクリプトステップパネル 438
11.2.4　スクリプト作成のショートカット 439
11.2.5　スクリプトの設計 440
11.2.6　スクリプトのテスト 441
11.2.7　スクリプトデバッガ 442
11.2.8　データビューア 444
11.3　スクリプトトリガ 445
11.3.1　スクリプトトリガの順番 446
11.3.2　レイアウトのスクリプトトリガ 447
11.3.3　レイアウトオブジェクトのスクリプトトリガ 450
11.3.4　ファイルのスクリプトトリガ 452
11.3.5　OnTimer 454
11.4　スクリプトの作成 455
11.4.1　スクリプト作成の注意点 456
11.4.2　「切り替え」や「移動」 457
11.4.3　スクリプト引数 458
11.4.4　[ウインドウの調整] スクリプトステップ 463
11.4.5　[関連レコードへ移動] スクリプトステップ 464
11.5　条件分岐、変数、引数 484
11.5.1　[If]、[Else If]、[Else]、[End If] スクリプトステップ 484
11.5.2　引数を使った条件分岐 489
11.5.3　スクリプトと変数 492
11.5.4　サブスクリプト 495
11.5.5　複数の引数 498
11.6　ユーザと対話する 510
11.6.1　[カスタムダイアログを表示] スクリプトステップ 510
11.6.2　Get (最終メッセージ選択) 関数 512
11.6.3　スクリプトで印刷する 512
11.7　エラーを処理する 517
11.7.1　[エラー処理] スクリプトステップ 520
11.7.2　スクリプトで PDF 保存 521
11.8　スクリプトの作成 その 2 525
11.8.1　パスを変数で指定する 525
11.8.2　検索をスクリプトで行う 528
11.8.3　[ユーザによる強制終了を許可] スクリプトステップ 534
11.8.4　[フィールド設定] スクリプトステップ 535
11.8.5　ウインドウでデータを選択 537
11.8.6　引数と変数を使ってターゲットフィールドを指定する 540
11.8.7　[レコード／検索条件確定] スクリプトステップ 540
11.8.8　ウインドウ 541
11.9　[Loop] で反復 548

11.10 データファイルの取り扱い.....558

第 12 章 集計とグラフ.....568

12.1 どのような集計やグラフが必要か.....568
12.1.1 設計が必要.....568
12.1.2 集計やグラフの要件定義と設計.....569
12.2 集計の作成.....572
12.2.1 準備.....572
12.2.2 集計の仕組み 1.....573
12.2.3 集計の仕組み 2.....581
12.3 グラフの作成.....593
12.3.1 グラフにどのようなデータを使うか.....594
12.3.2 グラフ表示の例.....597

第 13 章 モバイル／ FileMaker Go 608

13.1 FileMaker Go のクライアント.....608
13.1.1 カスタム App ファイルをどこに持つか.....608
13.1.2 FileMaker Pro との違い.....609
13.2 レイアウト.....611
13.3 スクリプト.....613
13.3.1 起動時に専用のレイアウトに切り替える.....614
13.3.2 デバイスからの挿入.....617
13.3.3 オブジェクトタイプフィールドの操作.....620
13.3.4 センサー情報とローカル通知.....623
13.3.5 その他.....625

第 14 章 運用を計画しよう.....626

14.1 運用準備.....626
14.1.1 カスタム App の使われ方.....626
14.1.2 ユーザが単独で使用（スタンドアロン）.....627
14.1.3 ユーザがホストにアクセスして使用（クライアント／サーバー）.....627
14.1.4 同期.....628
14.1.5 FileMaker Cloud または FileMaker Server によるホスト.....629
14.1.6 ハードウェアとアプリケーション.....631
14.2 データの準備.....633
14.2.1 ユーザ.....633
14.2.2 マスタ情報.....633
14.2.3 データ移行.....634
14.3 管理の準備.....636
14.3.1 管理者を決定する.....636
14.3.2 運用の 1 日を考える.....636
14.3.3 運用開始までの期間を考える.....637
14.4 複数のファイル.....638
14.4.1 データ分離モデル.....638
14.5 複数のカスタム App.....640

第 15 章 セキュリティ.....641

15.1 セキュリティの対象 641
15.2 アカウント 641
15.2.1 ユーザを決定 641
15.2.2 デフォルトアカウント 642
15.2.3 自動ログインの解除 643
15.2.4 アクセス権セット 644
15.2.5 アクセス権セットの設定 649
15.2.6 拡張アクセス権の設定 659
15.2.7 アカウント情報 661
15.3 ファイルアクセス 664
15.4 暗号化によるセキュリティ対策 672
15.4.1 ファイルを暗号化 672
15.5 ファイルの表示と非表示 674
15.5.1 ファイルを常に非表示にする 674
15.5.2 FileMaker WebDirect の設定 675
15.6 セキュリティを計画する 676

第 16 章 カスタム App の最適化 678

16.1 ユーザについて 678
16.2 ステータスツールバーの表示／非表示 679
16.3 カスタムメニュー 680
16.3.1 「メニュー」とは 681
16.3.2 カスタムメニューセットを切り替える 682
16.3.3 カスタムメニューの作成 682
16.3.4 メニューとメニュー項目 687
16.3.5 カスタムメニューを他のファイルにコピー 693
16.4 アカウントの管理 694
16.5 レコード削除の注意 700
16.6 排他 703
16.7 起動と終了の処理 705
16.7.1 起動：作業開始準備 705
16.7.2 終了 706
16.7.3 その他 707

追補 1 708

追補 2 709

索引 712

第 1 章 開発者になろう

本書の目指すところ「カスタム App」の「開発者」として必要なことの初歩、開発者の段階による技術目安などを確認します。

1.1 はじめに

本書は、Claris™ FileMaker Pro™ で「**カスタム App**」を作成したことがある方が『**ネットワーク上で共有されて、複数人が利用するカスタム App の作成について学ぶ**』ことを目的としています。モバイルデバイスからの利用の初歩も含みます。演習ファイルで FileMaker Pro の操作を体験し、FileMaker の良さ、カスタム App をつくることの楽しさを感じてください。

FileMaker は、『カスタム App の作成と使用のためのマルチプラットフォームアプリケーション』です。製品のラインナップは 3 種類で、それぞれ役割と性能が異なります。

用途	製品名
カスタム App の作成と使用	Claris FileMaker Pro
カスタム App の使用	Claris FileMaker Go® Claris FileMaker WebDirect™
カスタム App の共有と運用管理	Claris FileMaker Cloud® Claris FileMaker Server™

※ 詳しい製品情報や必要な動作環境、技術仕様などの最新情報はクラリス・ジャパン株式会社の Web サイト（https://www.claris.com/ja/）で確認してください。

本書は、「FileMaker Master Book：初級編」と同等の内容を理解し、FileMaker Pro の操作を一通り習得していることを前提として解説します。次のようなことです。

・画面の各部位やツールの名称や操作を理解している
・モードや表示形式を切り替えられる
・レコード間の移動、レイアウト間の切り替えができる
・検索を実行して対象レコードをソートできる
・ファイル、テーブル、レコード、フィールドを作成できる
・簡易なリレーションシップ、スクリプト、計算式について理解し作成できる
・インポートとエクスポートを操作できる
・印刷や集計、クイックグラフの基本を理解している
・アカウントへのパスワードの設定、アクセス権セットの基本を理解している
・共有の基本を理解している

まず、本書の文言などの前提についていくつか説明します。

1.1.1 業務

本書では「業務」という文言をたびたび使います。

業務は、簡単に言えば、企業や団体などで人が日常的に行っている「仕事」のことです。一連の流れがいくつか存在し、それらをさらにひとかたまりにしたものです。例えば、次のように呼ばれます。

・販売業務
・受付業務
・接客業務
・精算業務
・請求業務

1.1.2 カスタム App

企業や業務で発生する日々の問題や課題の解決のために作成、使用されるのが「**カスタム App**」です。使う人を適切にサポートするため、オリジナルな機能や画面を持つ特別な「**カスタムアプリケーション**」です。そして『人が使用する道具（ツール）』です。

1.1.3 システム

「**システム**」は、少し広い意味です。カスタム App だけでは、企業や団体における業務の問題や課題は解決できません。ほかにも、

・ユーザ
・アプリケーション
・サーバーやパソコン、デバイスやプリンタ
・ネットワーク
・その他、業務に必要な様々なもの（人や商品や情報）

などの要素が揃っている必要があります。これらは相互に関係し影響します。本書では、これらも含めてシステムと呼びます。

業務支援（サポート）
システムがつくられる理由は、主に業務における問題や課題の解決をサポートするための「**業務支援**」です。その中で、カスタム App は重要な位置にあります。カスタム App の作成をきっかけに業務の流れの改善が起きることもあるでしょう。

カスタム App の役割
システムにおけるカスタム App の役割の基本は、いつも同じです。対象業務の中で『必要な情報の入出力』をユーザが使用する画面や機能で実現し、作業をサポートすることです。

入出力については、次のことを考える必要があります。

＜データ入力＞
(1) いつ
(2) どこで
(3) 誰が
(4) なにを
(5) どのように
(6) どうやって

＜データ出力＞
(1) いつ
(2) どこで
(3) 誰が
(4) なにを
(5) どのように
(6) どうやって

入力、出力ともに同じ (1) から (6) の要素が並んでいます。さらに「なぜ」は、カスタム App がサポートする作業から設定されます。開発プロセスで言えば、「要件定義」と呼ばれる工程で決定します。

※ システムをつくりあげる手順は「開発プロセス」と呼ばれます。ごく簡単に本章のコラム（7 ページ）で後述します。

1.1.4 データ

カスタム App が取り扱う「**データ**」は、対象業務に必要な「**情報**」です。そして、データは大切な資産でもあります。日常業務で使用される以外にも、蓄積されたデータを目的に応じて集計したり、グラフで可視化して、なんらかの決定における判断材料にするなど活用可能です。

そして、データは主に

・フィールド
・レコード
・テーブル

を、単位として取り扱われます。

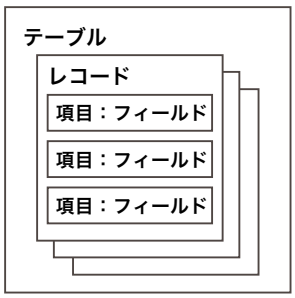


図 1-1 フィールド、レコード、テーブル

1.2 開発しよう

突然ですが、どのようなきっかけで、FileMaker Pro を使いはじめましたか？

カスタム App が作成される理由は、自分の作業を楽にしたり効率を良くしようとした結果であることがほとんどです。

うまく動作すると、同じ作業をしている周りの人々も使いはじめ、徐々にカスタム App が浸透するかもしれません。するとカスタム App が対象にする作業の範囲が増えていきます。当然、自分以外のユーザの要望が生まれます。それらが実現されていくとますます便利なものになり、機能が充実します。はじめは小さかったカスタム App は徐々に大きくなって、さらなる要望を実現し続けると、あるとき気づくのです。

『いろんなところがこんがらがってきて、だんだん大変になってきてしまった……』

あの機能やこの機能は、もっとすっきりできそうだとか、あっちでもこっちでも入力されているこの文言は共通にできそうだとか思えてきます。それは、FileMaker のことが今までよりも分かってきたからです。技術力が上がるほど、目につくところがでてきます。いっそのこと、つくり直したくなるかもしれません。

それは、少し進んだ「**開発者**」としての第 1 歩です。

1.2.1 考えておくこと

本書が目指す内容において考えておくべきことがあります。これまでの、

・身近な問題や課題の解決

に加えて、

・自分以外の複数のユーザが使用すること
・FileMaker Cloud や FileMaker Server で共有すること
・『いざというとき』のための運用

の 3 つです。また、次のことも考えます。

・自分以外の開発者がメンテナンスできるようにする

規模にもよりますが、企業や団体の業務で使用するカスタム App は、1 人ではなく複数の担当者がメンテナンス（保守）できることがベストです。

これらのことを考慮しながら、カスタム App を「**開発**」、つまり、新しく生み出しましょう。

メンテナンス
カスタム App やネットワークを正常に動作させ続けるために行う作業のことです。「**保守**」とも呼ばれます。

1.2.2 開発者

本書では、初級開発者が中級開発者を目指すために学習すべきことを解説しています。

一般的に、データベースをつくることを『データベースを開発する』と言い表すことがあります。今までよりも便利になるものを新しく生み出すからです。同じように、データベースをつくる人のことを「**開発者**」と呼びます。

これから、経験をどんどん積んで、上級、あるいはプロフェッショナルな開発者になりたいですか？それとも、より上位の開発者を目指しながら、ゆくゆくは部下に指示を出して、開発プロジェクトを統括する立場にシフトしていくかもしれません。

さまざまな開発者

技術の習熟度や経験度による段階の考え方を記述してみます（あくまでも、これは 1 つの考え方です）。

初級者とは

・ユーザとしてカスタム App を使ったり、自分のためにカスタム App を開発する
・最低限のセキュリティを考慮する
・1 つのファイルでカスタム App を構成する場合が多い

中級開発者とは

・（自分あるいは他の人を）ユーザとして想定し社内用のカスタム App を開発する
・カスタム App を FileMaker Cloud / FileMaker Server で共有する
・FileMaker Pro の標準メニューを使用するが、用途に合わせてある程度制御する
・セキュリティを考慮したアカウントを複数つくる（FileMaker Pro 標準機能を使用）
・1 つのファイルでカスタム App を構成する場合が多い

上級開発者とは

・（自分あるいは他の人を）ユーザとして想定し社内用のカスタム App を開発する
・カスタム App の FileMaker Cloud / FileMaker Server での共有を前提とする
・カスタム App の安全な運用計画（バックアップなど）を立てる
・カスタムメニューを使用するセキュリティを考慮したアカウントと、その管理機能を作成する
・状況に応じて、1 つあるいは複数のファイルでカスタム App を構成する
・後任者へ引き継ぎ可能な資料やスクリプトへのコメント記述などができる

プロフェッショナル開発者とは

・FileMaker の機能を熟知しており、特性を生かした適切なシステム構築ができる
・外部から委託されて、完成されたソリューションとしてのカスタム App を開発する
・要件の定義、仕様策定、設計を行い、その記録を残す
・カスタム App の FileMaker Cloud / FileMaker Server での共有を前提とする
・カスタムメニューを使用するセキュリティを考慮したアカウントと、その管理機能を設計、作成する
・他のデータベースやシステムとの連携など計画できる

必ずしも、上級、プロフェッショナルを目指さなくても良いのです。その都度必要なことができるようになれば充分です。

1.3 PDCA

インスピレーションの赴くままにアプリケーションを操作し、試行錯誤しながら開発できる手軽さや楽しさは FileMaker Pro の素晴らしい点ですが、本書では少し違う視点から「計画、実行、確認、改善」で、開発を進めることを提示します。いわゆる「**PDCA（= Plan Do Check Action）サイクル**」です。カスタム App 開発全体と、1 つ 1 つの機能を作成することの両方で意識することを想定します。

計画（Plan）
『どれぐらいの期間（スケジュール）で開発するか』だけでなく「**設計**」という概念も該当します。きちんとおこなっておくことでエラーを減らすことができるからです。

計画には重要な役割があります。『どのような結果であるべきなのか』、つまり「答え」を用意することです。

実行（Do）
『計画を実行すること』です。開発全体を進めること、実際に FileMaker Pro を操作して機能などの開発作業を行うことです。

確認（Check）
チェック、つまり「テスト」です。エラーがないか、正しい結果かどうかを確認します。「計画＝設計」で想定した「答え＝動作や結果」になっているかどうかを確認するということです。

改善（Action）
確認後、必要があれば改善します。結果にエラーがあれば訂正し再度確認します。ただし、計画時に用意したとおりの結果だったが状況が変わったために修正が必要になった、という場合、次の計画で新しい結果を用意します。エラーがなければ、そこで完了か、別の計画に移ります。

ここまでで 1 回のサイクルとなります。PDCA は 1 回で完了することもありますし、何度も反復して実施することもあります。

計画のメリット
計画することに慣れるまで時間がかかることがありますが、こう考えてみてください。計画しない場合、次の両方を考えながら、試しながら開発をおこなう必要があります。

・どのような結果であるべきなのか
・エラーを防ぐ

正しい結果という目的がはっきりしていれば次のことだけに集中できます。

・エラーを防ぐ

結果的に全体の作業時間の短縮が可能です。取り入れることをお勧めします。

システム開発プロセス

「システム開発プロセス」と呼ばれるものがあります。ユーザも機能もデータも少ないカスタム App の開発であれば、プロセスなしでも進め方に迷うことはほとんどありません。しかし、ある程度の規模が想定される場合、プロセスに沿って進めると確実です。

『規模が大きいからシステム開発プロセスでやってみよう』といきなり挑戦するよりも、規模にかかわらず普段からプロセスを取り入れることをお勧めします。

＜基本のプロセス＞
システム開発の歴史は長いので、プロセスの「モデル（型）」はいろいろあります。「どのモデルが良い」という答えはありません。開発したいシステムの規模や内容、開発者の特性、好みなどで選択します。基本的な流れは、概ね以下です。
(1) 企画
(2) 要件定義
(3) 設計
(4) 開発
(5) テスト
(6) リリース（完成）
(7) 運用

＜リリース（完成）＞
システムが完成することです。次のリリースがあるまで、その内容で使うものであると考えてください。なお、あくまでも器としての完成です。実際に業務で使い始めるまでには中身（データなど）の準備が必要です。

＜運用＞
システムがリリースされた後、業務で使っていくこと全般です。カスタム App をサーバーで共有してユーザが使用を開始すること、定期的にバックアップ、データ処理を実行したりすることです。一般的に「保守」も含んで運用と呼ぶことが多いようです。

＜プロセスと変更＞
特に注意するのは途中で変更があって対応のための作業が必要になる、しかもそれが何度も続くことです。使いやすいシステムにするには試行錯誤が必要です。しかし何度も変更すると、正しい動作や結果がはっきりしなくなり、何度も手直しをすることによりエラーも発生しやすくなります。そして、いつまでたってもできあがらないことにつながるのです。従って変更の決定には十分な考慮が必要です。

＜プロトタイプの勧め＞
『文字ばかりの書類だと想像できないから実際に見せて欲しい』と言われることがあります。そのようなとき、プロトタイプを利用することをお勧めします。検討段階で試行錯誤して最終的に決定した内容をリリースします（PDCA 的に、つくって、見せて、確認してもらって改善、というように検討します）。

まず、紙にワイヤーフレームと呼ばれる線画を描いて、画面構成や画面の遷移を検討するのが良いでしょう。検討後、「モックアップ」と呼ばれるプロトタイプを FileMaker Pro でつくるか、そのまま開発に入る場合もあります。

この方法は『どのような結果になれば良いのかはわかっているが、具体的な動作や操作、画面デザインがどのようなものがイメージが曖昧なとき』に有効です。なお、慎重に考えられるということは、それだけ期間が長くなる可能性があるので、対象を限定して取り入れることをお勧めします。そして、対象となる部分だけに集中し過ぎず、他の部分とのバランスを考える必要があります。ある部分でこれが良い、別の部分でこっちが良いとしたときに辻褄が合わなくなることがないようにします。

一番大切なことは決定する答え（目的）です。何を開発したいのかがはっきりしていれば、どのプロセスモデルを採用したとしても適切な進め方を考えることができます。FileMaker Pro は、プロトタイプの作成においても強力なツールです。レイアウトやスクリプトを修正しながら、意見の反映と結果の確認が可能です。

1.4　準備

本書を読み進めるための準備をします。

1.4.1　ソフトウェアの準備

次のソフトウェアをダウンロードし、インストールしてください。

FileMaker Pro
無料評価版がダウンロードできます。
https://www.claris.com/ja/trial/

FileMaker Go（App Store から無料ダウンロード）
https://content.claris.com/ja_fmgo_appstore

（次の 2 つは本書では解説しません。詳しくは、本書の中級編、上級編、プロダクトドキュメンテーションセンター (https://www.claris.com/ja/resources/documentation/）などを参照してください）

・FileMaker Cloud（「リソースとダウンロード」ページ）
　https://content.claris.com/FM_Cloud_Resources_LP_JA

・FileMaker Server（FileMaker Pro と同じリンクから無料評価版ダウンロード可）

1.4.2　演習ファイルの準備

演習ファイルをダウンロードしてください。デスクトップなど、開きやすい場所に保存しておくと便利です。

演習ファイル
https://content.claris.com/ja_fmb19_chukyu

※演習ファイルについて
演習ファイルは章ごとにフォルダに分かれています。章ごとのフォルダの中はさらに、章の開始時、完了時、演習前、演習後（完了時：確認用です）などのフォルダがあり、その中に各演習ファイルが入っています。本書では、必要に応じてそれらのファイルを使う旨を記述しています。

なお、ファイル名の頭に　2_　のように章の数字がついていますが本文中では省略されていることがあります。

※本書で使用している図版について
本書で使用している図版は、主に macOS 版の FileMaker Pro で作成しました。

※演習ファイルの展開について
演習ファイルを Windows 7 で展開するとファイル名が文字化けすることがあります。なお、マイクロソフトによる Windows 7 のサポート終了に伴い、FileMaker Pro では Windows 7 はサポートされなくなりました。

1.4.3 高度なツール

FileMaker Pro にはカスタム App 作成に役立つ「高度なツール」が含まれています。

・データベースデザインレポート（カスタム App の構造などの情報）の出力
・スクリプトデバッガ（スクリプトのテストに便利なツール）
・データビューア（計算式などの結果の確認などに便利なツール）
・カスタムメニュー（標準のメニュー以外のメニュー）の作成
・カスタム関数（標準の関数以外の関数）の作成
・フィールドやテーブルのコピーと貼り付け（ペースト）やテーブルのインポート
・レイアウトテーマのコピーとペースト（貼り付け）
・Developer ユーティリティ（ファイルの暗号化など）

これらのツールを利用するには、「高度なツールを使用する」設定がされている必要があります。もしメニューバーに「ツール」が表示されていない場合は、次の手順で設定してください。

手順：

(1) メニューから［環境設定］ダイアログを表示します。

macOS → [FileMaker Pro]–[環境設定 ...] メニューを選択します。
Windows → [編集]–[環境設定 ...] メニューを選択します。

(2) ［環境設定］ダイアログが表示されます。
(3) 「高度なツールを使用する」のチェックを **オン** にします。
（チェックボックスをオンにすると「FileMaker Pro の再起動が必要」と表示されます）

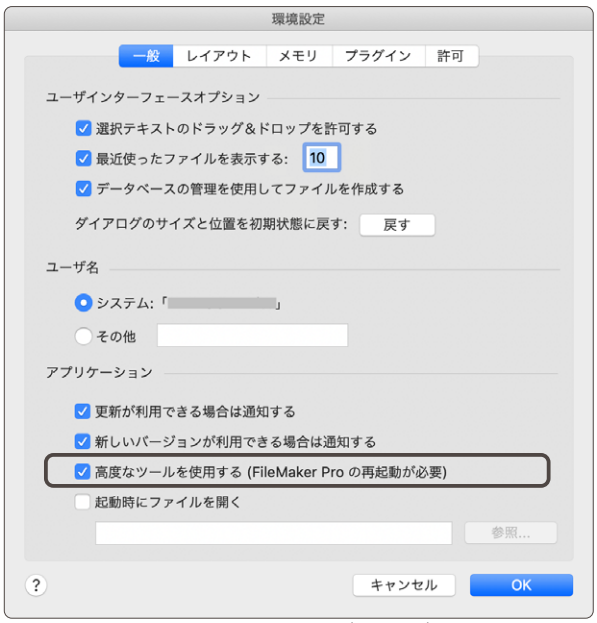


図 1-2［環境設定］ダイアログ

(4) ［OK］をクリックします。
(5) FileMaker Pro を終了し、再度起動します。
(6) メニューの「スクリプト」の右に「ツール」が増えたことを確認します。

1.4.4　開発環境

開発者として準備しておきたい環境（ハードウェア、アプリケーション、ネットワーク）は、次の内容がベストです。

・FileMaker Pro とクライアントパソコン
・FileMaker Go とモバイルデバイス
・FileMaker Cloud、あるいは FileMaker Server とサーバー機（サーバー環境）
・LAN などのサーバー機とクライアントパソコンを繋ぐネットワーク

FileMaker Cloud または FileMaker Server でホストしながら開発作業をすると次のメリットがあります。

・共有状態の動作をテストできる

これは、共有利用するカスタム App の作業で大事なことです。さらに、定期的にバックアップが保存されますので、

・不慮の電源喪失などにより強制終了でファイルが破損してしまった
・開発中の機能で致命的なエラーが起きて少し前のファイルに戻りたい
・間違ってレイアウトやテーブルやフィールドを消してしまい前のファイルに戻りたい

このような場合にも慌てずに対応できます。

ベストな環境が用意できなければ、開発中のカスタム App ファイルのバックアップをこまめに取って不測の事態に備えます。なお、動作の確認やユーザによるテストは、運用開始後と同様の環境で実施することを強くお勧めします。

1.4.5　リソース（参考資料）

FileMaker プラットフォーム製品には、それぞれヘルプのドキュメント（オンラインで利用可）が用意されており、クラリス・ジャパン株式会社の Web サイトで公開されています。また、同社の Web サイトでは、ナレッジベース、プロダクトドキュメンテーションセンターやテクニカルリソースなども提供されていますので、これらの情報も活用してください。

なお、本書以外にも、カスタム App の作成について学ぶために、クラリス・ジャパン株式会社や同社に認定されているパートナー企業の講師によるトレーニングが開催されています。

また、本書の内容を解説したビデオチュートリアル「FileMaker Master Book 中級編」が、YouTube のクラリス・ジャパン株式会社のチャンネルで公開されています。

詳細は、クラリス・ジャパン株式会社の Web サイト（https://www.claris.com/ja/）を参照してください。

第 2 章 企画しよう

『なにを目指すのか』を企画ではっきりさせます。目的を元に「基本の設計」を行うためです。

『ある程度の機能や画面を持つカスタム App を中心としたシステムを新しく作成すること』を想定し記述します。なお開発者は、対象とする業務について『概要を知っている』という想定です。もしも本書の方法を実際に取り入れる場合、規模や内容に応じてアレンジしたり簡略化してください。

2.1 なぜ企画するのか

新しいシステムを作成しようとするとき、一番最初に必要なのは「**企画**」することです。

これまで身近な問題の解決をきっかけにしてカスタム App の作成してきた場合、企画の必要性を感じないかもしれませんが、次のことを考えてみてください。

・自分以外のユーザが複数使用する

『みんなで使う』のですから『共通の目的（ゴール）を目指せるようにする』必要があります。自分だけではなく他のユーザの視点も考慮することがとても重要です。

企画は、様々な意見の中から、これから作成しようとしているシステムが『なにを対象にするか』という「**範囲**」を絞り込むことが目的です。範囲を決めることでゴールが決定し、そこを見据えて取り組めるようになるのです。

なお、目指す範囲が広い場合、すべてを一度で網羅する必要はありません。規模が大きかったり、多くの機能が必要だったりするのであれば、次のような決定をしたり、

・何回かに分けて、ゴールを複数設定する

あるいは、対象業務をとりまく環境がどんどん変わる（ゴールが変わる）ことが考えられるなら、

・定期的、継続的にゴールを設定する

と決定しても良いのです。さらに、複数人で共有利用するカスタム App では、

・FileMaker Cloud あるいは FileMaker Server で共有すること
・『いざというとき』のための運用

これらも念頭に入れて企画することが重要です。そして、関係者が同じように理解することを目指します。

2.2　問題や課題の収集

新しいシステムが必要とされる背景には、業務などに何らかの問題や課題があって、それを解決したいという要望があることがほとんどです。ですから、次のようなことを考えて意見や問題（だと思われていること）を集めます。

・どのような問題や課題を解決したいのか　→　**「目的」**（解決して得たい結果）

こうした収集では、○○の機能はどうなるのか？□□の入力はできるのか？など、ピンポイントな質問や要望も集まってきます。ここでは、個別の検討はせず、問題や課題を見つけ、どの業務のものか、どの作業のものか、など『対象範囲の候補になるかどうか』を中心に考えます。

2.2.1　整理する

集まった問題や意見などを整理します。箇条書きなどにして内容によってまとめます。これによって、包括的に眺めることができるようになりゴールが見えてきます。

次は、整理された意見を元に、問題や課題の解決のために必要なツールについて考えます。本書では、次の内容とします。

・解決のためにつくるもの（ツール）　→　本書では「**カスタム App**」
・ツールはどのような内容なのか　→　大まかな「**機能（役割）**」

2.2.2　対象と制約

ツール（カスタム App）の「**対象**」について考えます。『なにを対象にするか』です。

・どの業務（作業）か
・どのような情報を取り扱うか
・誰がどこで使うのか

次のことも考えます。「**制約**」などと呼びます。

・○○の期間だけ使えれば良い or 長く使いたい
・○月から使いはじめたい
・どのようなネットワーク環境が必要か、あるいは用意できるか
・クライアントは、パソコン、iPhone や iPad、Web ブラウザのどれか（1 つまたは複数）
・カスタム App だけで完結するのか、他のシステムとの統合があるのか
・セキュリティはどう考えるか
・ハードウェアやソフトウェアは新たに購入する必要があるかどうか

このようにして、企画でシステムの概要を把握していきます。意見を整理して対象や制約が見えてくるとカスタム App のイメージが浮かんできて、すぐにでも作成を開始したくなるかもしれません。しかし、もう少し企画で行うことがあります。

2.3　概要の把握と決定

整理した内容を元に、方向性などを確認し、まとめて決定します。

2.3.1　概要

次のことが企画内容となります。

（1）目的
（2）つくるもの　→　　本書では「カスタム App」
（3）対象範囲（業務、情報、誰がどこで使うか）
（4）制約
（5）作業期間、その他の制約

これらには、新しいシステムが必要とされる理由である次の点が含まれています。

・なぜ
・なにを
・どのように

企画は、あくまでも概要を把握するだけです。時間をかけすぎないために念頭においておくことがあります。それは、加減です。

詳細に入り込まない
あえて大まかに物事を取り扱います。具体的に機能の 1 つ 1 つの詳細を考えはじめると全体を把握することができません。

曖昧すぎてもいけない
『良いものが欲しい』という目的だけでは、「良い」が示すことが大きすぎます。

適切な加減には決定的な答えはありません。細かすぎない、曖昧すぎない、を考えてください。何度も述べますが、問題や課題を見つけだして目的、対象、制約を具体的にします。

2.3.2　決定

最終的に企画内容を決定します。『必要な人々の間で決定する』ということになります。

「必要な人々」の範囲については、確実な答えはありません。企業や業務により事情が異なるからです。決定者は、自分かもしれませんし、上司やリーダー的な人々かもしれません。意思決定に関わる必要がある人、あるいは人々のことを「**ステークホルダー**」と呼ぶことがあります。ステークホルダーが決定することを「**合意する**」と呼びます。

※ 企業や団体によっては企画内容が承認されることで新しいシステムをつくることが本格的に決定します。

企画はベース
企画で決定した内容は、今後の基本（ベース）になります。カスタム App の構造、機能、画面の設計時に重要な要素になります。例えば、『今回の対象範囲では無いけれど次や将来に△△業務でも使いたい』と書いてあれば、そこを考慮した構造設計が可能かもしれません。

2.3.3 書き残す

決定した企画の内容は、図や文字で書き表して関係する誰もが確認できるようにします。

書類で配布しても、PDF でどこかに保存しても、企業で使用している wiki サービス上のページでも良いのです。自分だけが使うのであればこのようなことは大げさかもしれませんが、業務で複数の人が使用する予定のシステムは情報を公開することも重要です。

2.3.4 期間

制約に含まれている項目の中から「期間」を決定することは重要です。

・○○の期間だけ使えれば良い or 長く使いたい
・○月から使いはじめたい

これらは、『新しいシステムがいつから使えるか（使えるようにならなければいけないか）』と関連するからです。システムの目的は問題や課題を解決することなのですから、それがいつになるのかを考えなければなりません。

また、「いつから使えるか」は『いつまでにカスタム App が完成しなければならないか』ということを表しています。

期間と作業量

作業期間に比較して、カスタム App 作成の作業量が多すぎるなら

・対象範囲のうち、次の開発に回すものを決める
・作成作業を実施する開発者を追加する
・作業期間を変更する

などの判断が必要になります。

本書で想定しているシステムの場合、中心となるのはカスタム App の開発期間です。それ以外にもシステムの使用開始までの準備期間が必要なことを忘れないでください。

2.3.5 その他

新しいシステムによる業務の変化は起きて当然です。特にカスタム App を導入することで手作業を自動化したり分散している情報を一元化することによる変化は必ず起こります。しかし、別の要素が見えてくることがあります。

業務改善

問題や課題の解決に『今とは違う業務の進め方』が必要になる可能性です。

時々、『業務の流れを大きく改善することが必要なのではないか？』などの意見がでることがあります。そのようなことが必要であると決定したら、担当者や担当チームが、何を優先するべきか順位を決めて取り組みます。そして業務の大きな改善とカスタム App の開発を同時進行することが可能なのかどうかも検討します。

本書では、次表の企画が決定したものとします。**システムの名前は「見積作成管理システム」です**。

見積作成管理システム	
目的	担当者がばらばらに作成している見積書を一元化したい （全員が見積書を同じデータベースで作成することで見積内容や注文されたかどうかなどの活用をできるようになる）
つくるもの	見積作成管理システム（カスタム App）
対象範囲	対象部門　営業部門 対象業務　見積業務 ユーザ　　営業部門の人員（10 人程度） 場所　　社内での使用
制約	FileMaker Pro で開発する ○ヶ月で完成想定 クライアントはパソコン FileMaker Cloud または FileMaker Server で共有する（検討中） 既存の社内ネットワークを使用する 営業部門の人員以外は使用できないようにする 他のシステムとは統合しない
その他	サーバー機は専用サーバー機の購入が必要かどうか要確認 ソフトウェアは追加購入不要 クライアントパソコンの追加購入不要

本書を読む方によっては「見積」に関する業務は身近ではないかもしれませんが、上表の「目的」は違う視点から考えるとどこにでもあることです。

・担当者がばらばらに作成している見積書を一元化したい
（全員が見積書を同じデータベースで作成することで見積内容や注文されたかどうかなどの活用が可能になる）

これは、次のような目的で新しいシステムが必要になることと同じシチュエーションです。

・社員がばらばらに管理している顧客情報を一元化したい
・保育園や幼稚園の職員が各自で作成している連絡ノートの記述を一元化したい
・病院の先生が研究のために収集する情報を一元化して記録したい

企画の内容ができました。次は設計のための材料集めをして、「**要件**」を定義します。企画の概要から、全体像をはっきりさせていきます。

カスタム App 以外の要素である、

・サーバー機やパソコンやプリンタ
・ネットワーク

これらもこの時点から考えていきますが、本章では記述しません。第 14 章を参照してください。

第 3 章 要件を定義しよう

企画の次は、ヒアリングで材料集めをし「要件」を「定義」します。設計のためのさらなる準備です。

「見積作成管理システム」の「要件定義」をしていきます。ここからは、主にカスタム App について記述します。

3.1 ヒアリング

「**ヒアリング**」で情報を集めます。聞いたり調べたりすることです。企画で集めた要望も参照します。

3.1.1 なにを聞くか

企画内容が決定されたことで、新しいシステムに関係する人々が同じ目的を目指せるようになりました。少し詳しいことを考えます。主に次のようなことを聞いていきます。

・カスタム App にどのような機能が必要か検討するためのこと
・入力する項目と書式（あれば）
・出力する項目と書式（あれば）

打ち合わせ、実際の業務の様子を見学、など、様々なやり方があります。企画段階の意見収集は、概要を掴むためでした。要件定義では、企画で決定した内容を前提としてさらに意見を収集していきます。

3.1.2 誰に聞くか

カスタム App を実際に使用するユーザの要望を聞きます。企画で対象範囲を決めました。その対象業務に携わる人、対象ユーザの人、必要があれば業務の責任者などにもヒアリングを実施します。

業務に携わっている現場からのみヒアリングをすれば良いかどうかは微妙なところです。例えば、対象の業務が複数ある場合、それぞれの現場の人にだけ話を聞くと業務の連携部分が曖昧になるかもしれません。業務の責任者同士に決定してもらう必要がでてきます。

出力（印刷や集計結果、グラフなど）項目は、責任者や責任者が出席する会議のメンバーが必要だと考える情報を把握する必要があるかもしれません。

ヒアリングは、対象者の数に比例して様々な意見がでてきます。ここでも、○○の機能はどうなるのか、□□は入力できるのか？などのピンポイントの疑問や要望が集まってきます。それらの話について企画のときよりも少し詳しく概要を聞きますが、決定はしません。まずは、要望を集めましょう。

3.1.3 業務の流れ

対象業務について具体的なことがわかっていますか？どのような作業があって、どのような流れで進める業務なのか、を把握してください。

対象業務を理解することはとても重要です。すでによく知っている業務であれば、なにを要望されているのかわかりやすいかもしれません。しかし知らない業務だったら、おおまかな流れ（必要があれば詳細な作業も）を確認してください。

フローチャート（流れ図）

業務の流れをわかりやすく記述する方法として「**フローチャート**」があります。いくつかの記号（箱）に意味を持たせて並べ、矢印線で繋いで流れを表現するものです。詳しく解説しませんが、フローチャートはとても便利です。機能の設計にも使えます。詳しい情報は、書籍などでたくさん解説されていますので参照してください。対象である「見積業務」の流れは下表です。これをフローチャートにすると右図になります。

見積業務（現在）の流れ
お客様から連絡がくる （開始）
見積書を作成する （担当者名で）
印刷する
上司が承認する
お客様に見積書を提示する （持っていくか郵送する）
お客様が検討する
（結果がでる） 注文をもらう （FAX か電話かメール） または 注文されない
結果を記録する （注文されなければここで完了）
注文をもらった場合、 注文を処理する部署に連携 （完了）

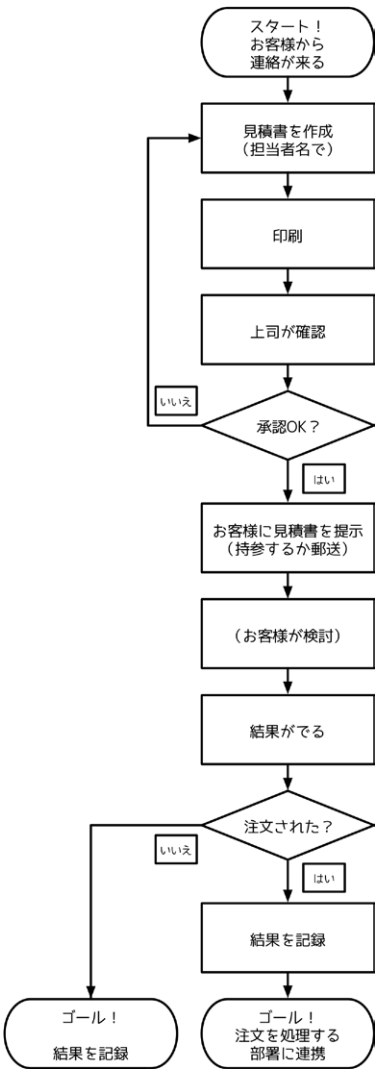
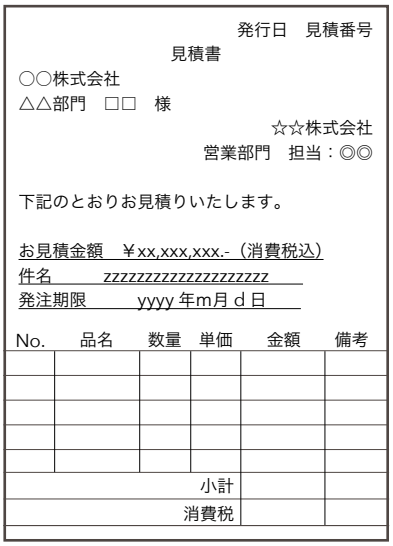


図 3-1 業務の流れをフローチャートで書き表した例

3.1.4 入出力の項目

この時点でわかっている入力、印刷や出力するものについて項目を確認します。例えば、

・入力は、申請書や申込書
・出力は、見積書や請求書



発行日　見積番号

見積書

○○株式会社
△△部門　□□　様

☆☆株式会社
営業部門　担当：◎◎

下記のとおりお見積りいたします。

お見積金額　￥xx,xxx,xxx.-（消費税込）
件名　zzzzzzzzzzzzzzzzzzz
発注期限　yyyy 年m月 d 日

No.	品名	数量	単価	金額	備考
			小計		
			消費税		

図 3-2 見積書書式例

などです。データで出力する場合は、ファイル形式や必要な項目、項目順なども併せて聞いておきます。聞くだけではなく実際に業務で利用しているものを収集するのがベストです。それらは、フィールドを設計するためにとても役立ちます。

3.1.5 できないこともある

ヒアリングで集めた要望は「**要件**」にする必要があります。重要なのは、必ずしもすべてが要件にできるわけではないことです。次のものは要件にする検討対象から外します。

・システムでは明らかに実現できないもの
・企画で決定した対象範囲から逸脱しているもの
・企画の制約では実現できないもの

例えば、営業部門の「見積作成管理システム」の企画についてヒアリングしていたら『倉庫の在庫管理がしたい』という要望がでてきたら……。

見積書の作成時、商品の在庫数も関係あるかもしれませんが、一般的に倉庫の在庫管理はカスタム App の一機能として考えるには大きすぎます。別の企画として「在庫管理システム」が必要になります。ですから『この企画では対象外』となります。しかし、将来の新しいシステム開発のための検討事項としておくと良いかもしれません。

3.1.6 期限を切る

後から後から意見が出てくるとヒアリングが終わらなくなることがあります。そのような場合は、期限を切ってそれまでに代表する人にまとめてもらいます。期限後の要望は次の開発時に収集する形にすることも考えます。

ここでは「見積作成管理システム」の業務の流れとヒアリングした内容を以下に想定して続けます。

見積業務（現在）の流れ

- お客様から連絡がくる（開始）
- 見積書を作成する（担当者名で）
- 印刷する
- 上司が承認する
- お客様に見積書を提示する（持っていくか郵送する）
- お客様が検討する
- （結果がでる）注文をもらう（FAX か電話かメール）または注文されない
- 結果を記録する（注文されなければここで完了）
- 注文をもらった場合、注文を処理する部署に連携（完了）

発行日　見積番号

見積書

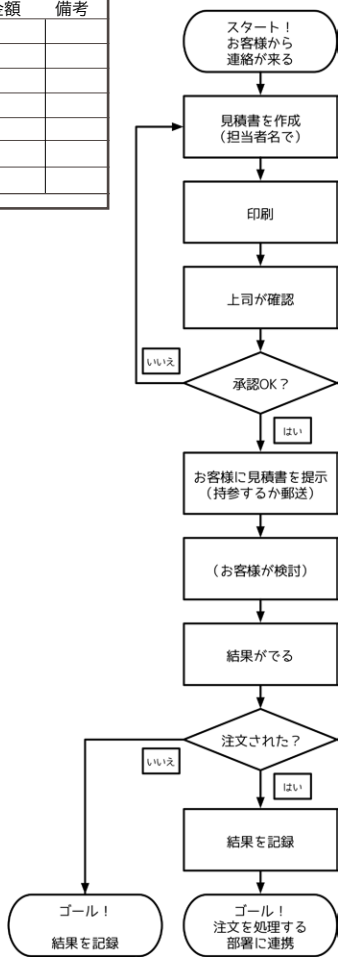
○○株式会社
△△部門　□□　様

☆☆株式会社
営業部門　担当：◎◎

下記のとおりお見積りいたします。

お見積金額　￥xx,xxx,xxx.-（消費税込）
件名　zzzzzzzzzzzzzzzzzzz
発注期限　yyyy 年m月ｄ日

No.	品名	数量	単価	金額	備考
			小計		
			消費税		



見積作成管理システムのヒアリングで出た意見

見積書をいまよりも簡単につくりたい（月に 100-200 件程度）
見積書と同じ内容の発注書も一緒に印刷できると良い
見積書の印刷以外に、PDF は作成できるか？
他の人のつくった見積書を見て参考にしたい
見積書を印刷したい
お客様ごとに自社の担当者がいる
見積がいまどういう状況なのかわかるようにしたい
~~倉庫の在庫管理がしたい~~　→　今回の対象外
前回の見積書を複製して作成できるようにしてほしい
見積書は A4 サイズで印刷する
担当者が外出しているとき他の人が代理で見積作成したい
番号で検索したい
~~外出先で情報を確認したい~~　→　今回の対象外
見積書は今の書式で問題ない
お客様が紛失したときは再発行する
お客様にメールで見積書を送信したい
お客様ごとの注文実績を見たい
見積は訂正することがある
発注書が新しく欲しい

3.2　要件を定義する 1

「**要件**」は「必要な条件」です。「**要件定義**」では、企画からヒアリングまでに集めた要望の中から、システムで実現すること、そして、実現のために必要な条件を企画内容を大前提として決定します。主な要件定義は、カスタム App の作成作業の開始前に決定します。機能の確認、ユーザテストなどに必要な内容が含まれるからです。土台のようなものであると考えてください。

まず以下を決定します。

・どのような入力や出力の項目が必要か
・業務の流れ中の、どの作業部分で、入力や出力をするのか

3.2.1　なぜ要件にするのか

要望を形にするには『たぶんこんな感じかな……』ではいけません。誰かに当てはまっていても、他の人にとっては違う可能性があります。複数ユーザが 1 つのシステムを使用するためには、企画と同じように共通の「これ」という決定が必要です。そして、具体的な文言として要件を表現できるようにします。各種のテスト時に、定義した要件と合致するかどうか確認するために利用します。

みんなで決める
複数の人が係わっていると、全員が納得する決定はできないかもしれません。『作業者が良いと思う内容でつくってもらって、できあがったものを見て、あとから最終的な決定することにしよう』という意見がでてくるかもしれません。その「最終的な決定」は、できあがったカスタム App に対してどのような修正作業になるのか、この時点では誰もわかりません。期間に影響する可能性があります。なかなか要件が決まらなければ、少しずつでも要件を定義できるよう「プロトタイプモデル」の導入の検討をおこなってください。

3.2.2　規模感

要件定義時に役立つ情報として「規模感」があります。いくつかの角度から規模感を掴むことで、カスタム App ファイルや必要なハードディスクの大きさ、運用開始後に気をつけることなどを計画、決定する際にとても役立ちます。

データ増加数
わかりやすい方法として、使用する人数と 1 日、1 ヶ月、1 年の単位で、どのぐらいのデータが作成されるか確認します。「**データの増加数**」を把握するためです（本書では 1 ヶ月間に業務を行う日数を 20 日として想定しています）。

「見積作成管理システム」では、次のような状況を考えます。

・見積書は 1 ヶ月に最高で 200 件程度である
・印刷される書式を調べたら明細行が 1 件から 10 件程度のものまでである

1 ヶ月に明細行データは 200 件 ×10 行＝ 2,000 件増加する想定になります。1 年間では見積書は 2,400 件、明細行は 24,000 件です。

さらに考えます。

ユーザ数とデータ件数
ユーザ数とデータ件数の関係も考えます、1 日、1 ヶ月、1 年間にどのぐらいの件数を取り扱うのか、によってユーザの「利用頻度」の把握が容易になります。例えば、

1 年間に 3 人で 5,000,000 件を取り扱う
1 ユーザあたり 1 ヶ月 138,900 件／ 1 日に 6,945 件
→ データの増加、ユーザの利用頻度共に多い、ユーザ数が少ない

1 年間に 20 人で 50,000 件を取り扱う
1 ユーザあたり月に 210 件／ 1 日に 10 件
→ データの増加、ユーザの利用頻度、ユーザ数、いずれも多くも少なくもない

1 年間に 100 人で 12,000 件を取り扱う
1 ユーザあたり月に 10 件／ 1 日に 0 から 1 件
→ データの増加は少なめ、ユーザの利用頻度少ない、ユーザ数が多い

極端な例もありますが、取り扱い件数とユーザの関係と特徴を考えてみてください。なお、

・特に作業が集中する日や期間の有無

などの条件も合わせて考えると、さらにわかりやすくなります。「見積作成管理システム」では次のとおりとします。

・ユーザは 10 人程度
・1 ユーザが扱う見積書は 10 ～ 20 件／月（0 ～ 2 件／ 1 日、120 ～ 240 件／ 1 年間）
・作業が集中する日は特に無し、期間としては年度末には集中しがち

さらに確認します。

バイナリデータの有無
文字以外のデータ形式（バイナリデータ）が必要かどうかです。

・取り扱うデータに画像や動画などが必要かどうか

必要な場合は、形式や平均的なファイルサイズも確認します。例えば、データの増加件数は少なくても、大きな動画を取り扱うとシステムの取り扱うデータサイズが大きくなるかもしれません。

これら以外にも、別の方向から規模感を考えることも必要です。機能の数、レイアウト数などです。いくつかの調査結果を含めて全体を把握し、各種の計画や決定事項に役立てます。

3.2.3 項目

構造を設計するために、フィールドとテーブルの元になる「**項目**」を考えます。入力項目と、出力する書式や項目の内容から項目がわかります。「見積作成管理システム」では、次表の項目が必要だとわかりました。

見積作成管理システムの項目
発行日
見積番号
お客様情報（会社名　郵便番号　住所　所属　肩書　名前）
自社情報（会社名　郵便番号　住所　所属　担当者名　電話番号　FAX 番号　電子メール）
見積の総額（￥xx,xxx,xxx.-　の形式）
件名
見積期限
見積の内容（明細：品名　数量　単価　　金額　備考）1 見積書に複数必要
小計（xx,xxx,xxx　の形式）
消費税額（xx,xxx,xxx　の形式　1 円以下は、切り捨て）
見積の状況（作成／お客様検討中／結果確定済）
結果（注文あり／注文なし）

3.3　要件を定義する 2：作業を考える

規模感と必要な項目はわかりました。次は、要望を一覧にし、同じ内容のものは 1 つにまとめるなどして整理します。可能であれば大まかな「**作業**」ごとにわけます。例えば、

・前回の見積書を複製して作成したい　→　見積作成作業
・見積を印刷したい　→　見積出力作業
・番号で検索したい　→　？？？？

このような感じです。最後の要望は、何の番号で、何を検索したいのか、検索後、どうしたいのかがわかりません。さらにヒアリングしたところ、これは

・お客様の電話番号や見積書の管理番号で検索したい

という要望でした。理由は『問い合わせ時の対応などに使用したいから』でした。このように、必要であれば何度かヒアリングを実施し、曖昧な要望を補いながら作業へ切り分けます。

見積作成管理システムのヒアリングした内容	切り分け先の作業
~~見積書をいまより簡単につくりたい　→　具体的要望不明　確認事項あり~~ →　見積番号の自動付与、発行日、期限日の自動入力がしたい	見積作成
前回の見積書を複製して作成できるようにしてほしい →　見積書 1 件ずつで良い	
担当者が外出しているとき他の人が代理で見積作成したい	
見積は訂正することがある →　商品の増減、値引き →　見積番号は新しく付与したい、発行日も新しく付与する	見積訂正
お客様が紛失したときは再発行する →　見積書 1 件ずつで良い →　見積番号、発行日も同じままにしたい	見積再出力
他の人のつくった見積書を見て参考にしたい	見積管理
見積がいまどういう状況なのかわかるようにしたい （作成／お客様検討中／結果確定済）（注文あり／注文なし）	
お客様ごとの売り上げを見たい	
~~見積書の印刷以外に、PDF は作成できるか？~~ →　紙の印刷と同じ書式の PDF を作成する	見積出力
~~見積書を印刷したい　見積書は A4 サイズで印刷する~~ →　見積書を A4 サイズで印刷したい	
見積書は今の書式で問題ない	
見積書と同じ内容の発注書も一緒に印刷できると良い	
お客様にメールで見積書を送信したい	見積提示
~~番号で検索したい~~ →　お客様の電話番号や見積書の管理番号で検索したい	~~作業対象不明　確認事項あり~~ 見積検索
お客様ごとに自社の担当者がいる	その他　→　見積作成時に自動入力
お客様ごとの注文実績を見たい （件数と金額）（毎月、月初に前月分 1 日から末日分を集計する） （注文受付日を対象とする）（消費税抜きの金額の合計）	見積集計

3.3.1 業務の流れのどこか

大まかに切り分けた作業が実行されるのは、業務の流れのどこでしょうか。さらに、作業にカスタム App が必要かどうかも考えます。これで、カスタム App の持つべき役割が段々わかってきます。

(1) いつ → 業務フローのどこか
(2) どこで → 場所（部署や、社内か社外か、など）
(3) 誰が → ユーザ
(4) なにを → 対象のもの
(5) どのように → 作業
(6) どうやって → カスタム App かどうか

業務の流れに作業を加え、カスタム App が必要かどうかも加えました。

見積業務（カスタム App 導入後）の流れ	作業名	カスタム App	場所	ユーザ
お客様から連絡がくる（開始）		×	基本的に作業は社内の営業部門で実施する	基本的に作業は営業部門の人員が実施する
見積書を作成する（担当者名で）	見積作成	○		
印刷する（印刷または PDF にする）	見積出力	○		
上司が承認する	承認	×		
お客様に見積書を提示する （持っていく 　または　郵送（窓あき封筒） 　または　PDF をメール添付送信）	お客様に提示 　見積検索 　見積提示	○ メールの場合		
お客様が検討する	注文有無伺い	×		
	問い合わせ対応 　見積検索 　見積訂正＆出力 　見積再出力	○		
（結果がでる） 注文をもらう （FAX　または　電話　または　メール） または 注文されない		×		
結果を記録する （注文されなければここで完了）	結果記録 　見積検索 　見積管理	○		
注文をもらった場合、 注文を処理する部署に連携 （完了）	注文受付	×		

次のことがわかるようになりました。

・業務の流れがどうなるか
・業務の流れには、どのような作業があるか
・カスタム App がどこで必要か、どの作業をサポートするのか

業務フローをアップデートしました。そして、**作業から機能を考えていきます。**

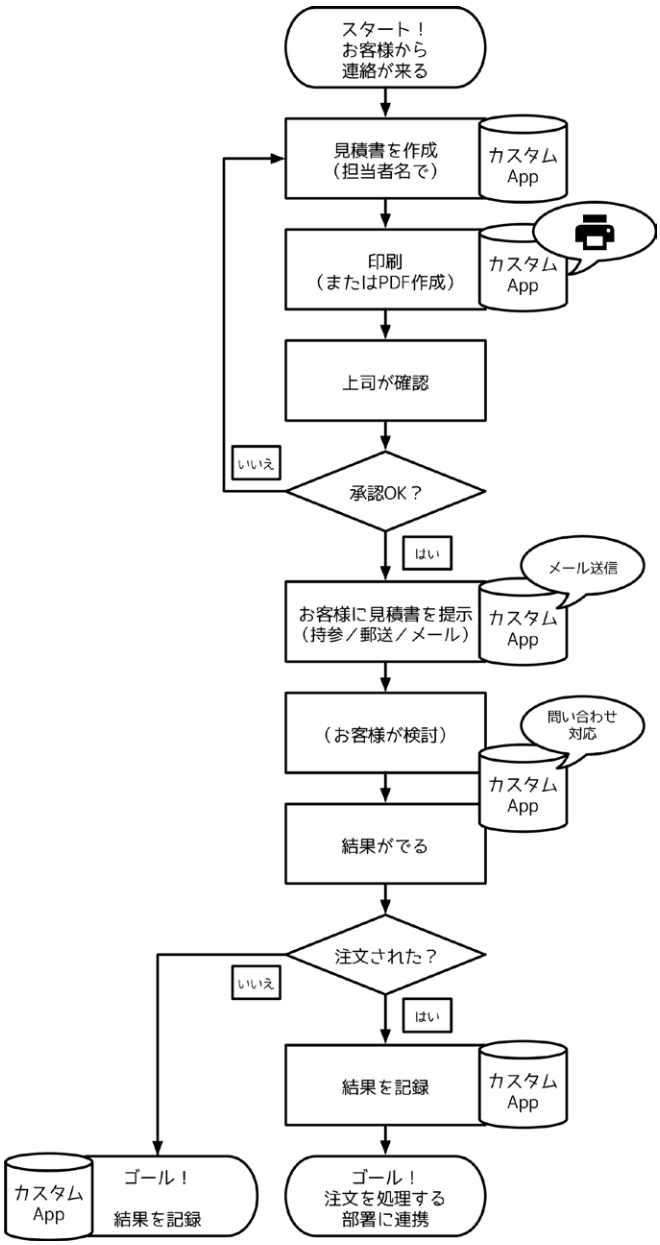


図 3-3 業務のフローチャートにシステムの必要部分を追記した例

3.4　要件を定義する 3：機能を考える

カスタム App が業務をサポートするために必要な機能は、どのようなものでしょうか。これまでの内容から、次表に「見積作成管理システム」の機能を記述しました。参照してください。

見積業務（カスタム App 導入後）	作業名	カスタム App	主要な機能と要件
お客様から連絡がくる（開始）		×	
見積書を作成する （担当者名で）	見積作成	○	・新しい見積データを作成する （見積の作成は月に 200 件程度） ・一部の見積項目を自動入力する （見積番号、発行日、期限日、担当名） （見積状況は　作成） ・既存の見積データを検索し指定の 1 件を複製し新しい見積データ作成する （検索は　見積検索　を参照）
印刷する （または PDF にする）	見積出力	○	・指定の 1 件の見積データを印刷 ・指定の 1 件の見積データを PDF 化 （用紙サイズは A4 縦）
上司が承認する	承認	×	
	状況記録 　見積検索 　見積管理	○	・項目に検索条件を入力して見積データを検索する （お客様電話番号、お客様名、見積管理番号など） ・指定の 1 件の見積データに状況を入力する （見積状況は　お客様検討中）
お客様に見積書を提示する （持っていく または　郵送（窓あき封筒） または　PDF をメール添付送信）	お客様に提示 　見積検索 　見積提示	○ （メールの場合）	・項目に検索条件を入力して見積データを検索する ・指定した 1 件の見積データの PDF を添付しメールを送信する
お客様が検討する	注文有無伺い	×	
	問い合わせ対応 　見積検索 　見積訂正＆出力 　見積再出力	○	・項目に検索条件を入力して見積データを検索する ・指定の 1 件の見積データを訂正する （商品の増減、値引き） （新しいデータが必要） ・指定の 1 件の見積データ検索して出力する （出力は　見積出力を参照）
（結果がでる） 注文をもらう （FAX　または　電話　または　メール） または　注文されない		×	・注文の連絡は 1 日に 0-10 件程度
結果を記録する （注文されなければここで完了）	結果記録 　見積検索 　見積管理	○	・項目に検索条件を入力して見積データを検索する ・指定の 1 件の見積データに結果を入力する
注文をもらった場合、 注文を処理する部署に連携 （完了）	注文受付	×	

3.4.1 作業を細分化する

前表のように、作業から機能を考えるには、例えば、

・見積作成作業

が、機能として、さらに細分化できるかどうかを考えます。例えば、このようになります。

・新しい見積データ作成
・自動入力
・既存見積の複製による新しい見積データ作成

カスタム App をいくつも作成するうちに、経験からだんだん思いつくようになります。具体的な図や文言で記述して要件定義の決定時に関係する人々が共通のイメージを持つことを目指します。

なお、前表には「見積作成管理システム」のヒアリング内容で含まれていないものがあります。

・見積集計　→　お客様ごとの注文実績を見たい

です。これは通常の見積業務ではなく、日々の業務の結果、入力されたデータを使う作業です。別にしておきます。

集計（カスタム App 導入後）	作業名	カスタム App	主要な機能と要件
お客様ごとの売り上げを確認する	見積集計	◯	・お客様ごとに注文をもらった件数と金額を集計して一覧で表示する （前月分 1 日から末日の注文受付日が対象） （金額に消費税を含めない） （用紙サイズは A4 縦）

その他にも、ヒアリング時に今回の対象外とした要望がありました。将来の検討事項として記述します。

・営業部門が見積時に在庫を確認できるようにしたい
・外出先からカスタム App にアクセスしたい

優先順位を考える

要望の数が思ったよりも多いと感じるならば、企画の目的（解決するべき問題や課題）にとって、どれぐらい重要であるかを考えて優先順位をつけ、要件にするかどうかを決定します。

3.4.2　機能を定義する

作業を細分化した内容から機能を定義するには、どのようにすれば良いのでしょうか。例えば、

・新しい見積データを作成する

であれば、

(1) いつ　　→　お客様から見積書の提示を求められたら
(2) どこで　　→　企業内で／カスタム App で／見積を作成できる画面で
(3) 誰が　　→　通常使用ユーザが
(4) なにを　　→　見積書を
(5) どのように →　1 件ずつ（1 件には、複数の品名が含まれる／一部項目自動入力　など）
(6) どうやって →　新しく作成する
(7) 何故　　→　お客様に提示するため
(8) 備考　　→　自動入力は、発行日、品名ごとの単価、計、小計、消費税額、見積額

のように分解していきます。

ヒアリングで受け取る要望の内容は、人の気持ちを伝えるための言葉です。その言葉を機能とするために細かい要件を確認していかなければなりません。

「どこで」と画面
機能を実行する場所としての「どこで」を考えると、最終的には「画面」になることがほとんどです。どのような画面にどのような機能が必要になるかを考えるのに役立ちます。

機能の一覧
定義した機能について、一覧にしておくと作業時に便利です。

並行して進む
要件を定義していくと、データ項目からテーブルが想像できますし、作業内容、機能定義からは画面が思い浮かびます。これらは次章以降で解説します。常に 1 つしか工程を進めてはいけないわけではありません。どんどん設計につなげていきます。

3.4.3　定義内容の確認

定義した機能の内容を、新しいシステムに関係する人で検討します。ヒアリングの要望が機能として実現されたかどうかを複数の目で確認してもらうためです。プロトタイプを導入することも効果的です。

なお、すべての人が必要な機能の要望を、余すところなく完璧に伝えられることはほとんどありません。機能を一覧で提示し、説明すると、『要望を訂正したい』『不要だった』『他にも必要なことがあったけれど忘れていた』といった意見がでることがあります。

要件定義の段階では、なるべく柔軟に取り入れます。その結果、機能が大きくなったり増えすぎたら次期の開発に振り分けるなどの決定をすれば良いのです。

3.5　要件を定義する 4：その他

主要なことがわかってきました。さらに、業務で使用する場合や、複数の人で使用する場合に必要なことを考えます。

3.5.1　セキュリティ

どのようなセキュリティが必要か考えます。

使う人とレベル

共有を前提とするカスタム App では、誰でも使用できると、

・データを作成したのは誰か
・データを最後に修正したのは誰か

などを正しく記録できません。ですから、

・ログインして使用できるのは誰か

を決めておきます。企画時に対象ユーザを決定しました。その内容で良いか確認し、セキュリティのレベル（区分）がいくつあるべきかを考えます。基本は 2 種類です。次のような内容です。

・普段の作業をする
・システムで使用する共通のデータを管理する

必要に応じてレベルを増やします。

通信

接続（通信）の暗号化を考えます。原則として、暗号化することをお勧めします。通信の暗号化は本書では取り扱いません。本書の上級編やヘルプを参照してください。

ファイル

カスタム App ファイルを暗号化するかどうかを考えます。

※ FileMaker Cloud で共有する場合はファイルの暗号化は必須です。

セキュリティについて、「見積作成管理システム」では次表とします。

セキュリティ		要件
アカウント	共通データの管理	作業に応じて必要なアカウント名とパスワードでログインする
	通常使用	
通信	暗号化する	SSL 暗号化通信
ファイルの暗号化	暗号化しない	社内でのみ使用する想定のため

3.5.2　現在のデータはどうするか

これまでのデータをどうするかも決めます。データを新しいシステム（本書ではカスタム App）へ移すことを「**データ移行**」と呼びます。本書では詳細に解説しませんが、いくつかのポイントだけ提示します。

移行対象
それまでのデータのうち、どのデータを移行するかを決めます。

例えば、各自がばらばらのアプリケーションのファイルを使って作業をしていたデータの場合、移行はとても大変です。対象として集められたデータが全く同じ項目から構成されていることはあまりないからです。既存の各項目を新しいカスタム App の項目やテーブルにあてはめて（マッピング）からデータを移行しなければなりません。

データの種類も考慮します。

マスタ的なデータ
例えば、お客様情報や商品情報は、移行する必要があることがほとんどです。それらの情報があらかじめ入力されていないと通常の作業（例えば、見積書作成）ができないからです。

イベント的なデータ
例えば、これまでの見積データは移行しないか、一部分だけ移行する決定をすることがあります。これは、次のようなことがあるからです。
・過去データは利用する必要がない
・過去データの量が多すぎる
・データ化されていないため再入力する必要がある

移行手段
入力方法です。手入力にするか、データインポートにするか、別のシステムとデータ連携するかの、いずれかです。

カスタム App への影響
データ移行を実施する場合、とくにイベントデータの作成日や修正日で動作する機能について注意が必要です。移行時に過去の日付にするなどの対応を検討してください。

データ移行は、移行の有無、移行データの範囲、項目の検討、データの収集、準備、移行計画作成、実行……と、それなりの時間がかかります。充分な期間を設けて実施します。

「見積作成管理システム」では、次表とします。

移行候補データ	移行対象	主要な機能と要件
見積データ	移行しない	
お客様情報	移行する	・A さんのまとめたお客様情報を、実際に業務で使用を開始する直前に移行する
商品情報	移行する	・A さんのまとめた商品情報を、実際に業務で使用を開始する直前に移行する

3.5.3 他のシステムは関係あるか

企画の段階で『他システムと統合しない』と決めていたとしても再度確認します。例えば、「見積作成管理システム」では、

・注文をもらった場合、注文を処理する部署に連携

とあります。

現在は、『受注した見積書、あるいは発注書を注文を処理する部署に手渡しするだけ』とします。しかし、データをエクスポートすると連携先の部署で役立つかもしれません。提案すると新たな機能として検討対象になるかもしれません。

当然ですが、他システムと連携する場合の打ち合わせには注文を処理する部署の人も呼ぶ必要があります。他の部門や部署の業務フローの変更を伴うため関係者が揃っていないといけません。

「見積作成管理システム」では、次のようにします。

・注文を処理する部署のためのデータ出力は、今回は実施しない
・将来はエクスポートによる連携を検討する

3.5.4 環境

企画で考えたカスタム App の使用環境などについても確認して決定します。パソコンのスペックが大丈夫かどうか、ネットワークは想定したように使えるか、必要なソフトウェアは足りているか、などです。

・ハードウェア（サーバー機、クライアントパソコン、プリンタ）
・ソフトウェア
・ネットワーク

3.6　要件定義を合意する

企画と同じく、必要な合意をします。

3.6.1　合意

要件も『必要な人々の間で合意する』ということになります。企画と同じく企業や業務により事情が異なるため、確実な答えはありません。

問題は解決できそうか

要件定義では、あれがしたいこれがしたいと機能に集中しすぎるあまりに、企画の問題や課題の解決という目的の達成についての考慮が曖昧になることがあります。決定したことで問題や課題が解決できるかどうか最終的な確認を行ってください。

企画したことが変わることも

要件定義をしているうちに、企画で決めたことが変わることがあります。変えても良いのです。気をつけることは、変えた影響を考えずに進めてしまうことです。企画内容が変わればヒアリングがやり直しになったり全体の作業期間にも影響があるかもしれません。

3.6.2　書き残す

要件定義で決定した内容も文字で書き表して、システムに関係する誰もが見られるようにします。書類で配布しても、PDF でどこかに保存しても、企業で使用している wiki サービス上のページでも良いのです。自分だけが使うのであれば、このようなことは大げさかもしれませんが、業務で複数の人が使用する予定のシステムは情報を公開することも重要です。

3.6.3　期間

要望が多岐に渡っていて、しかも量が多い場合、全部を 1 回で実現しなければならないわけではありません。優先順位をつけてどこまでやるかを検討します。1 回ですべてを実現することだけが問題や課題の解決になると考えてしまうと、カスタム App の完成に長い期間がかかってしまうかもしれません。

3.6.4　さらに知りたいなら

要件定義についてはたくさんの書籍が発行されています。方法論、どこまで詳細に決めるか、どのようなまとめを作成するかなど自分に合った方法を模索してください。もっと詳細なところまで確認していきますが、ここでは省略します。

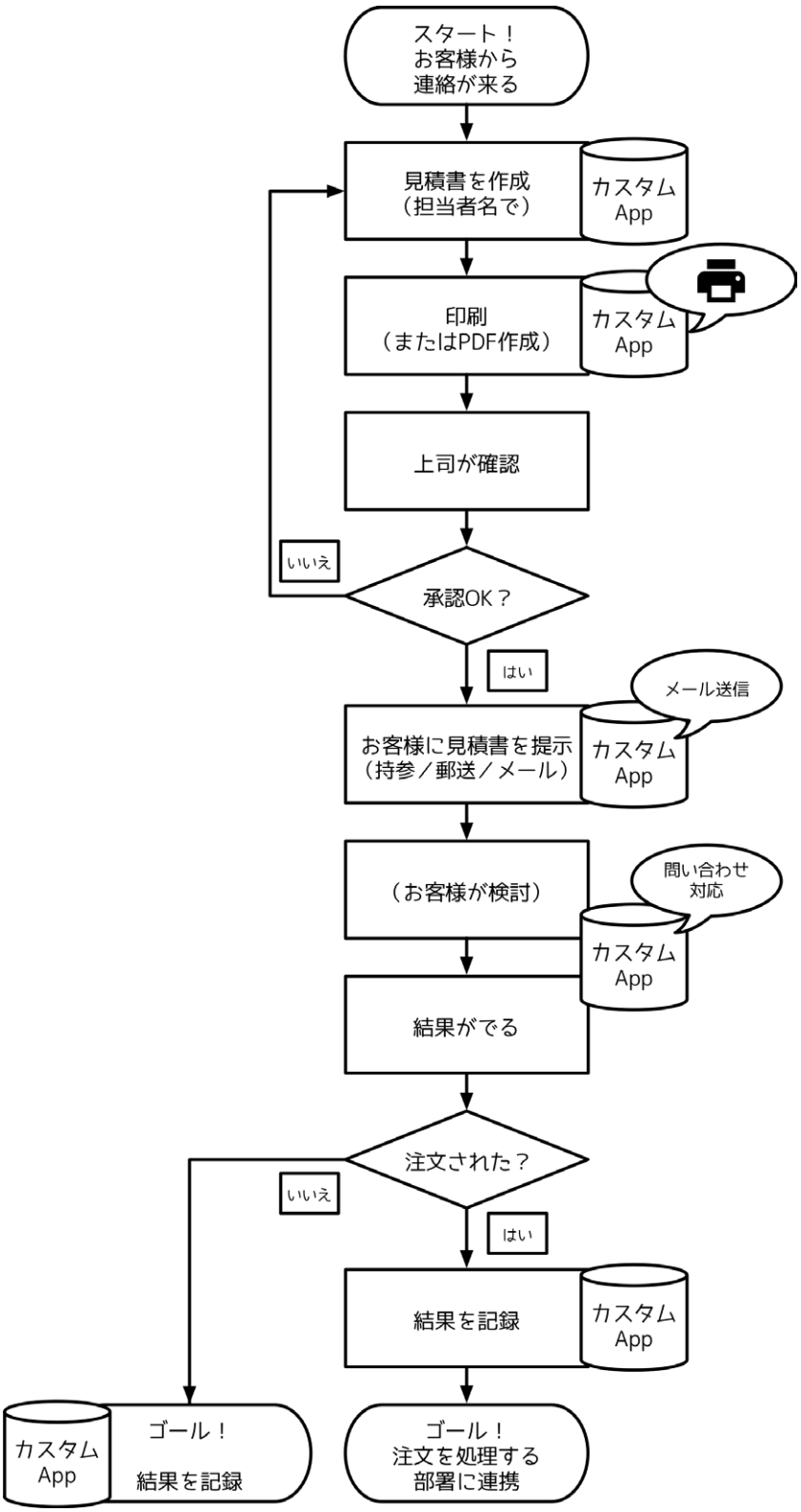
「見積作成管理システム」は、次ページからの図表の内容で決定したこととします。

要件の概要	項目	内容
つくるもの	見積作成管理システム	FileMaker Pro で開発されたカスタム App を中心とする
計画	作業期間	○ヶ月（YYYY 年 M 月リリース予定）
	使用開始	リリース後○ヶ月（データ移行と試用テスト期間）
目的	情報の一元化と活用	全員が見積書を同じデータベースで作成することで見積内容や 注文されたかどうかなどを活用できるようになる
対象	対象部門	営業部門
	対象業務	見積業務
	人＝ユーザ	営業部門の人員（10 人程度）
	場所	社内での使用　※将来検討あり
	時間	営業時間を中心にする
ユーザ	使用方法	作業に応じて必要なアカウントとパスワードでログインする
	アカウント	1 人に 1 つ（種類によって操作できる機能が違う）
	アカウント種類	共通データの管理（アカウントの管理も含む）
		通常使用
セキュリティ	通信	暗号化（SSL 暗号化通信）
	ファイルの暗号化	暗号化しない
データ	件数	見積作成は 200 件程度／ 1 ヶ月（1 見積あたりの明細行は 10 行程度） 注文は 0-10 件／ 1 日
	お客様数	現在は 50 件
	商品数	現在は 100 件
出力書式	見積書	A4 縦、1 枚（書式は別紙参照）
	発注書	A4 縦、1 枚（書式は別紙参照）
	お客様ごと集計結果	A4 縦、複数枚（新書式なので○月までに決定）
アプリケーション	サーバー	FileMaker Server ライセンスあり。購入必要なし
	クライアント	FileMaker Pro　ライセンスあり。購入必要なし （FileMaker Go の利用も検討中）
ハードウェア	サーバー機	以下のどちらかを検討中 ・実機を新規購入（機器はアプリケーションの推奨環境を元に機器 　購入部門と検討予定）でサーバー室に設置予定 ・ホスティングか FileMaker Cloud を新規契約
	クライアントパソコン	現在のものをそのまま使用可能
ネットワーク	社内の LAN	現在のものをそのまま使用可能
展開（共有）	あり	検討中　FileMaker Cloud または FileMaker Server
結合（連携）	なし	※将来検討あり
データ移行	見積データ	移行しない
	お客様情報	移行する A さんがまとめて実際に業務で使用する前に移行する
	商品情報	
その他	将来検討など	将来、注文を処理する部署への注文テキストデータ出力を検討する
		倉庫の在庫管理システムが欲しいという意見があった
		将来、モバイルデバイスで社外から接続し使用を検討する

カスタム App 管理作業	作業名	カスタム App	主要な機能と要件
共通のデータの管理	システム管理	○	・共通データの管理ユーザが、共通データの 新規作成、編集、削除をおこなう

集計（カスタム App 導入後）	作業名	カスタム App	主要な機能と要件
お客様ごとの売り上げを確認する	見積集計	○	・お客様ごとに注文をもらった件数と 金額を集計して一覧で表示する （前月分 1 日から末日の注文受付日が対象） （消費税を含めない）（用紙サイズは A4 縦）

見積業務（カスタム App 導入後）	作業名	カスタム App	主要な機能と要件
お客様から連絡がくる（開始）		×	
見積書を作成する （担当者名で）	見積作成	◯	・新しい見積データを作成する （見積の作成は月に 200 件程度） ・一部の見積項目を自動入力する （見積番号、発行日、期限日、担当名） （見積状況は　作成） ・既存の見積データを検索し指定の 1 件を複製し新しい見積データ作成する （検索は　見積検索　を参照）
印刷する （または PDF にする）	見積出力	◯	・指定の 1 件の見積データを印刷 ・指定の 1 件の見積データを PDF 化 （用紙サイズは A4 縦）
上司が承認する	承認	×	
	状況記録 見積検索 見積管理	◯	・項目に検索条件を入力して見積データを検索する （お客様電話番号、お客様名、見積管理番号など） ・指定の 1 件の見積データに状況を入力する （見積状況は　お客様検討中）
お客様に見積書を提示する （持っていく または　郵送（窓あき封筒） または　PDF をメール添付送信）	お客様に提示 見積検索 見積提示	◯ （メールの場合）	・項目に検索条件を入力して見積データを検索する ・指定した 1 件の見積データの PDF を添付しメールを送信する
お客様が検討する	注文有無伺い	×	
	問い合わせ対応 見積検索 見積訂正＆出力 見積再出力	◯	・項目に検索条件を入力して見積データを検索する ・指定の 1 件の見積データを訂正する （商品の増減、値引き） （新しいデータが必要） ・指定の 1 件の見積データ検索して出力する （出力は　見積出力を参照）
（結果がでる） 注文をもらう （FAX　または　電話　または　メール） または　注文されない		×	・注文の連絡は 1 日に 0-10 件程度
結果を記録する （注文されなければここで完了）	結果記録 見積検索 見積管理	◯	・項目に検索条件を入力して見積データを検索する ・指定の 1 件の見積データに結果を入力する
注文をもらった場合、 注文を処理する部署に連携 （完了）	注文受付	×	


スタート！
お客様から
連絡が来る
見積書を作成
（担当者名で）
カスタム
App
印刷
（またはPDF作成）
カスタム
App
上司が確認
いいえ
承認OK？
はい
メール送信
お客様に見積書を提示
（持参／郵送／メール）
カスタム
App
（お客様が検討）
問い合わせ
対応
カスタム
App
結果がでる
注文された？
いいえ
はい
結果を記録
カスタム
App
カスタム
App
ゴール！
結果を記録
ゴール！
注文を処理する
部署に連携

見積作成管理　システム機能一覧		
作業名	**機能名**	**条件**
見積作成	新規見積作成	いつ　→　見積作成作業でお客様から希望があったら どこで　→　見積を作成できる画面で 誰が　→　通常使用ユーザが なにを　→　見積書を どのように　→　1 件ずつ（複数の品名が含まれる／一部項目自動入力　など） どうやって　→　新しく作成する 何故　→　お客様に提示するため 備考　自動入力項目は、発行日、品名ごとの単価、計、小計、消費税額、見積額
以下省略		

見積作成管理システム　項目一覧
見積項目
発行日
見積番号
お客様情報（お客様番号　会社名　郵便番号　住所　所属　肩書　名前）
自社情報（会社名　郵便番号　住所　所属　担当者名　電話番号　FAX 番号　電子メール）
見積の総額（￥xx,xxx,xxx.-　の形式）
件名
見積期限
見積の内容（明細：商品番号　品名　数量　単価　　金額　備考）1 見積書に複数必要
小計（xx,xxx,xxx　の形式）
消費税額（xx,xxx,xxx　の形式　1 円以下は、切り捨て）
見積の状況（作成／お客様検討中／結果確定済）
結果（注文あり／注文なし）
以下、お客様情報、商品情報の項目については詳細は省略

発行日　見積番号

見積書

○○株式会社
△△部門　□□　様

☆☆株式会社
営業部門　担当：◎◎

下記のとおりお見積りいたします。

お見積金額　￥xx,xxx,xxx.-（消費税込）
件名　zzzzzzzzzzzzzzzzzzzz
発注期限　yyyy 年m月ｄ日

No.	品名	数量	単価	金額	備考
			小計		
			消費税		

発行日　見積番号

発注書

発注日　年　月　日
☆☆株式会社　行
（担当：◎◎）

○○株式会社
△△部門　□□　様

下記のとおり発注します。

お見積金額　￥xx,xxx,xxx.-（消費税込）
件名　zzzzzzzzzzzzzzzzzzzz
発注期限　yyyy 年m月ｄ日

No.	品名	数量	単価	金額	備考
			小計		
			消費税		

第 4 章 モデリングと正規化

企画と要件定義を元に設計をはじめます。「構造の設計」について解説します。

概要と必要なものを決定しました。その内容を元にしてカスタム App の構造を設計します。なお、本章では「カスタム App」ではなく「データベース」と記述します。モデリングや正規化の作業は一般的なデータベースの設計でも実施することだからです。

4.1 構造設計

要件定義は土台でしたが、設計は土台の上の基礎部分と言えます。

・データモデリング
・テーブル設計

これらを、次のステップで行います。

・エンティティを探す
・エンティティの関連付けをする
・エンティティを正規化する

できあがった設計に従って FileMaker Pro でファイルにテーブルを作成し、リレーションシップグラフ、コンテキストを生み出すことで、レイアウト越しにデータをやりとりすることが可能になります。

設計はとても重要です。データベースにおけるデータ取り扱いの基本ルールだからです。

4.2 データモデリング

データモデリングとは、データを整理整頓してデータベースで利用できるように「**概念的な構造(スキーマ)**」を決めることです。

データベースはデータを取り扱っていますが、通常、ユーザは、データの全体像について考えたり、確認することはありません。画面や出力結果を通し、部分的なデータが確認できるだけで充分だからです。一方、開発者は、データベースで取り扱う必要があるすべてのデータが、うまく管理され、入力と出力がされるように全体を把握する必要があります。データモデリングにより全体を把握することが可能です。

4.2.1 Entitiy Relationship モデル

本章では、データを次のように考えて設計します。

・**実体＝** Entity **＝エンティティ**に分ける
・**エンティティ同士の　関連＝** Relationship **＝リレーションシップ**を考える

この関係を図示することを、本書では次の手法を利用します。

・**ER 図**（Entity Relationship Diagram）

ER 図は、直訳すると「実体関連図」です。ここでの「**実体**」は、テーブルのことを意味しています。データベースに必要なデータと、その関連を図示したものです。リレーションシップグラフを作成するための設計図と考えてください。リレーションシップグラフと似ていますが違います（FileMaker Pro では、設計内容そのものを定義する場所はありません)。

ER 図は書き方のルールが複数あります。本書の書き方は 1 つの例です。興味があれば調べてみてください。

手書きで
最初は紙に書いて気軽に作業する方が発想が自由にできます。エンティティを表現するのに付箋を活用しても良いかもしれません。ER 図を描く専用のアプリケーションもありますので、好きな道具を使ってください。

4.2.2 エンティティを探す

データモデリングを開始します。まず、要件定義で集まった入出力が必要なデータ項目から「**エンティティ**」を探し出します。次のように考えてください。

・同じ分類ができるような情報の集まり

「見積作成管理システム」の項目を次表にまとめました（これまでより詳細な記述になっています)。ここからデータのエンティティを見つけるために、どの項目が同じ分類にできるのか考えます。はじめは、だいたい直感的に考えます。

「見積作成管理システム」の、すべての項目
発行日
見積番号
お客様　お客様番号
お客様　会社名
お客様　郵便番号
お客様　住所
お客様　所属
お客様　肩書
お客様　名前
自社　会社名
自社　郵便番号
自社　住所
自社　所属
自社　担当者名
自社　電話番号
自社　FAX 番号
自社　電子メール
見積の総額（￥xx,xxx,xxx.-　の形式）（消費税込み）
件名
見積期限
見積の内容（明細）　商品番号　1 見積書に複数必要
見積の内容（明細）　商品名　　1 見積書に複数必要
見積の内容（明細）　商品単価　1 見積書に複数必要（消費税含まず）
見積の内容（明細）　商品金額　1 見積書に複数必要
見積の内容（明細）　商品数量　1 見積書に複数必要
見積の内容（明細）　商品備考　1 見積書に複数必要
小計（xx,xxx,xxx　の形式）
消費税額（xx,xxx,xxx　の形式　1 円以下は、切り捨て）
備考
見積の状況（作成／お客様検討中／結果確定済）
結果（注文あり／注文なし）

上表の項目は、次の 4 つの情報の集まりであるとわかります。あまり意識しなくても見分けることができるはずです。人は、現実にある物事（実体があってもなくても）を共通の分類を持つ「属性」として認識できるからです。

・見積書
・お客様
・自社情報
・商品

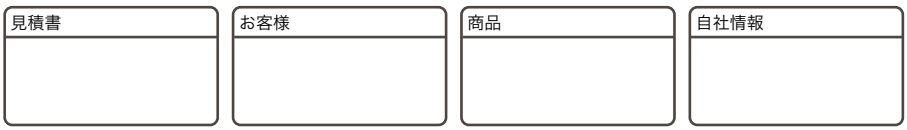


図 4-1 4 つのエンティティ

属性

エンティティに含まれる項目の名前のようなものです。以下はすべて「果物」という属性になり果物エンティティとして ER 図を描きます。

・リンゴ
・バナナ
・ミカン

4.2.3 エンティティの種類

エンティティは、そのデータの性格から大きく 2 種類に分けられます。イベントとマスタです。

イベントエンティティ

イベント＝出来事です。『○○する』という文脈にあてはまります。例えば、『見積りする』と表現できるものです。このようなデータを「イベント」「トランザクション」などと呼びます。なお、日付項目が必要なことがほとんどです。本書では「**イベント**」エンティティと記述します。

「見積書」は、毎日の業務で発生するデータです。イベントエンティティです。そして発行日などの日付項目が必要です。ただし、依頼されたときだけ発生するものです。依頼する「お客様」、見積る内容である「商品」、差出人としての「自社情報」が必要です。

マスタ（リソース）エンティティ

リソース＝資源です。『○○する』ではありません。このようなデータは「静的」「マスタ」などと呼びます。本書では「**マスタ**」エンティティと記述します。

「お客様」「商品」「自社情報」はマスタエンティティです。「見積書」の有無に関わらず存在します。

「見積作成管理システム」のエンティティの関係は、次図です。

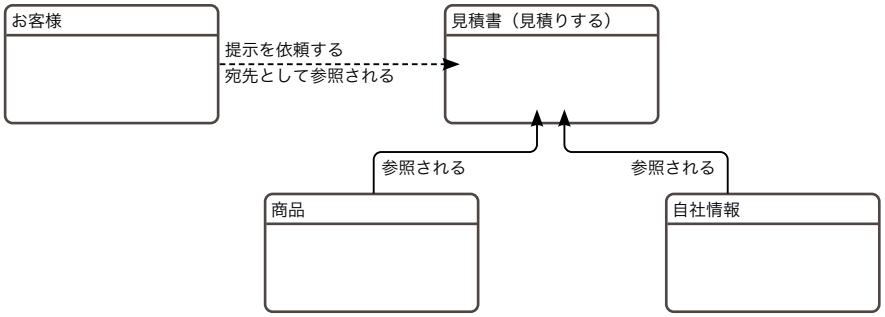


図 4-2 エンティティの関係

ER 図に書くときイベントとマスタを色分けすると、わかりやすくなります。次図のようになります。

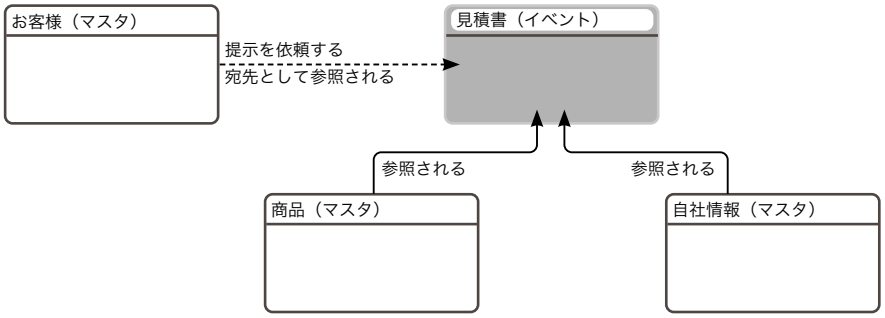


図 4-3 色をつけてエンティティの種類をわかりやすくする

この段階では、まだ ER 図ではありません。「**関連**」がわかりません。関連は、あるエンティティの 1 レコードに対する他のエンティティのレコード数について図示することが必要です。

4.2.4 関連の種類

エンティティの関係はわかりましたが、関連を設計しなければなりません。後述する「**主キー**」「**外部キー**」に関わるからです。主に 3 種類の関連があります。

・1 対多（親子）
・1 対 1
・多対多

4.2.5 1 対多

関連では最も多く設計されるものです。ポイントとしては、必ず 1 対多（2 つ以上）でなければいけないわけではありません。1 対 1 の状態も含まれます。次の例示では 2 つのエンティティのどちらが 1 でしょうか。わかりやすい方法は双方向で考えることです。

＜お客様と見積書＞
・1 つのお客様は、見積書を何度でも依頼をすることができる
・見積書は、1 件につき 1 つのお客様に対して作成する

＜商品と見積書＞
・1 つの商品は、見積書から何度でも参照されることができる
・見積書は、1 件につき複数の商品に対して作成する

＜自社情報と見積書＞
・1 つの自社情報は、見積書から何度でも参照されることができる
・見積書は、1 件につき 1 つの自社情報に対して作成する

どのエンティティの組み合わせが 1 対多になるかわかりましたか？ 1 対多は、このような線であらわします。



図 4-4 1 対多の線

先ほどの例示を図示すると次図のようになります。

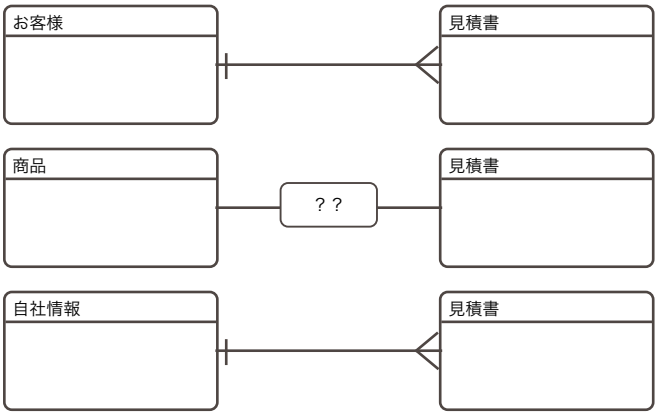


図 4-5 お客様、商品、自社情報と、見積書の関連

商品と見積書の関連は、1 対多ではありません。それは後ほど解説します。1 対多は、「親子」とも呼ばれます。そして、必ずしもイベントエンティティとマスタエンティティの関連ではありません。



図 4-6 親子　1 対多

業務などにおける 1 対多はたくさん存在します。いくつか例を挙げます。

・会社と従業員
・人と住所（家、会社など）
・従業員と資格
・顧客と請求書
・クラスと生徒

どちらのエンティティの立場からみるか、の違いはありますが、「1 対多」と「多対 1」は、原則として同じ関連性です。

4.2.6　多対多

これも多く設計されます。実は、「見積書」と「商品」は多対多です。

・商品は、見積書から何度でも参照されることができる
・見積書は、1 件につき複数の商品に対して作成する

次図の線であらわします。そして、必ずしもイベントエンティティとマスタエンティティの関連ではありません。



図 4-7 多対多の線

「見積作成管理システム」の ER 図は次図になります。

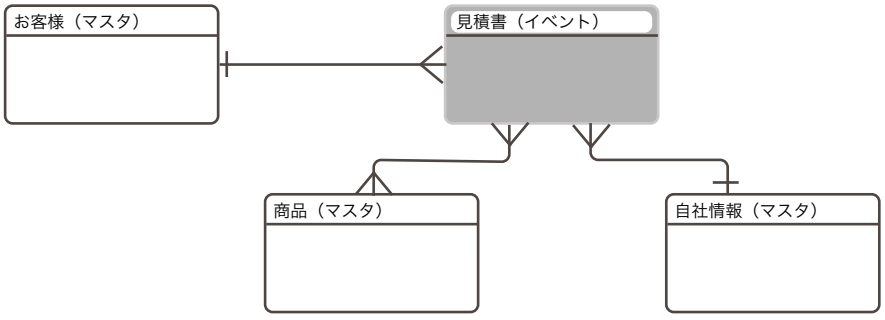


図 4-8 残りの関連が図示　ER 図

他の多対多は、考え方も一緒に例示します。

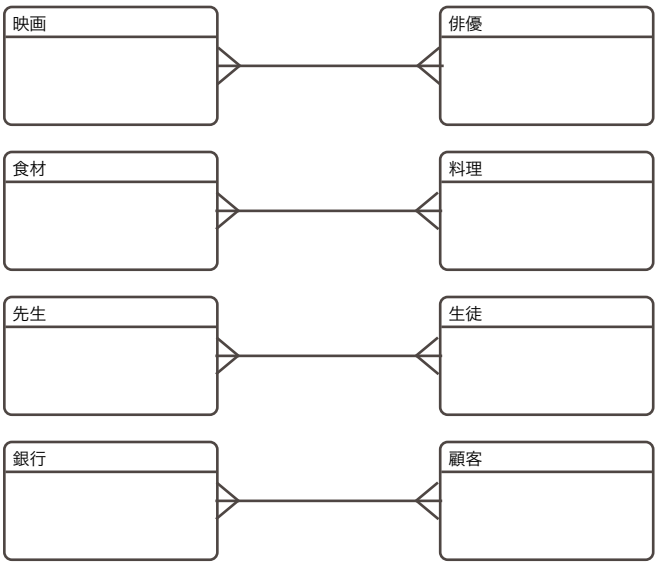


図 4-9 様々な多対多

＜映画と俳優＞
・俳優は何度も映画に出演することができる
・映画は、1 つに複数の俳優が出演する

＜料理と食材＞
・食材はいろんな料理に使われる
・料理は、1 回に複数の食材を使用する

＜先生と生徒＞
・先生は、複数の生徒（クラス）に授業する
・生徒（クラス）は、複数の先生の授業を受ける

＜銀行と顧客＞
・銀行は複数の顧客（口座）を持つ
・顧客は複数の銀行（口座）を持つことができる

多対多は、ER 図で図示された 2 つのエンティティをそのまま 2 つのテーブルとしてデータベース上に関連性を作成することはできません。本章の後半で考え方を記述します。

4.2.7 1 対 1

特別な関連になります。1 対 1 にしかならない関係は 1 つのエンティティにまとめるのが普通です。しかし、特別な事情があって 1 対 1 が必要になることがあります。例えば

・1 人と国の発行するマイナンバー

この関係を 2 つのエンティティで表そうとすると、1 対 1 です。1 対 1 は次の線で表します。



図 4-10 1 対 1 の線

そしてこうなります。



図 4-11 1 対 1 の例

他にも、複数のシステムが連携する場合、例えば、A システムの顧客マスタと B システムの会員情報を関連付けるには、1 対 1 を選択することがあります。

4.2.8 他のシステムとの関係

エンティティ同士の関連で、慎重に考えなければならないことがあります。他のシステムとの統合です。すでに存在している他のカスタム App や、他のデータベースシステムのデータを利用する場合です。

例えば、

・企業の基幹システムが管理している顧客情報
・病院のカルテシステムにある患者情報

のようなデータを、

・一部あるいはすべてを出力して定期的にインポートして使用する
・ESS（外部 SQL ソース）などで結合して直接使用する

などの場合です。ほとんどの場合、他のシステムの設計を変更することは必要ありません。そのまま利用することを考えます。連携するテーブルの主キー（あるいは外部キー）を見つけ、ER 図に加えます。例示したような顧客や患者のテーブルは、1 対 1 で関連することが多いはずです。

「見積作成管理システム」のエンティティには、いまは、1 対 1 の関連はなさそうです。

ここまでで、大まかなエンティティの ER 図ができあがりました。

次は、**項目を「正規化」してテーブル設計を行います。**

4.3 正規化

エンティティと関連が設計できました。その概念を念頭に置いて、次は、項目からテーブルを正規化していきます。モデリング前にある程度、正規化しても良いのですが、本書では大まかな概念から掴んで欲しかったので、あまり正規化せず大きな括りのエンティティ同士の関連を先に考えました。

4.3.1 正規化とは

簡単に言えば、項目を一定のルールに従って整理整頓し、必要に応じてひとかたまりに分け、それらを関連付けることです。この作業の目的は、データを効率良く管理できるようにすることです。

大原則があります。この原則に従うと、わかりやすく作業できます。

・One Fact In One Place（1 つの場所に 1 つの事実）

エンティティを考えたとき「属性」という文言を使いました。そして『属性は、エンティティに含まれる項目の名のようなもの』と記述しました。ほとんどの場合、エンティティには同じ属性を持った複数の項目が含まれています。その項目は、ほぼそのまま「**フィールド**」になります。

まず、これまでに見つけた各エンティティに含まれる項目に、名前をつけます。難しく考える必要はありません。要件定義時にだいたいわかっています。名前は、そのままフィールド名にできます。

「見積作成管理システム」には、次表の項目がありました。すでに大きな括りのエンティティは探し出してありますが、改めてすべての項目から、エンティティを正規化のルールに従って見つけ出して関連を設計をします。

ここでは、3 段階で正規化します。各段階では次のルールに従って正規化を進めていきます。

＜はじめの正規化＞
・繰り返す属性は別のエンティティにする

＜ 2 番目の正規化＞
・主キーに部分的に従属する属性は別のエンティティにする

＜ 3 番目の正規化＞
・主キーに従属しない属性は別のエンティティにする

下表を正規化していきます。39 ページと同じ内容ですが、属性ががわかりやすいように項目名を少し変更しました（お客様、自社、見積、などを付加しています）。

「見積作成管理システム」のすべての項目
発行日
見積番号
お客様　お客様番号
お客様　会社名
お客様　郵便番号
お客様　住所
お客様　所属
お客様　肩書
お客様　名前
自社　会社名
自社　郵便番号
自社　住所
自社　所属
自社　担当者名
自社　電話番号
自社　FAX 番号
自社　電子メール
見積の総額（￥xx,xxx,xxx.-　の形式）（消費税込み）
件名
見積期限
見積の内容（明細）　商品番号　1 見積書に複数必要
見積の内容（明細）　商品名　　1 見積書に複数必要
見積の内容（明細）　商品数量　1 見積書に複数必要
見積の内容（明細）　商品単価　1 見積書に複数必要（消費税含まず）
見積の内容（明細）　商品金額　1 見積書に複数必要
見積の内容（明細）　商品備考　1 見積書に複数必要
小計（xx,xxx,xxx　の形式）
消費税額（xx,xxx,xxx　の形式　1 円以下は、切り捨て）
備考
見積の状況（作成／お客様検討中／結果確定済）
見積の結果（注文あり／注文なし）

4.3.2 はじめの正規化

はじめのルールは『繰り返す属性は別のエンティティにする』です。

例えば、前表では「見積の内容（明細）」という 1 件の見積書に複数必要な属性があります。これは繰り返す属性であると考えられます。ですから、別のエンティティにします。

「見積情報」と「見積明細」という名前にしました。そして「見積書」はイベントエンティティでした。そのまま、両方ともイベントエンティティとします。



図 4-12 第 1 正規化後

1 つの項目から、2 つのエンティティができました。そのエンティティはどういう関係でしょうか。

・1 件の見積書には、複数の見積内容（商品ごとに 1 つ）が含まれる
・1 つの見積内容は、1 件の見積書に対して作成する。

つまり、1 対多です。

4.3.3 主キーと外部キー

各エンティティにはリレーションシップのための「キー」がまだないので、作ります。キーは、大きく分けて次の 2 種類です。

主キー（プライマリキー = primary key）
リレーショナルデータベースでは、テーブルから他のテーブルのある 1 つのレコードを識別できるように設計します。そのために作成されるのが「**主キー**」です。Primary Key とも呼ばれます。主キーは 1 レコードごとにデータベースが自動採番するユニークな数字にすることをお勧めします。また、主キーの内容をユーザが変更できないようにします。なお、主キーは「id」という名前の場合があります。キーのことを「識別子＝ identifier」と呼ぶことがあるからです。

※ ユニークとは一意、つまり、唯一のという意味です。

「見積情報」エンティティには、「見積番号」がありますが、別に主キーをつくります。「見積番号」のような業務のために用意されたものは、将来、業務上の都合で内容やルールが変わってしまうかもしれません。主キーは常に不変で『あてになる』ものでなければなりません。

外部キー（フォーリンキー = foreign key）
他のテーブルの主キーを保存するためのフィールドをつくります。「**外部キー**」と呼びます。Foreign Key とも呼ばれます。原則として、外部キーから見た主キーは常に 1 つです。1 対多では「多」側のエンティティに『「1」側の主キーを外部キーとして持つレコード』が作成されることになります。

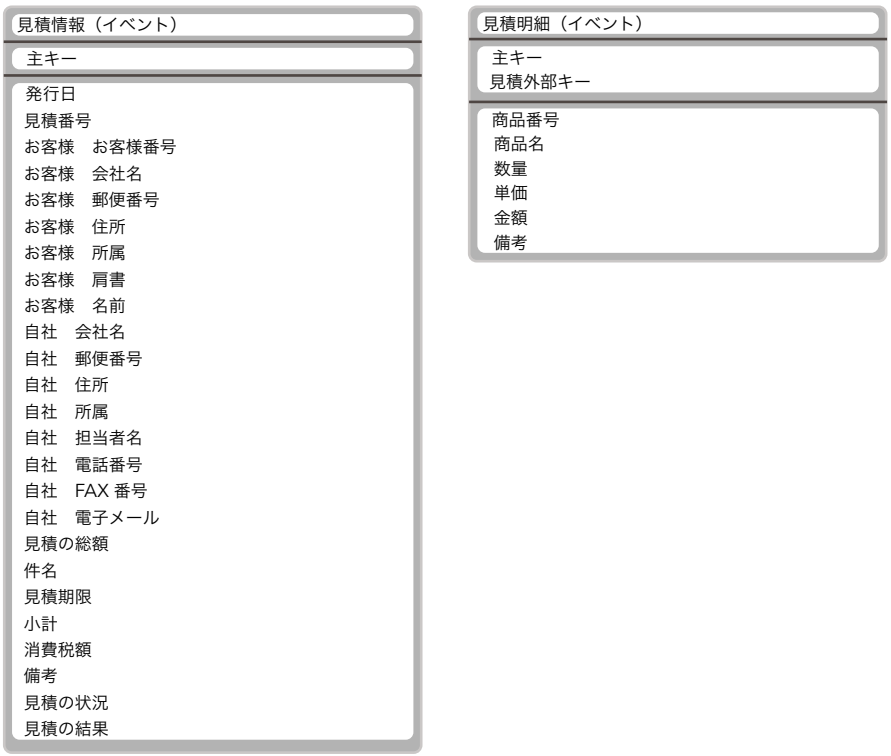


図 4-13 主キーと外部キーを設定した

次の正規化にはキーが必要です。ここからは、すべてのエンティティに主キーを作成し、正規化に応じて外部キーも作成していきます。

4.3.4 2 番目の正規化

次のルールは、『主キーに部分的に従属する属性は別のエンティティにする』です。どういうことでしょうか。やってみます。

まず「見積情報」に注目します。「見積情報」の主キーに関係しているけれど、見積自体についてのものではない項目が「主キーに部分的に従属する属性」なのです。「お客様」と「自社」が、そうです。これらも別のエンティティにします。

データモデリングでは「お客様」と「自社情報」はマスタのエンティティでした。

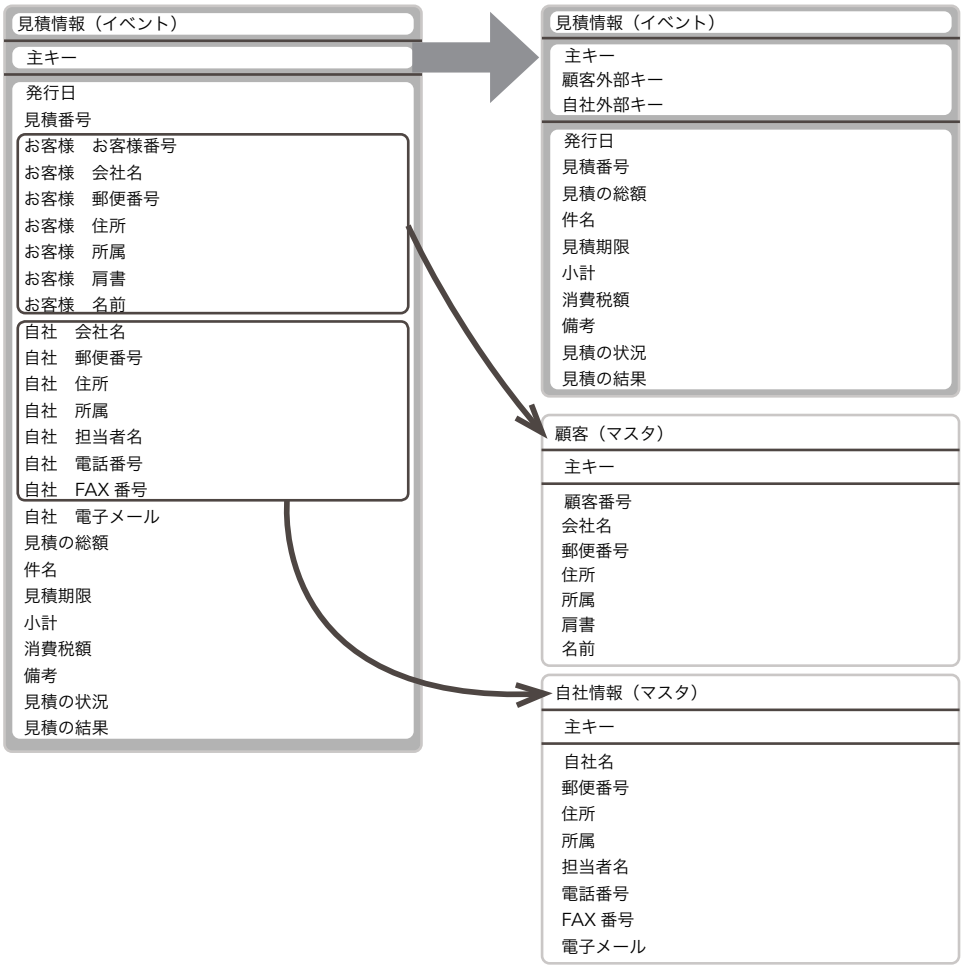


図 4-14 第 2 正規化 1

「お客様」は、「顧客」という名称に変更します。これで「見積情報」「見積明細」「顧客」「自社情報」4 つのエンティティとなりました。

次は「見積明細」に注目します。同じように考えます。商品情報は参照情報でもよさそうです。「商品」もマスタのエンティティでした。そして、2 つはどのような関連があるでしょうか。

・1 つの見積明細は、1 つの商品に対して作成する
・商品は、見積明細から何度でも参照されることができる

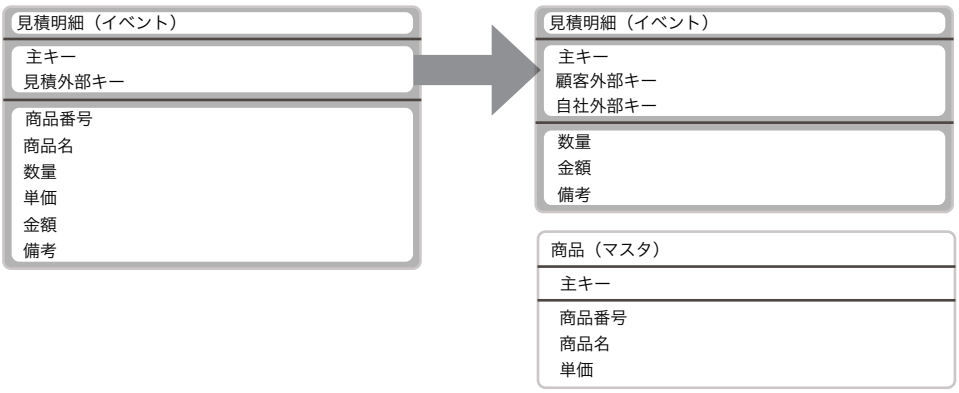


図 4-15 第 2 正規化 2

そして「顧客」には、企業や団体としての情報と顧客側の担当者の情報が含まれていました。

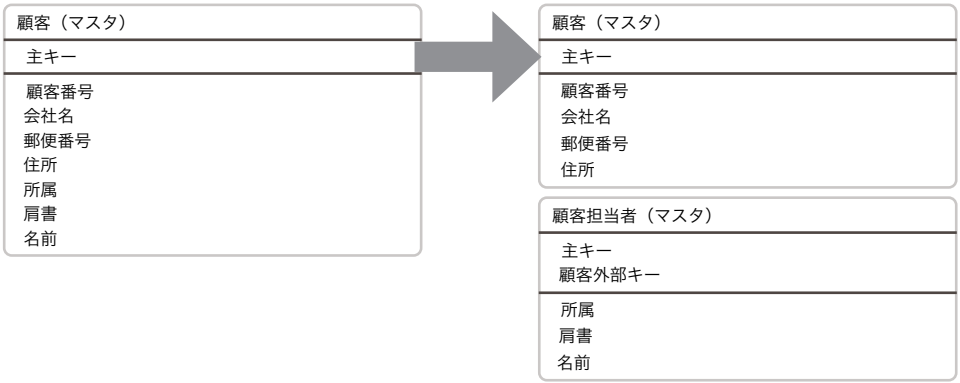


図 4-16 第 2 正規化 3

「商品」「顧客担当者」を加えて 6 つのエンティティとなりました。

4.3.5 3 番目の正規化

3 番目のルールは『主キーに従属しない属性は別のエンティティにする』です。

実は、少し変更する必要があるエンティティがあります。「担当者名」は、現在「自社情報」の中にありますが、お客様ごとに担当者が決まっているとヒアリング時に聞いていました。ですから、「顧客」に所属するべき項目です。

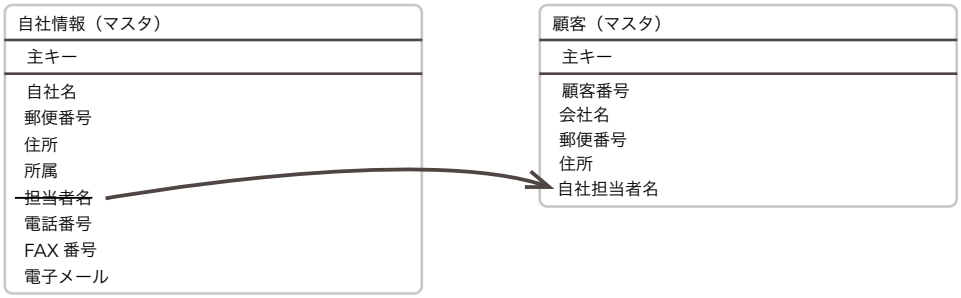


図 4-17 担当者情報は、企業に従属する

さらに「顧客」に注目します。担当者は、1 人 1 社ではありません。

・1 つの顧客企業には、1 人の担当者がいる
・1 人の担当者には、複数の担当顧客企業がある

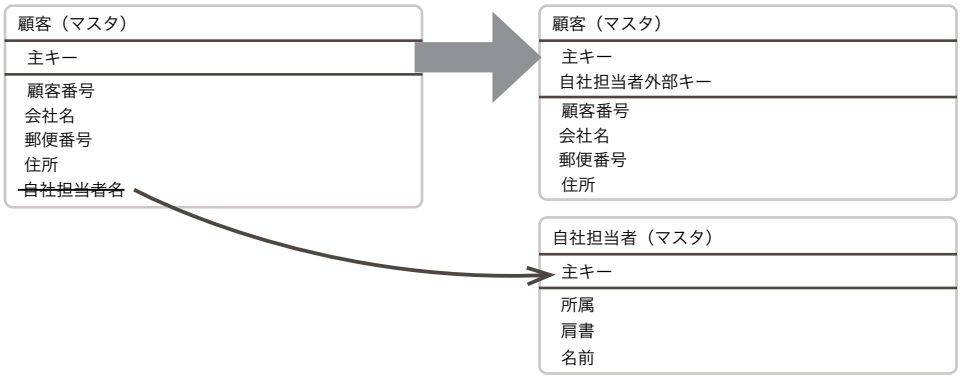


図 4-18 第 3 正規化

さらにエンティティが分かれました。「自社担当者」です。このエンティティは、顧客自体とは直接関係ありません。これが、主キーに従属していない属性です。そしてマスタです。

改めて全体を確認し「自社情報」から「自社担当者」へ「所属」を移しました。「所属」は担当者に従属すると考えたためです。

これでデータベースの構造の設計はできました。

正規化が終わった現在の ER 図は、原則として

・エンティティがテーブル
・エンティティに含まれる項目はフィールド

となります。さらに正規化するルールもありますが、ここまで正規化しておけば充分です。ER 図はどうなったでしょうか。

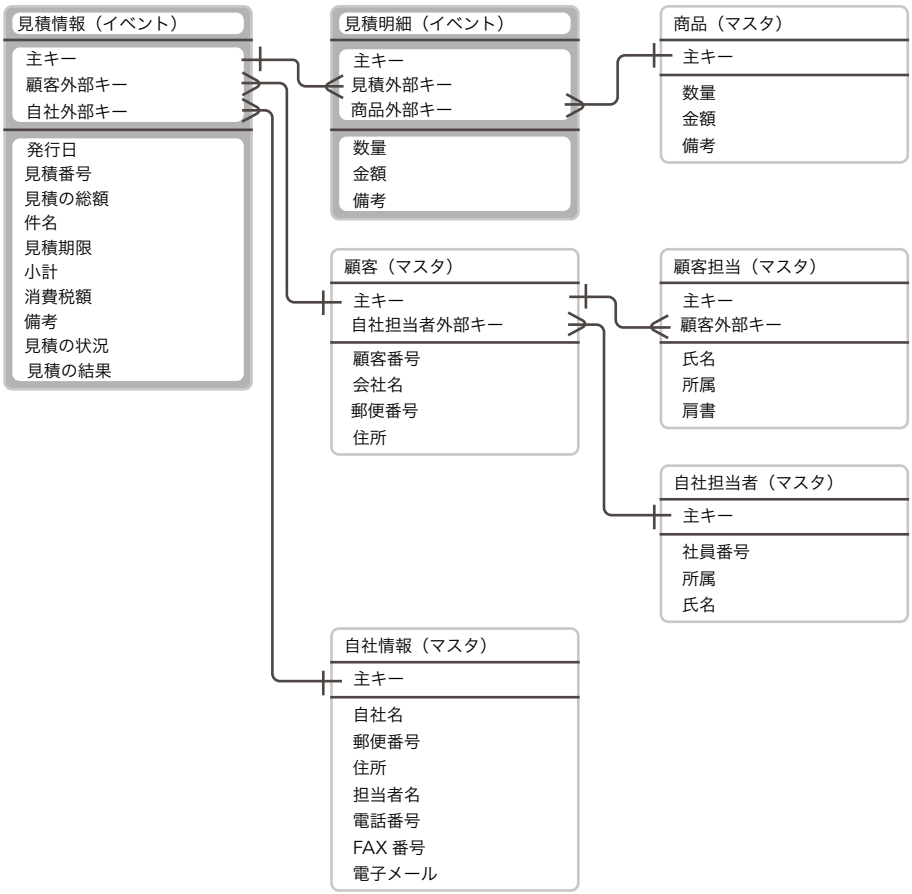


図 4-19 正規化後の ER 図

具体的になってわかりやすくなりました。関連が変わった部分があります。多対多に変化がありました。3 つのエンティティに変わりました。

・「見積書」と「商品」 → 「見積情報」と「見積明細」と「商品」

つまり、多対多は、

・2 つの 1 対多に挟まれたエンティティ　つまり　3 つのエンティティ

で、実現できることがわかります。このように考えると、ER 図を参考にしてリレーションシップグラフを作成するときに、どのようにテーブル同士を関連させるべきか見えてきます。

4.3.6 多対多の実現

「見積作成管理システム」では、「見積情報」と「商品」の間に「見積明細」が挟まって、多対多の関連が成立しました。

交差・中間

多対多の実現のために 2 つのエンティティの間にあるエンティティのことを「交差エンティティ」と呼びます。この役割を果たすテーブルのことを「中間テーブル」と呼びます。

映画と俳優を題材にして多対多について考えます。

・俳優は何度も映画に出演することができる
・映画は、1 つに複数の俳優が出演する

この関係性は、どちらに外部キーを置いても成り立ちません。データベースでは、外部キーに入力できるデータは、常に 1 つだからです。One Fact In One Place（1 つの場所に 1 つの事実）です。

・「映画」側に外部キーを置くと常にドラマ出演する俳優は 1 人だけ
・「俳優」側に外部キーを置けば今度は、俳優は 1 つの映画にしか出演できない

すると、次図のような関係を成立させることができません。

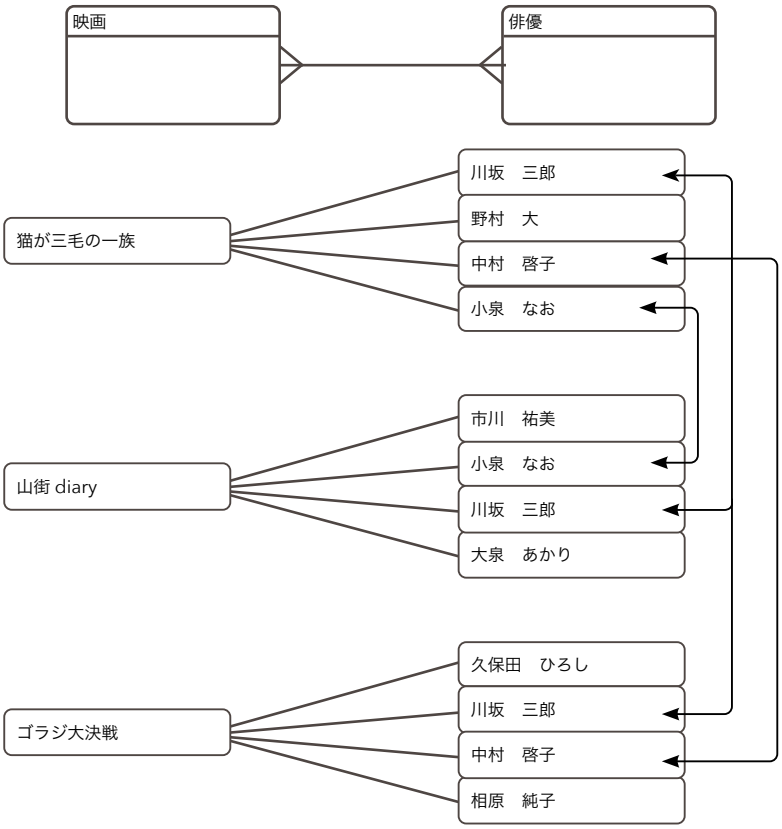


図 4-20 同じ人が違う映画に出演する

そこで、交差エンティティ＝中間テーブルを使用して、この関係を成立させるのです。1 対多を 2 つ使います。

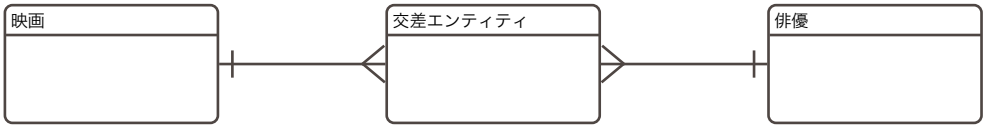


図 4-21 交差エンティティを置いた関連

もう少し具体的には、次図になります。One Fact In One Place が実現されています。

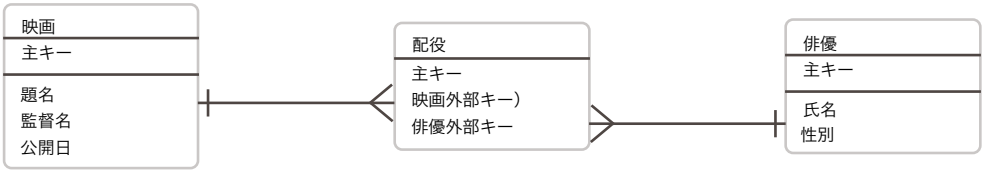


図 4-22 キーがどのように各エンティティに作成されるか

交差エンティティ＝中間テーブルは、多対多の関連を成立させるために考えられますが「見積作成管理システム」の「見積明細」のように、結果として交差エンティティが正規化の過程で生まれます。ですから意識をせずに使用していることがあるはずです。

交差エンティティは便利です。この例では「配役」という名前をつけました。すると、

・役名
・出演料

のような「映画」と「俳優」の組み合わせに対して持ちたい項目を持たせることができます。多対多が実現できなくて困ったときは、交差エンティティをつくれないかを考えてみてください。

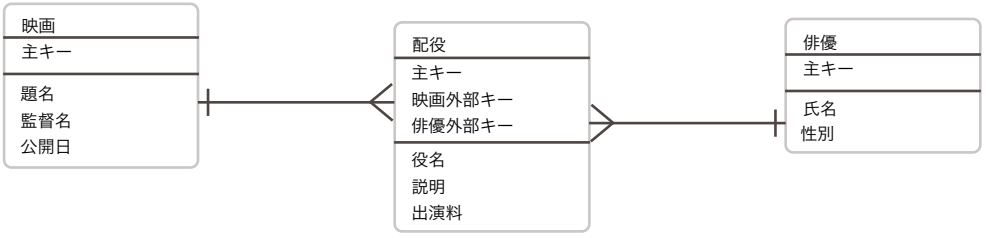


図 4-23 配役に従属する項目を追加した例

第 5 章で、この設計を実際にデータベースで使用する演習を行います。

4.4 その他

データベースの構造の設計で、その他に注意したい点がいくつかあります。

4.4.1 依存

データは関連に従った依存を持つことがあります。

多対多では、交差エンティティ＝中間テーブルに作成できるデータは他の 2 つのテーブルの外部キーを持つ必要があります。1 対多では、多（子）の側に 1（親）の外部キーを持たないといけません。

このような関係を「**依存**」と呼びます。

データベースの機能として、依存を無視したデータを作成することも可能です。しかし、それは ER 図で考えた設計と異なります。設計上依存が必要なのに、それを無視したデータが存在するようなことを通常は許しません。依存を無視したデータが必要であるとしたら、想定外のイレギュラーなデータか、設計時に間違いや見逃しがあるのかもしれません。もう一度、正規化を見直します。

4.4.2 正規化しすぎない

正規化を考えていると、いくつかの 1 対多の関係を多対多にしたくなってくることがあります。例えば、『今は自社担当者は 1 顧客ごとに 1 名だけれど、もしかしたら複数名になることがあるかも……』と思い浮かぶかもしれません。その考慮は重要で素晴らしいことです。

しかし、いくつか注意があります。

交差エンティティ＝中間テーブルは、依存先の 2 つの外部キーを持たなければいけないため、データベース上でのデータの入力や更新、削除の取り扱いに気をつける必要があります。ER 図のあちこちに多対多の関連を設定すると設計が複雑になり機能を作成するための作業量も増えます。

作業の期間や、データベースの複雑さに加えて 1 対多を多対多にするかもしれない可能性の高さを総合的に考慮します。まずは必要な機能について優先して考えます。

4.4.3 ルールは絶対ではない

正規化で 1 対多にするべき項目でも、あえてテーブルを分けない場合があります。

例えば、インポートやエクスポートでデータを取り扱いやすくしたり、『この属性に従属するのはずっと 3 つしかありえない』というような状況から、テーブルを分ける代わりにフィールドを必要な数だけ作成することでテーブル数を少なくし、データベースを必要以上に複雑にしないためであったりします。

4.4.4 設計と変更

運用を開始した後に設計上の大きな変更を実施することはお勧めしません。大きな設計変更は、多くの機能に影響します。動作確認や変更に対応した作業をする必要があります。

しかし、業務を取り巻く環境が激変するなど、『変更しなければならない』状態になることはありえます。そのような場合、現在のシステムを変更するか、別の新システムを作成して置き換えるか、検討してください。

3 段階の正規化を実施しました。その基本を理解した上で設計してください。

数字と UUID

通常、FileMaker Pro のデフォルトのフィールドの自動作成では「主キー」フィールドに計算値自動入力で Get(UUID) 関数　が設定されています（計算値自動入力については後述）。新規レコードを作成すると「主キー」フィールドに次のような文字列が自動入力されます。

67574151-F1FC-4ECB-8787-3BE8AA9B7D3F

UUID（Universally Unique IDentifier）は、全世界でユニークになる（重複する確率が限りなく 0 である）識別子です。FileMaker Pro 独自の考え方ではなく標準化がなされている世界的なものです。

主キー数字の連番にするか、UUID を使うか、の選択はカスタム App の役割などによって設計段階で決定します。なお、「主キー」の仕様を途中で変更することは避けてください。主キー＝識別子は、レコードをユニークにするためのとても重要なデータです。関連により外部キーにも入力されている可能性が高いものです。すでに動作している仕様を別な仕様で置き換えることはとても難しい作業です。予期しないエラーや、関連が成立しなくなってしまった……などの障害が発生するかもしれません。

ユニークな数字は、充分に主キーとして動作します。リレーションシップ成立の仕組みがすぐに確認できること、などを重視する場合は数字の連番を採用してもまったく問題ありません（本書では、一部の演習ファイルで数字の連番を採用しています）。

UUID が良いけれど、フィールドタイプを数字にしたい場合は　Get(UUID 番号) 関数　を利用してください。

第 5 章 リレーションシップ（関連）

テーブル設計の結果（ER 図など）がリレーションシップでどのように実現されるかなどについて解説します。

5.1 関連の定義

データベースの構造を決めることを「**定義**」と呼びます。

・企画と要件定義の内容

そして、設計され、ER 図などで表された

・エンティティ
・エンティティに含まれる項目
・エンティティのリレーションシップ（関連）

を元に、必要な

・テーブル
・フィールド
・リレーションシップ（関連）

を定義します。カスタム App ファイルを作成した後、［データベースの管理］ダイアログで、データベース構造の作成、変更、削除、設定、確認をします。

本章では、リレーションシップを「**関連**」と記述します。これは、リレーションシップについての作業を実施する場所が［データベースの管理］ダイアログの「リレーションシップ」タブの「リレーションシップグラフ」なので紛らわしいためです。

まず、リレーションシップグラフと関連の定義について解説します。

本章よりファイルでの作業を開始します。「環境設定」の「レイアウト」タブ内の［新規フィールドを現レイアウトに加える］チェックが　**オフ**　になっていることを確認してください。

（ここから本書に付属している演習ファイルの「5 章開始時」フォルダの中の「5_ 見積作成管理 .fmp12」ファイルをベースに進めます）

「見積作成管理 .fmp12」ファイルを開いて［データベースの管理］ダイアログを表示してください（ヘルプが表示されている場合は、右上のアイコンをクリックして閉じてください）。

5.1.1 ［データベースの管理］ダイアログ

［データベースの管理］ダイアログの上部には３つのタブがあります。



図 5-1［データベースの管理］ダイアログボックスの「リレーションシップ」タブ

［データベースの管理］ダイアログについて改めて確認します。

「テーブル」タブ
作成されているすべてのテーブル名と情報が表示されます。テーブルの作成や削除など、テーブルを管理するタブです。

「フィールド」タブ
テーブルごとに作成されているすべてのフィールド名と情報が表示されます。フィールドの作成や設定、削除など、あるテーブルのフィールドを管理するタブです。

「リレーションシップ」タブ
設定されているリレーションシップのグラフが表示されます。関連の作成、定義、削除など、関連を管理するタブです。

主な開き方は次の通りです。

方法：メニュー
［ファイル］–［管理］–［データベース ...］メニューを選択します。

方法 2：ショートカット
ショートカットキー［Ctrl］＋［Shift］＋［D］（Windows）あるいは［command］＋［Shift］＋［D］（macOS）を使います。

5.1.2 リレーションシップグラフ

「リレーションシップ」タブをクリックしてください。「リレーションシップ」タブに表示されているのは「**リレーションシップグラフ**」です。



図 5-2 リレーションシップグラフ

リレーションシップグラフのツール

主なツールについて図示します。さらに詳しいツールの使い方はヘルプを参照してください。

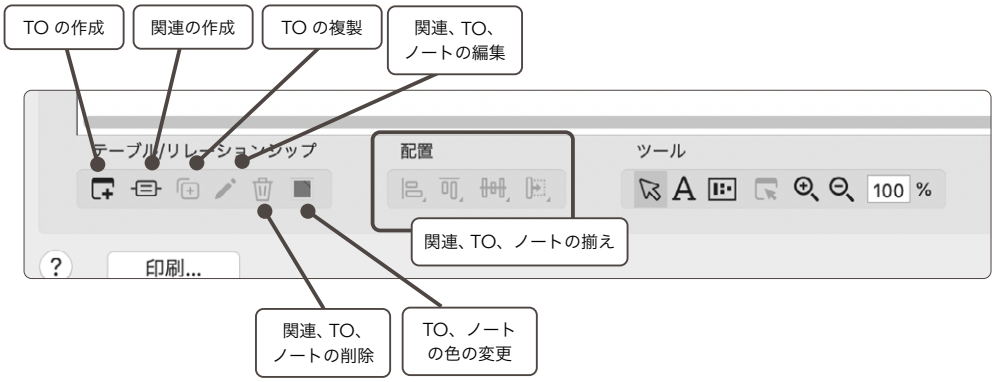


図 5-3 リレーションシップグラフのツール

5.1.3 テーブルオカレンス（TO）

「見積作成管理 .fmp12」のリレーションシップグラフには「見積情報」「見積明細」など四角い枠が全部で 7 つあります。この四角い枠が「**テーブルオカレンス**」です。今後は「TO」と呼びます。テーブル間の関連付けには、この TO を使います。



図 5-4 リレーションシップグラフ（関連がなにも設定されていない状態）と TO

TO は、ファイルに実際に作成されているテーブル（実体テーブル）の仮想テーブルです。

実体テーブル
ファイルに実際に作成されているテーブルのことを本書では「実体テーブル」と呼んで区別することがあります。

仮想テーブル
実体テーブルを仮想的に表現したものです。

FileMaker Pro は、ファイルにテーブルを作成すると同じ名前の TO を 1 つ作成しリレーションシップグラフ上に表示します。TO とは「仮想のテーブル」です。「どの実体テーブルの情報を取り扱うのか」という定義が必ずなされています。

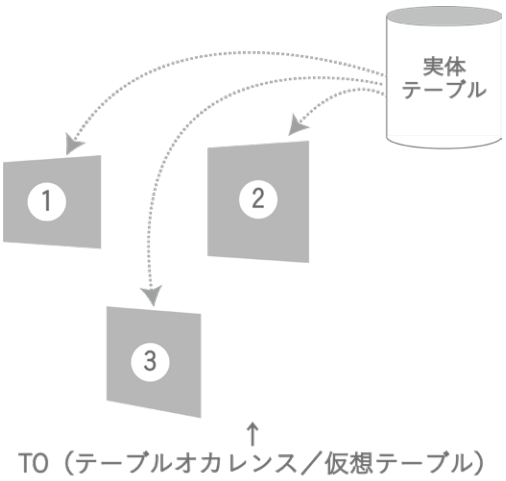


図 5-5 実体テーブルと TO（仮想テーブル）

1 つの TO に定義される実体テーブルは常に 1 つです。逆に、1 つの実体テーブルに対して用途に応じて複数（いくつでも）TO を作成可能です。

5.1.4 テーブルオカレンスグループ（TOG）

リレーションシップグラフを下の方へスクロールすると、すでに作成済みの、関連付けが定義された複数の TO の集まりがあります。このような TO の集まりを、「**テーブルオカレンスグループ (TOG)**」と呼びます。TOG はリレーションシップグラフ上にいくつでも作成可能です。

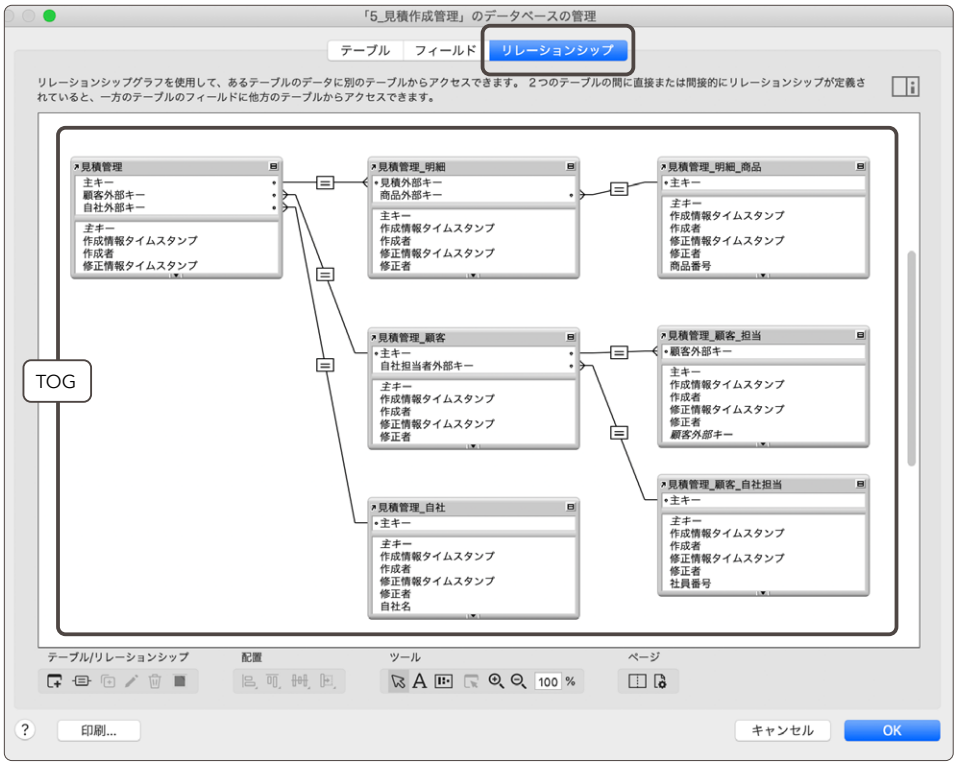


図 5-6 TOG

リレーションシップグラフの表現

2 つの TO に関連が定義されるとリレーションシップグラフでは線で表現されます。

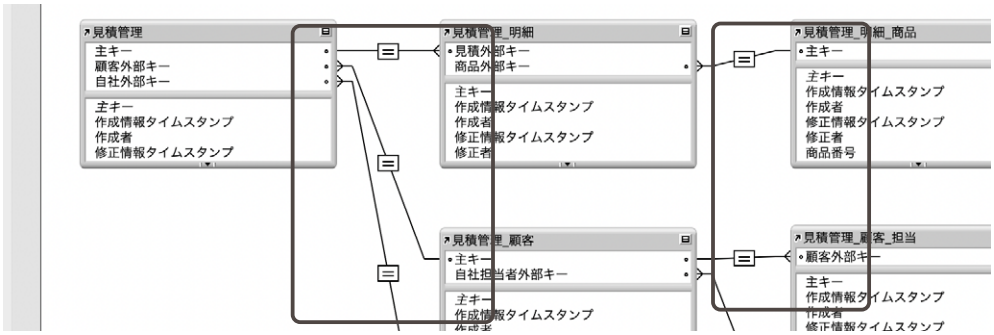


図 5-7 関連を表す線

TO を繋いでいる線の端子は、関連の定義に従って、自動的に描画されています。

5.1.5 ［リレーションシップの編集］ダイアログ

「見積管理」TO と「見積管理 _ 明細」TO を繋いでいる線や、四角形（次ページ「方法 1：四角形をクリック」の図も参照）をダブルクリックしてください。

［リレーションシップの編集］ダイアログが表示されます。関連の設定と確認が可能です。関連の照合フィールド、条件の指定と、必要に応じてオプションの設定が可能です。

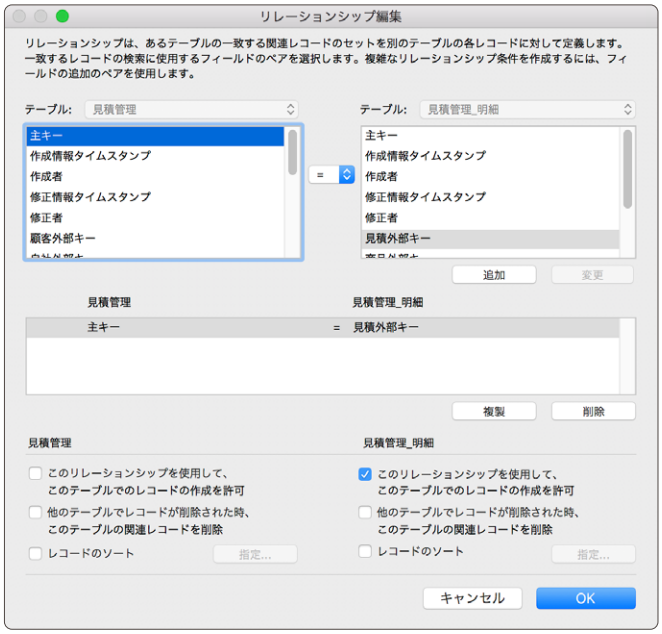


図 5-8［リレーションシップの編集］ダイアログ

改めて確認します。

演算子

関連条件の演算子は「＝」だけではありません 7 種類が用意されています。

・＝（同一）
・≠（違う）
・＜（未満）
・≦（以下）
・＞（超過）
・≧（以上）
・×（すべて）

［このリレーションシップを使用して、このテーブルでのレコードの作成を許可］

テーブルの照合フィールドにデータが入力されたとき、その参照先テーブルにそのデータに基づいた新しい関連レコードが作成されます

［他のテーブルでレコードが削除された時、このテーブルの関連レコードの削除を許可］

テーブルのレコードが削除されたとき、その参照先のテーブルの関連レコードも削除されます。

※特に取り扱い注意のオプションです。削除を実行するレイアウトに関連レコードが表示されていなくても削除されます（第 16 章も参照してください）。

［レコードのソート］
関連レコードをソートする順を設定します。設定しなければレコードの作成順になります。

2 つのテーブルオカレンスにそれぞれ設定がある
ダイアログには、設定場所が 2 つあります。右側と左側の TO のどちら側に設定するかを選択するためです。例えば、関連レコードの削除では、

・「見積情報」テーブルの 1 レコードを削除したら「見積明細」テーブルにある関連レコードも削除する

この場合、次のように参照先の関連レコード側の TO に設定します。

・「見積管理 _ 明細」TO の［他のテーブルでレコードが削除された時、このテーブルの関連レコードの削除を許可］を　**オン**

もしも、参照元と参照先で共通の 1 設定しかできなかったら「見積明細」テーブルのデータを削除したら「見積情報」テーブルのレコードも削除されてしまいます。それは困ります。そのために 2 つの TO のどちらに設定するか選べるようになっているのです。

これらのオプションは、左右の TO のどちらに設定するか、そして、設定でどのようなことが起きるのか、を確認した上で設定をおこなってください。

ダイアログの開き方を記述します。

方法 1：四角形をクリック
TO を繋いでいる線や四角形部分をダブルクリックします。

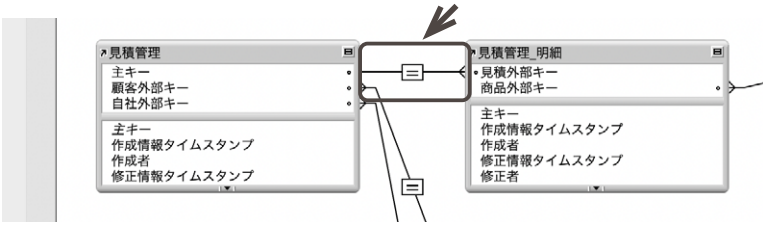


図 5-9 四角形部分

方法 2：鉛筆アイコン
TO を繋いでいる線を選択して［データベースの管理］ダイアログの画面下の鉛筆アイコンをクリックします。



図 5-10 鉛筆アイコン

5.1.6 関連定義のルール

関連を定義する場合に、気をつけることがあります。

TO 名はユニークでなければならない

TO の名前はユニークでなければなりません。2 つ以上の TO に同じ名前はつけられません。

ループできない

TO 間をぐるぐる回るような関連を定義することはできません。

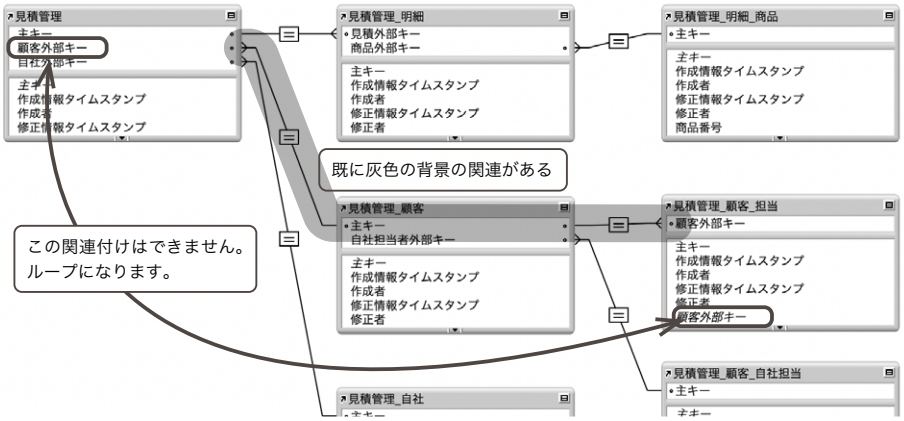


図 5-11 ループになるような関連はつくれない

テーブルの関連は、2 つのテーブル間に 1 つとは限りません。2 つのテーブル間に 2 つの関連が必要な場合は、TO を複数作成して定義します。

ただし、別の TO としてこの関係を設定できます。

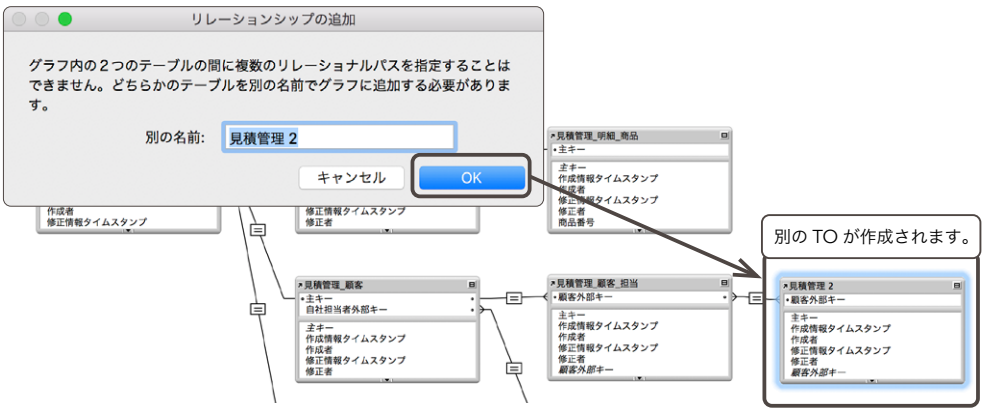


図 5-12 それでもつくろうとすると別の TO になる

なぜこのようになるのでしょうか。それは『TO 同士の関連の定義は 1 つだけ』だからです。ある TO 間に 2 つの関連はつくれません。図の例では、「見積管理」TO と「見積管理 _ 顧客 _ 担当」TO は「見積管理 _ 顧客」TO を通してすでに関連しています。

ある TO 間に 2 つの関連があったら、どちらの関連を元にデータを扱うべきか判断できません。

前述した One Fact In One Place（1 つの場所に 1 つの事実）が、ここにも隠れています。TO 間の関連は 1 つでなければなりません。

自分同士を関連付けできる
テーブルの関連は、別のテーブル間とは限りません。自分と自分の関連が必要なことがあります。その場合、同じ実体テーブルを持つ別々の 2 つの TO を関連付ければ可能です。「**自己リレーションシップ**」と呼ばれます。この関連を使用した機能などの演習を後述します。

5.1.7 ER 図とリレーションシップグラフの違い

ER 図のエンティティとリレーションシップグラフの TO は違うものです。

ER 図は 1 つの図に 1 つのエンティティを 1 回のみ描きます。テーブル間の関係を表現する図だからです。一方、リレーションシップグラフでは、実体テーブルが持ついくつかの「**コンテキスト**」に従って関連を定義します。そのため、**1 つの実体テーブルから複数の TO（仮想テーブル）を作成して登場させることができます**。ここが大きく違います。

全体像の把握
ある実体テーブルの TO がリレーションシップグラフに複数登場するとき、矛盾が生じるような関連が定義されてしまうとデータに不整合がでるかもしれません。実体テーブル同士の関連全体を確認するためにも設計と ER 図は重要です。

ER 図で『あるテーブルが、どのようなコンテキストを持ち得るのか』が確認できると、リレーションシップグラフで関連を定義するときに迷うことがありません。

コンテキスト
リレーションシップグラフ中の TO が、どのような「立ち位置」なのか、ということです。データベースから離れて考えてみます。ある人（「A」さんとします）は、両親にとっては子供です。会社では従業員、配偶者や子からはお父さん、趣味のサークルではメンバーです。どれも A さんですが、立ち位置によって違います。

FileMaker Pro は、リレーションシップグラフで設定したユニークな TO 名を通じて、どのコンテキストから、どの関連の定義を使って表示や計算や処理を実行するのかを判断します。

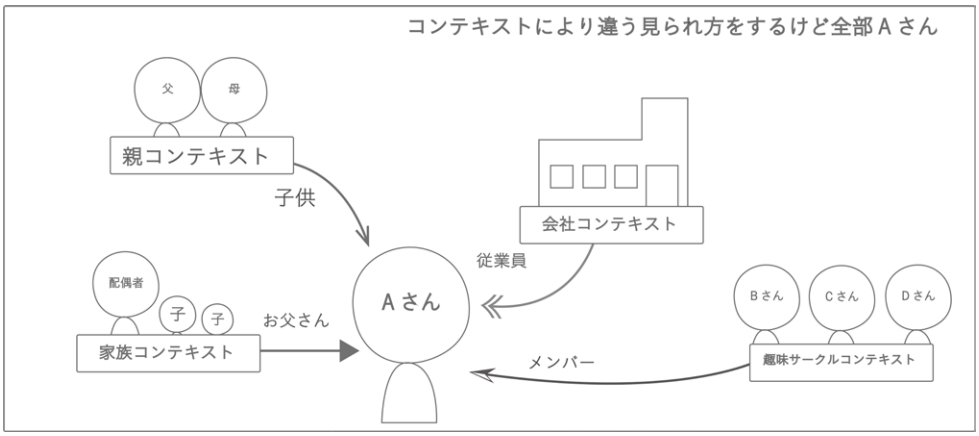


図 5-13 様々なコンテキスト

5.2　設計からカスタム App へ

ファイルに構造を作成するとき、テーブルやフィールドの作成で迷うことはないと思います。関連はどうでしょうか。ER 図は 1 つの図であることがほとんどですが、1 つのカスタム App で、TOG が 1 つとは限りません。

5.2.1　テーブルオカレンスグループ（TOG）のわけ方

どのような基準で TOG を作成し、複数に分けるのでしょうか。次のような考え方があります。

カスタム App のメニューを目安にする

わかりやすい目安として「**メニュー**」を考えます。ユーザがカスタム App を使用するとき、はじめに表示されるメニュー画面にどのようなボタンがあるべきか、です。

例えば「見積作成管理システム」では、

・見積管理

だけが思いつくかもしれませんが、要件定義のまとめを見ていくと

・共通のデータの管理

という作業もありました。この作業に含まれるものは、

・顧客情報管理
・商品情報管理
・自社情報管理

が、思いつきます。他にもありそうです。それは、システムの管理です。

・ログインのためのアカウント管理

です。まとめて 5 つのメニュー＝ TOG が必要であると考えられます。

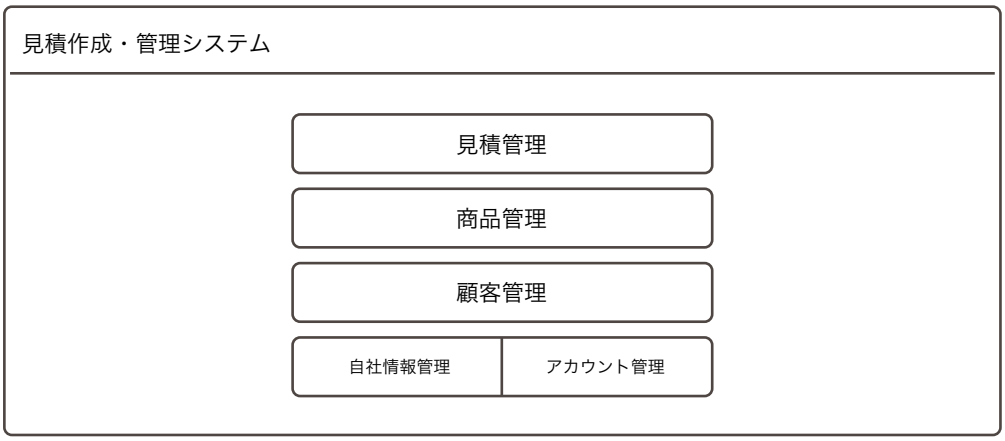


図 5-14 見積作成管理システムのメニュー画面イメージ

5 つのメニューごとに別々の TOG を作成すると、作業時にメニューごとにレイアウトや機能を考えられるようになります。つまり、

・このレイアウトはどの TOG（メニュー）の TO のものか
・この機能はどの TOG（メニュー）について動作するものか

が、わかりやすくなります。各 TO のコンテキストもはっきりします。ここでは一番大きな単位でメニューを考えましたが必要な場合には、さらに TOG を分けることがあります。

なお、注意があります。**通常、TOG 同士は関連づけません。**

リレーションシップグラフでは TO の関連の定義だけではなく、関連のオプション（レコードの新規作成やソートなど）を設定可能です。TOG 同士を関連づける場合も、設定済みのオプションも含めてなにが起きるのかを慎重に考える必要があります。想定していない動作や、不整合なデータ作成が起きる原因になるかもしれないからです。

5.2.2 アンカー / ブイ方式

TOG の見やすさについて考えます。コンテキストがわかりやすいように TO を配置します。

FileMaker Pro では、リレーションシップグラフでの TOG の記述方法として「**アンカー / ブイ**」と呼ぶ方式があります。TOG では、

・**中心、あるいは基本となる** TO を**一番左**に置く（アンカー＝錨）
・**関連するテーブル**を**右**に広げていく（ブイ＝浮標）

といったルールで TOG を表現します。

本書では、リレーションシップグラフとの比較がわかりやすいように、ER 図をアンカー / ブイに近い形で記述しています。

※ ER 図にたくさんのエンティティがある場合には、アンカー / ブイのように記述することが難しくなります。ER 図は、わかりやすく見やすいことを念頭に置いて作成してください。

「見積作成管理システム」の ER 図を元に「見積管理」メニュー、「顧客管理メニュー」で使用する想定の TOG 作成しました（次ページ図）。確認してください。

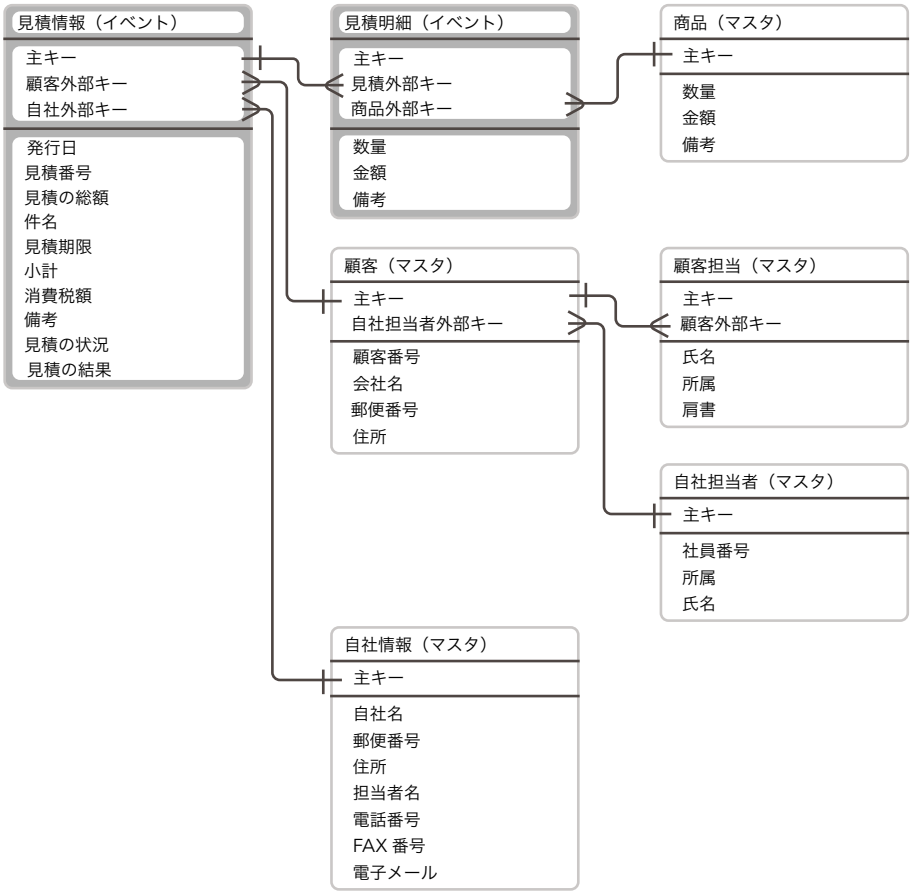


図 5-15「見積作成管理システム」の ER 図

次ページの図は、先ほどのメニューを元にしたアンカー / ブイの TOG 例です。次の 2 つ機能（矢印）のための TOG です。

- **見積管理** ←
- **顧客情報管理** ←
- 商品情報管理
- 自社情報管理
- ログインのためのアカウント管理

（「見積作成管理 .fmp12」ファイルの［データベースの管理］ダイアログを大きく縦長に広げると、次図の TOG を確認できます。確認後、［キャンセル］をクリックしてダイアログを閉じてください）

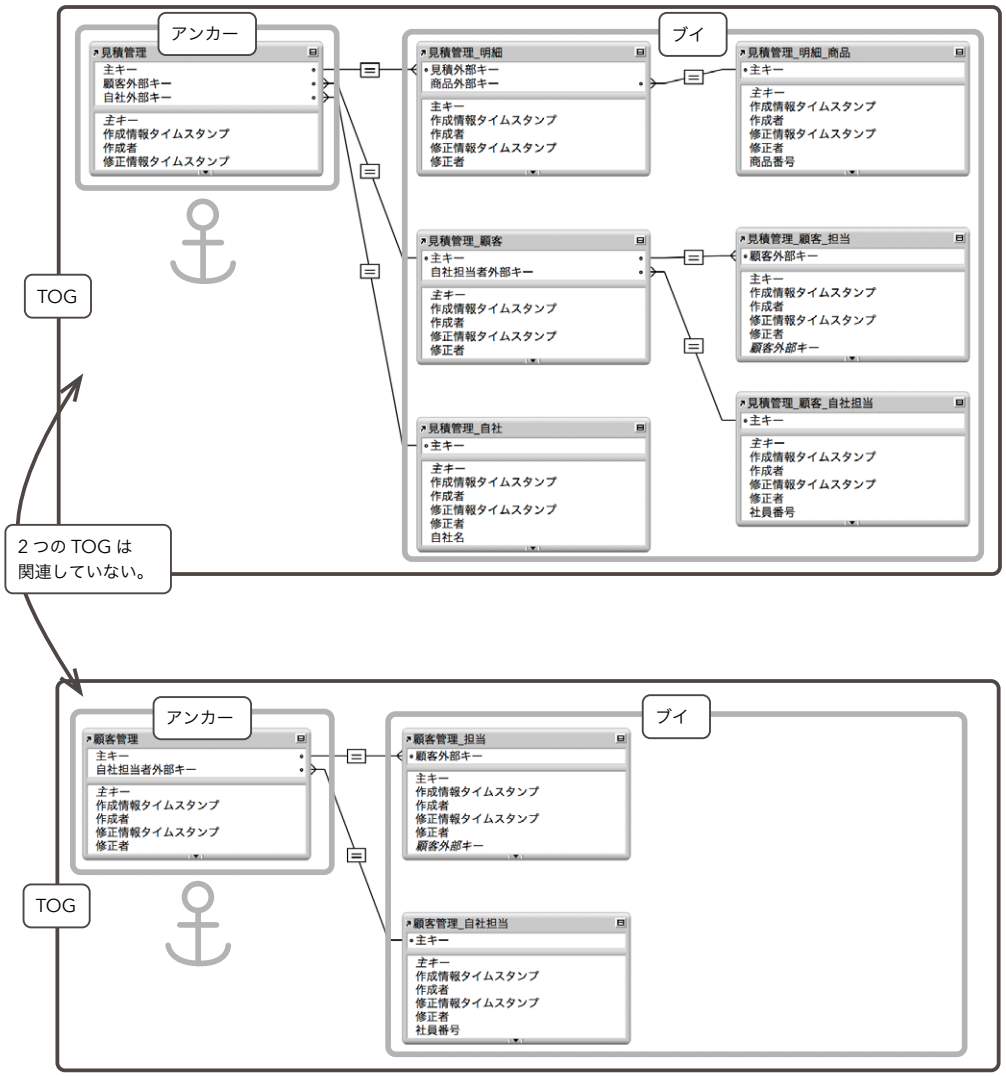


図 5-16「見積管理」と「顧客情報管理」2 つの TOG（アンカー / ブイ）

ER 図のエンティティ、テーブル、TO がどのような関係になったかを表にしました。エンティティと実体テーブルは同じ 1 つですが、TO の数が違います。

「見積作成管理システム」の ER 図のエンティティ名と FileMaker のテーブル名と TO 名		
エンティティ名	**実体テーブル名**	**TO 名**
見積情報	見積情報	見積管理
見積明細	見積明細	見積管理 _ 明細
商品	商品情報	見積管理 _ 明細 _ 商品
顧客	顧客情報	見積管理 _ 顧客 顧客管理
自社担当者	自社担当者	見積管理 _ 顧客 _ 自社担当 顧客管理 _ 自社担当
顧客担当	顧客担当者	見積管理 _ 顧客 _ 担当 顧客管理 _ 担当
自社情報	自社情報	見積管理 _ 自社

また、設計された内容を元に定義しているので、「見積管理」TO からはじまる TOG と、「顧客管理」TO からはじまる TOG では、顧客と顧客側の担当者、自社の担当者について、

・見積管理 _ 顧客
・見積管理 _ 顧客 _ 担当
・見積管理 _ 顧客 _ 自社担当

の、3 つの TO と

・顧客管理
・顧客管理 _ 担当
・顧客管理 _ 自社担当

の、3 つの TO の関連が同じです。TOG が異なっていても同じ関連が必要な箇所だからです。先に設計しておけば、関連の定義に迷うことが少なくなります。

同じ関連が登場する場合
複数の TOG で、同じ関連が必要となる事があります。例えば「見積作成管理システム」では上記の、ような場合です。2 つの TOG を繋げずに、それぞれに関連を定義することをお勧めします。

5.2.3 TO の命名ルール

それでは、TO は、どのように命名すると良いでしょうか。次の 3 つのことが思い当たります。

・ユニークであること
・TOG（コンテキスト）が推測しやすいこと
・どの実体テーブルの TO なのか推測しやすいこと

作業中にレイアウト、計算式やスクリプトで TO 名を一覧から選択、指定する場面は、何度もあります。そのための工夫として命名ルールを決めておきます。例えば、次のようなルールです。

アンカー / ブイ方式を思い出してください。アンカーになっている基本の TO からの関連をわかりやすいように命名するには、次のような方法があります。これまでの図でも使っています。

・アンカーの TO　→　見積管理　　（←見積情報テーブルの TO）
・ブイの TO　　→　見積管理 _ 明細　　（←見積明細テーブルの TO）
・ブイの TO　　→　見積管理 _ 明細 _ 商品（←商品テーブルの TO）

アンダーバーでアンカーとブイをつないでいます。TO の属する TOG や階層、コンテキストがわかりやすいので、TO 選択時の迷いが少なくなります。ユニークな TO 名になりやすい方法です。

本書ではこのルールで TO 名を付与していきます。「見積作成管理 .fmp12」を閉じてください。

（ここから、本書に付属している演習ファイルの「5 章開始時」フォルダの「5_ 映画と人 .fmp12」ファイルをベースに進めていきます）

演習ファイル「映画と人 .fmp12」ファイルを開いてください。

5.3　多対多とデータ

多対多の関連について設計とリレーションシップグラフを確認します。映画と俳優の関係を表しています。まず、次の要件があります。

・俳優は複数の映画に出演することができる
・映画は、1 つに複数の俳優が出演する

2 つのエンティティが考えられます。多対多です。

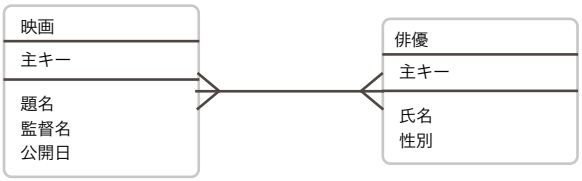


図 5-17 映画と俳優は多対多

そして､1 つの映画の出演者のために「配役」エンティティを中間のテーブルとして付け加えました。

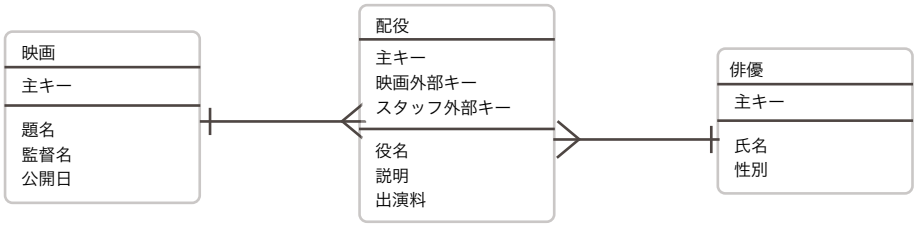


図 5-18 中間テーブルを加えた

さらに要件があります。映画の監督についてです。監督と映画は 1 対多です。

・映画にとって監督は常に 1 人である
・1 人の監督は、複数の映画を監督することができる

新しいエンティティ「監督」を増やすべきでしょうか。しかし、もう 1 つ要件がありました。

・映画監督は俳優として映画にでることがある。俳優が映画監督となることもある。

エンティティを追加せず、「俳優」エンティティが俳優と監督が含まれた「スタッフ」というエンティティになりました。ER 図は次のようになりました。

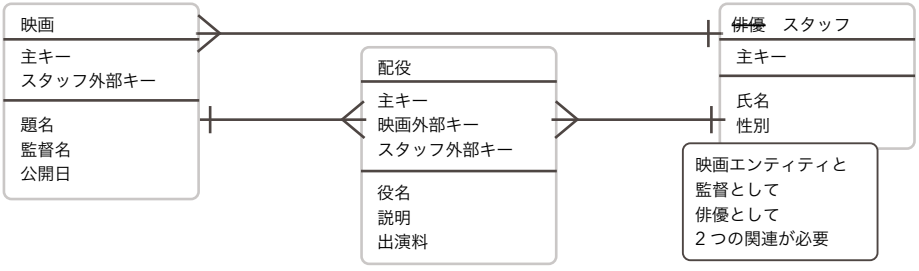


図 5-19 スタッフは監督でもあり、俳優でもある

「スタッフ」という 1 つのエンティティは、「映画」エンティティと 2 種類の関連があり、それぞれ、「俳優」と「監督」の 2 つのコンテキストを持つことがわかりました。カスタム App でもそのように取り扱う必要があります。

※ ER 図では、2 つのエンティティ間にいくつでも関連を記述可能です。

関連を、カスタム App ではどのように実現するかを確認していきます。

［データベースの管理］ダイアログを開いてください。「テーブル」タブを確認すると、次図のようになっています。ER 図のエンティティとテーブル数は同じ 3 です。



図 5-20 設計内容から作成されたテーブル

「フィールド」タブも確認してください。「主キー」と「外部キー」といくつかの項目が作成されています。



図 5-21 設計内容から作成されたフィールド

「リレーションシップ」タブは、どうなっているでしょうか。次図のようになっています。1 つの ER 図から、2 つの TOG が作成されています。

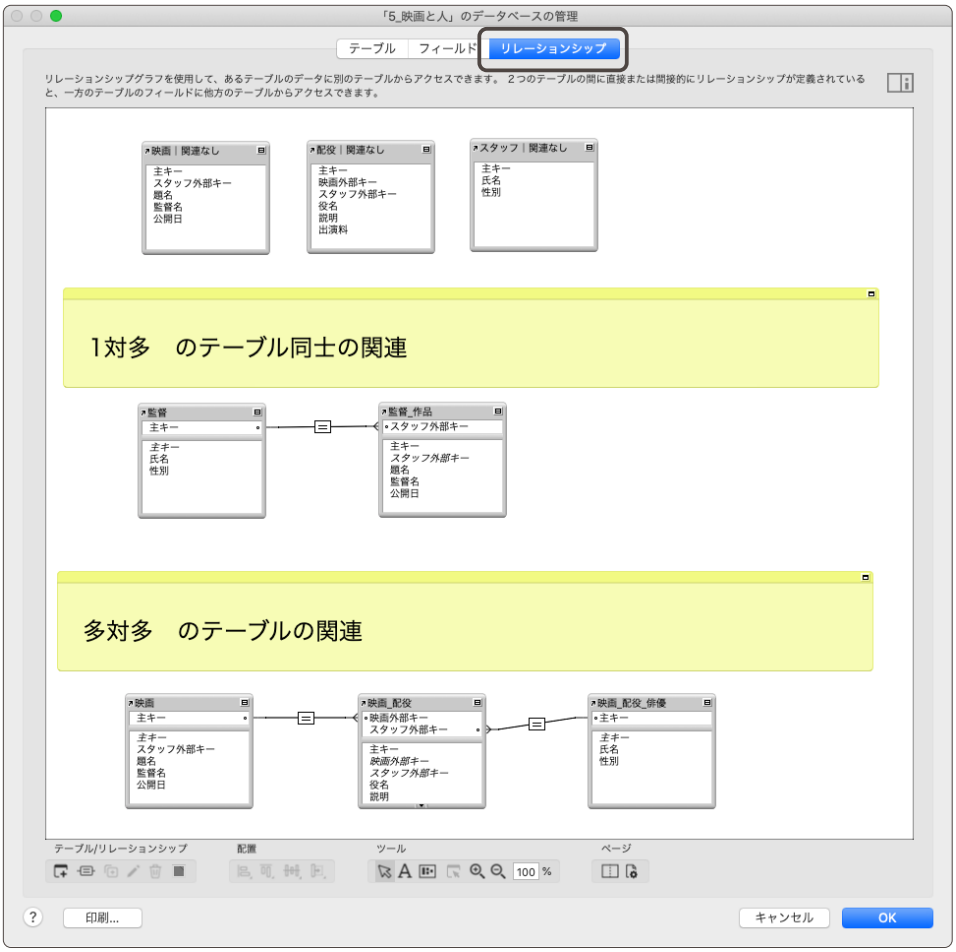


図 5-22 設計内容からリレーションシップグラフを作成

確認後、［キャンセル］をクリックしてダイアログを閉じてください。

「主キー」フィールドについて

本章で使用する演習ファイル「映画と人 .fmp12」「料理と食材 .fmp12」は、関連を確認しやすいように、各テーブルの「主キー」フィールドを、次のように設定しています。

・数字タイプフィールド
・シリアル番号の自動入力

5.3.1 1 対多とレイアウト

「多対多」の前に「1 対多」を確認します。「監督コンテキスト」レイアウトに切り替えてください。「スタッフ」エンティティと「映画」エンティティの関連を確認します。9 レコード目を表示してください。

レイアウトモードでレイアウト設定を確認してください。このレイアウトには「監督」TO が設定されています。つまり「監督」TO のコンテキストから、関連テーブルとレコードを表示しています。

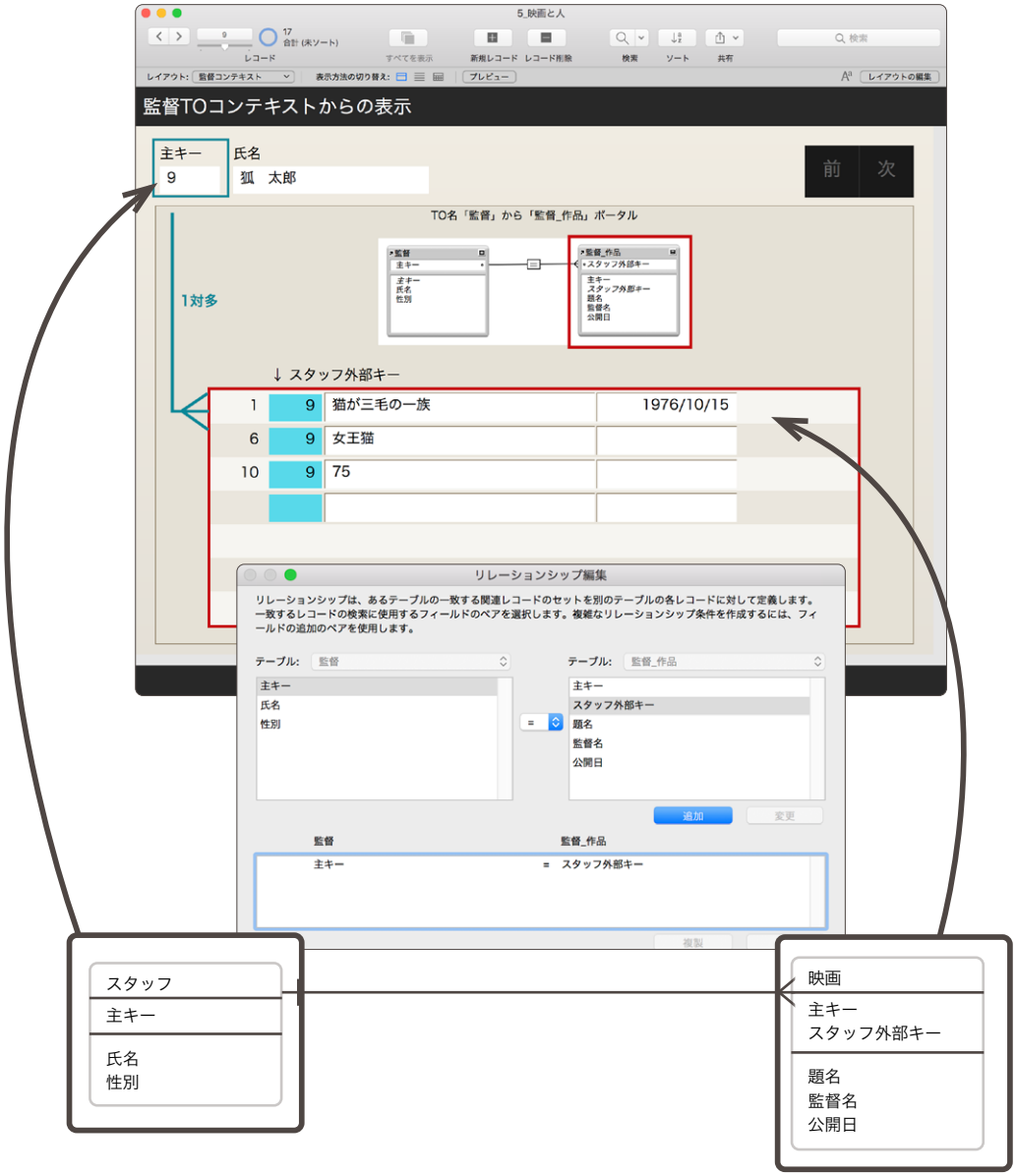


図 5-23「監督コンテキスト」レイアウト

ER 図の「スタッフ」エンティティと「映画」エンティティの関連は「1 対多」です。

・1 側は、「スタッフ」テーブル
・多側は、「映画」テーブル

と、いうことになります。次のことで関連を成立させることができるように設計されています。

・「スタッフ」テーブルの「主キー」フィールドと同じ数字を「映画」テーブルの「スタッフ外部キー」フィールドが持つ

これを、リレーションシップグラフ上で、

・1 側、つまり「スタッフ」テーブルは「監督」TO
・多側、つまり「映画」テーブルは「監督 _ 作品」TO

として関連を定義しています。レイアウト上には、1 人の監督について 1 つ以上の作品が入力できるようにポータルが配置されています。

ポータル

作品名を入力する部分はポータルオブジェクトです。表示されるのはレイアウトに設定された TO の「関連テーブル」の「関連レコード」です（関連が成立しないとデータが表示されません）。

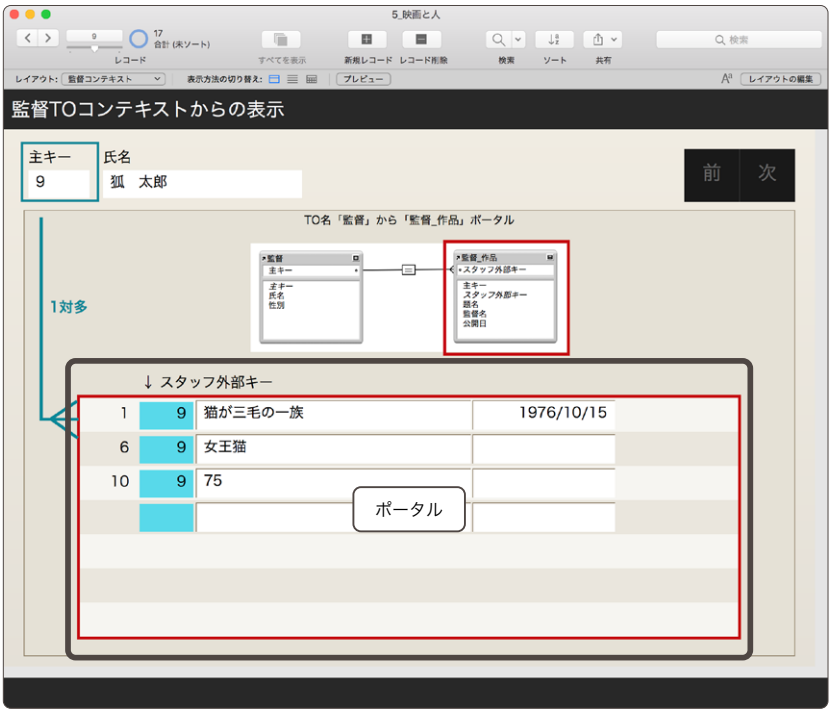


図 5-24「監督コンテキスト」レイアウトのポータル

「映画」エンティティと「スタッフ」エンティティの関連、「1 対多」が、実現されている様子を確認しました。

5.3.2 多対多とレイアウト

「多対多」を確認します。「映画コンテキスト」レイアウトに切り替えてください。「スタッフ」エンティティと「映画」エンティティの 2 つめの関連を確認します。レイアウトモードでレイアウト設定を確認してください。このレイアウトには「映画」TO が設定されています。1 レコード目を表示してください。

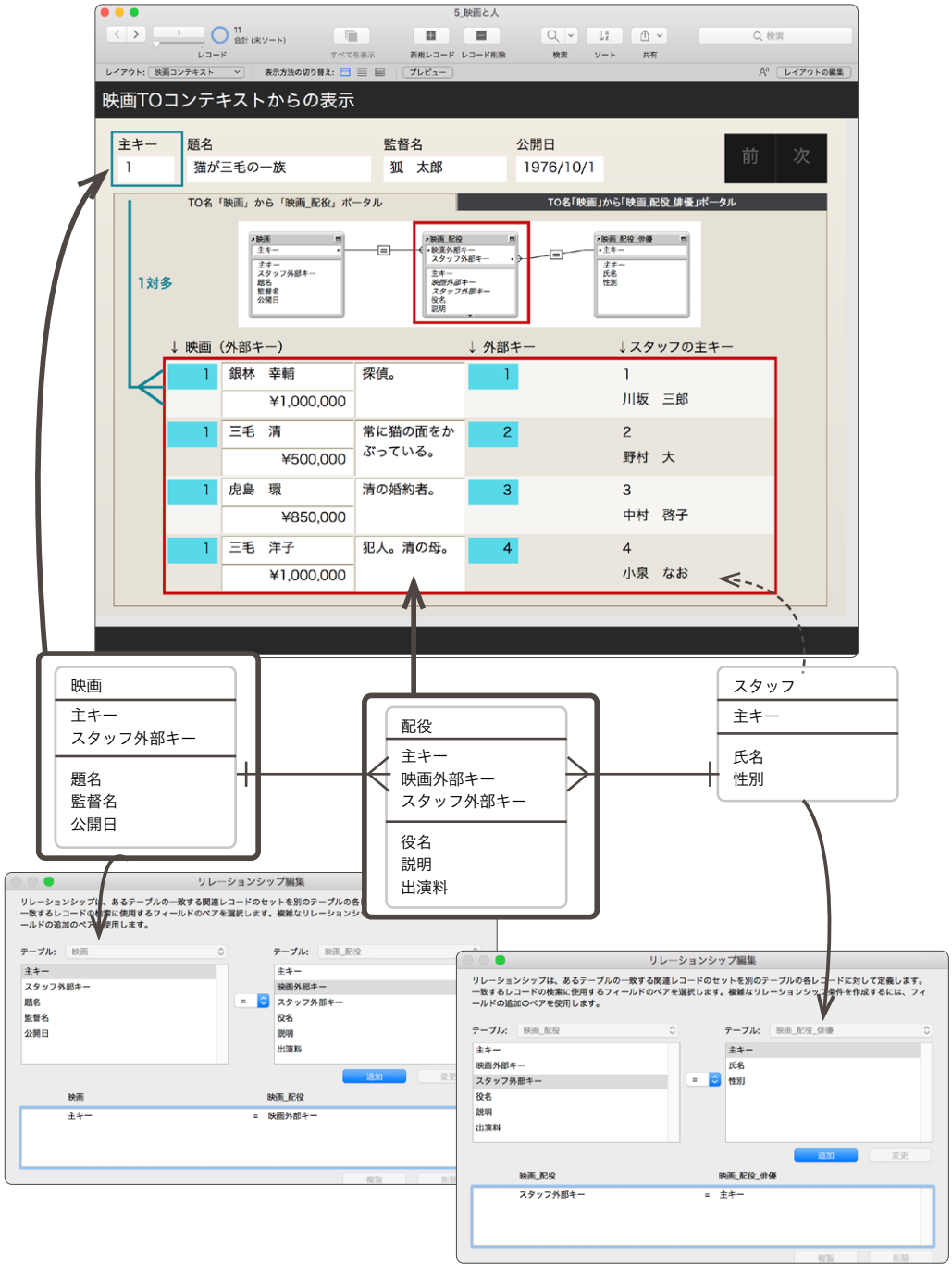


図 5-25 中間テーブルを使って多対多がレイアウト上で表示されている（映画コンテキスト）

ER 図の「スタッフ」エンティティと「映画」エンティティのもう 1 つの関連は「多対多」でした。中間テーブルとして「配役」エンティティを設計しました。

・1 側は、「スタッフ」テーブル、多側が「配役」テーブル
・1 側は、「映画」テーブル、多側が「配役」テーブル

次のことで、多対多の関連を成立させることができます。

・「スタッフ」テーブルの「主キー」フィールドと同じ数字を「配役」テーブルの「スタッフ外部キー」フィールドが持つ
・「映画」テーブルの「主キー」フィールドと同じ数字を「配役」テーブルの「映画外部キー」フィールドが持つ

リレーションシップグラフ上では、アンカーを「映画」TO として、次のように定義しています。

・「映画」テーブルは「映画」TO
・「配役」テーブルは「映画 _ 配役」TO（中間テーブル）
・「スタッフ」テーブルは「映画__配役 _ 俳優」TO

レイアウト上には、1 作の映画に出演している 1 人以上の俳優が表示できるようにポータルが配置されています。例えば、1 レコード目の「川坂　三郎」さんは、2、3 レコード目の映画にも違う役名で出演していることが確認できます。

「俳優コンテキスト」レイアウトへ切り替えてください。このレイアウトは「映画__配役 _ 俳優」TO が設定されています。1 レコード目を表示してください。「川坂　三郎」さんは、複数の役名で違う映画に出演していましたが、それが確認できました。

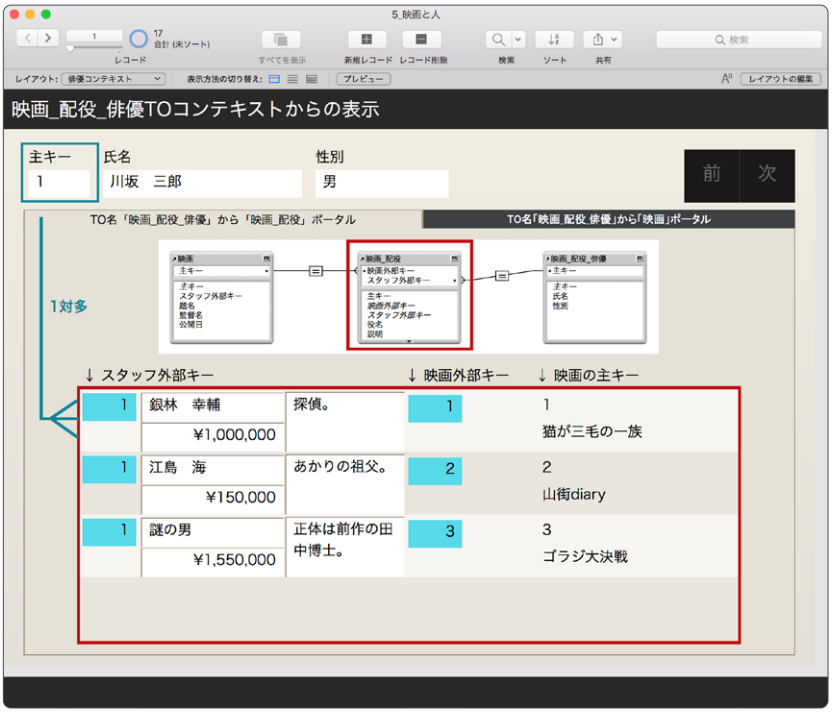


図 5-26 中間テーブルを使って多対多がレイアウト上で表示されている（俳優コンテキスト）

「映画」エンティティと「スタッフ」エンティティの 2 つめの関連、「多対多」が、実現されている様子を確認しました。

多対多とポータルの設定

「多対多」は、レイアウトにポータルを配置する場合に注意する点があります。次図は正しくデータが表示されている状態です。

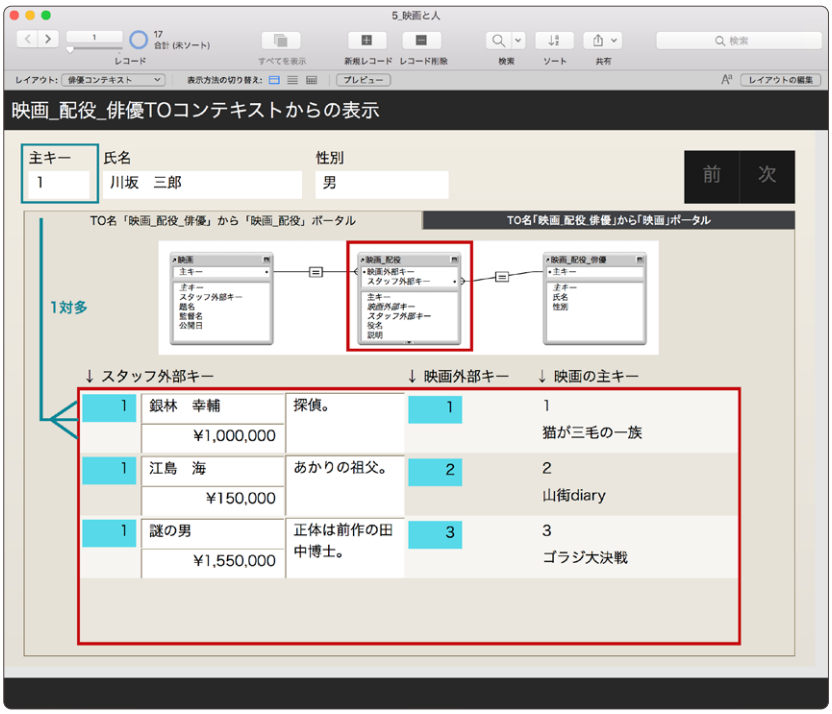


図 5-27 映画の配役が正しく表示されている

しかし、ポータルに指定する TO によっては、正しく関連レコードを表示できません。その例を 2 つのレイアウトのタブパネル上に作成しました。

・「映画コンテキスト」レイアウトの「TO 名「映画」から「映画 _ 配役 _ 俳優」ポータル」タブ
・「俳優コンテキスト」レイアウトの「TO 名「映画 _ 配役 _ 俳優」から「映画」ポータル」タブ

タブをクリックしてください（次ページの図も参照）。どちらも中間テーブル（「映画 _ 配役」TO）ではなく、その先の TO をポータルに指定しています。

ポータルに設定されている TO とは別の TO のフィールドを配置していますがデータが正しく表示されません。関連先の TO（とデータ）であっても次図のような事が起きることに注意が必要です。

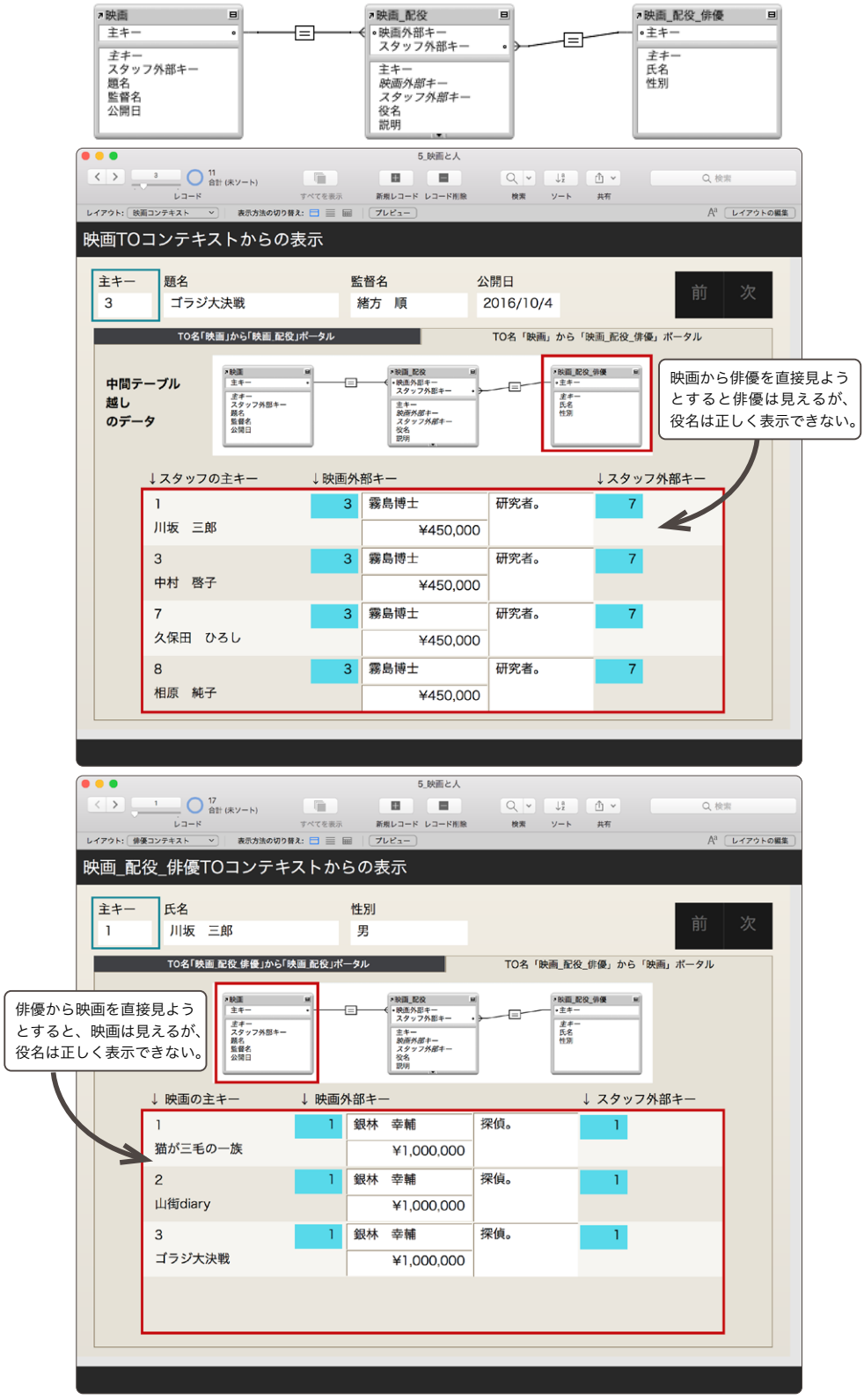


図 5-28 中間テーブルではなく、その先の TO をポータルとしてレイアウトに配置した場合

5.3.3　設計とデータの表示

「どの TO からデータを表示するのか」が、とても重要です。そして、設計した関連をカスタム App で実現するときには、次のような作業が必要だとわかります。

・正規化したエンティティを実体テーブルにする
・実体テーブルに、主ーと外部キーなど、照合フィールドとなる項目を作成する
・関連を定義するのは実体テーブルではなく仮想テーブルの TO
・リレーションシップグラフで関連を定義する（設計を元にする）
・カスタム App のメニューなどに応じて TOG を分ける
・定義した関連の論理に従ってデータを表示するには、適切な TO、つまりコンテキストが設定されたレイアウトやレイアウトオブジェクトなどで指定する必要がある

FileMaker Master Book：初級編の図を（読んだことがある方は）思い出してください。

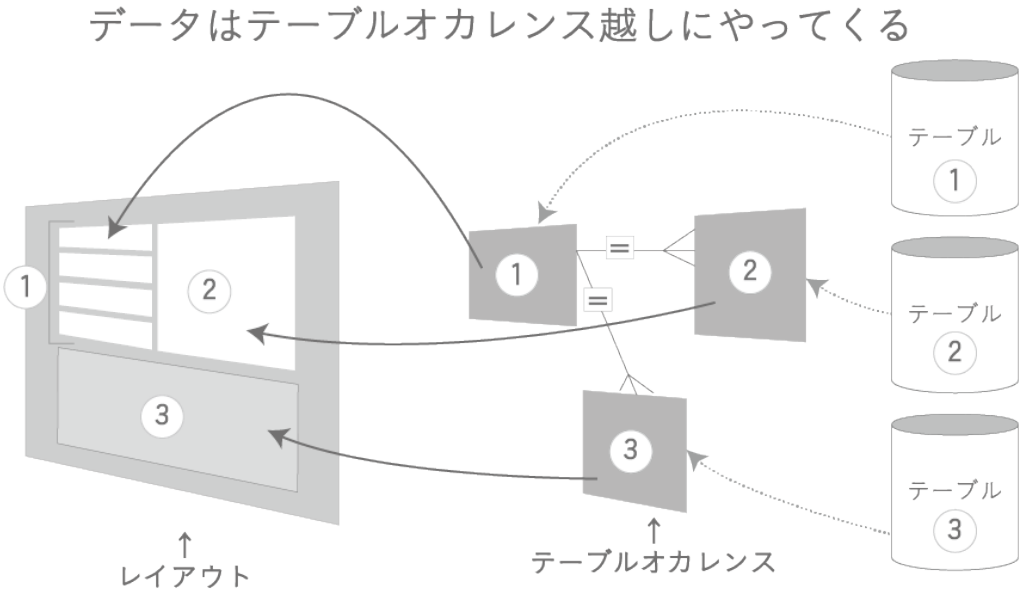


図 5-29 レイアウトとデータの関係

5.4　関連とデータ

関連について改めて確認します。

関連テーブル
ある TO にリレーションシップグラフ上でなんらかの関連の定義がなされている TO のことを、「**関連テーブル**」と呼びます。

関連レコード
関連テーブルのレコードの内、関連の定義による照合の結果、条件に合致して「関連がある」とされたものは「**関連レコード**」と呼びます。

非関連○○
必ずしも、関連テーブルに入力されているすべてのレコードが関連レコードになるわけではありません。関連でないものは「**非関連○○**」と呼ばれます。

5.4.1　照合条件

関連を設計するときに考えるのは、2 つのテーブルのデータがどのような関係（1 対多、多対多など）であるべきかということです。その関連を成立させるために照合条件があります。例えば、次のようなことです。

・あるテーブルのフィールドのデータを、他のテーブルのフィールドのデータと「**照合**」したとき「**共通の値（＝）**」だったら関連がある

照合条件は、リレーションシップグラフ上の関連の定義で『関連があるかないか』を FileMaker Pro が判断するために指定するものです。

・照合フィールド
・照合の演算子

の 2 つを指定します。これらを変更すると関連レコードが変わります。

5.4.2　照合フィールド

照合に使われるフィールドが「**照合フィールド**」です。［リレーションシップ編集］ダイアログで設定します。設定にはルールがあります。

・1 つのフィールドには必ず相手先テーブルの 1 つのフィールドをペアとして指定し照合の条件を指定する

※ 2 つ以上のフィールドのデータを計算式などで結合したフィールドを照合フィールドとして指定することも可能です。「複合キー」などと呼ばれます。

照合しているのは索引
照合には、照合フィールドの索引情報が使われます。索引は第 7 章で解説します。テキスト、数字、日付、時刻、タイムスタンプのフィールドタイプと、これらの結果タイプが設定されている計算タイプフィールド、グローバル格納フィールドも照合フィールドとして指定可能です。

複数の条件を指定する関連

これまで、照合フィールドは 1 ペアでしたが、複数のフィールドを照合フィールドとして指定することが可能です。設定ルールは同様です。

・複数のフィールドを照合フィールドとして指定する場合は、複数のペアがテーブル間に指定され、照合の条件もペアごとに指定する

例えば、「顧客情報」テーブルと「見積情報」テーブルがあるとします。「1 対多」の関連です。

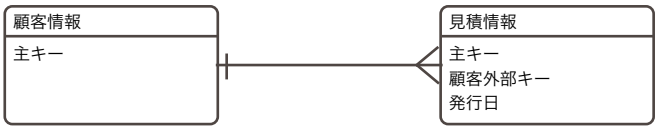


図 5-30「顧客情報」テーブルと「見積情報」テーブルの関係

顧客ごとの見積を確認するために、リレーションシップグラフに 1 ペアの関連が設定されています。

ここから、照合キーの記述で、「フィールド」を省略します。

・「顧客管理」TO の「主キー」＝「顧客管理 _ 見積」TO の「顧客外部キー」

「顧客管理」TO から、顧客レコードごとの見積レコードが関連レコードとなります。

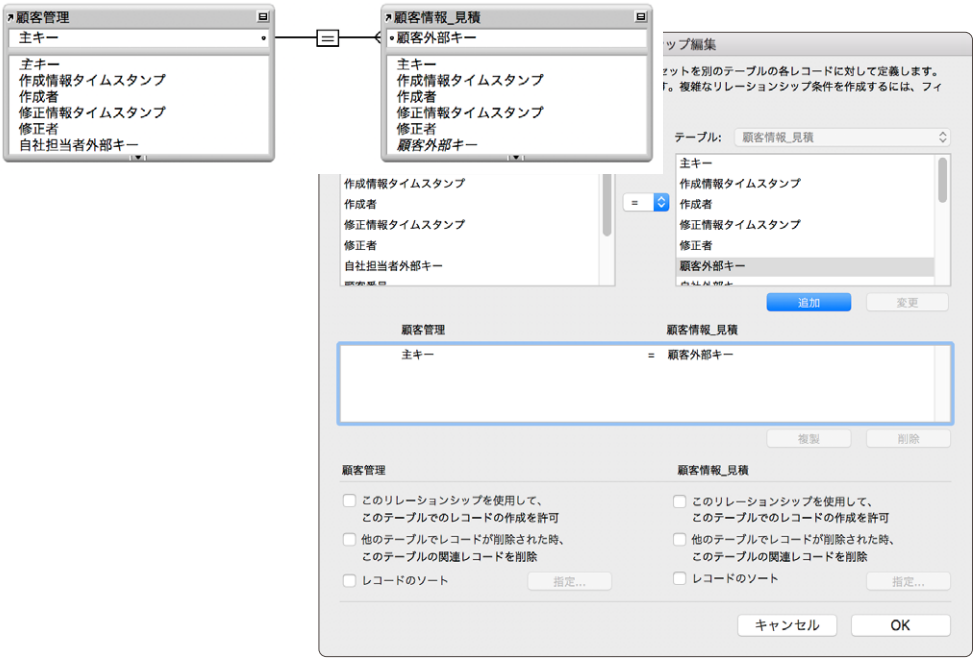


図 5-31 1 つ目の条件

これに、

・顧客ごとに、2018/4/1 以降に発行した見積レコードを表示したい

という要件が必要になったら、「顧客情報」テーブルに「発行日基準」フィールド（日付タイプ）を作成して「2018/4/1」を入力します。そして、次のように 2 ペアの照合条件を設定します。

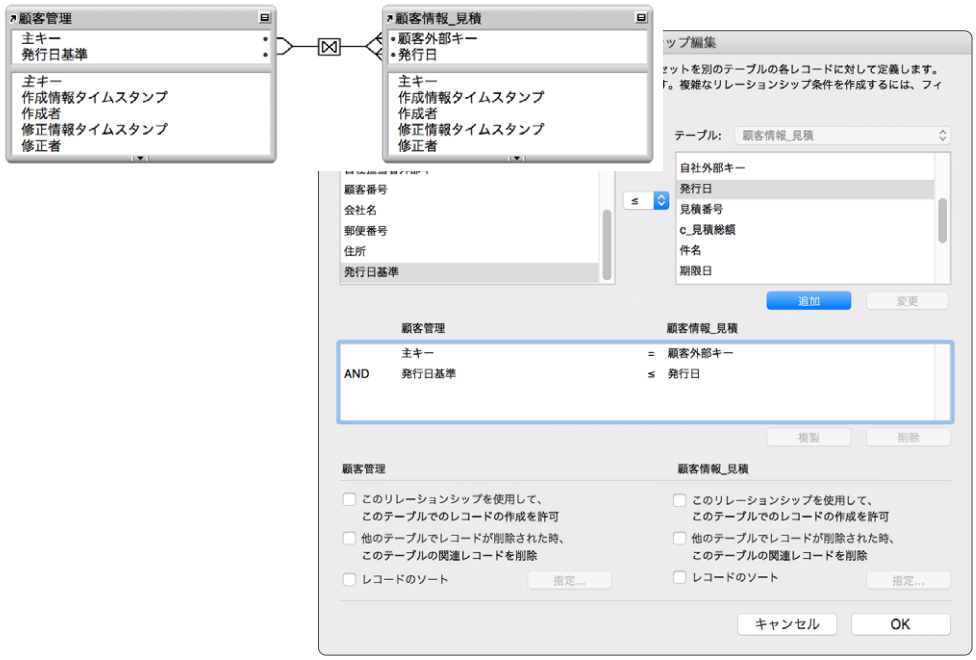


図 5-32「顧客情報」テーブルと「見積情報」テーブルの関係（2 つの照合条件）

・「顧客管理」TO の「主キー」　　＝「顧客管理 _ 見積」TO の「顧客外部キー」
・「顧客管理」TO の「発行日基準」≦「顧客管理 _ 見積」TO の「発行日」

2 ペアの照合フィールドが設定された関連で、2018/4/1 以降の発行日を持つ見積レコードに絞り込むことができました。

範囲で絞り込む

範囲を指定することもあります。次のような要件では、
・顧客ごとに、2018/4/1 から 2019/3/31 の範囲で絞り込んで見積レコードを表示したい

この場合、「顧客情報」テーブルに 2 つのフィールドを作成します。
・「発行日年度開始」フィールド（日付タイプ）を作成して「2018/4/1」を入力
・「発行日年度終了」フィールド（日付タイプ）を作成して「2019/3/31」を入力

そして次のように関連を設定します。
・「顧客管理」TO の「主キー」　　＝「顧客管理 _ 見積」TO の「顧客外部キー」
・「顧客管理」TO の「発行日年度開始」　≦「顧客管理 _ 見積」の「発行日」
・「顧客管理」TO の「発行日年度終了」　≧「「顧客管理 _ 見積」の「発行日」

これで、2018/4/1 から 2019/3/31 の範囲の発行日で、関連レコードが絞り込めるようになります。

これらの方法では、「顧客情報」テーブルの「発行日基準」、「発行日年度開始」、「発行日年度終了」フィールドに、すべてのレコードで「2018/4/1」などが入力されている必要があります。

「発行日基準」、「発行日年度開始」、「発行日年度終了」フィールドをグローバル格納フィールドにして、日付を入力できるようにすることもできます。例えば、カスタム App の起動時にスクリプトを使って適切な日付を入力すると、年度が切り替ったときなど対応が容易になります。（グローバル格納フィールドは、参照元の照合フィールドとしてであれば指定可能です。第 7 章を参照してください）。

※ 複数の条件は日付と組み合わせるだけではなく「取引中の顧客」で「自分の担当である」なども可能です。

なお、ポータルフィルタでもポータルに表示される関連レコードを絞り込むことが可能です。但しポータルオブジェクトごとに設定が必要となります。

5.4.3 参照元と参照先

関連が定義されている TO をどの立場から見るかによって次のように呼び分けます。

- **参照元**
- **参照先**

次図の関係では、例えば「監督」TO から「監督 _ 作品」TO のデータを見るとき、

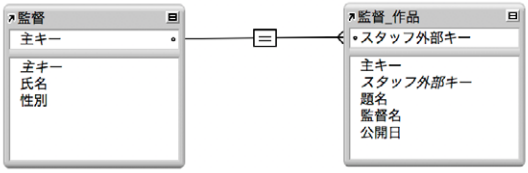


図 5-33 1 対多

- 「監督」TO が参照元
- 「監督 _ 作品」TO が参照先

となります。「監督 _ 作品」TO から「監督」TO を見ると、

- 「監督 _ 作品」TO が参照元
- 「監督」TO が参照先

どの参照元から参照先を見ているのか、はとても重要です。参照先のテーブルと関連レコードについてなにが見えるはずなのかを意識しながら作業します。

多対多の場合は、

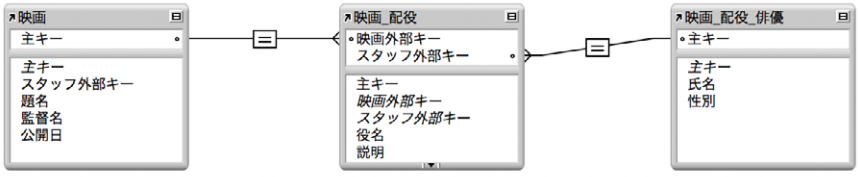


図 5-34 多対多

・「映画」TO（参照元）から見ると「映画 _ 配役」TO、「映画 _ 配役 _ 俳優」TO は両方とも参照先
・「映画 _ 配役」TO（参照元）から見ると「映画」TO、「映画 _ 配役 _ 俳優」TO は両方とも参照先
・「映画 _ 配役 _ 俳優」TO（参照元）から見ると「映画」TO、「映画 _ 配役」TO は両方とも参照先

参照先であっても、ポータルに指定する参照先によっては正しくデータが表示されません。

※ 本書で採用しているアンカー / ブイの TO の並べ方と TO の命名ルールは、左側が参照元 TO、右側が参照先 TO、とわかりやすくするためのものです。特に TO 名は常に左から右を参照していると考えられるので、計算式やスクリプトでの作業時に便利です。

5.4.4 関連によるレコード作成

関連のオプションを設定すると、関連テーブルに関連レコードを作成可能です。このようなレコード作成には、

・リレーションシップグラフで関連の定義とオプション設定をする
・レイアウトに上記の設定先の TO のポータルを置く

これらが必要です。まず、関連を定義し［リレーションシップ編集］ダイアログで次の設定をします。

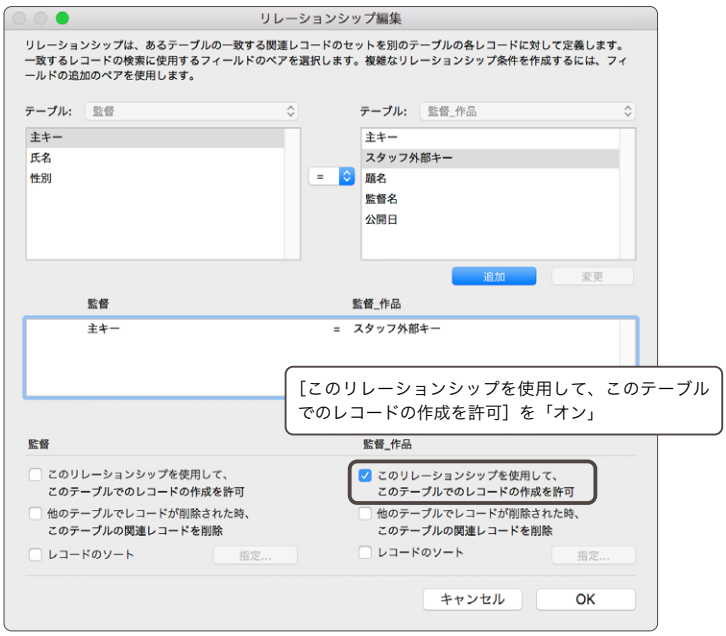


図 5-35［リレーションシップ編集］ダイアログ

レイアウトを作成します。次のように指定します。

・レイアウトに指定する TO　→　主キー側／参照元
・ポータルに指定する TO　→　外部キー側／参照先

ポータルでレコードを作成すると、

・現在のレコード（参照元テーブル）の主キーのデータが参照先テーブルに新しく作成されたレコードの外部キーフィールドに自動入力される

そして、関連が成立します。「映画と人 .fmp12」に同様の設定をしました。「監督コンテキスト」レイアウトへ切り替えて、次図を参照しながら確認してください。

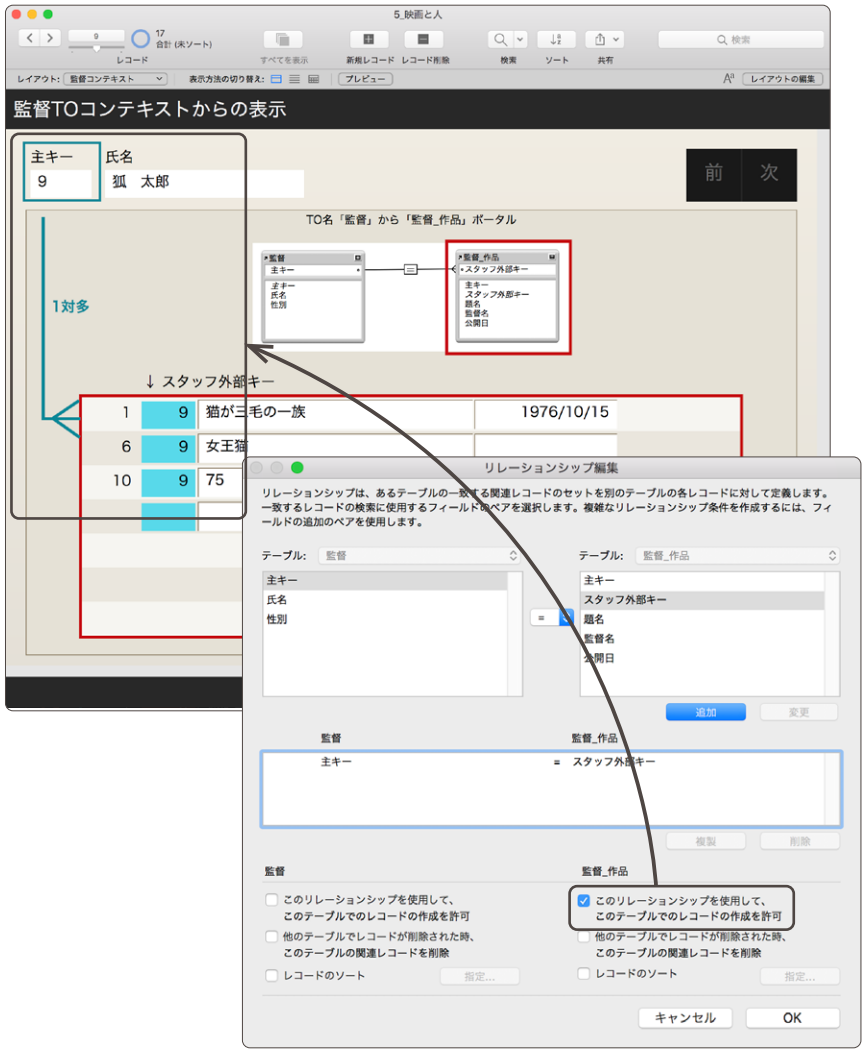


図 5-36 自動で入力される

やってみよう

ポータルに新しい作品名を入力すると、外部キーに現在表示されている監督名の左にある「主キー」の数値が入力されることを確認してみましょう。

その他の方法

自動入力以外にも参照元レコードの主キーを参照先レコードの外部キーに入力する方法があります。ユーザが入力することです。例えば、値一覧を使用したり、スクリプトを利用したりします。

（ここから、本書に付属している演習ファイルの「5 章開始時」フォルダの「5_ 料理と食材 .fmp12」ファイルをベースに進めていきます）

多対多と関連のオプション（新規レコード作成）を実際に作業します。

題材は料理と食材です。「料理と食材 .fmp12」ファイルを開きます。次の要件があります。

・食材はいろんな料理に使われる
・1 つの料理には、1 回に複数の食材を使用する

2 つのエンティティが考えられます。多対多です。



図 5-37 食材と料理は　多対多

中間テーブルを加え、簡単な項目も考え、エンティティの種類も考えました。

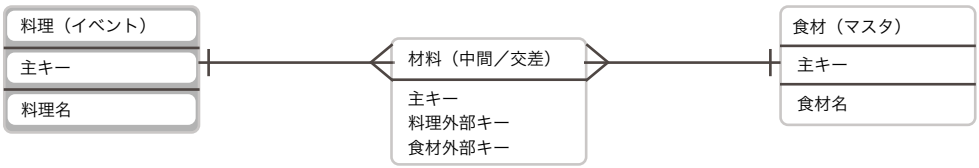


図 5-38 正規化したテーブルの ER 図

「5 章開始時」フォルダ

演習 5-1　多対多と関連のオプションを確認する 1：関連を定義する

- **使用ファイル：5_ 料理と食材 .fmp12**

(1)　ファイルになにが準備されているか確認します。
a.　［データベースの管理］ダイアログを開きます。「テーブル」タブを確認します。
b.　「料理」「食材」の 2 テーブルが作成されています。

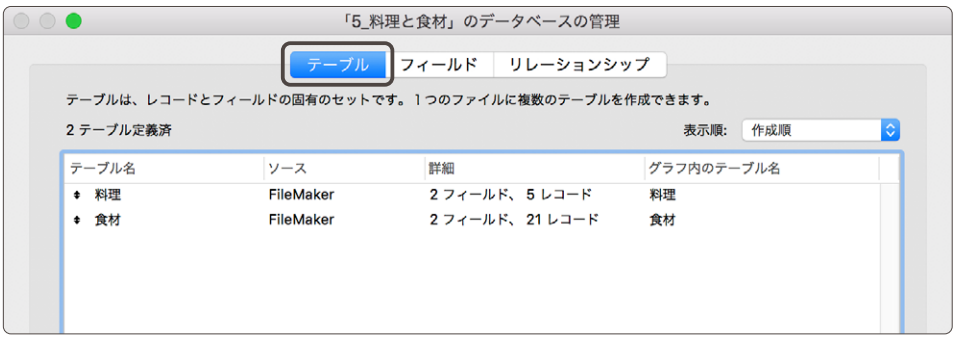


図 5-39 料理と食材 .fmp12

(2)　中間テーブルとフィールドを作成します。

a.　「テーブル」タブで「材料」テーブルを作成します。

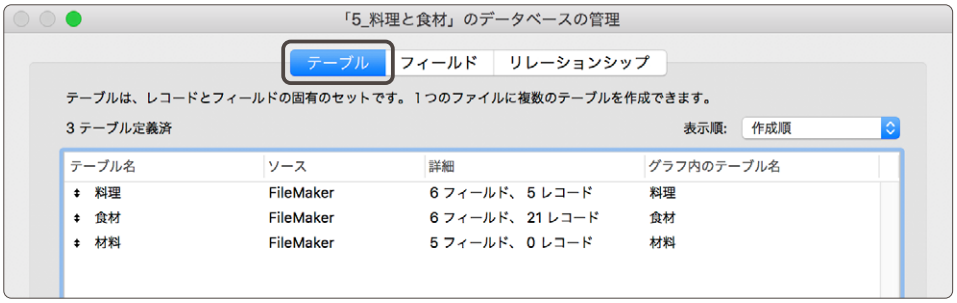


図 5-40「材料」テーブルを作成

b.　「材料」テーブル行をダブルクリックして「フィールド」タブに移動します。

c.　自動作成された「主キー」フィールドのタイプを「数字」に変更します。

d.　続いて、「主キー」フィールドのオプションを［シリアル番号］に変更します。

1)　「主キー」フィールドをクリックします。

2)　［フィールド名］を確認して［オプション ...］をクリックします。

3)　［フィールド「主キー」のオプション］ダイアログが開きます。

・［シリアル番号］にチェックを　**オン**

・［作成］は　「作成時」を選択

・［次の値］は　そのまま（「1」になっているはずです）

・［増分］は　そのまま（「1」になっているはずです）

4)　［計算値］のチェックを　**オフ**　にします。

5)　［データ入力時の値変更の禁止］は　**オン**　のままにします。



図 5-41　フィールドオプション設定

e.　次表の灰色の行のフィールドを作成します。

材料テーブルのフィールド			
フィールド名	**タイプ**	**入力値の自動化オプション**	**入力値の制限オプション**
主キー	数字	シリアル番号 値変更禁止	空欄不可 ユニークな値
作成者	テキスト	作成アカウント名 値変更禁止	空欄不可
修正者	テキスト	修正アカウント名 値変更禁止	空欄不可
作成情報タイムスタンプ	タイムスタンプ	作成情報タイムスタンプ 値変更禁止	西暦 4 桁の日付、空欄不可
修正情報タイムスタンプ	タイムスタンプ	修正タイムスタンプ 値変更禁止	西暦 4 桁の日付、空欄不可
料理外部キー	数字	なし	なし
食材外部キー	数字	なし	なし
説明	テキスト	なし	なし
分量	数字	なし	なし

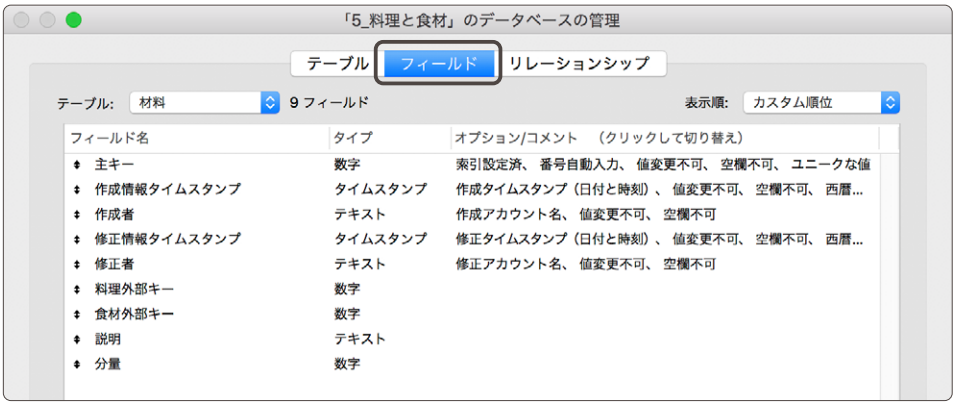


図 5-42 中間テーブル「材料」を作成し、キーのフィールドも作成

(3)　「リレーションシップ」タブに移動します。関連を定義します。3 つの TO を
料理　　材料　　食材
のように並べ替えます（ER 図の順番と同じです）。次図も確認してください。

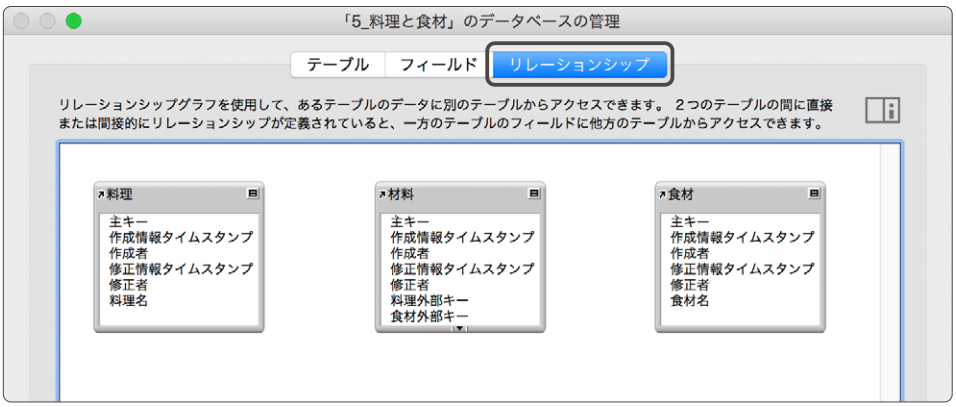


図 5-43 TO を配置

(4)　「料理」TO と「材料」TO を関連させます。
「料理」TO の「主キー」フィールドと「材料」TO の「料理外部キー」フィールドを繋げます。

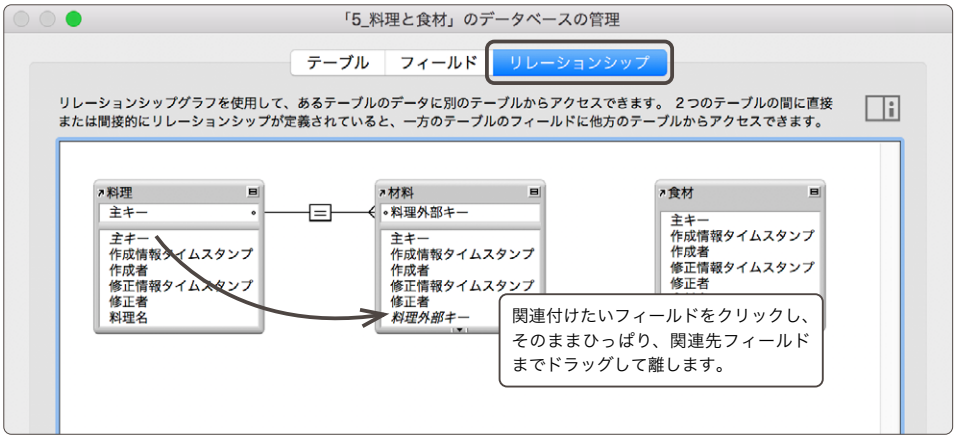


図 5-44 関連付ける

(5)　関連の定義とオプションの設定をします。

a.　繋いでいる線か、四角形をダブルクリックして［リレーションシップ編集］ダイアログを開きます。

b.　照合フィールドを確認します（違っていたら直してください）。

・左側が「料理」TO で　「主キー」フィールド
・右側は「材料」TO で　「料理外部キー」フィールド
・照合条件の演算子は　「=」

c.　オプションを設定します。
右側の「材料」TO の［このリレーションシップを使用して、このテーブルでのレコード作成を許可］を　**オン**　にします。

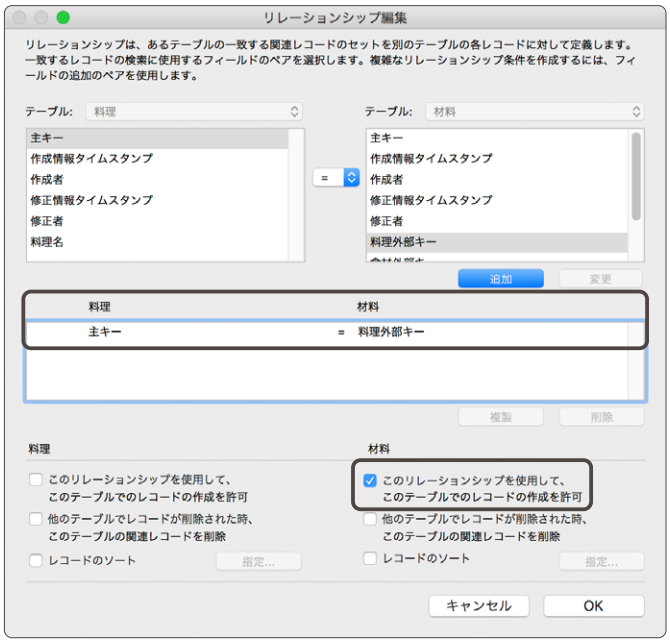


図 5-45 リレーションシップの編集

(6)　「材料」TO の名前を「料理 _ 材料」に変更します。

a.　「材料」TO をダブルクリックします。

b.　［テーブルを指定］ダイアログが表示されます。

c.　テーブルは「材料」を指定したままで、名前を入力します。

・［名前］は　　「料理 _ 材料」

d.　「OK」をクリックします。

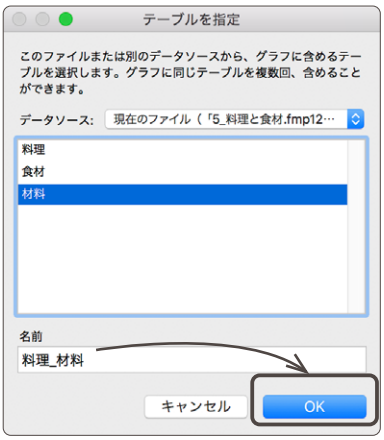


図 5-46　TO の名前を変更

(7)　引き続き関連を定義します。「食材」TO と「料理__材料」TO を関連させます。「食材」TO の「主キー」フィールドと「料理__材料」TO の「食材外部キー」フィールドを繋げます。

(8)　関連の定義を確認します。

a.　［リレーションシップ編集］ダイアログを開きます。

b.　照合フィールドを確認します（違っていたら直してください）。

・左側が「料理__材料」TO で　　「食材外部キー」フィールド

・右側は「食材」TO で　　「主キー」フィールド

・照合条件の演算子は　　「=」

c.　オプションはなにも設定しません。

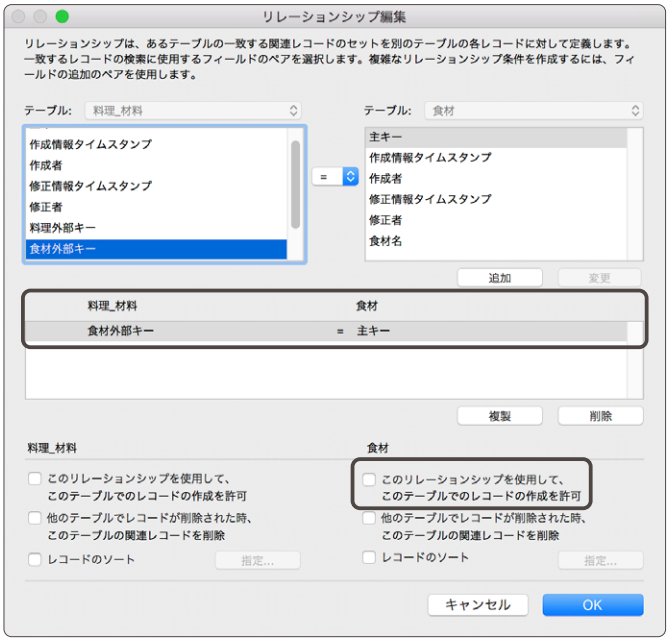


図 5-47　関連付けとリレーションシップの編集

（9） 「食材」TO の名前を「料理 _ 材料 _ 食材」に変更します。

（10） 構造ができました。
TO を揃えたり、等間隔に並べたりなど、整えてください。

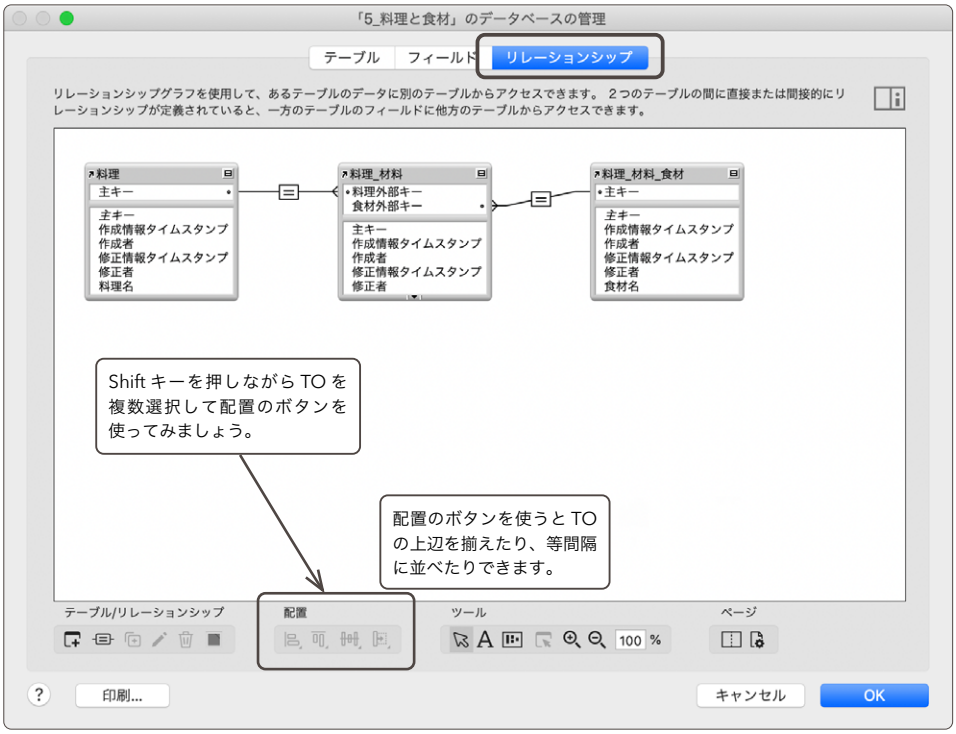


図 5-48 できあがった TOG

（11） ［OK］をクリックして［データベースの管理］ダイアログを閉じます。

演習ここまで

設計からカスタム App の構造をつくりました。次は、**定義した関連を使ったレイアウトをつくります。**

演習 5-2　多対多と関連のオプションを確認する 2：レイアウトを作成する

- **使用ファイル：5_ 料理と食材 .fmp12**

(1)　レイアウトモードに切り替えます。

(2)　レイアウトを作成します。

a.　［レイアウト］–［新規レイアウト / レポート］メニューを選択します。
b.　［新規レイアウト / レポート］ダイアログが表示されます。
c.　［レコードを表示］は「料理」TO を選択します。
d.　レイアウト名は「料理と材料」にします。
e.　「コンピュータ」の「フォーム」を選択して［完了］をクリックします。



図 5-49　レイアウトを作成

(3)　作成された「料理と材料」レイアウトにフィールドを配置します。

a.　左側のパネルを表示します。
b.　「フィールド」タブをクリックして次の設定を行います。
・現在のテーブルは　「料理」
・フィールド配置は　「縦」
・フィールドラベルは　「左」
・コントロールスタイルは　「編集ボックス」

c.　「主キー」と「料理名」フィールドを配置します。
フィールドをクリックし、マウスでレイアウト上のボディパートにドラッグします。
(下にポータルを配置するので、やや上に置いてください／次ページの図も参照してください)

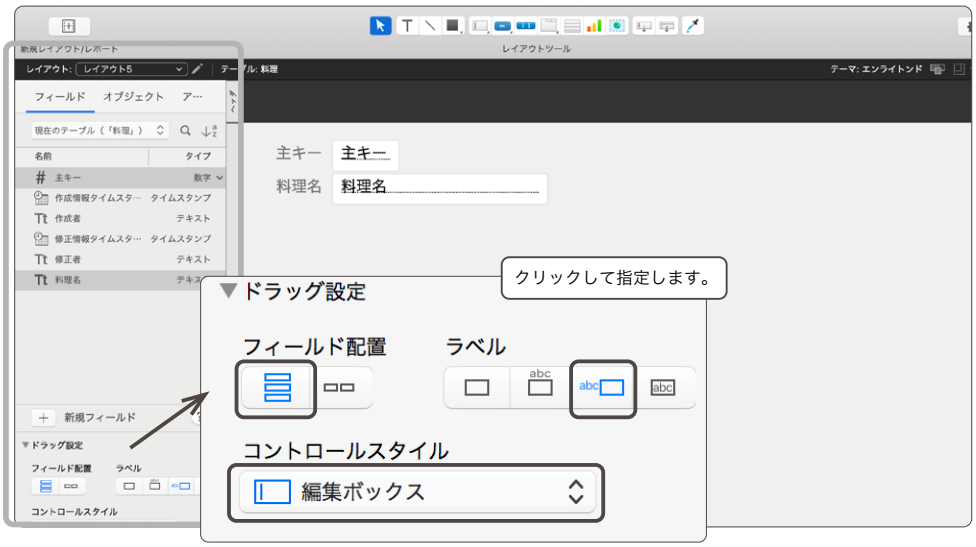


図 5-50 フィールドを配置

(4) 「料理 _ 材料」TO のポータルを配置します（次図も参照してください）。

a. ［ポータルツール］アイコンをクリックします。
ポータルを配置したい位置でドラッグして四角形を描くようにしてマウスボタンを離すと［ポータル設定］ダイアログが表示されます。



図 5-51 ステータスツールバーの［ポータルツール］アイコン

b. ［ポータル設定］ダイアログでは次の設定を行います。

・［レコードを表示］は　「料理 _ 材料」TO を選択
・［垂直スクロールを許可］を　**オン**
・［代替の行状態を使用］を　**オン**
・［アクティブな行状態を使用］を　**オン**



図 5-52 ポータル設定ダイアログ

c. ［OK］をクリックします。
d. ［ポータルにフィールドを追加］ダイアログが表示されます。
「料理 _ 材料」TO のフィールドを指定します。
・主キー
・料理外部キー
・食材外部キー
・説明
・分量

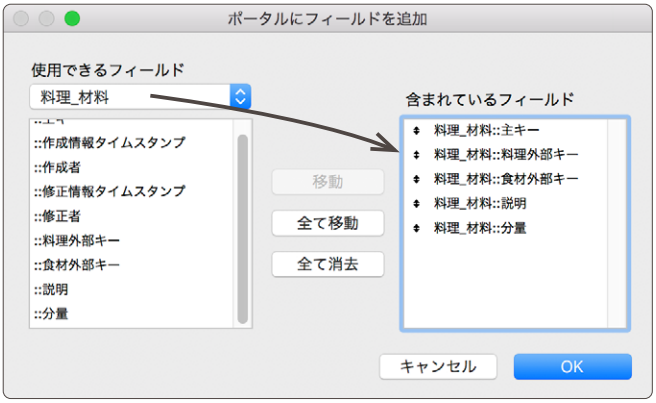


図 5-53「材料」TO のフィールドを指定する

e. 別の TO のフィールドを指定します。
［使用できるフィールド］の TO 名を「料理 _ 材料 _ 食材」に切り替えます（関連テーブルとして表示されているはずです）。次のフィールドを指定します。
・食材名

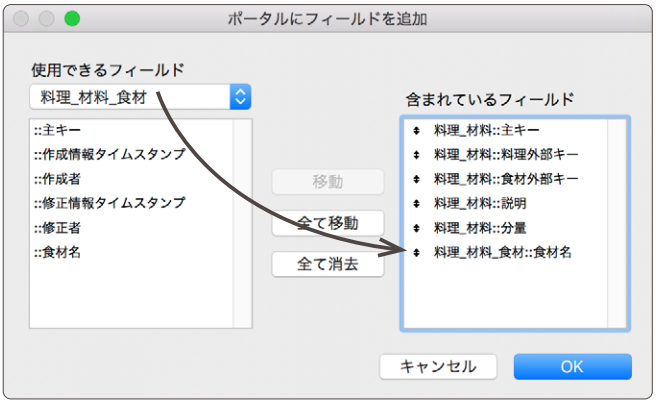


図 5-54「食材」TO のフィールドを追加指定する

f. ［OK］をクリックします。
g. ポータルが配置されます。

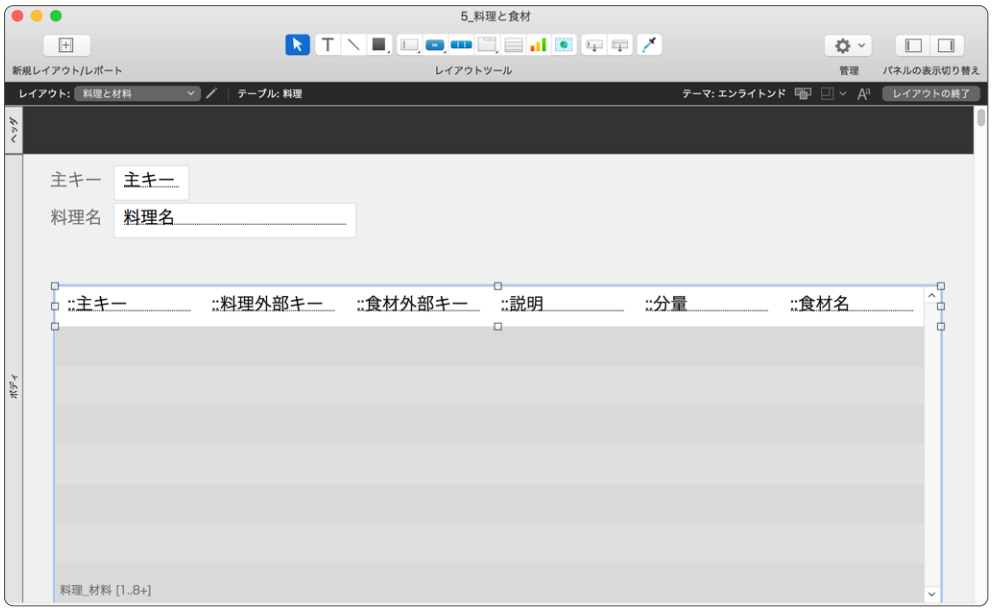


図 5-55 ポータルが配置された

(5)　入力が必要なフィールドがわかりやすいように色をつけます。
右パネルのインスペクタの「外観」タブのグラフィックを使います（次図も参照してください）。

a.　［Shift］キーを押しながら「説明」「分量」「食材名」の 3 つのフィールドを選択します。

b.　インスペクタの「外観」タブの［塗りつぶし］で、任意の色を設定します。

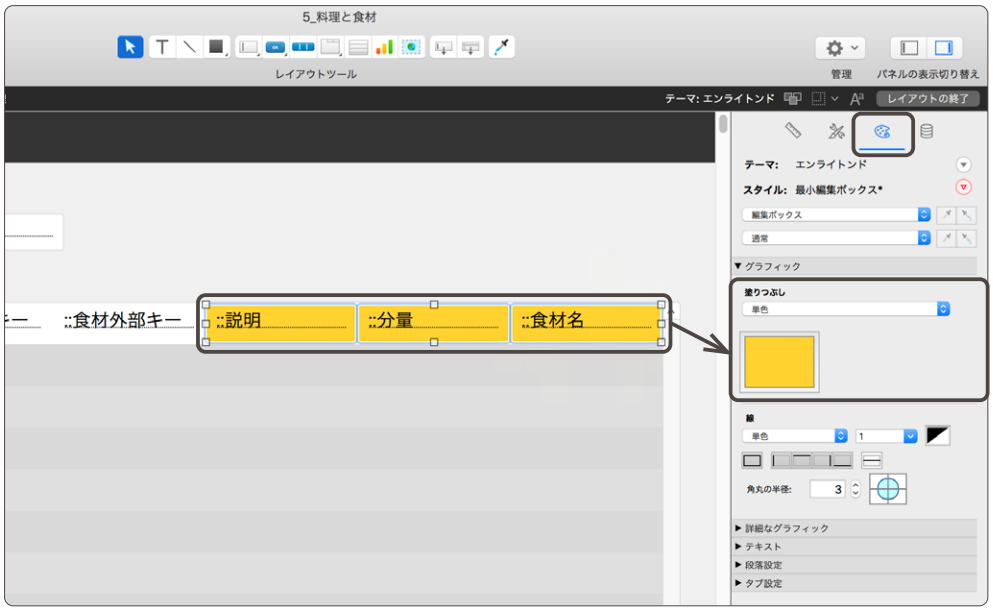


図 5-56 ポータルのフィールドに色をつける

(6)　作業に余裕があれば、次図のようにフィールドのラベルをポータルの上に置きます。

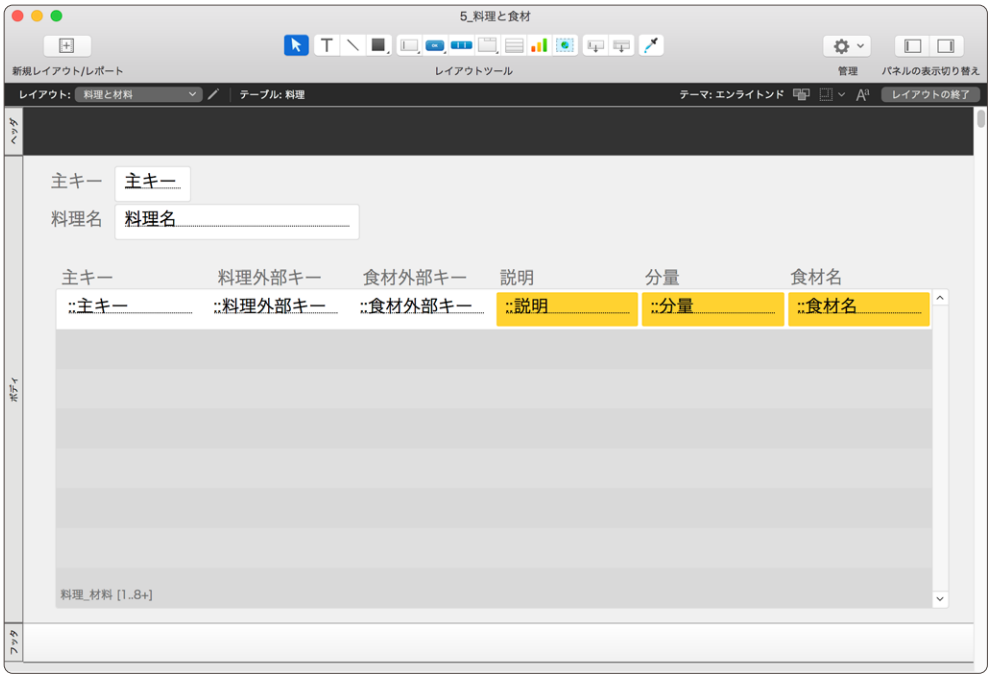


図 5-57 フィールドのラベルを配置

(7)　レイアウトを保存してブラウズモードに戻ります。

演習ここまで

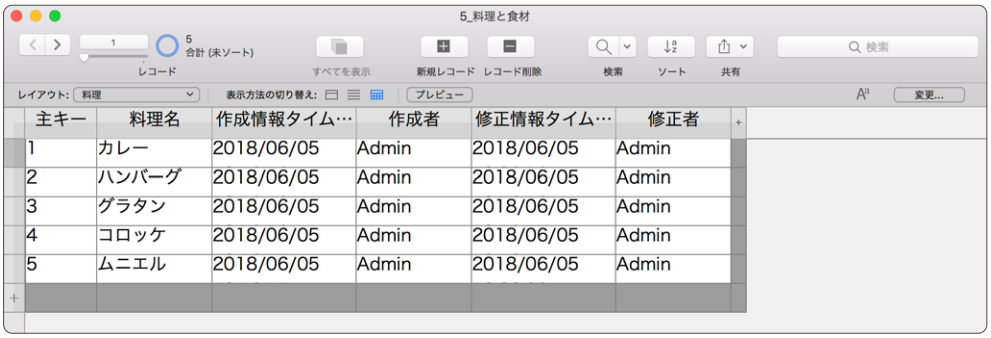
これで、構造とレイアウトができました。ポータルにレコードが入力できそうです。

「料理と食材 .fmp12」の「料理」テーブルには、あらかじめマスタとなるデータを入力してあります。**データを確認してテスト入力をします。**

演習 5-3　多対多と関連のオプションを確認する 3：データの確認と入力

- **使用ファイル：5_ 料理と食材 .fmp12**

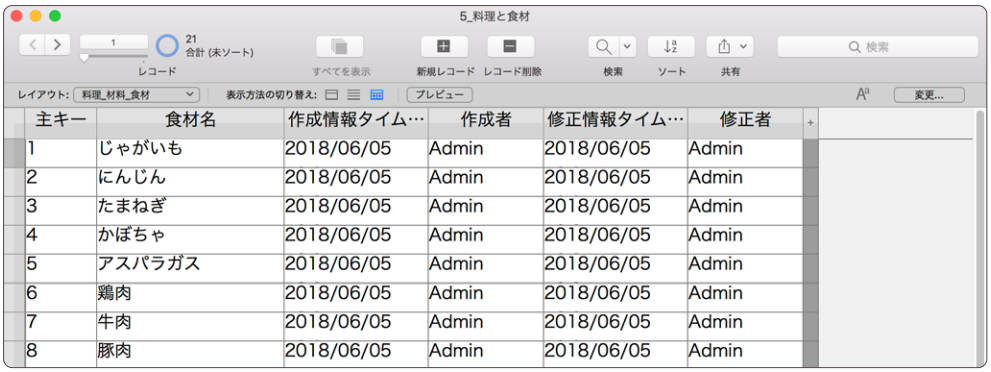
(1)　「料理」レイアウトへ切り替えます（このレイアウトは「料理」TO のデータが表示されています）。
5 つの料理が登録されていることを確認します。



主キー	料理名	作成情報タイム…	作成者	修正情報タイム…	修正者
1	カレー	2018/06/05	Admin	2018/06/05	Admin
2	ハンバーグ	2018/06/05	Admin	2018/06/05	Admin
3	グラタン	2018/06/05	Admin	2018/06/05	Admin
4	コロッケ	2018/06/05	Admin	2018/06/05	Admin
5	ムニエル	2018/06/05	Admin	2018/06/05	Admin

図 5-58「料理」レイアウトのデータ

(2)　「料理 _ 材料 _ 食材」レイアウトへ切り替えます（このレイアウトは「料理 _ 材料 _ 食材」TO のデータが表示されています）。
21 の食材が登録されていることを確認します。3 件目に「たまねぎ」が入力されていることを確認します。



主キー	食材名	作成情報タイム…	作成者	修正情報タイム…	修正者
1	じゃがいも	2018/06/05	Admin	2018/06/05	Admin
2	にんじん	2018/06/05	Admin	2018/06/05	Admin
3	たまねぎ	2018/06/05	Admin	2018/06/05	Admin
4	かぼちゃ	2018/06/05	Admin	2018/06/05	Admin
5	アスパラガス	2018/06/05	Admin	2018/06/05	Admin
6	鶏肉	2018/06/05	Admin	2018/06/05	Admin
7	牛肉	2018/06/05	Admin	2018/06/05	Admin
8	豚肉	2018/06/05	Admin	2018/06/05	Admin

図 5-59「料理 _ 材料 _ 食材」レイアウトのデータ

(3)　「料理 _ 材料」レイアウトへ切り替えます（このレイアウトは「料理 _ 材料」TO のデータが表示されています）。なにも入力されていません

(4)　わかりやすいように表示を変更します。

a.　表形式に切り替えてください。
b.　レイアウトバーの右側にある［変更 ...］をクリックします。
c.　［表形式の変更］ダイアログが表示されます。
d.　左下の［+］をクリックします。
e.　［フィールドを追加］ダイアログが表示されます。
f.　［Shift］キーを使ってすべてのフィールドを選択します。
g.　［OK］を 2 回クリックします。



図 5-60 フィールドを設定後の［表形式の変更］ダイアログ

(5)　表形式にフィールドが表示されます。



図 5-61「料理 _ 材料」レイアウト

(6)　料理に必要な食材を入力して多対多が成立するか試します。

(7)　「料理と材料」レイアウトへ切り替えます。
1 件目のレコードを表示してください。料理名は「カレー」です。
ポータルに食材として　たまねぎ　について入力することにします。

(8)　「食材名」に　たまねぎ　と入力してみます。

(9)　しかし「食材名」には入力できません。他の項目ではどうでしょうか。
・「説明」に　　みじんぎり
・「分量」に　　1
と、入力すると「主キー」と「料理外部キー」フィールドにデータが入力されます。

演習ここまで

たまねぎ　を入力するにはどうしたら良いのでしょうか。

演習で入力したポータルの TO は「料理 _ 材料」TO（「材料」テーブル）です。多対多の中間テーブルなので、新しいレコードを作成するには照合フィールドとして設定されている 2 つの外部キーにデータが入力される必要があります。

正しく入力されるようにするには、どうしたら良いのでしょうか。

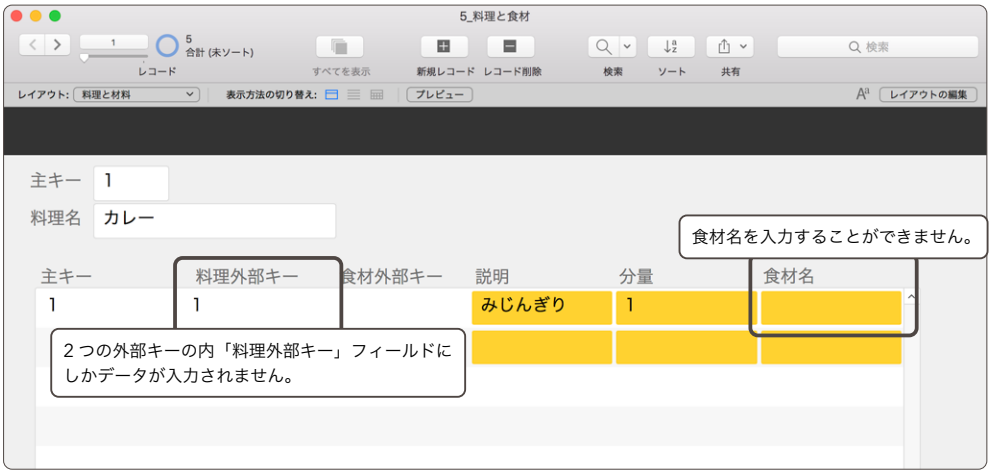


図 5-62 入力できないフィールドがある

演習 5-1 の作業を思い出してください。関連を設定したとき［このリレーションシップを使用して、このテーブルでのレコード作成を許可］を **オン** にしたのは、次の内容です。

・「料理」TO と「料理 _ 材料」TO の間の関連
・「料理 _ 材料」TO 側のオプションを設定

では、「料理 _ 材料」TO と「料理 _ 材料 _ 食材」TO の間にも設定が必要なのでしょうか？違います。実はこの設定をすると新しい食材レコードが登録されてしまうのです。

演習 5-3 で確認しましたが**既に「たまねぎ」は登録されています。そのレコードを利用する必要があります。**「料理と材料」レイアウトにあるポータルでは、

・どの料理なのか
・どの料理の材料なのか

は、FileMaker Pro にわかりますが、

・どの食材なのか（中間テーブルの「料理 _ 材料」TO を越えた先の「料理 _ 材料 _ 食材」TO のこと）

は、わからないのです。ユーザが食材を指定して教えなければなりません。そのために、関連で設定した照合フィールド（「食材外部キー」フィールド）へ適切な外部キーのデータを入力します。

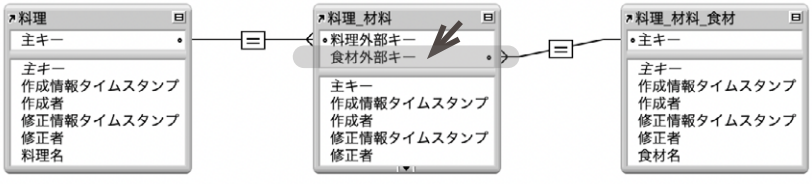


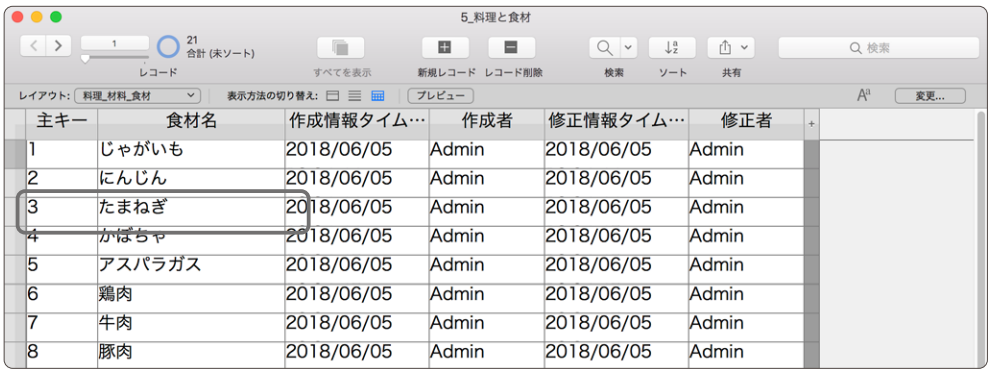
図 5-63「料理 _ 材料」TO の「食材外部」キー（照合フィールド）

次の演習で、**外部キーの手入力をおこなって確認します。**「カレー」の材料として、中間テーブルのレコードに「たまねぎ」の外部キーを入力します。

演習 5-4　多対多と関連のオプションを確認する 4：外部キーの手入力 1

- **使用ファイル：5_ 料理と食材 .fmp12**

(1)　「料理 _ 材料 _ 食材」レイアウトへ切り替えます。
3 件目の「たまねぎ」の「主キー」フィールドのデータは「3」であることを確認します。これで「食材外部キー」フィールドになにを入力したら良いかわかりました。



主キー	食材名	作成情報タイム…	作成者	修正情報タイム…	修正者
1	じゃがいも	2018/06/05	Admin	2018/06/05	Admin
2	にんじん	2018/06/05	Admin	2018/06/05	Admin
3	たまねぎ	2018/06/05	Admin	2018/06/05	Admin
4	かぼちゃ	2018/06/05	Admin	2018/06/05	Admin
5	アスパラガス	2018/06/05	Admin	2018/06/05	Admin
6	鶏肉	2018/06/05	Admin	2018/06/05	Admin
7	牛肉	2018/06/05	Admin	2018/06/05	Admin
8	豚肉	2018/06/05	Admin	2018/06/05	Admin

図 5-64「食材」レイアウトのデータ

(2)　「料理と材料」レイアウトに切り替えます。

(3)　前の演習で片方だけ外部キーが入ったポータルのレコードの「食材外部キー」フィールドに「たまねぎ」の主キーである「3」を入力します。



図 5-65 たまねぎの主キーをポータルに入力すると「たまねぎ」が表示された

(4)　食材名に「たまねぎ」が表示されます。これで多対多の中間テーブルのレコードの関連が成立しました。

演習ここまで

先に中間テーブルにレコードが作成されていて後から「食材」の外部キーを入力しました。本来は、次の演習のように**食材について先に入力する**方が自然です。

演習 5-5　多対多と関連のオプションを確認する 5：外部キーの手入力 2

- **使用ファイル：5_ 料理と食材 .fmp12**

(1)　「にんじん」の外部キーを入力します。にんじんの「主キー」フィールドのデータは「2」です。

(2)　「料理と材料」レイアウトで「たまねぎ」の下の行の「食材外部キー」フィールドに「2」を入力します。

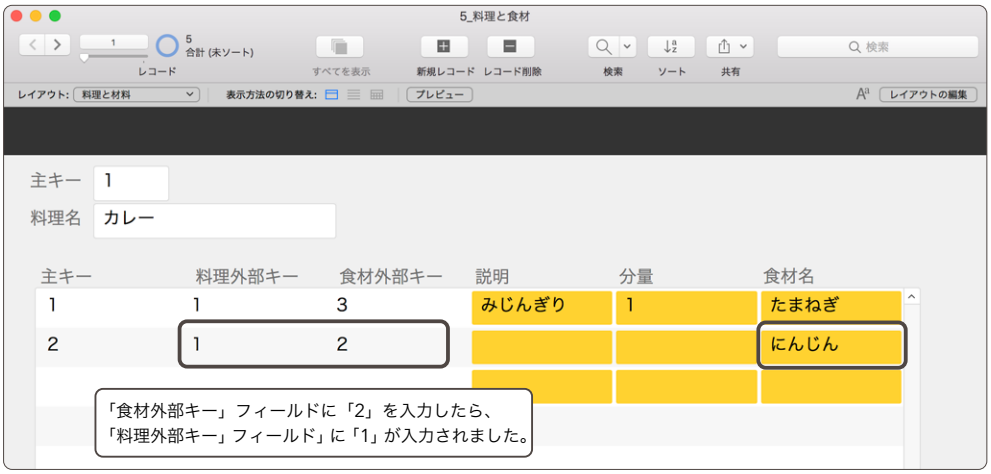


図 5-66 にんじんの主キーをポータルに入力すると 2 つの外部キーが入力された

(3)　自動的に「料理外部キー」フィールド」に「1」が入力され、食材名に「にんじん」が表示されます。

(4)　多対多の中間テーブルのレコードの関連が成立しました。

演習ここまで

料理のための材料を中間テーブルに登録するためには、「食材外部キー」フィールドへ必要な食材の主キーを入力することが必要だとわかりました。外部キーに入力すれば関連が成立するデータの内容がわかっていれば問題ありません。しかし、毎回主キーを調べたり覚えるのは現実的ではありません。

値一覧を使って入力できるようにします。

演習 5-6　多対多と関連のオプションを確認する 6：外部キーを値一覧から入力する

- **使用ファイル：5_ 料理と食材 .fmp12**

(1)　「フィールドの値を使用」した「値一覧」を作成します。

a.　［ファイル］–［管理］–［値一覧 ...］メニューを選択します。

b.　［値一覧の管理］ダイアログが表示されます。

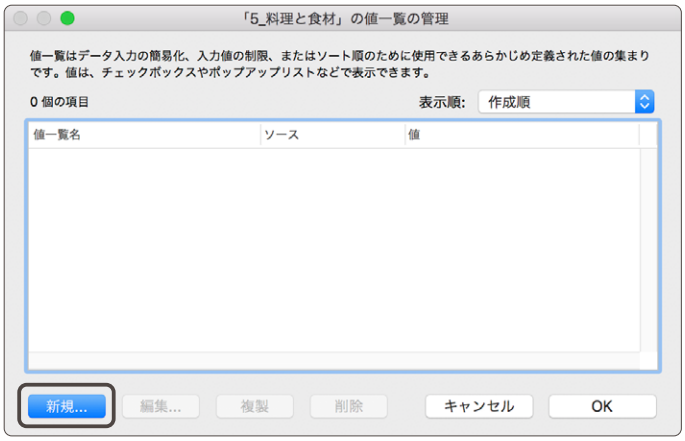


図 5-67［値一覧の管理］ダイアログ

c.　左下の［新規］をクリックします。

d.　［値一覧の編集］ダイアログが表示されます。

e.　［値一覧名］に「食材リスト」と入力します。

f.　［フィールドの値を使用］ラジオボタンをクリックします。

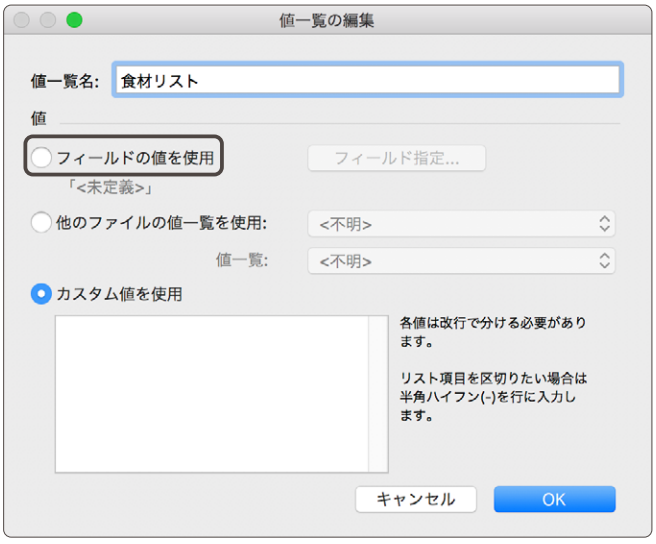


図 5-68 値一覧の作成開始

g.　［値一覧に使用するフィールドの設定］ダイアログが表示されます。

h.　［最初のフィールドの値を使用］は「料理 _ 材料 _ 食材」TO を選択します。

i.　「主キー」フィールドをクリックします。

j. ［2 番目のフィールドの値も表示］を　**オン**　にして「「料理 _ 材料 _ 食材」TO を選択します。

k. 「食材名」フィールドをクリックします。

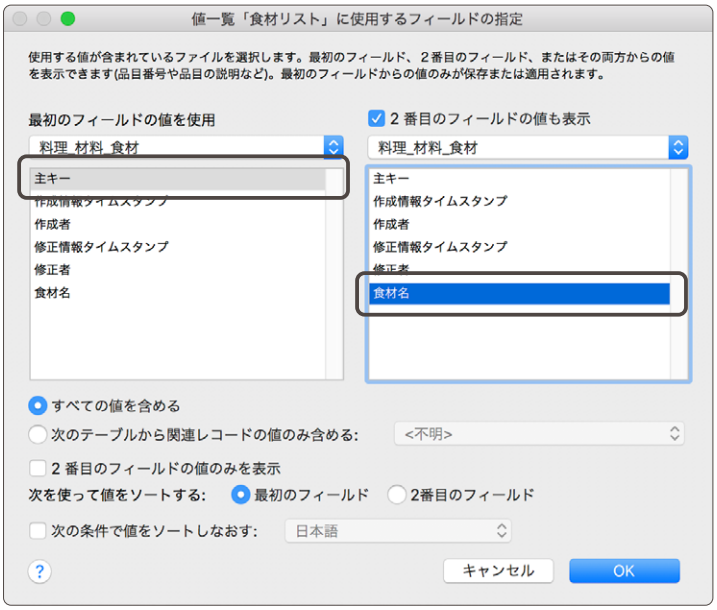


図 5-69 フィールドを指定する

l. ［OK］を何度かクリックしてダイアログを閉じます。値一覧ができました。

(2) フィールドに割り当てます。

a. 「料理と材料」レイアウトでレイアウトモードに切り替えます。

b. ポータルの「食材外部キー」フィールドに値一覧を割り当てます。
右パネルのインスペクタの「データ」タブのフィールドを使います（次図も参照してください）。

・［コントロールスタイル］は　「ポップアップメニュー」
・［値一覧：］は　「食材リスト」
・［一覧の表示切り替え用矢印を表示］を　**オン**
・［値一覧でデータ書式を上書き］を　**オン**



図 5-70 値一覧の割り当て

(3)　レイアウトを保存してブラウズモードに戻ります。

(4)　「食材外部キー」フィールドをクリックすると「食材」テーブルに入力されているすべてのレコードが値一覧として表示されます。

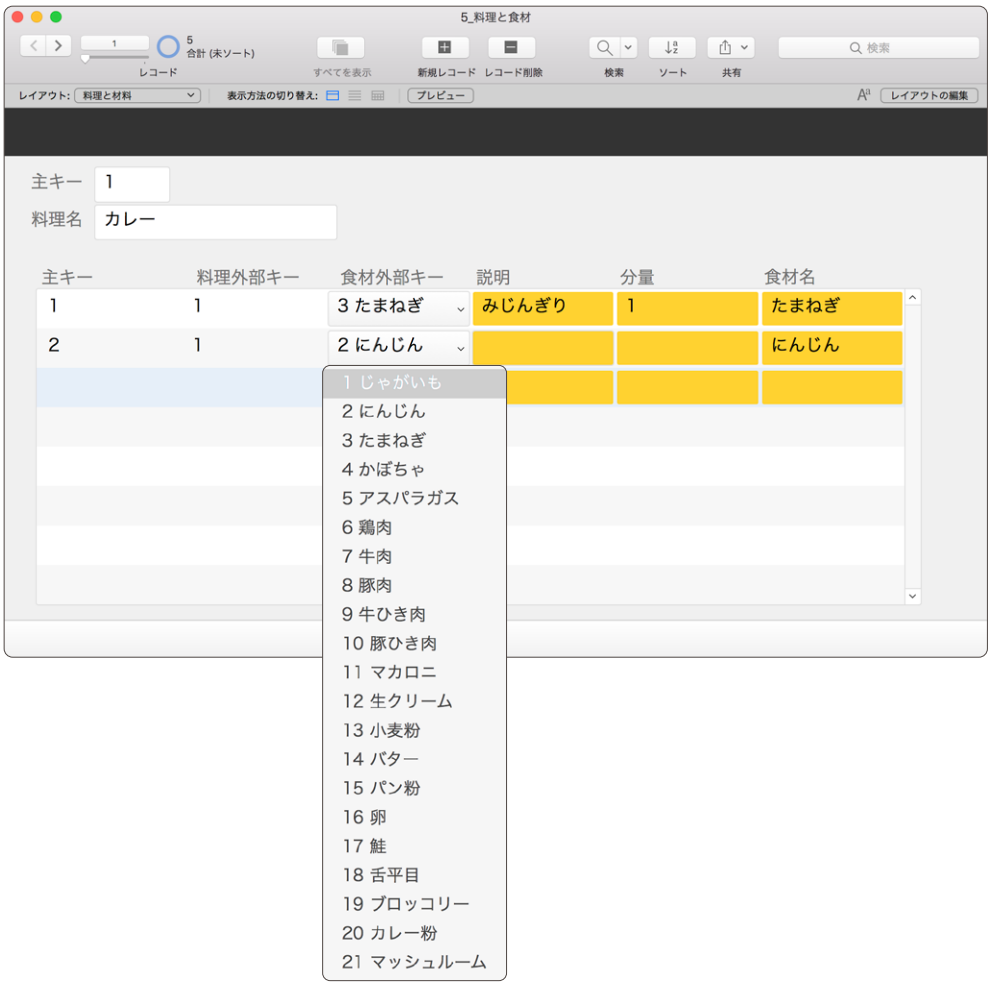


図 5-71 値一覧で食材を選択できるようになった

(5)　動作を確認します。
a.　3 行目の食材外部キーの値一覧で「3 じゃがいも」を選択します。
b.　レコードが作成されます。

(6)　「料理」テーブルの「主キー」の「1」が「料理外部キー」フィールドに入力されていることを確認してください。

演習ここまで

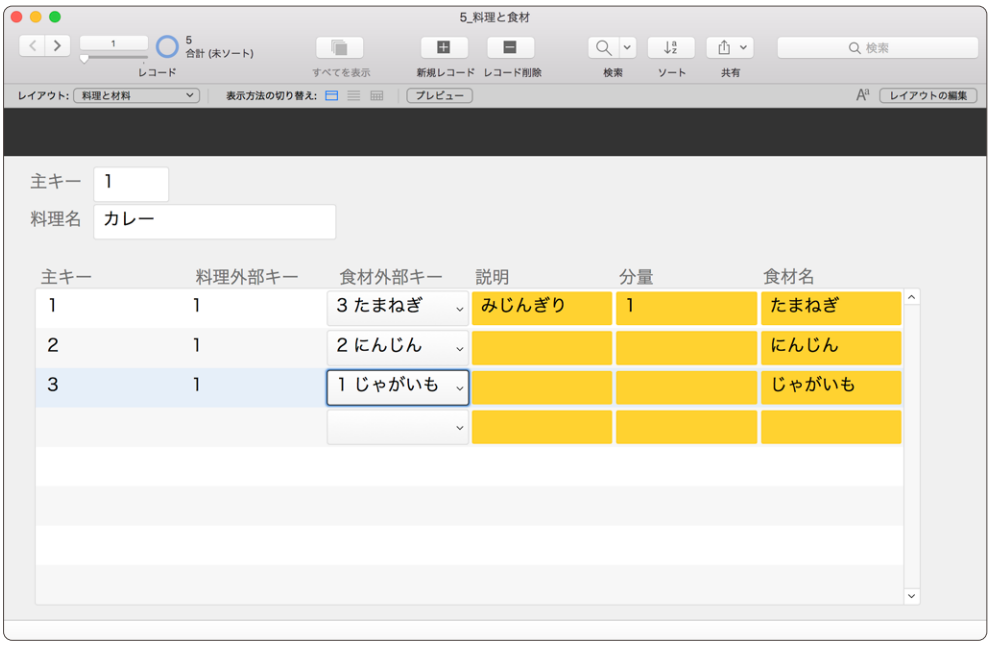


図 5-72 値一覧の選択でポータルにレコードが作成される

多対多の関連に限らず、値一覧を使って外部キーをポータルに入力することで、新規レコードを作成して関連を成立させる方法はユーザにとってわかりやすい入力手段の 1 つです。

5.4.5 関連の表示

FileMaker Pro では、次のようにテーブルオカレンス名とフィールドを表示します。

TO名 :: フィールド名

例えば、 料理 _ 食材 :: 分量は、料理 _ 食材 TO の「分量」フィールド ということです。この表示をフィールドの「**完全修飾名**」と呼びます。どのコンテキストのフィールドなのか、ということです。

レイアウト上、インスペクタ、計算式などで、フィールドの完全修飾名を確認すると正しいコンテキストを指定しているかどうかがわかります。

また、関連がない TO のフィールドをレイアウトに置くとブラウズモードで **< 非関連テーブル >** と表示されます。

設計からリレーションシップグラフへ、そしてレイアウトで入力することへつなげました。次に、カスタム App とデータについて説明します。

演習ファイルを閉じてください。

第 6 章 データとフィールド

カスタム App でデータ（テーブル、レコード、フィールド）を取り扱う際の、フィールドオプションの設定（入力値の自動化、入力値の制限）について解説します。

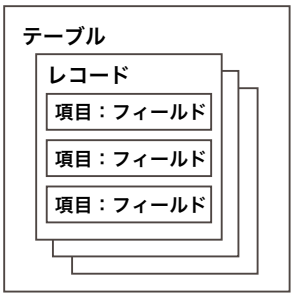
6.1 データ

基本事項を改めて確認します。カスタム App に入力されるデータは、フィールドに保存（格納）されます。

テーブル
ひとつの分類ができる情報の集まりです。設計で正規化されます。

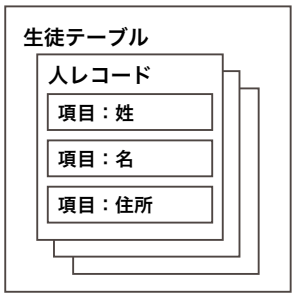
レコード
テーブルの項目（フィールド）の 1 セットごとです。

フィールド
レコードの 1 つずつの項目です。設計でどのテーブルに属するかを決定します。



データを使った作業
レコード単位で次の作業を行います。

・作成
・更新（修正）
・削除



入力と出力
レイアウト（TO）を通してレコード単位で次のことが行われます。

・入力
・出力

どのように複雑で長いスクリプトも、ほとんどの場合、データを入出力するために作成されています。1 つのスクリプト内で何度も入出力をしながら結果を得ることもあります。

入力は、ブラウズモードでのレコードの作成や更新のためだけではなく、検索モードで検索条件を指定することなども当てはまります。出力は、印刷や書き出しだけではなく、集計結果やグラフ、検索結果の表示なども当てはまります。このようにして取り扱われるデータを保存しているフィールドとオプションについて解説します。

6.2 フィールド定義

フィールドには型（タイプ）が用意されています。保存されるデータに従って適切なタイプを指定します。また、オプションを設定することも可能です。フィールド定義の内容、つまり

・名前
・タイプ
・入力値の自動化
・入力値の制限
・格納と索引

は、設計時に決めることが可能です。

フィールド定義の場所
フィールド定義が可能な場所は、3 つあります。

・レイアウトモードの左パネルの「フィールド」タブ
・ブラウズモードの「表形式の列見出し」
・[データベースの管理] ダイアログ

複数のテーブルやフィールドを続けて作成する場合は、[データベースの管理] ダイアログを使用するのが便利です。

6.2.1 フィールドの名前

企画、要件定義には、ユーザも参加する検討（打ち合わせ）のために、わかりやすい名称を使うことがあります。その後、設計時に別の名称にすることがあります。

例えば、フィールド名を英語（半角アルファベット）にすることもあります。わかりやすい見出しがレイアウト上で表示されていれば、ユーザが迷うことはありません。そして、ユーザがフィールドの名前を意識する必要があることはほとんどありません。

本書のフィールド名は、なるべく日本語で端的に内容を表すものとします。

フィールドの命名ルール
取り扱うデータを端的で内容がすぐにわかるように命名します。計算式やスクリプトの作成中に、フィールドを選択しやすく、タイプも簡単に見分けられると便利です。

・計算タイプフィールドは「c_ フィールド名」（calculation の頭文字）
・集計タイプフィールドは「s_ フィールド名」（summary の頭文字）
・グローバル格納フィールドは「g_ フィールド名」（global の頭文字）

本書ではこのようなフィールド名も使用します。

6.2.2 フィールドタイプ

データには型（タイプ）があります。データを適切に扱えるようにタイプごとの保存方法、索引方法が用意されています。また、ソート結果にも影響します。

フィールドタイプは次表の 8 タイプです。それぞれの特徴を確認します。

フィールドのタイプ		
タイプ名		格納するデータの種類
基本タイプ	テキスト	オプションのテキストスタイル、および行揃えを含め、10,000,000 文字まで（使用可能なメモリによって制限されます）
	数字	10^-400 から 10^400 までの値と同じ範囲の負の値をサポート
	日付	日付のみ。グレゴリオ暦で 0001 年 1 月 1 日から 4000 年 12 月 31 日までの範囲の日付
	時刻	時刻のみ。時、分、秒を保存
	タイムスタンプ	暦上の日時の特定日付と時刻の組み合わせ（2003/10/25 2:39:46 など）
	オブジェクト	最大 4GB の画像、Microsoft Word や Excel などのファイル、PDF、マルチメディアファイル（写真、ムービー、サウンドなど）など
計算		現在のレコードまたは関連レコードのフィールド値を使用する計算式の計算結果（計算の結果は、テキスト、数字、日付、時刻、タイムスタンプ、またはオブジェクトのいずれかのタイプ）
集計		同じテーブル内の複数のレコードのフィールド値を集計して得られた値

郵便番号と電話番号はテキスト
記号を除けば数字だけなのですが、「0」からはじまる場合があるのでテキストです。

業務上の管理番号では可能な場合は数字
業務上付与している番号（商品番号、見積番号など）は「0」からはじまるものでない限り、数字の方が検索やソートでの取り扱いが楽です。

特殊なタイプとして「計算」「集計」があります。他のフィールドなどのデータを元に算出するものです。

計算タイプ
結果のデータにあわせたタイプを設定します。

集計タイプ
集計結果に対してはタイプ設定はありません（結果は主に数値です）。

フィールドと入力
設定されたタイプに則したデータを入力します。

例えば、テキストタイプのフィールドに、数字、日付、時間、タイムスタンプを入力すると一見、表示上は問題がなく思えても、様々なタイミングで期待とは違う結果になってしまいます。

6.2.3 ハウスキーピングフィールド

データは、いつ誰が作成または修正したのか、本当に最新なのか、などの情報が必要です。レコード単位の次の内容です。

・識別子
・作成情報
・修正情報

これらのような項目のことを「ハウスキーピングフィールド」と呼びます。各テーブルに作成することをお勧めします。ユーザが意識することなく記録できるように自動入力を設定します。次表に、フィールドオプションの設定例を記述します。

ハウスキーピングフィールドの設定例				
フィールド名	**フィールドタイプ**	**入力値の自動化**	**入力値の制限**	**備考**
主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値	識別子
作成者	テキスト	作成情報（アカウント名） データ入力時の値変更禁止	空欄不可	
修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可	
作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付 空欄不可	
修正情報タイムスタンプ	タイムスタンプ	修正情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付 空欄不可	

なお、FileMaker Pro のバージョン 17 から、新しいテーブルを作成するたびに上表のハウスキーピングフィールドが自動作成されるようになっています。

※ データをインポートしてテーブルを作成する場合、別のアプリケーションのファイルを変換してファイルを作成する場合には、デフォルトのフィールドは作成されません。

自動作成されたハウスキーピングフィールドは、フィールドオプションでユーザが入力できないように設定されています。これは、ユーザが本来とは違う日付をわざと入力してしまうことを防ぎ、自動入力された『いつもあてになる』データがフィールドに設定されているようにするためです。

自動入力される内容について
デフォルトのフィールドの自動入力される内容を変更すること、自動入力されないようにできます。詳細はヘルプやナレッジベースの情報を確認してください。

「環境設定」の［新規フィールドを現レイアウトに加える］チェックが **オフ** になっていることを確認してください。

（ここから、本書に付属している演習ファイルの「6 章開始時」フォルダの中の「6_ 見積作成管理 .fmp12」ファイルをベースに進めます）

「見積作成管理 .fmp12」ファイルを開いて［データベースの管理］ダイアログを表示してください。

まず、デフォルトのフィールドの自動作成を確認してみます。

「6 章開始時」フォルダ

演習 6-1　デフォルトのフィールドの自動作成を確認する

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　［データベースの管理］ダイアログを表示します

(2)　「テーブル」タブを選択します。

(3)　新しいテーブルを作成します。
a.　テーブル名は「ハウスキーピング」としてください。
b.　［作成］をクリックしてください。
c.　「フィールド」タブを表示してください。
d.　デフォルトのフィールドが作成されています。

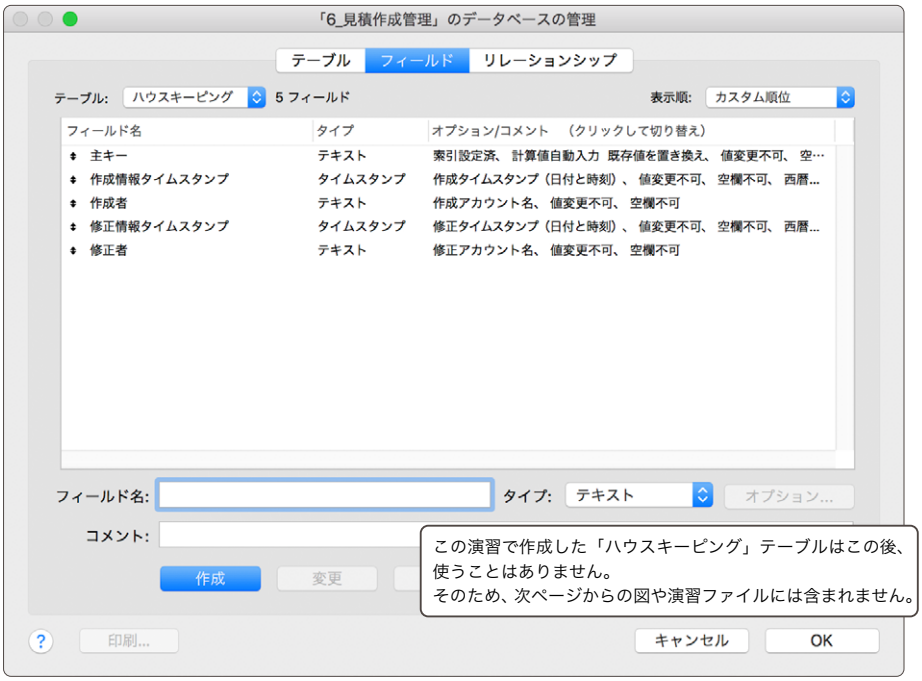


図 6-1 デフォルトのフィールドができている

(4)　デフォルトのフィールドの内容を確認してください。確認後、［キャンセル］をクリックしてダイアログを閉じてください

演習ここまで

改めて「見積作成管理システム」を作成していきます。第 4 章までに設計された内容から、テーブルが 7 つ、フィールド約 70 を作成する必要があります。

テーブルとフィールドの作成を演習で行うのは時間がかかるので、「見積作成管理 .fmp12」に、テーブル、フィールドを作成しておきました（フィールドタイプも設定済みです）。

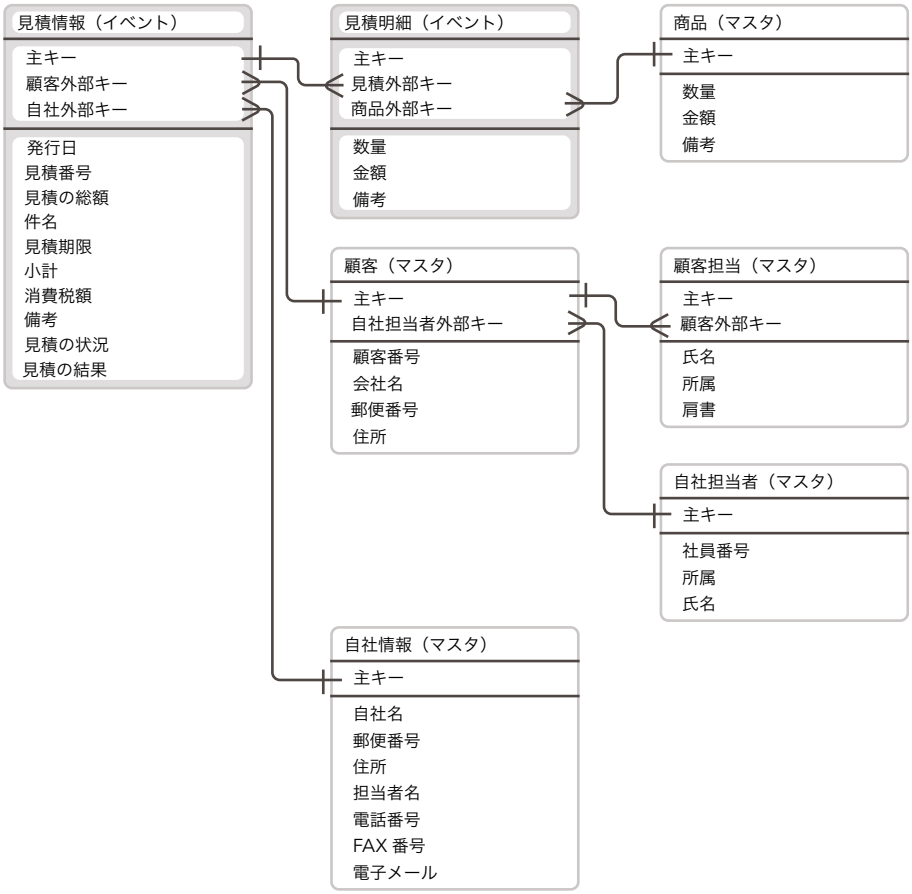


図 6-2 ER 図

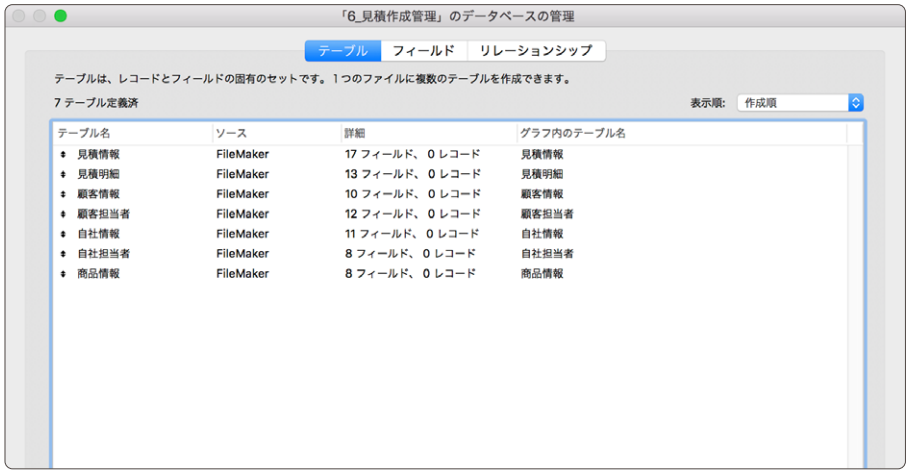


図 6-3［データベースの管理］ダイアログ

テーブル名は、設計時の ER 図から少し変わっています。フィールド名をわかりやすくするため、いくつか変更しています。第 3 章の要件定義時と変更した点は、次のとおりです。

＜「見積明細」テーブル＞
・見積書作成当時の商品情報を残す要望があったため、必要なフィールドを追加
＜「見積情報」テーブル＞
・見積書作成当時の消費税を残すという要望があったため、フィールドを追加
・見積番号は「10001」顧客番号は「20001」商品番号は「30001」から自動採番する
＜「顧客担当者」テーブル＞
・常に「現在の顧客担当」は参照とする（集計は「顧客ごと」だけを対象とするため）
・連絡先も登録したいという要望があったため、フィールドを追加

	見積情報	
	フィールド名	**タイプ**
1	主キー	テキスト
2	作成情報タイムスタンプ	タイムスタンプ
3	作成者	テキスト
4	修正情報タイムスタンプ	タイムスタンプ
5	修正者	テキスト
6	顧客外部キー	テキスト
7	自社外部キー	テキスト
8	発行日	日付
9	見積番号	数字
10	~~見積の総額~~ → c_ 見積総額	計算（結果は　数字）
11	件名	テキスト
12	~~見積期限~~ → 期限日	日付
13	~~小計~~ → c_ 見積小計	計算（結果は　数字）
14	~~消費税額~~ → c_ 消費税	計算（結果は　数字）
15	備考	テキスト
16	~~見積の状況~~ → 見積状況	テキスト
17	~~見積の結果~~ → 見積結果	テキスト

	見積明細	
	フィールド名	**タイプ**
1	主キー	テキスト
2	作成情報タイムスタンプ	タイムスタンプ
3	作成者	テキスト
4	修正情報タイムスタンプ	タイムスタンプ
5	修正者	テキスト
6	見積外部キー	テキスト
7	商品外部キー	テキスト
8	数量	数字
9	~~金額~~ → c_ 金額	計算（結果は　数字）
10	備考	テキスト
11	商品番号	数字
12	商品名	テキスト
13	単価	数字

	商品情報	
	フィールド名	**タイプ**
1	主キー	テキスト
2	作成情報タイムスタンプ	タイムスタンプ
3	作成者	テキスト
4	修正情報タイムスタンプ	タイムスタンプ
5	修正者	テキスト
6	商品番号	数字
7	商品名	テキスト
8	単価	数字

	顧客情報	
	フィールド名	**タイプ**
1	主キー	テキスト
2	作成情報タイムスタンプ	タイムスタンプ
3	作成者	テキスト
4	修正情報タイムスタンプ	タイムスタンプ
5	修正者	テキスト
6	自社担当者外部キー	テキスト
7	顧客番号	数字
8	会社名	テキスト
9	郵便番号	テキスト
10	住所	テキスト

	顧客担当者	
	フィールド名	**タイプ**
1	主キー	テキスト
2	作成情報タイムスタンプ	タイムスタンプ
3	作成者	テキスト
4	修正情報タイムスタンプ	タイムスタンプ
5	修正者	テキスト
6	顧客外部キー	テキスト
7	所属	テキスト
8	肩書	テキスト
9	氏名	テキスト
10	電話番号	テキスト
11	FAX 番号	テキスト
12	電子メール	テキスト

	自社情報	
	フィールド名	**タイプ**
1	主キー	テキスト
2	作成情報タイムスタンプ	タイムスタンプ
3	作成者	テキスト
4	修正情報タイムスタンプ	タイムスタンプ
5	修正者	テキスト
6	自社名	テキスト
7	郵便番号	テキスト
8	住所	テキスト
9	電話番号	テキスト
10	FAX 番号	テキスト
11	電子メール	テキスト

	自社担当者	
	フィールド名	**タイプ**
1	主キー	テキスト
2	作成情報タイムスタンプ	タイムスタンプ
3	作成者	テキスト
4	修正情報タイムスタンプ	タイムスタンプ
5	修正者	テキスト
6	社員番号	テキスト
7	所属	数字
8	氏名	テキスト

（追加したフィールドはセルの色を変えています）

計算タイプフィールドについては、計算結果のデータのタイプを決定してありますが、後ほどの演習で、計算タイプに変更して計算式を設定します。

6.3　フィールドオプション

フィールドオプションを設定します。データの格納については第 7 章で取り扱います。

・入力値の自動化
・入力値の制限
・データの格納

すべてのフィールドタイプにオプションが設定可能です。

※ 計算タイプと集計タイプは、これから説明する内容とは違う特別な内容になります。

「見積作成管理 .fmp12」の［データベースの管理］ダイアログで「見積情報」テーブルの「件名」フィールドの［フィールドオプション］ダイアログを表示してください。

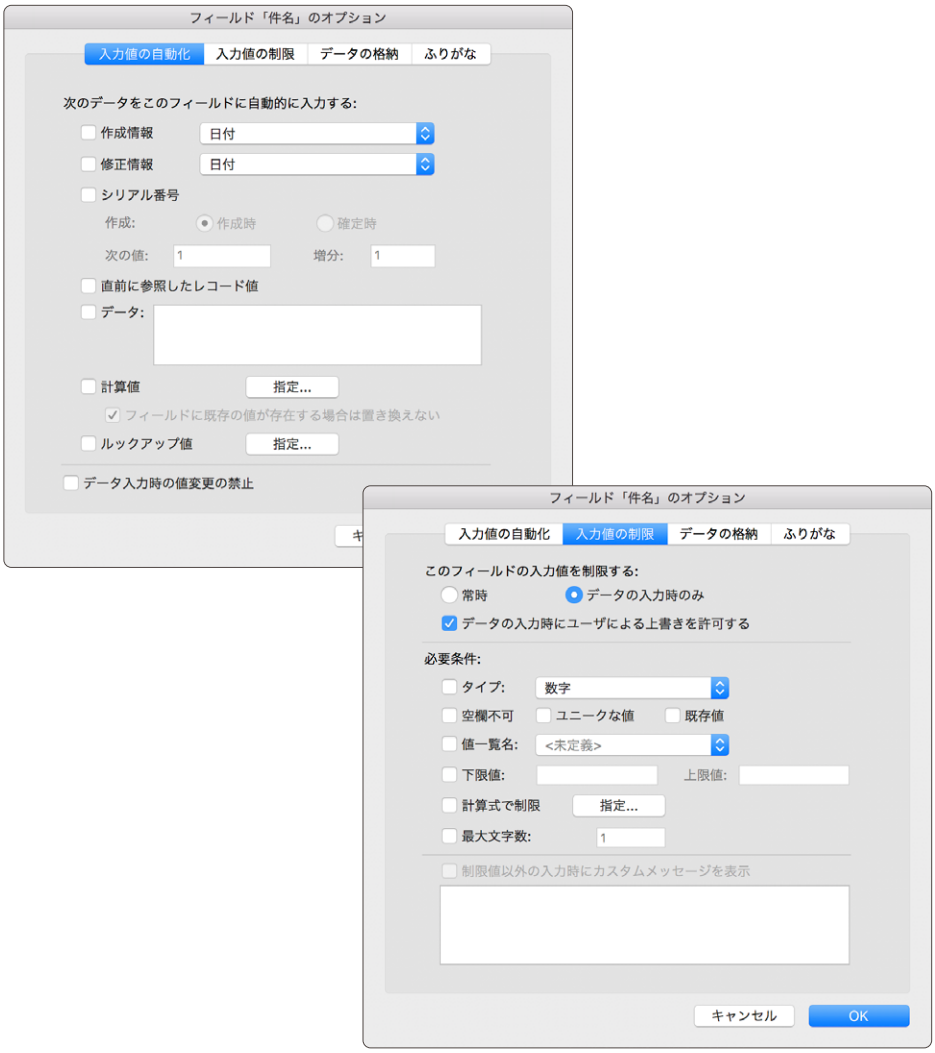


図 6-4［フィールドオプション］ダイアログ

現在はなにも設定されていません。設定できる内容を見ていきます。

6.4　フィールドオプション：入力値の自動化

このタブでは、次のような設定をします。

・ユーザに意識させずにフィールドにデータを保存する
・レコード作成時の「初期値」を入力する
・あるタイミングで必要なデータを入力する
・入力したデータを変更できないようにする
・ユーザのデータ入力を禁止する

フィールドには、あらかじめデータが入力されると便利なものがあります。例えば、日付の入力が必要なフィールドがあって 7 日後の日付を入力することが多いのであれば、新規レコード作成時に 7 日後の日付が自動入力されていると便利です。ユーザは、それ以外の日付を入力したい場合のみ、入力すればよいのです。

自動入力はフィールドへ入力されるデータの精度を高めることにも使用できます。英数字を入力するフィールドで、データを必ず半角英数字にしたいのであれば、データの入力後に全角の英数字でも半角に変換されて自動入力されるように設定できます。

次の 3 点を意識して設定します。

(1) なにを　　…　どのフィールドに設定するか
(2) どのように　…　設定内容
(3) どの程度　　…　ユーザの変更の許可／不許可



図 6-5［フィールドオプション］ダイアログの「入力値の自動化」タブ

自動入力は、レイアウト上から行われるものではありません。きっかけはユーザのレイアウト上での入力だったりしますが、FileMaker Pro がデータを入力します。そのため、スクリプトトリガの［OnObjectModify］が有効になることはありません。

レコードの作成時に、自動で入力されるデータを「**初期値**」または「**デフォルト（default）値**」と呼びます。設計時に決定しておくことが多いものです。「入力値の自動化」タブで設定できる内容を確認します。

6.4.1 作成情報／修正情報

レコードの作成または修正された時点の情報を入力する設定です。日付、時刻、タイムスタンプ、名前、アカウント名から選択し、内容に合ったフィールドタイプを設定します。

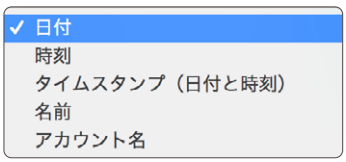


図 6-6 設定できる内容

日付
「今日」が入力されます。

時刻
現在の時刻が入力されます。

タイムスタンプ（日付と時刻）
現在の日時が入力されます。

なにが入力されるか
自動入力される日付、時刻、タイムスタンプは、レコードの作成／修正が実行されたパソコンの OS（オペレーティングシステム）の情報です。正しく入力されないようなら設定を確認します。

名前
［環境設定］ダイアログで設定されている「ユーザ名」が自動入力されます。

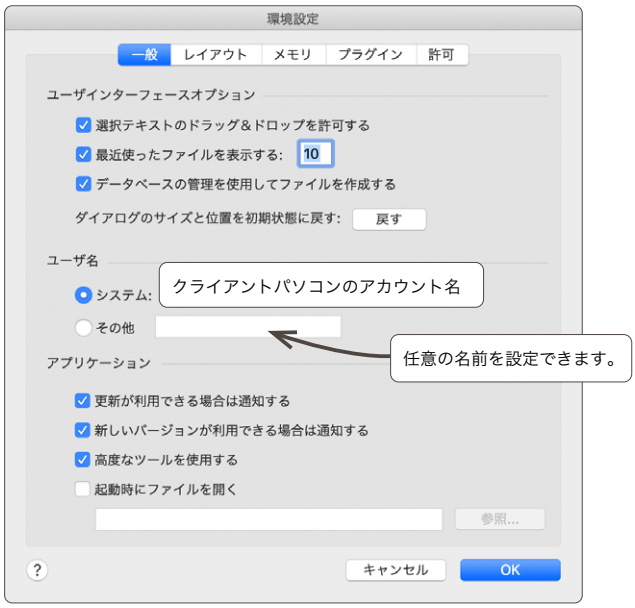


図 6-7［環境設定］ダイアログのユーザ名

アカウント名
ログインしているアカウント名が入力されます。デフォルトのフィールドでは、作成者と修正者は「アカウント名」が設定されています。

ログインを前提にすること
共有利用されるカスタム App では、ユーザがログインして使用することをなるべく前提とします。アカウントはユーザ 1 人ずつに配布します。誰が作成／修正したのか特定しやすくなります。

環境設定のユーザ名は、ログインに関係なく取得できる便利なデータですが、アカウント名よりも個人の識別が確実ではありません。

6.4.2 シリアル番号

レコードの作成または確定時に、自動で連番を生成して入力する設定です。主キーに数字を使う場合などに便利なオプションです。選択したフィールドタイプに従って、保存できる文字数が違います。数字タイプの場合、保存できる桁数で充分機能するはずです。

［次の値］は、次に生成されるデータです。［増分］は連番が増加する数です。例えば、［次の値：2］で［増分：5］だったら、2 の次に 7 が生成されます。以降、12、17 と続きます。

頭になにかつける
接頭語をつけることができます。例えば［次の値：A1］［増分：1］とすると、A1、A2、A3……と続きます。ただし、数字タイプフィールドにそのような設定をおこなうと、フィールドにデータは入力されるがテキストの検索ができない、という状態になります。

作成時と確定時
作成時を選択するとレコードが作成と同時に入力されます。確定時を選択すると、レコード作成後、最初に確定されるまで入力されません。「レコードの確定」は、様々なタイミングで行われるのでユーザの入力方法により、バラバラのタイミングで入力されることになるかもしれません。

本書ではレコードの作成時にシリアル番号を自動入力する方法をお勧めします。

6.4.3 ［直前に参照したレコード値］

レコード作成の直前にアクティブにしていたレコードの、同フィールドのデータを入力する設定です。

6.4.4 ［データ：］

255 文字までのデータ（決まった文字列）を指定して入力する設定です。任意の初期値を入力したいときに便利です。

例えば、レコード作成時に「見積状況」フィールドに「作成／お客様検討中／結果確定済」の 3 つのうち「作成」を入力したい場合などです。

6.4.5 ［計算値］

計算式の結果を自動で入力するための設定です。設定したフィールドのタイプは計算結果のタイプになります。

組み合わせ

他のフィールドのデータが変更されたときに計算式を組み合わせて設定することが可能です。

例えば［データ：］で初期値を入力してユーザがそのデータを変更したり、消去した後、初期値に戻したいことがあります。しかし、［データ：］による入力は新規レコード作成時にしか実行されません。そのような場合、［データ：］の代わりにこのような計算式で自動入力できます。

```
If ( IsEmpty ( Self ) ; "あいうえお" ; self )
```

新規レコード作成時とフィールドが空欄で確定されると「あいうえお」が入力されるようになります。

フィールドに既存の値が存在する場合は置き換えない

このオプションには、次のような違いがあります。

・**オン**　既存の値が存在する場合、データは置き換わらない
・**オフ**　計算式の評価が行われるたびにデータが置き換わる

評価

設定されている計算式は、次のタイミングで評価されます。
・レコード作成時
・計算式で参照しているフィールドのデータが変更されたとき

例えば 2 つのフィールド「データ 1」と「データ 2」があります。そして「データ 1 と 2」フィールドには、次の計算式が設定されているとします（いずれもテキストタイプのフィールドです）。

```
データ1 & データ2 & "  " & Get ( 日付 )
```

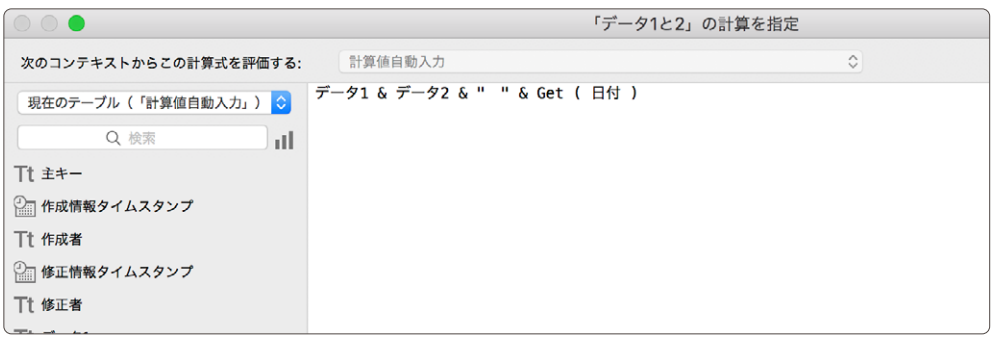


図 6-8［計算を指定］ダイアログ

［フィールドに既存の値が存在する場合は置き換えない］のオン／オフで違いがあります。この特性を理解して使い分けてください。

なお［計算を指定］ダイアログで［すべての参照フィールドが空の場合は評価しない］のチェックも関係しますので、4 パターンの設定があります。

フィールドに既存の値が存在する場合は置き換えない　と　すべての参照フィールドが空の場合は評価しない				
1	フィールドに既存の値が存在する場合は置き換えない	オン	すべての参照フィールドが空の場合は評価しない	オン
2		オン		オフ
3		オフ	すべての参照フィールドが空の場合は評価しない	オン
4		オフ		オフ

設定によるデータの違いを表にしました。もしも作業を 2018 年 7 月 28 日におこなうと、

			データ 1 と 2	
作業	**データ 1**	**データ 2**	**置き換えオン 空評価オン**	**置き換えオン 空評価オフ**
新規レコード作成	（空欄）	（空欄）	（空欄）	2018/7/28
データ 1 に「あああ」と入力	あああ	（空欄）	あああ　2018/7/28	2018/7/28
データ 2 に「いいい」と入力	あああ	いいい	あああ　2018/7/28	2018/7/28
データ 1 の内容を消去	（空欄）	いいい	あああ　2018/7/28	2018/7/28
データ 2 の内容を消去	（空欄）	（空欄）	あああ　2018/7/28	2018/7/28
「データ 1 と 2 置き換えオン空評価オン」「データ 1 と 2 置き換えオン空評価オフ」の内容を変更	（空欄）	（空欄）	ううう（と直接入力）	ううう（と直接入力）
データ 2 に「えええ」と入力	（空欄）	えええ	ううう	ううう
データ 2 の内容を消去	（空欄）	（空欄）	ううう	ううう

			データ 1 と 2	
作業	**データ 1**	**データ 2**	**置き換えオフ 空評価オン**	**置き換えオフ 空評価オフ**
新規レコード作成	（空欄）	（空欄）	（空欄）	2018/7/28
データ 1 に「あああ」と入力	あああ	（空欄）	あああ　2018/7/28	あああ　2018/7/28
データ 2 に「いいい」と入力	あああ	いいい	あああいいい　2018/7/28	あああいいい　2018/7/28
データ 1 の内容を消去	（空欄）	いいい	いいい　2018/7/28	いいい　2018/7/28
データ 2 の内容を消去	（空欄）	（空欄）	（空欄）	2018/7/28
「データ 1 と 2 置き換えオフ空評価オン」「データ 1 と 2 置き換えオフ空評価オフ」の内容を変更	（空欄）	（空欄）	ううう（と直接入力）	ううう（と直接入力）
データ 2 に「えええ」と入力	（空欄）	えええ	えええ　2018/7/28	えええ　2018/7/28
データ 2 の内容を消去	（空欄）	（空欄）	（空欄）	2018/7/28

※［データの入力時にユーザによる上書きを許可する］はいずれも　**オン**　です。

計算タイプフィールドとの違い

計算値で自動入力されたデータはレイアウトやインポートなどで直接変更することが可能です。計算タイプのフィールドは、直接変更することはできません。

「既存の値」とは、計算値で自動入力されたデータだけではなく、他の方法で入力されたデータでもあります。

※ ユーザが変更できるようにするには［データ入力時の値変更の禁止］をオフにする必要があります。

6.4.6 ［ルックアップ値］

関連テーブルの関連レコードの指定したフィールドからデータをコピーして入力する設定です。関連レコードを参照するのではありません。コピーされたデータがフィールドに入力されます。

ルックアップ設定には、関連が必要です。次のようにデータがコピーされます。

(1)「開始するテーブル（参照元 TO)」の照合フィールドのデータが新しく入力または変更される
(2)「関連テーブル（参照先 TO)」の関連レコードの「値のコピー元のフィールド」で指定したフィールドのデータがコピーされる

そして、次のポイントがあります。

・参照先の関連レコードの一番目のレコードのデータがコピーされる
・コピーされたデータを変更してもコピー元のデータに反映されない

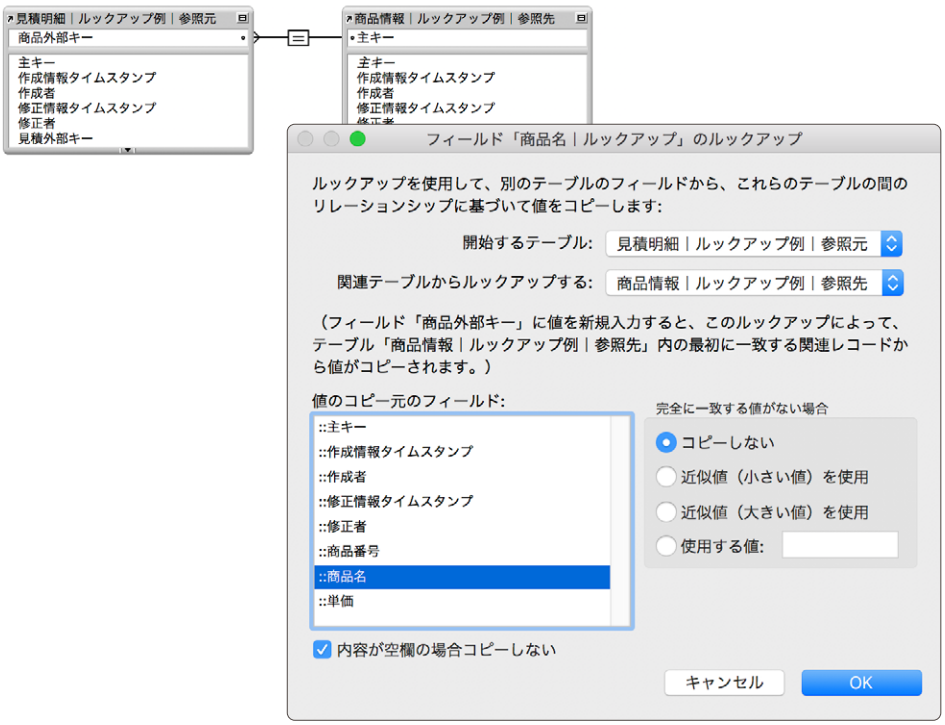


図 6-9 ルックアップの設定例

関連レコードのフィールドのデータを自動入力するにはルックアップではなくフィールドオプションの［入力値の自動化］の［計算値］でも可能です。

保存と参照
関連レコードのデータを参照するのではなく、［ルックアップ値］のようにコピーしてフィールドに入力したくなる状況はよくあります。

例えば、「見積作成管理システム」の「見積明細」テーブルでは「商品名」と「単価」は「商品外部キー」を使用した関連で「商品情報」テーブルのデータを参照することが可能な設計になっています。ただし、次のような可能性がある場合、参照だけで良いかどうか考えます。

・商品自体は変わらないが商品名が変わる
・商品自体は変わらないが単価が変わる

商品名や単価に変更が起きても、現在の見積書で問題になることはあまりありません。しかし、過去の見積書では、当時の商品名や単価がわからなくなってしまいます。それらのデータを元にした計算結果も変わります。それでよいでしょうか？

よろしくないのであれば、データを別途保存する必要があります。
※ 消費税率についても同じように考えられます。

『レコードを作成した当時のデータを残しておきたいかどうか』は、要件定義時に確認するべき重要なポイントです。ユーザからは、作成時や印刷時に正しい情報が表示されていれば構わないという意見がでることがありますが、カスタム App として保存するデータは、集計などに備える必要があります。集計までの間に単価や消費税が変わってしまったら、期待した結果にならないかもしれません。慎重に検討をおこなってください。

保存されているデータの更新
参照ではなく別途入力しているデータでも、関連レコードのデータ変更を反映させたいときがあります。例えば、

・「見積明細」テーブルのレコードで商品 A を指定したけれど、翌日、価格が変わった。
まだ顧客に提示していないので、新しい価格が表示されるようにしたい。

などです。この作業は注意する必要があります。

例えば、［ルックアップ］には「再ルックアップ」がありますが、対象レコードに注意しないと反映させたくないレコードにまで新しいデータが反映されてしまいます。データを反映する際は、常に対象レコードに気をつけなければなりません。

6.4.7 ［データ入力時の値変更の禁止］

このチェックを **オン** にすると、自動入力されたデータをユーザが変更できなくなります。しかし、スクリプトで変更することは可能なので注意してください。

6.4.8 設定の保存

［データベースの管理］ダイアログを［OK］をクリックして閉じないと、設定が保存されません。［キャンセル］をクリックと、それまでのすべての変更が破棄されます。
※ 何も変更しなかった場合、［キャンセル］をクリックすると想定しない変更などが起きません。

「見積作成管理システム」の入力値の自動化は次表の内容で設定します。

	見積情報	タイプ	入力値の自動化
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
6	顧客外部キー	テキスト	
7	自社外部キー	テキスト	
8	発行日	日付	今日　→　作成日
9	見積番号	数字	10001 から自動採番 データ入力時の値変更禁止
10	c_ 見積総額	計算（結果は　数字）	
11	件名	テキスト	
12	期限日	日付	発行日＋ 1 ヶ月後
13	c_ 見積小計	計算（結果は　数字）	
14	c_ 消費税	計算（結果は　数字）	
15	備考	テキスト	
16	見積状況	テキスト	「作成」
17	見積結果	テキスト	「注文なし」
18	消費税	数字	発行日の消費税 8%だったら 0.08

演習で設定するオプションはこの枠の中です。

	見積明細	タイプ	入力値の自動化
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
6	見積外部キー	テキスト	
7	商品外部キー	テキスト	
8	数量	数字	「0」
9	c_ 金額	計算（結果は　数字）	
10	備考	テキスト	
11	商品番号	数字	商品テーブルから同名フィールドを計算値自動入力
12	商品名	テキスト	商品テーブルから同名フィールドを計算値自動入力
13	単価	数字	商品テーブルから同名フィールドを計算値自動入力

	顧客情報	タイプ	入力値の自動化
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
6	自社担当者外部キー	テキスト	
7	顧客番号	数字	20001 から自動採番 データ入力時の値変更禁止
8	会社名	テキスト	
9	郵便番号	テキスト	
10	住所	テキスト	

	顧客担当者	タイプ	入力値の自動化
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
6	顧客外部キー	テキスト	
7	所属	テキスト	
8	肩書	テキスト	
9	氏名	テキスト	
10	電話番号	テキスト	
11	FAX 番号	テキスト	
12	電子メール	テキスト	

	自社情報	タイプ	入力値の自動化
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
6	自社名	テキスト	
7	郵便番号	テキスト	
8	住所	テキスト	
9	電話番号	テキスト	
10	FAX 番号	テキスト	
11	電子メール	テキスト	

	自社担当者	タイプ	入力値の自動化
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
6	社員番号	数字	
7	所属	テキスト	
8	氏名	テキスト	

社員番号は、自社で別途に管理しているため自動採番する必要はありません。

	商品情報	タイプ	入力値の自動化
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止
6	顧客外部キー	テキスト	
7	商品番号	数字	30001 から自動採番 データ入力時の値変更禁止
8	商品名	テキスト	
9	単価	数字	「0」

「見積情報」テーブルの［入力値の自動化］を設定します。他のテーブルの一部のフィールドは、演習ファイルに設定済みです（関連を利用した自動入力は後ほど演習します）。

演習 6-2　入力値の自動化の設定：初期値

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　［データベースの管理］ダイアログの「テーブル」タブで、「見積情報」テーブルをダブルクリックして「見積情報」テーブルの「フィールド」タブへ移動します。

(2)　「発行日」フィールドを選択し［オプション］をクリックします。

(3)　［フィールドオプション］ダイアログで「入力値の自動化」タブを選択します。

(4)　表で計画した内容に基づいて自動化する入力値を設定します。

「発行日」フィールド
- ［作成情報］は　「日付」を選択

(5)　［OK］をクリックします。

(6)　さらに 3 つのフィールドにオプションを設定します。

「見積番号」フィールド
- ［シリアル番号］を　**オン**
- ［作成］は　作成時のまま
- ［次の値］は　10001
- ［増分］は　1 のまま
- ［データ入力時の値変更の禁止］を　**オン**

「見積状況」フィールド
- ［データ］は　「作成」を入力

「見積結果」フィールド
- ［データ］は　「注文なし」

(7)　設定が終わったら、［OK］をクリックして［データベースの管理］ダイアログを閉じます。
入力値の自動化が設定されました。

（設定内容については、次ページの図も参照してください）

演習ここまで



図 6-10 フィールドオプション設定例

計算式を使う入力値の自動化設定を行います。

演習 6-3　入力値の自動化の設定：計算値の自動入力

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　「期限日」フィールドに 1 ヶ月後の日付を入力する設定をします。

(2)　［フィールドオプション］ダイアログで「入力値の自動化」タブを選択します。

(3)　［計算値］の右の［指定 ...］をクリックして次の計算式を入力します。

```
Date ( Month ( 発行日 ) + 1 ; Day ( 発行日 ) ; Year ( 発行日 ))
```

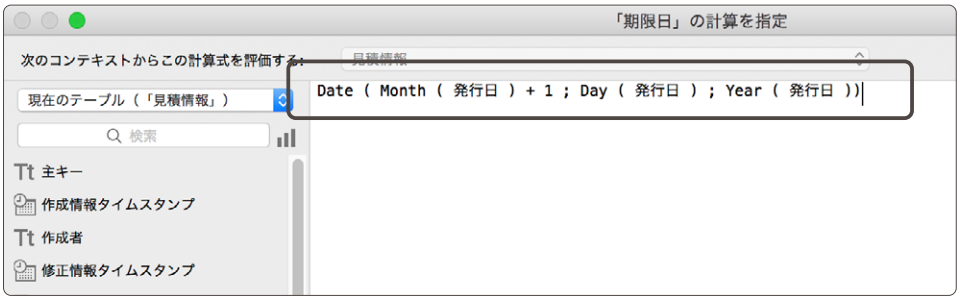


図 6-11　計算式を入力

(4)　［OK］をクリックして［フィールドに既存の値が存在する場合は置き換えない］を　**オフ**　にします

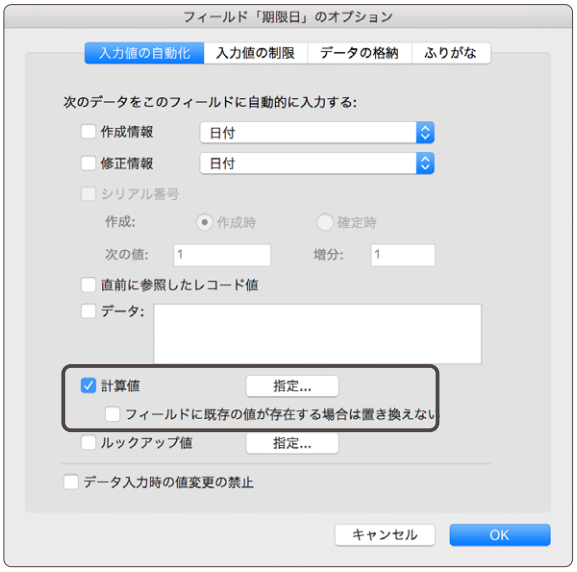


図 6-12　フィールドオプション設定

(5)　［OK］をクリックします。

(6)　続けて「消費税」フィールドに消費税率（2019/9/30 までが 8%、2019/10/1 以降が 10%）を設定します。詳細な手順は省略します。
計算式は、次のとおりです。[フィールドに既存の値が存在する場合は置き換えない] を **オフ** にします。

```
Case ( 発行日 < Date ( 10 ; 1 ; 2019 ) ; 0.08 ; 0.1 )
```



図 6-13 計算式を入力

(7)　[OK] をクリックして［データベースの管理］ダイアログを閉じます。入力値の自動化が設定されました。

演習ここまで

これで 1 テーブル分の入力値の自動化が終わりました。

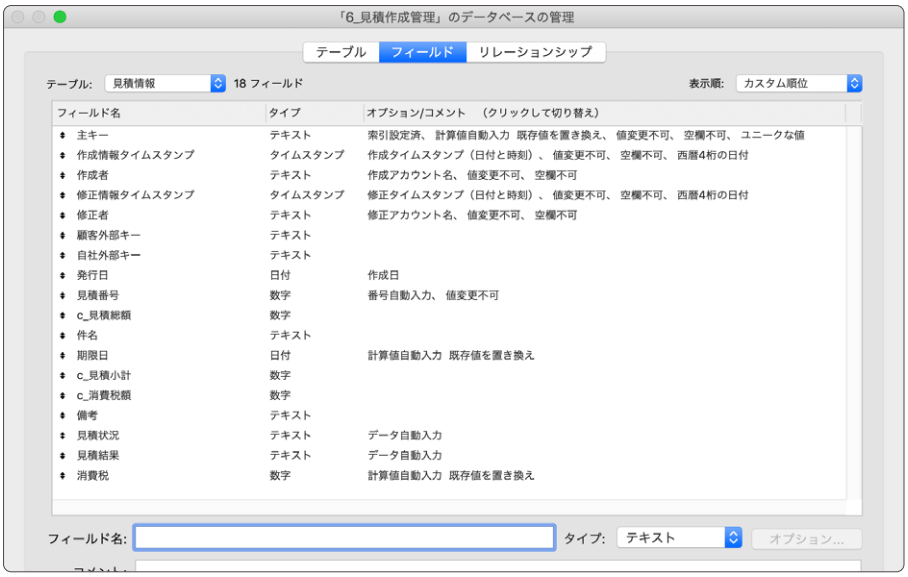


図 6-14「見積情報」テーブルのフィールド（オプション設定後）

他のテーブルも設定するとそれなりに時間がかかります。フィールド数が多くなるほど作業時間がかかりますので、少しでもスムーズに進むようにするために先に設計することをお勧めします。

設計をせずにいきなりフィールド定義をすると、間違いや設定忘れがあるかもしれません。想定する内容があれば、作業後にその通りにできているか確認できます。計画して実行することは間違った作業をするリスクを減らすことにつながります。

6.5　フィールドオプション：入力値の制限

「入力値の制限」タブは、次のような設定が可能です。

・レコード確定時にフィールドに保存されるデータを一定のルールでコントロールする
・設定したコントロールを無視できないようにする

フィールドに入力されるデータは、設計で期待した内容であることが必要です。すると、入力値の自動化だけではカバーしきれないことがあります。例えば、「今日」より新しい日付が入力されるべきフィールドでは、初期値として今日を入力するだけではなく「昨日より古い日付」を入力させないように制限することで、間違ったデータ入力を防ぐことが可能です。

制限も自動化と同じく、フィールドへ入力されるデータの精度を高めることができます。

(1) なにを　…　どのフィールドに設定するか
(2) いつ　…　どのタイミングで
(3) どの程度　…　ユーザの変更の許可／不許可
(4) どの条件で　…　設定内容
(5) 知らせ方　…　エラーの表示方法

この 5 点を意識して設定します。



図 6-15［フィールドオプション］ダイアログの「入力値の制限」タブ

6.5.1 ［このフィールドの入力値を制限する］

『いつ制限を評価するか』を設定します。初期値は［データの入力時のみ］です。次の入力があります。

・手入力
・置換、再ルックアップなどのメニューコマンドによるもの
・スクリプト
・インポート
・「入力値の自動化」オプションによる自動入力

常時：
すべての入力で制限します。もちろん自動入力も含まれます。常に制限を評価します。例えば、この制限によりインポート時にエラーになったレコードはスキップされます。

データの入力時のみ：
ユーザの手入力時のみ制限されます。全置換、インポートや［フィールド設定］スクリプトステップなどは制限されません。

6.5.2 ［データの入力時にユーザによる上書きを許可する］

設定した制限をユーザが無視できるようにします。確実に制限したい場合　**オフ**　にします。

6.5.3 ［必要条件］

制限する内容です。1 つのフィールドに複数の条件を設定できます。ただし、オプションが衝突しないように気をつけます。

複数選択できない設定があります。例えば、「ユニークな値」と「既存値」は一緒に選べません。また、設定する条件により評価するタイミングが異なります。他のレコードと比較する場合などがあるためです。

入力値の制限・計算式による制限	
項目	**入力値の必要条件**
タイプ：［数字］	数字のみ
タイプ：［西暦 4 桁の日付］	月と日が数字で、年が 4 桁の数字である日付（2015/12/5 など）
タイプ：［時刻］	時間と分が 00:00 から 23:59 までの数値である時刻
空欄不可	空欄のフィールドを不可とする
ユニークな値	他のレコードの同じフィールドで値が重複しないようにする
既存値	他のレコードの同じフィールドにすでに入力された値に一致させる
値一覧名	値一覧を選択　指定した値一覧の値に一致する
下限値／上限値	特定の範囲内の文字（数字、ひらがな、カタカナは五十音順、英単語はアルファベット順、漢字はコード順）、数字、日付、また時刻
［計算式で制限］	計算結果と一致する ［フィールドに変更があった場合のみ制限する］を選択または選択解除してから［OK］をクリックします
［最大文字数］	指定した文字数以下にする
［最大キロバイト数］	特定のファイルサイズを超えないようにする （オブジェクトタイプのフィールドの場合）

6.5.4 ［制限値以外の入力時にカスタムメッセージを表示］

制限の評価がエラーになると標準のエラーメッセージが表示されます。

［制限値以外の入力時にカスタムメッセージを表示］を設定すると、独自のメッセージを表示できます。具体的な内容をユーザに表示したい場合に便利です。最大 255 文字まで設定可能です。

［データの入力時にユーザによる上書きを許可する］の設定により、メッセージ上に表示されるボタンが変わります。標準／カスタムどちらのメッセージでも同じボタンが表示されます。

［データの入力時にユーザによる上書きを許可する］が　オン

（アクセス権セットの［ユーザによるデータ入力警告の無視を許可］が　**オン**　の必要あり）

- はい ..... 入力されているデータ（制限値以外のデータ）をそのままにしてレコード確定をおこないます。
- いいえ..... レコードは確定されずにフィールドが編集中の状態に戻ります。
- フィールド復帰..... 入力されているデータが元に戻ります。

［データの入力時にユーザによる上書きを許可する］チェックが　オフ

- OK ..... レコードは確定されずにフィールドが編集中の状態に戻ります。
- フィールド復帰..... 入力されているデータが変更前の内容に戻ります。

次図のようにボタンが変わります（標準のエラーメッセージは設定条件により内容が変わります）。



図 6-16 文字数制限で、標準のエラーメッセージを表示、上書きを許可する場合の設定

6.5.5 設定の保存

［データベースの管理］ダイアログを閉じないと、設定が保存されません。［キャンセル］を押すと、それまでのすべての変更が破棄されます。

カスタム App の役割も含めて考慮し、適切な制限（と思う内容）を設定したら、実際に入力して試すことでユーザにわかるかどうかを改めて確認します。

エラーの多さに注意
たくさんの制限を設定すると、エラーメッセージが頻繁に表示されてユーザが入力作業を面倒だと考えるかもしれません。

「見積作成管理システム」では入力値の制限は、次ページからの表のように設定します。

（ここから、本書に付属している演習ファイルの「6_4 から 5」フォルダの中の「6_ 見積作成管理 .fmp12」ファイルをベースに進めます）

	見積情報	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	顧客外部キー	テキスト		
7	自社外部キー	テキスト		
8	発行日	日付	今日　→　作成日	西暦 4 桁の日付、空欄不可
9	見積番号	数字	10001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
10	c_ 見積総額	計算（結果は　数字）		
11	件名	テキスト		最大文字数 30 カスタムメッセージ データの入力時上書き不可
12	期限日	日付	発行日＋ 1 ヶ月後	西暦 4 桁の日付、空欄不可
13	c_ 見積小計	計算（結果は　数字）		
14	c_ 消費税	計算（結果は　数字）		
15	備考	テキスト		
16	見積状況	テキスト	「作成」	空欄不可
17	見積結果	テキスト	「注文なし」	空欄不可
18	消費税	数字	発行日の消費税 8%だったら 0.08	数字、空欄不可

演習で設定するオプションはこの枠の中です。

	見積明細	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	見積外部キー	テキスト		
7	商品外部キー	テキスト		
8	数量	数字	「0」	数字
9	c_ 金額	計算（結果は　数字）		
10	備考	テキスト		
11	商品番号	数字	商品テーブルから同名フィールドを計算値自動入力	
12	商品名	テキスト	商品テーブルから同名フィールドを計算値自動入力	
13	単価	数字	商品テーブルから同名フィールドを計算値自動入力	

	顧客情報	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	自社担当者外部キー	テキスト		
7	顧客番号	数字	20001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
8	会社名	テキスト		
9	郵便番号	テキスト		
10	住所	テキスト		

	顧客担当者	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	顧客外部キー	テキスト		
7	所属	テキスト		
8	肩書	テキスト		
9	氏名	テキスト		
10	電話番号	テキスト		
11	FAX 番号	テキスト		
12	電子メール	テキスト		

	自社情報	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	自社名	テキスト		
7	郵便番号	テキスト		
8	住所	テキスト		
9	電話番号	テキスト		
10	FAX 番号	テキスト		
11	電子メール	テキスト		

	自社担当者	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	社員番号	数字		数字
7	所属	テキスト		
8	氏名	テキスト		

	商品情報	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	顧客外部キー	テキスト		
7	商品番号	数字	30001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
8	商品名	テキスト		
9	単価	数字	「0」	数字

「見積情報」テーブルの［入力値の制限］を設定します。「見積状況」と「見積結果」フィールドは、後の章で設定します。他のテーブルは、演習ファイルに設定済みです。

「6_4 から 5」フォルダ

演習 6-4　入力値の制限の設定 1

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　［データベースの管理］ダイアログの「テーブル」タブで、「見積情報」テーブルをダブルクリックして「見積情報」テーブルの「フィールド」タブへ移動します。

(2)　「発行日」フィールドを選択し［オプション］をクリックします。

(3)　［フィールドオプション］ダイアログで「入力値の制限」タブを選択します。

(4)　表で計画した内容に基づいて入力値の制限を設定します。

「発行日」フィールド
・［タイプ］は　**オン**　にして　西暦 4 桁の日付　を選択
・［空欄不可］は　**オン**

(5)　［OK］をクリックします。

(6)　さらに 4 つのフィールドにオプションを設定します。

「期限日」フィールド
・［タイプ］は　**オン**　にして　西暦 4 桁の日付　を選択
・［空欄不可］は　**オン**

「見積番号」フィールド
・［空欄不可］［ユニークな値］を　**オン**

「見積状況」フィールド
・［空欄不可］は　**オン**

「見積結果」フィールド
・［空欄不可］は　**オン**

「消費税」フィールド
・［タイプ］は　**オン**　にして　数字　を選択
・［空欄不可］は　**オン**

(7)　設定が終わったら、［OK］をクリックして［データベースの管理］ダイアログを閉じます。
入力値の自動化が設定されました。

（設定内容については、次ページの図も参照してください）

演習ここまで



図 6-17 フィールドオプション設定例

演習 6-5　入力値の制限の設定 2

- **使用ファイル：6_ 見積作成管理 .fmp12**

続けて「件名」フィールドにオプションを設定します。詳細な手順は省略します。

・[データの入力時にユーザによる上書きを許可する] のチェックを　**オフ**
・[最大文字数：] は　「30」
・[制限値以外の入力時にカスタムメッセージを表示] のチェックを　**オン**
　次のように入力します。

件名は 30 文字以内で入力してください。
(印刷範囲を超えてしまいます)



図 6-18 フィールドオプション設定例

演習ここまで

設定できました。

自動入力、制限がどのように動作するか確認します。

演習 6-6　フィールドオプションの確認 1：入力値の自動化

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　「見積情報」レイアウトに切り替えて、新規レコードを 1 件作成します。

(2)　［入力値の自動化］に設定されている内容を確認します。
下表が設定内容です（設定されていないフィールドは省略しています）。

見積情報	タイプ	入力値の自動化	入力値の制限
主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
発行日	日付	今日　→　作成日	西暦 4 桁の日付、空欄不可
見積番号	数字	10001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
期限日	日付	発行日＋ 1 ヶ月後	西暦 4 桁の日付、空欄不可
見積状況	テキスト	「作成」	空欄不可
見積結果	テキスト	「注文なし」	空欄不可
消費税	数字	発行日の消費税 10%だったら 0.1	数字、空欄不可

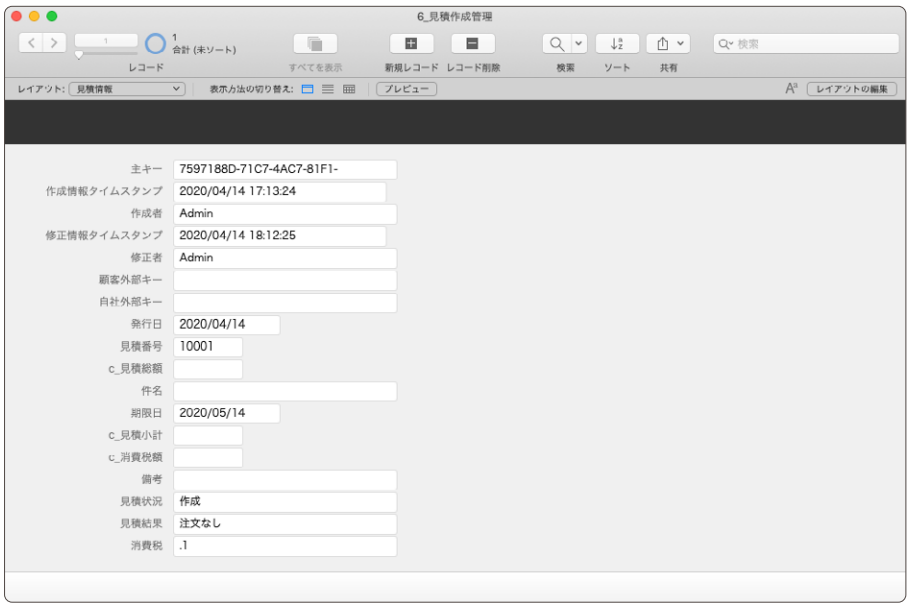


図 6-19　見積情報の新規レコード作成後の例

「主キー」フィールドに UUID
「作成情報タイムスタンプ」「修正情報タイムスタンプ」フィールドに「現在の日時」
「作成者」、「修正者」フィールドに「Admin」
「発行日」フィールドに「今日」が入力されています。
「見積番号」フィールドにシリアル番号の［次の番号：］に設定した「10001」
「期限日」フィールドに「発行日から 1 ヶ月後の日付」
「見積状況」フィールドに「作成」
「見積結果」フィールドに「注文なし」
「消費税」フィールドに「.1」

これらが入力されているはずです。設定したオプションが正しく動作したか確認してください。

(3)　「主キー」、「見積番号」フィールドなど［データ入力時の値変更禁止］ **オン** のフィールドにカーソルを入れて編集しようとしてみてください。



図 6-20 データ入力時の値変更禁止が設定されている場合のエラーメッセージ

(4)　前図のエラーメッセージが表示されれば、［データ入力時の値変更禁止］ **オン** が正しく設定されています。

演習ここまで

※ 主キーは、計算値の自動入力 Get (UUID) の結果です。UUID については、第 10 章も参照してください。

演習 6-7　フィールドオプションの確認 2：入力値の制限：空欄不可 1

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　引き続き、［入力値の制限］が設定された内容を確認します。
下表が設定内容です（設定されていないフィールドは省略しています）。

見積情報	タイプ	入力値の自動化	入力値の制限
主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
発行日	日付	今日　→　作成日	西暦 4 桁の日付、空欄不可
見積番号	数字	10001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
件名	テキスト		最大文字数 30 カスタムメッセージ データの入力時上書き不可
期限日	日付	発行日＋ 1 ヶ月後	西暦 4 桁の日付、空欄不可
見積状況	テキスト	「作成」	空欄不可
見積結果	テキスト	「注文なし」	空欄不可
消費税	数字	発行日の消費税 8%だったら 0.08	数字、空欄不可

(2)　「発行日」フィールドを消去してレコードを確定しようとすると、空欄不可のエラーメッセージが表示されます。

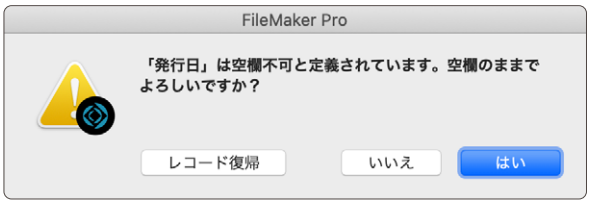


図 6-21 空欄不可／データの入力時にユーザによる上書きを許可するがオンのエラーメッセージ

(3)　［はい］をクリックすると日付は消えます。続けて「期限日」フィールドの同様のメッセージが表示されますので、［はい］をクリックします。

(4)　［はい］をクリックすると続けて「消費税」フィールドの同様のメッセージが表示されますので、［はい］をクリックします。

(5)　「発行日」フィールドの日付を消去すると、計算式で計算値自動入力をしていた「期限日」「消費税」の 2 つのフィールドのデータも消えました。

演習ここまで」

3 つのフィールドが連動しました。なぜでしょうか。空欄になった理由は［計算を指定ダイアログ］の設定です。

・［すべての参照フィールドが空の場合は評価しない］が　**オン**

「期限日」フィールドの計算式は次の内容でした。「発行日」フィールドだけが参照されています。

```
Date ( Month ( 発行日 ) +  1 ; Day ( 発行日 ) ; Year ( 発行日 ))
```

「発行日」フィールドへ新しいデータ（空欄）が入力されましたが、「すべての参照フィールドが空」だったので計算は実行されませんでした。その結果、「空欄不可」の制限が評価後エラーになり、メッセージが表示されました。

※「消費税」フィールドも同様の理由で空欄になりました。



図 6-22［すべての参照フィールドが空の場合は評価しない］がオン

［すべての参照フィールドが空の場合は評価しない］のチェックがオフだとどうなるのかを試します。

演習 6-8　フィールドオプションの確認 2：入力値の制限：空欄不可 2

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　「期限日」フィールドの［入力値の自動化］の［計算を指定］ダイアログで、［すべての参照フィールドが空の場合は評価しない］を　**オフ**　にします。

(2)　「発行日」フィールドにカーソルを入れて［挿入］–［現在の日付］を選択してレコードを確定します。

(3)　今日が入力されて、再度「期限日」「消費税」フィールドが計算されます。

(4)　「発行日」フィールドを空にしてレコードを確定しようとすると、空欄不可のエラーメッセージが表示されます。

(5)　「はい」をクリックします。

(6)　「期限日」フィールドが「？」になりました。

(7)　続けて「消費税」フィールドの空欄不可のメッセージが表示されますので、［はい］をクリックします。

演習ここまで

今回は「期限日」フィールドのエラーは表示されませんでした。しかし「？」になりました。「発行日」フィールドへ新しいデータ（空欄）が入力されましたが計算式がエラーになって「？」が表示されています。

設定はそのままにして違うエラーを試します。

入力値の制限で「日付」「時刻」「タイムスタンプ」の制限を設定したフィールドでは、その書式と違う書式のデータを入力することはできません。**設定と違うデータを入力**して試します。

演習 6-9　フィールドオプションの確認 3：入力値の制限：西暦 4 桁

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　「発行日」フィールドに半角で「20/8/25」と入力します。レコードを確定しようとすると、西暦 4 桁ではないので、エラーメッセージが表示されます。



図 6-23 西暦 4 桁　が設定されている場合のエラーメッセージ

(2)　［フィールド復帰］をクリックします。

(3)　続けて、空欄不可のエラーメッセージが表示されます。

(4)　［はい］をクリックすると空欄になります。

(5)　続けて「消費税」フィールドの空欄不可のメッセージが表示されますので、［はい］をクリックします。

(6)　もう一度、「発行日」フィールドに「あいうえお」と入力してレコードを確定しようとすると、エラーメッセージが表示されます。

(7)　今度は、［OK］をクリックすると、カーソルがフィールドに入ったままとなります。

(8)　2020/8/25 などの西暦が 4 桁の日付を入力して、レコードを確定します。

(9)　「期限日」「消費税」フィールドにもデータが入力されます。

演習ここまで

［入力値の自動化］と［入力値の制限］の設定の組み合わせで起きることを把握して設定する必要があることがわかります。

次は「件名」フィールドの**文字数制限**を試します。設定したカスタムメッセージが表示されるか確認します。

演習 6-10　フィールドオプションの確認 3：入力値の制限：最大文字数

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　「件名」フィールドに 31 文字以上を入力します。レコードを確定しようとすると、カスタムした独自のエラーメッセージが表示されます。



図 6-24 カスタムメッセージが設定／ユーザによる上書きを許可するがオンのエラーメッセージ

(2)　メッセージを確認したら［フィールド復帰］をクリックします。

演習ここまで

いくつかのエラーメッセージと、メッセージの選択肢とその挙動について確認しました。

［入力値の自動化］と［入力値の制限］の設定は大切なことです。『どのようになるべきか』を考えて設定します。

6.6 その他の制限方法

入力値の制限について違う方法も可能です。

スクリプトトリガ

OnObject…ではじまるスクリプトトリガを使用して、より細かく入力値を制限することができるかもしれません。しかし、スクリプトトリガは原則としてレイアウト上で動作します。インポートでは制限されません。あるレイアウトにだけ設定し、別のレイアウトでは設定しないと、制限した意味がなくなる場合があります。

計算値の自動入力を制限に利用する

［入力値の自動化］の［計算値］を使用すると、間接的に入力されたデータに制限をかけることが可能です。

`Self`関数

`Self`関数を使って式に自分自身を指定できます。例えば、入力されるデータの英数字を常に半角にしたいなら次のように`RomanHankaku`関数と`Self`関数を組み合わせて使用します。

```
RomanHankaku ( Self )
```

Self 関数と、テキスト関数、日本語関数を使って決まった文字数だけにしたり、英字、数字の全角半角、ひらがな、カタカナの全角、半角、電話番号の xxx-xxxx-xxxx から数字だけをフィルタして取り出すこともできます。

この方法はエラーメッセージを表示せずに入力した内容を加工するので、ユーザのストレスが少なくなります。しかし、ユーザが意図しない加工になる可能性があるため、要件定義でルールを明確にする必要があります。

条件付き書式／プレースホルダテキストをエラーメッセージ代わりに使う

［入力値の制限］のエラーメッセージ以外の方法で、どのようなデータを入力するべきかをユーザに知らせることが可能です。

レイアウト上で条件付き書式で、制限したいフィールドの文字数をカウントし「30 文字より多かったら」という条件で、赤いテキストオブジェクトの「制限を超えています。印刷可能な文字数は 30 文字までです」と表示するというようなことです。

空欄不可の代わりに、フィールドが空欄だと表示されるプレースホルダテキストを使うこともできます。

いずれもエラーメッセージではありませんので、ユーザのストレスが少なくできます。しかし、スクリプトトリガと同じくレイアウト上でのみ使えるやり方です。また、ユーザがそれらを無視すると期待されているデータが入力されません。

6.7　発行日と期限日

「見積作成管理システム」の「発行日」「期限日」フィールドを例に、どのような設定が良いのか考えてみます。

いま、「期限日」フィールドは「？」になる可能性があるフィールドオプションのままです。下表の薄く色を塗った設定です。

発行日	期限日	
	すべての参照フィールドが空の場合は評価しない	内容
日付が入力されている	オン	1 ヶ月後が正しく計算される
	オフ	1 ヶ月後が正しく計算される
空欄	オン	空欄
	オフ	エラー（？になる）

「期限日」フィールドを「？」のエラーにしないためには、

・「期限日」フィールドの［すべての参照フィールドが空の場合は評価しない］を　**オン**

元に戻せばエラーになりません。それだけで良いでしょうか？ 2 つのフィールドのフィールドオプションは下表です。

フィールド名	タイプ	入力値の自動化	入力値の制限
発行日	日付	作成日（今日）	西暦 4 桁の日付 空欄不可 データの入力時にユーザによる上書きを許可する　オン
期限日	日付	発行日＋ 1 ヶ月後（計算式） すべての参照フィールドが空の場合は評価しない　オン	西暦 4 桁の日付 空欄不可 データの入力時にユーザによる上書きを許可する　オン

実際に使われることを考えます。そもそも「？」の原因は「発行日」フィールドが「空欄」の場合でした。しかし、発行日も期限日もない見積書は困ります。

・両フィールドとも空欄にできないようにする

このほうがよさそうです。

フィールド名	タイプ	入力値の自動化	入力値の制限
発行日	日付	作成日（今日）	西暦 4 桁の日付 空欄不可 **データの入力時にユーザによる上書きを許可する　オフ**
期限日	日付	発行日＋ 1 ヶ月後（計算式） **すべての参照フィールドが空の場合は評価しない　オン**	西暦 4 桁の日付 空欄不可 **データの入力時にユーザによる上書きを許可する　オフ**

これでどうでしょうか。**上表の設定で試します。**

演習 6-11　入力値の自動化と制限の設定

- **使用ファイル：6_ 見積作成管理 .fmp12**

詳細な手順は省略します。

「発行日」フィールドに入力値の制限オプションを設定します。
・［データの入力時にユーザによる上書きを許可する］のチェックを　**オフ**

「期限日」フィールドでは、
入力値の制限オプションは、
・［データの入力時にユーザによる上書きを許可する］のチェックを　**オフ**
入力値の自動化オプションは、［計算を指定］ダイアログの、
・［すべての参照フィールドが空の場合は評価しない］のチェックを　**オン**

演習ここまで

確認します。

演習 6-12　発行日と期限日のフィールドオプション確認 1

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　「見積情報」レイアウトで新規レコードを 1 件作成します。

(2)　「発行日」フィールドに今日の日付が入力されます。

(3)　「発行日」フィールドを空にすると、エラーメッセージが表示されます。



図 6-25 空欄不可／ユーザによる上書きを許可するがオフのエラーメッセージ

(4)　［OK］をクリックすると、カーソルがフィールドに入ったままとなります。
空欄にすることはできません。

(5)　2018/9/25 などの西暦が 4 桁の日付を入力します。

演習ここまで

次は**「期限日」フィールドを空にしてみます。**

演習 6-13　発行日と期限日のフィールドオプション確認 2

- **使用ファイル：6_ 見積作成管理 .fmp12**

(1)　「期限日」フィールドを空にすると、エラーメッセージが表示されます。



図 6-26 空欄不可／ユーザによる上書きを許可するがオフのエラーメッセージ

(2)　[OK] をクリックすると、カーソルがフィールドに入ったままとなります。
空欄にすることはできません。

(3)　2018/10/25 などの西暦が 4 桁の日付を入力します。

演習ここまで

現在、「期限日」フィールドだけを空欄にしても空欄不可であるため、そのエラーが表示されてフィールドの確定ができません。

また、[データの入力時にユーザによる上書きを許可する] が　**オフ**　の場合、エラーメッセージに [フィールド復帰] ボタンが表示されません。そのため、もとのデータに戻せません。

常に「発行日」フィールドの日付から「期限日」フィールドの日付を計算するのであれば、「期限日」フィールドをブラウズモードで入力不可にするほうが良いのかもしれません。

フィールド名	タイプ	入力値の自動化	入力値の制限
発行日	日付	作成日（今日）	西暦 4 桁の日付 空欄不可 **データの入力時にユーザによる上書きを許可する　オフ**
期限日	日付	発行日＋ 1 ヶ月後（計算式） **すべての参照フィールドが空の場合は評価しない　オン**	西暦 4 桁の日付 空欄不可 **データの入力時にユーザによる上書きを許可する　オフ**

期限日はユーザがレイアウト上で入力できない方がよい？

なお、「期限日」フィールドの空欄不可を解除してから、演習 6-13 を試しても計算式は評価されません。空欄のままになります。参照フィールドである「発行日」フィールドのデータが変更されないからです。

やってみよう

上記の作業をやってみましょう。

また、この設定では、次のようなことが起こります。

(1)「発行日」フィールドを空欄にする
(2) 空欄不可のエラーメッセージが表示される
(3)［OK］をクリック
(4) 日付を入力しようとして「20/8/25」のような文字列を入力する
(5) 西暦 4 桁のエラーメッセージが表示される
(6)［フィールド復帰］をクリック
(7)（フィールドが空になる）
(8) 空欄不可のエラーメッセージが表示される……［OK］をクリック……

このように、エラーメッセージが何度も表示されることがあります。エラーにならないデータが入力されるまでフィールドからカーソルを移動することができません。

やってみよう

上記の作業をやってみましょう。

それとも、このような考え方もできます。制限を厳しくしない方法です。

見積書の「発行日」や「期限日」のような項目は、業務上重要なのでユーザが空欄であることに気づきやすいものです。さらに入力用のレイアウトでの示唆や、印刷用のスクリプトで空欄であること伝えることで、空欄にならないようにカバーすることができます。

「4 桁でない西暦が入力されないようにする」ということが重要であれば、それだけを制限として、空欄不可は止めてしまうということも可能です。

または「期限日」が必ずしも「発行日の 1 ヶ月後」と限らない場合があるかもしれません。

要件定義では、発行日と関係なく『期限日なしでも良いかどうか』『期限日に任意の日付を入力できるようにしたいかどうか』も確認しておくことも必要です。これは、業務上の通常ルールではないイレギュラーなルールに柔軟に対応することに繋がります。

本書では、次表の内容でフィールドオプションを設定します。

フィールド名	タイプ	入力値の自動化	入力値の制限
発行日	日付	作成日（今日）	西暦 4 桁の日付 空欄不可 **データの入力時にユーザによる上書きを許可する　オフ**
期限日	日付	発行日＋ 1 ヶ月後（計算式） **すべての参照フィールドが空の場合は評価しない　オン**	西暦 4 桁の日付 ~~空欄不可~~ **データの入力時にユーザによる上書きを許可する　オン**

（第 7 章以降はこの内容をベースに進めていきます）

変更になったフィールドオプションは薄く塗った部分です。

	見積情報	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	顧客外部キー	テキスト		
7	自社外部キー	テキスト		
8	発行日	日付	今日（作成日）	西暦 4 桁の日付、空欄不可 データの入力時上書き不可
9	見積番号	数字	10001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
10	c_ 見積総額	計算（結果は　数字）		
11	件名	テキスト		最大文字数 30 カスタムメッセージ データの入力時上書き不可
12	期限日	日付	発行日＋ 1 ヶ月後	西暦 4 桁の日付
13	c_ 見積小計	計算（結果は　数字）		
14	c_ 消費税	計算（結果は　数字）		
15	備考	テキスト		
16	見積状況	テキスト	「作成」	空欄不可
17	見積結果	テキスト	「注文なし」	空欄不可
18	消費税	数字	発行日の消費税 8%だったら 0.08	数字、空欄不可

消費税

「消費税」フィールドも計算式で参照しているので「発行日」フィールドと連動しています。ただし、税率切り替えの過渡期などは、発行日と関係なく税率を設定したいという要望があるかもしれません。現在の計算式は次の内容です。

```
Case ( 発行日 < Date ( 10 ; 1 ; 2019 ) ; 0.08 ; 0.1 )
```

発行日ではなく、新規レコードを作成した時点の日付を使って初期値の税率を入力するには次の内容となります。

```
Case ( Get ( 日付 ) < Date ( 10 ; 1 ; 2019 ) ; 0.08 ; 0.1 )
```

さらに、ユーザが値一覧などで消費税率を選択できるようにしても良いかもしれません。

様々に試してきました。どのような設定がユーザにとって有益であるかを考えてみてください。

第 7 章 データと索引・格納

引き続きフィールドオプションについて解説します。フィールドの索引、データ格納の種類とバイナリファイルのためのオブジェクトタイプフィールドについて解説します。

7.1 「データの格納」タブについて

データは通常、1 テーブル、1 レコードの 1 フィールドに 1 つのデータが保存されます。保存されているデータの次の設定は、フィールドオプションの「データの格納」タブを使います。

・索引
・グローバル格納
・バイナリデータの格納（オブジェクトタイプフィールド）

これらの設定を活用することでカスタム App を強化できます。



図 7-1［フィールドオプション］ダイアログ

（オブジェクトタイプフィールドは［フィールドオプション］ダイアログの設定内容が異なります。まず、通常の設定について確認します）

繰り返し
このオプションを設定すると 1 つのフィールドに複数のデータを格納できるようになります。「繰り返しフィールド」と呼ばれます。複数のデータを一括で取り扱ったり、そのうちの 1 つのデータだけを取り出したりするための機能や関数が用意されています。本書では詳しく解説しません。FileMaker Pro のヘルプを参照してください。

7.2 索引

フィールドに保存されているデータには、**「索引」**を作成できます。索引は本の最後についている「索引」と同じようなものです。文言を基準に従って並べて、文言ごとに表示されている場所を整理しておいて利用するものです。

索引が作成されているフィールドを対象に検索を実行すると、索引が作成されていないフィールドを検索するよりも時間が短くなります。索引は、

・検索
・関連（リレーションシップ）
・フィールドのデータを使用した値一覧
・既存のデータを使用してオートコンプリート
・入力値の制限におけるユニークなデータ、既存値

で使用されています。通常、ユーザの目に見えないところに作成、保存されています。

7.2.1 索引内容

［索引設定］で種類を選択します。次の 3 種類から選択できます。フィールドタイプにより選択できる内容が異なります。

・なし
・最小限
・すべて

フィールドが作成されたときに自動で設定されるのは以下の内容です。

・［索引設定］は 「なし」
・［必要時に索引を自動設定］のチェック　**オン**

通常は、この設定を特に変更する必要はありません。「必要時」とは、

・検索や関連などでフィールドが「使用（評価）」されるとき

です。評価されたタイミングで索引が作成され、自動的に次の設定に変わります。

・［索引設定］は「最小限」

使用を開始したばかりのカスタム App では、検索実行時に時間がかかることがあります。それは索引を作成しているためかもしれません。その場合、2 回目以降は作成された索引を使用して検索するので、検索時間が短くなります。

索引とファイルサイズ
索引が作成されるとカスタム App のファイルサイズは大きくなります。当然ですが、レコード数やフィールドに格納されるデータが増加するに従って索引のサイズも大きくなります。

索引と更新

フィールドに保存されているデータが、ユーザやスクリプトなどにより任意のタイミングで内容に変更（入力、更新、消去）が起きると索引が作成、更新されます。もしも、大量レコードのデータ更新をしたとき、カスタム App の応答速度が遅いと感じることがあったら、それは索引の処理のためかもしれません。インポートでも索引の作成、更新が起きますし、レコード削除も不要になった索引の削除が行われるので影響があるかもしれません。

デフォルト言語

デフォルト言語の設定は索引とソート順、デフォルトフォントの適用時に使用されます。通常、日本であれば「日本語」が設定されています（ファイルを作成したパソコンの OS の言語設定に基づいています）。

7.2.2 フィールドタイプと索引

フィールドタイプによって、作成される索引の内容が異なります。

テキストタイプフィールド

テキスト、または計算タイプフィールドの計算結果がテキストの設定であるフィールドは、索引設定で「**なし**」「**最小限**」「**すべて**」を選択できます。また、2 種類の索引を作成できます。

値の索引

フィールドに格納されたデータの最初の 100 文字までを使用して作成されます。

単語の索引

フィールドに格納されたデータからユニークな単語を切り出すものです。検索実行に使用されます。

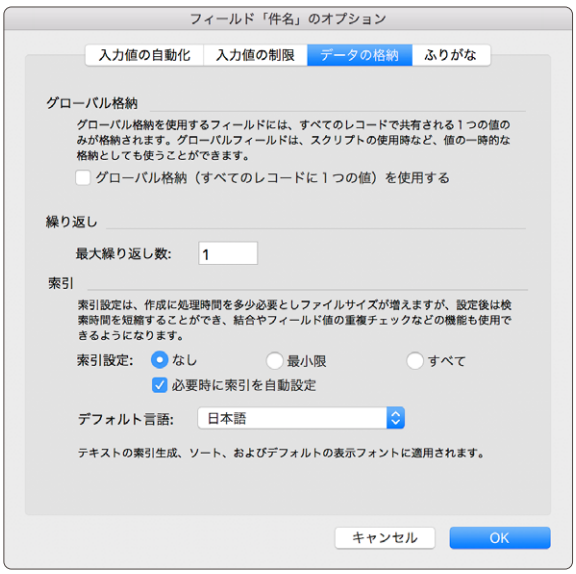


図 7-2 テキストタイプのフィールドの索引設定

[索引設定] で「すべて」を選択すると、値と単語の両方の索引が作成されます。「最小限」の場合、値か単語のどちらかの索引が作成されます。

数字、日付、時刻、タイムスタンプタイプ

数字、日付、時刻、タイムスタンプのフィールドタイプと、これらの結果タイプが設定されている計算タイプフィールドでは、索引設定で「**なし**」あるいは「**すべて**」のどちらかを選択できます。なお、値の索引だけが作成されます。



図 7-3 数字タイプのフィールドの索引設定

（ここから、本書に付属している演習ファイルの「7 章開始時」フォルダの中の「7_ 見積作成管理 .fmp12」ファイルをベースに進めます）

「見積作成管理システム」では、商品の説明文と画像を入力する項目の要望があり設計が変更になりました。「商品情報」テーブルにフィールドが追加されています。下表の薄く塗った部分です。

	商品情報	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	顧客外部キー	テキスト		
7	商品番号	数字	30001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
8	商品名	テキスト		
9	単価	数字	「0」	数字
10	商品説明	テキスト		
11	商品画像	オブジェクト		

※ 商品画像と説明にはダミーデータが入力されています。

索引の内容（値の索引と単語の索引の違い）を確認します。

「7 章開始時」フォルダ

演習 7-1　値の索引と単語の索引の違いを確認する

- **使用ファイル：7_ 見積作成管理 .fmp12**

(1)　［データベースの管理］ダイアログを開いて「商品情報」テーブルの「フィールド」タブへ切り替えてください。

(2)　「商品情報」テーブルに「商品説明」フィールドが作成されています。

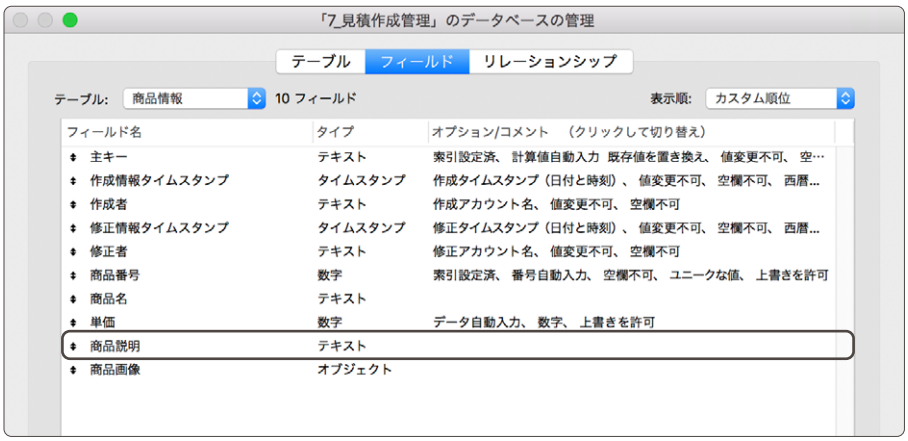


図 7-4「商品説明」フィールド

(3)　「商品説明」フィールドをダブルクリックして［フィールドオプション］ダイアログを開きます。「データの格納」タブをクリックします。索引は作成されていません。



図 7-5 索引が作成されていない状態

(4)　確認後、［キャンセル］を何度かクリックして［データベースの管理］ダイアログを閉じます。

(5)　「商品情報」レイアウトが表示されます（わかりやすいように表形式で表示しています）。

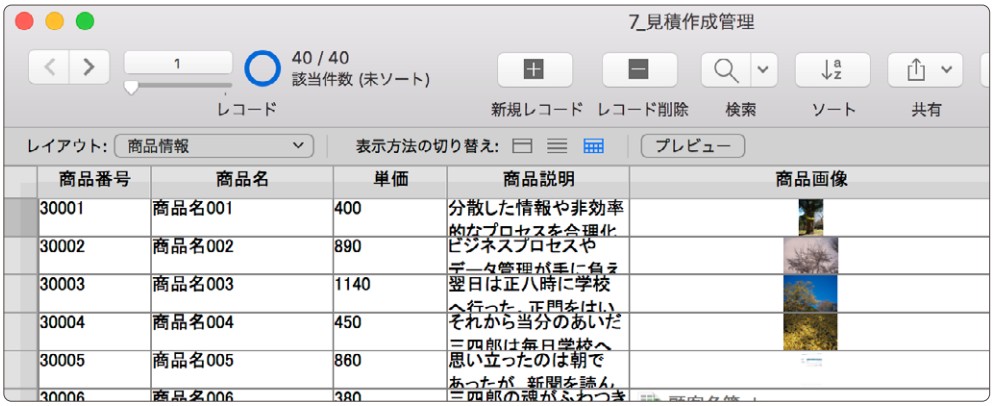


図 7-6「商品情報」レイアウト

(6)　検索を実行します。

a.　検索モードに切り替えます。
b.　「商品説明」フィールドに「翌日」と入力して実行します。
c.　結果は 4 件です。

(7)　検索を実行したので「商品説明」フィールドには索引が作成されました。確認します。

a.　「商品説明」フィールドの」[フィールドオプション] ダイアログの「データの格納」タブを表示します。
b.　索引が作成されていることを確認します（「最小限」に切り替りました）。
c.　[キャンセル] を何度かクリックして [データベースの管理] ダイアログを閉じます。



図 7-7　索引が作成された

(8)　すべてのレコードを表示します。

(9)　索引の内容を確認します。

a. 「商品情報」レイアウトで「商品説明」フィールドにカーソルを入れます。
b. 索引一覧を確認します。[挿入]–[索引一覧 ...] メニューを選択します。
c. [索引一覧] ダイアログが表示されます。
d. 「単語一覧で表示」にチェックをいれます。

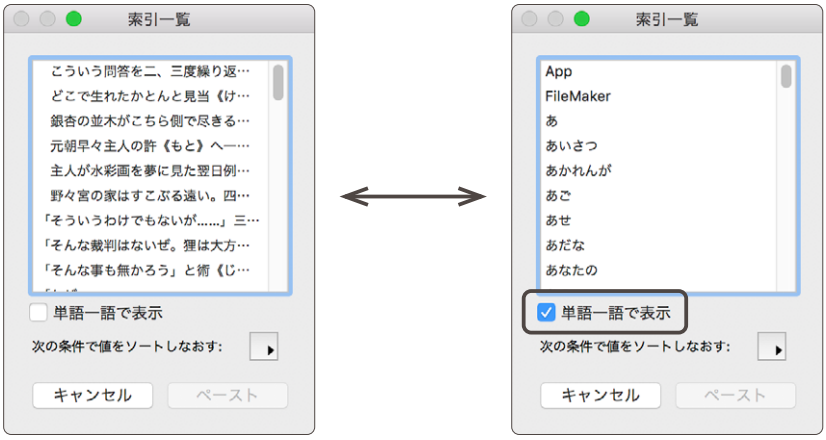


図 7-8［索引一覧ダイアログ］の「値の索引」と「単語の索引」

（10）［キャンセル］をクリックしてダイアログを閉じます。

演習ここまで

テキストタイプのフィールドは、単語の索引が検索対象です。［索引一覧］ダイアログで「単語一語で表示」をチェックしてスクロールすると、演習で検索した「翌日」があります

［索引一覧］ダイアログ

演習で表示を確認した［索引一覧］ダイアログは、フィールドへの入力補助機能として用意されているものです。どのように索引が作成されているか確認するためにも利用できます。

索引と検索

『ある文言ごとに、その文言が出て来る場所を整理してあるもの』と前述しましたが、索引は、どのように検索に使用されるのでしょうか。［索引一覧］ダイアログに表示された索引内容はソートされていましたが、実際の索引もソートされた状態で保持されています。フィールドのデータの変更後に索引を更新し、それが今まで索引になかった文言であれば適切な位置に並べられるということになります。その中を二分して、どちらが検索条件に合致するかを判断し、合致したものを更に二分して……を繰り返すのです。このように検索が実行されます。

(1)　索引が 10000 ある。
(2)　「田中」が検索実行される。
(3)　5000 番目の索引が「田中」と「等しい」か、等しくなければ「田中より　大きいか」を確認。
(4)　大きければ 1 から 5000 までを対象として……と繰り返す。
(5)　「田中」を発見したら終了。
(6)　「田中」の索引にある場所、入力されているレコードが対象レコードとして表示。

すべてのレコードを 1 件ずつ探すよりも早く見つかります。この方法では索引がユニークな単語として整理されていてソートされていることが前提です。

7.2.3 索引が作れないもの

オブジェクトタイプや、非保存の計算、集計タイプには索引は作成されません。グローバル格納のフィールドにも索引は作成できません。

非保存の計算タイプフィールド

計算タイプのフィールドは計算式の中で次のものが参照されていると自動的に非保存になります。

- 関連先のフィールド
- 集計タイプフィールド
- 非保存の計算タイプフィールド
- グローバル格納フィールド

非保存の計算タイプフィールドには索引が保存されませんが、索引が必要な動作に対して、都度作成されます。そのため、パフォーマンス低下などの影響があります。計算式を変更後に計算タイプフィールドが非保存に変わっていたら注意します。

非保存と検索

索引は持てませんが検索できないわけではありません。検索が実行された時点の計算結果に対して索引を作成し、検索が実行されます（そのために結果が表示されるまで少し時間がかかることがあります）。

索引を作らない設定

［索引設定］を「なし」に選択して［必要時に索引を自動設定］のチェックを **オフ** にすると、索引が作成されなくなります。この設定は注意が必要です。例えば、照合フィールドであれば関連が正しく機能しなくなる場合があります。

7.3 データの格納

索引の次は、データの格納を解説します。独特の格納設定が用意されています。

7.3.1 グローバル格納

「グローバル」はフィールドのタイプではありません。特徴のある格納方法です。この特徴をよくわかっていると、様々な活用が可能です。

・グローバル格納のフィールドに格納されるデータは、ファイル全体に対して 1 つ
・索引が作成されない



図 7-9 グローバル格納の設定

通常、レイアウトに設定されている TO（コンテキスト）により表示可能なデータが決まりますが、「**グローバル格納**」を設定したフィールドは、テーブル、レコード、フィールド、関連、コンテキストを意識する必要がなく、どこからでも入力、変更が可能となります。

・テキスト
・数字
・日付
・時刻
・タイムスタンプ
・オブジェクト
・計算

のフィールドタイプにグローバル格納の設定が可能です。

ファイル全体に対して 1 つ、つまり、**どこからでも同じデータが見えることについて確認**します。

演習 7-2　グローバル格納フィールドの見え方を確認する

- **使用ファイル：7_ 見積作成管理 .fmp12**

(1)　「ユーティリティ」レイアウトに切り替えます。

(2)　[データベースの管理] ダイアログを開きます。

(3)　「ユーティリティ」テーブルに「共通情報」フィールドが作成されています。

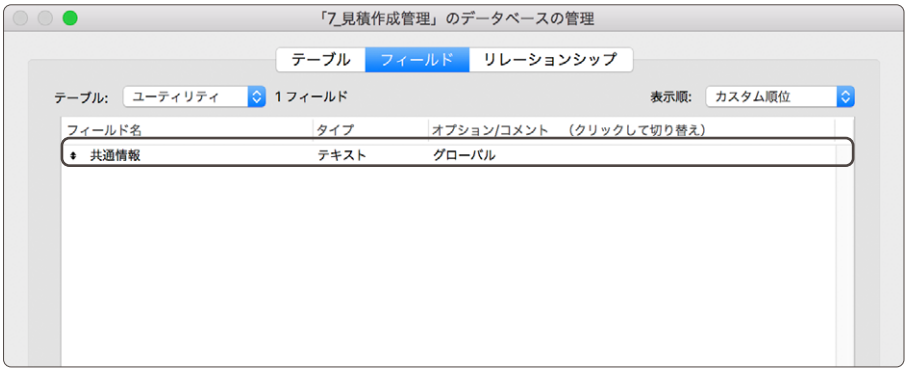


図 7-10「共通情報」フィールド

(4)　「共通情報」フィールドをダブルクリックして [フィールドオプション] ダイアログを開きます。「データの格納」タブをクリックします。次の設定を確認します。

・[グローバル格納（すべてのレコードに 1 つの値）を使用する]　が　**オン**



図 7-11「共通情報」フィールドのフィールドオプション

(5)　リレーションシップグラフで関連がなにも設定されていないことを確かめます。



図 7-12 関連はどのテーブルにも設定されていない

(6)　確認後、[キャンセル] をクリックして [データベースの管理] ダイアログを閉じます。

(7)　「ユーティリティ」レイアウトには「あいうえお」と入力されているフィールド (「共通情報」フィールド) が置いてあります。

(8)　同じフィールドが他のレイアウトにも置いてありますのでレイアウトを行き来します。
すべて同じデータであることを確かめます (「商品情報」レイアウトでは、フォーム形式に切り替えて確認してください)。

(9)　どのレイアウトでも構いませんので「共通情報」フィールドを別のデータに変更します。
(例えば　かきくけこ)。

(10)　レイアウトを行き来します。どこで見ても変更したデータに変わっています。

演習ここまで

これが、グローバル格納の特性の 1 つです。

やってみよう

ファイルを一旦閉じてまた開いても、閉じる前と同じデータが「共通情報」フィールドに格納されていることを確認しましょう。

7.3.2 共有とグローバル格納

グローバル格納フィールドのデータが関連が定義されていなくても、どのコンテキストでも表示されることを確認しました。演習 7-2 は、ローカル上（自分のパソコン上）にファイルでした。

FileMaker Cloud や FileMaker Server で共有されているファイルの場合、

・共有ファイルへ接続している複数のクライアント（ユーザと考えてください）が、1 つのグローバル格納フィールドのデータを同時に表示することが可能

と、なります。さらに、次のような特殊な取扱になります。

・あるユーザが、グローバル格納フィールドのデータを書き換えても他のユーザに表示される内容には反映されない

グローバル格納フィールドでは、クライアントが接続中に保持されている「**セッション**」ごとに、別のデータを扱うことができます。接続（セッション）が終了すると、グローバル格納フィールドのデータは破棄されます。

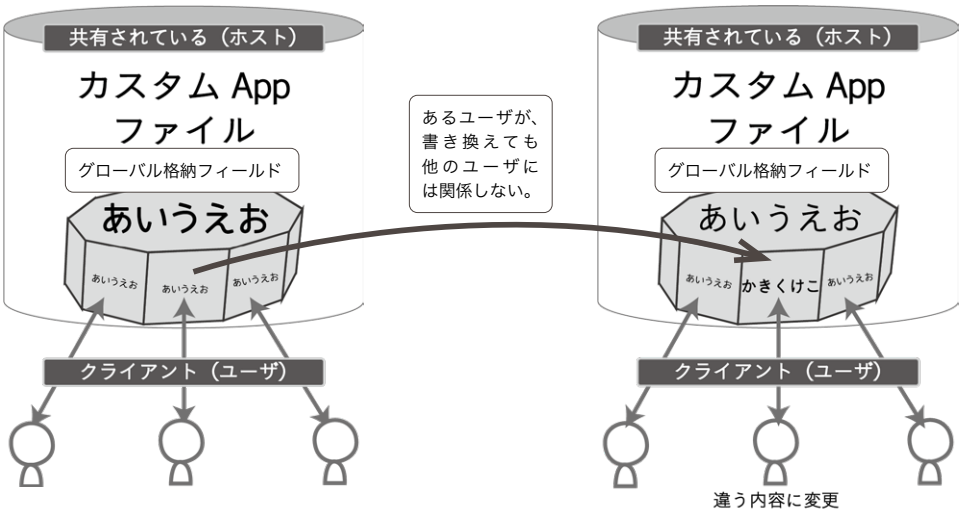


図 7-13 同じデータを見られる／違うデータにできる

セッション
FileMaker Cloud や FileMaker Server は複数のクライアントと正しく通信（やりとり）をできるように「セッション」を保持しています。

セッションごとに別のデータであっても、ユーザにはローカル上の場合と同じく、どこでも同じデータが見えます。

グローバル格納フィールドへのデータの保存
グローバル格納フィールドにデータを保存するには、ファイルをローカル上で開いてデータを入力する必要があります。

共有されているファイルのグローバル格納フィールドにデータを入力しても、データは保存されません。初期値が必要な場合はあらかじめ入力しておく必要があります。

共有されているファイルの場合、グローバル格納フィールドの初期データは様々考えられますし、それらは時々変更が必要になることがあります。その度に共有を停止して、ローカル上にファイルを持ってきて開いて入力するのは手間です（ユーザの使用を一時停止しなければなりません）。ユーザがログインするタイミングで動作するスクリプトなどを作成してグローバル格納フィールドに初期値を設定することをお勧めします。

7.3.3 どこにつくるか

グローバル格納フィールドは、どこにつくるのが良いでしょうか。原則として、グローバル格納フィールドは、どのテーブルにあっても動作は変わりません（一部例外があります。7.3.5 を参照してください）。

1 つのテーブルにあると機能の作成作業などで便利です。『あのグローバル格納フィールド、どこにつくったかな……』ということがありません。テーブル名は「ユーティリティ」などわかりやすい名前にしてください。

7.3.4 検索とグローバル格納

グローバル格納フィールドには、索引が作成されないため検索できません。

やってみよう

検索モードに切り替えても、検索条件を入力することができません。

7.3.5 関連とグローバル格納

関連の定義では、次の場合にグローバル格納フィールドを照合フィールドとして指定できます。

・参照元テーブルの照合フィールド

なお、照合フィールドとして使用する場合、そのテーブルに作成されていなければなりません。

例えば、グローバル格納フィールドの特徴を利用してユーザごとに一時的に保存したいデータの置き場として利用する場合、次のような仕組みがあるとします。

「ユーティリティ」テーブル
・「ログインアカウント」フィールド　→　グローバル格納フィールド
・「使用者名」フィールド　→　グローバル格納フィールド

「ユーザ情報管理」テーブル
・「アカウント名」フィールド
・「氏名」フィールド

「ユーティリティ」テーブルと「ユーザ情報管理」テーブルには次のような関連が定義されています。

「ユーティリティ」TO の「ログインアカウント」＝「アカウント」TO の「アカウント名」
※ グローバル格納フィールドを参照元の照合フィールドとして利用しています。

次のような内容のスクリプトも作成されています。

(1) ユーザのアカウント名を取得して「ログインアカウント」フィールドに保存
(2) 上記の関連を使って「ユーザ情報管理」テーブルの「氏名」フィールドの内容を「使用者名」フィールドに入力する

このスクリプトがユーザのログイン後に実行されると、「使用者名」フィールドにユーザの氏名が入力されます。すると、関連を気にせずに「使用者名」フィールドの氏名を様々なレイアウト上で活用できます。例えば、「送付状」レイアウトで差出人氏名として使うことができます。

さらに、グローバル格納フィールドを関連元テーブルの照合フィールドに使う場合の確認をします。次のような機能です。

・一時的なデータを表示する

（ここから、本書に付属している演習ファイルの「7 章開始時」フォルダの中の「7_ 料理と食材 .fmp12」ファイルをベースに進めます）

「料理と食材 .fmp12」では、料理に食材を関連付けていました。**「料理と材料」レイアウトで食材の説明を見たい場合に、その場で調べられるような機能**をつくります。

まず、関連を定義します。

「7 章開始時」フォルダ

演習 7-3　グローバル格納フィールドの関連：関連の定義

- **使用ファイル：7_ 料理と食材 .fmp12**

(1)　「料理と食材 .fmp12」を開いて、［データベースの管理］ダイアログの「料理」テーブルの「フィールド」タブへ切り替えてください。

(2)　「料理」テーブルに「g_ 食材説明用照合キー」フィールドが作成されています。

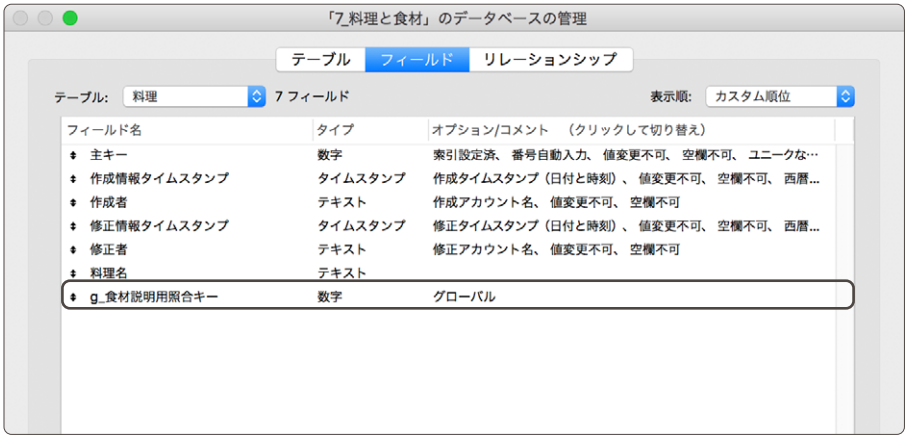


図 7-14「g_ 食材説明用照合キー」フィールド

(3)　「g_ 食材説明用照合キー」をダブルクリックし［フィールドオプション］ダイアログを開きます。「データの格納」タブをクリックします。次の設定を確認します。

・［グローバル格納（すべてのレコードに 1 つの値）を使用する］　が　**オン**



図 7-15「g_ 食材説明用照合キー」のフィールドオプション

(4)　リレーションシップグラフを表示します。
グローバル格納フィールドを使用した関連用の TO があります。「食材」テーブルのものです。

・「料理 _ 食材｜説明」TO



図 7-16「料理 _ 食材｜説明」TO

(5)　関連を定義します。
「料理」TO の「g_ 食材説明用照合キー」と「料理 _ 食材｜説明」TO の「主キー」フィールドを関連付けます。

しかし、TO の四角の中に繋げたい「g_ 食材説明用照合キー」フィールドが表示されていません。

a.　TO の四角の下の▼をクリックして「g_ 食材説明用照合キー」フィールドを表示します。
b.　ドラッグアンドドロップで繋げます。

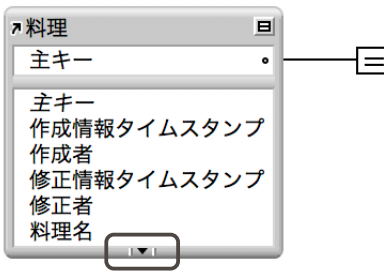


図 7-17 四角の▼

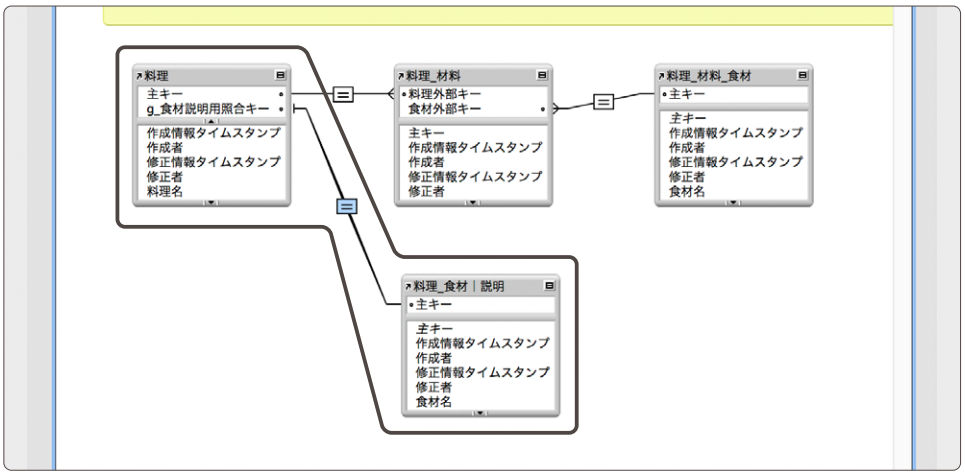


図 7-18 関連付ける

(6) 次は関連の定義とオプションの確認を行います。

a. ［リレーションシップ編集］ダイアログを開きます。

b. 照合フィールドを確認します（違っていたら直してください）。

・左側が「料理」TO で「g_ 食材説明用照合キー」

・右側は「料理 _ 食材｜説明」TO で「主キー」フィールド

・照合条件の演算子は「=」

c. オプションの設定は行いません。

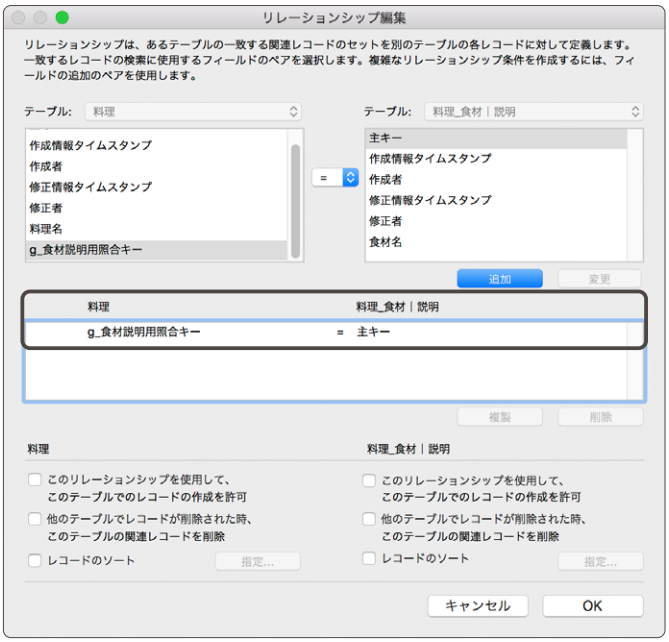


図 7-19［リレーションシップ編集］ダイアログ

(7) ［OK］を何度かクリックして［データベースの管理］ダイアログを閉じます。

演習ここまで

レイアウトに必要なレイアウトオブジェクトを作業します。

演習 7-4　グローバル格納フィールドの関連：レイアウトの作業

- **使用ファイル：7_ 料理と食材 .fmp12**

(1)　「料理と材料」レイアウトでレイアウトモードに切り替えます。

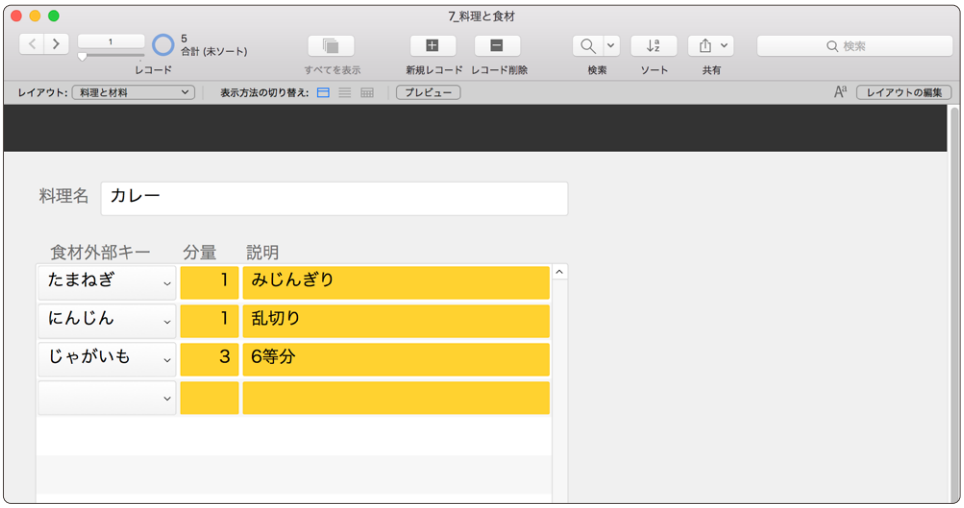


図 7-20「料理と材料」レイアウト

(2)　レイアウトの右側の空きスペースに、新しく「料理 _ 食材 | 説明」TO のポータルを配置します。

a.　［ポータルツール］アイコンをクリックします。

b.　ポータルを配置したい位置でドラッグして四角形を描くようにしてマウスボタンを離すと［ポータル設定］ダイアログが表示されます。

c.　［ポータル設定］ダイアログでは次の設定をします。

・［行数：］は　「1」

・［アクティブな行状態を使用］は　**オフ**

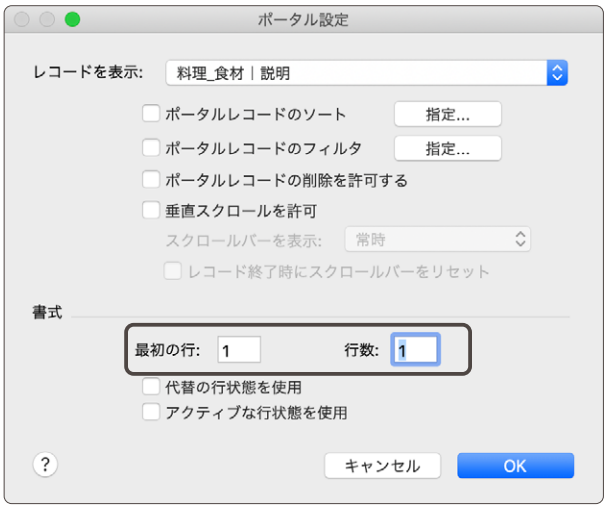


図 7-21　ポータル設定ダイアログ

d. ［OK］をクリックします。
e. ［ポータルにフィールドを追加］ダイアログが表示されます。次のフィールドを指定します。
・「料理 _ 食材｜説明」TO の「写真」「特徴」フィールド



図 7-22「料理 _ 食材｜説明」TO のフィールドを指定する

f. ［OK］をクリックします。
g. ポータルが配置されます。

(3) 次図のように整えます。
a. ポータルを四角形にします。
b. 「写真」フィールドは大きい四角にします。
c. その下に「特徴」フィールドを置きます。

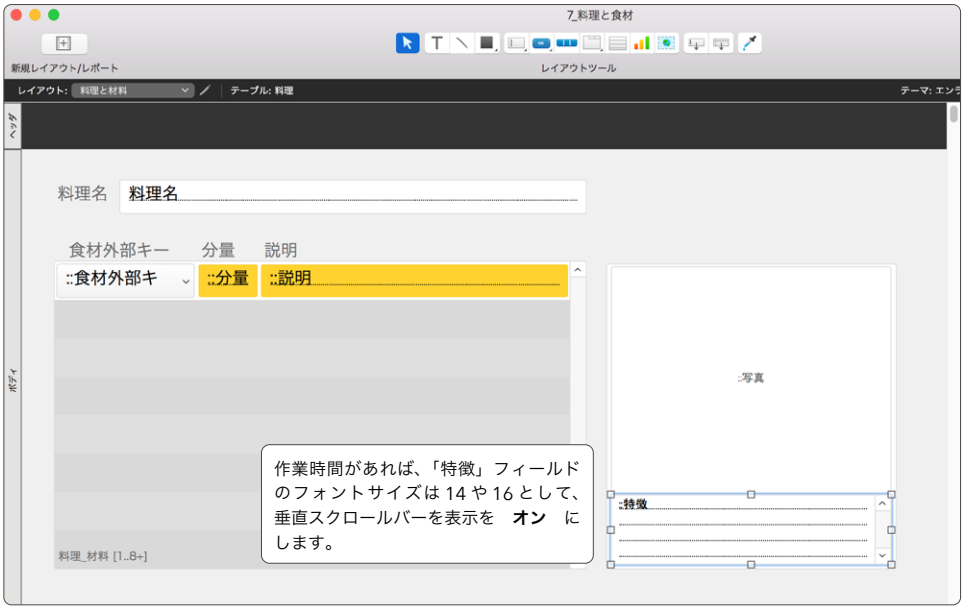


図 7-23 ポータルを調整

(4) レイアウトを保存してブラウズモードに切り替えます。ポータルにはなにも表示されません。現在、「g_ 食材説明用照合キー」にデータが入っていないので関連レコードはありません。

(5)　もう一度レイアウトモードに切り替えます。

(6)　照合フィールドにデータを入力するために「g_ 食材説明用照合キー」を配置します。
(右パネルの「フィールド」タブを表示してください)

・[現在のテーブル] を　「料理」
・[フィールド配置] は　「縦」
・[ラベル] は　「なし」

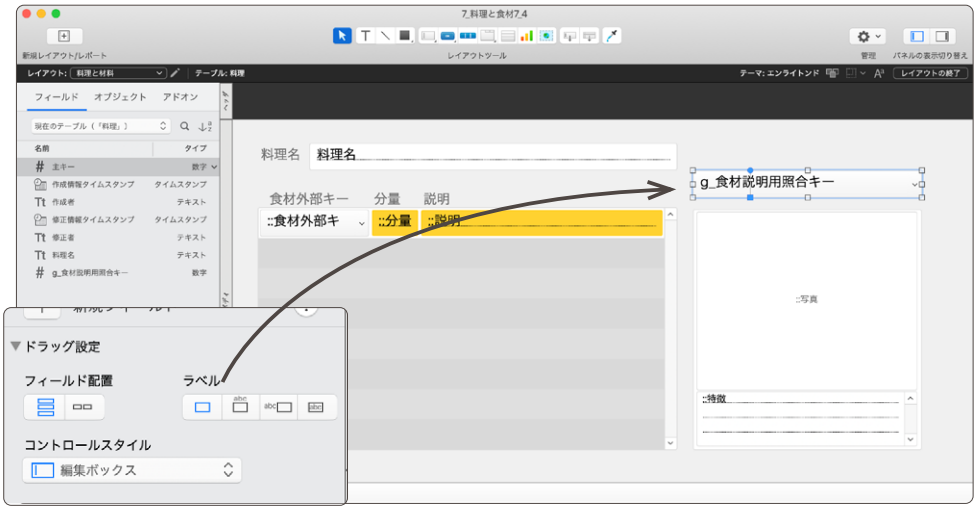


図 7-24　フィールドを配置

(7)　「g_ 食材説明用照合キー」に値一覧を割り当てます（第 5 章の値一覧を利用します)。
(右パネルのインスペクタの「データ」タブを表示してください)

・[コントロールスタイル] は　「ポップアップメニュー」
・[値一覧：] は　「食材リスト」
・[一覧の表示切り替え用矢印を表示] を　**オン**
・[値一覧でデータ書式を上書き] を　**オン**

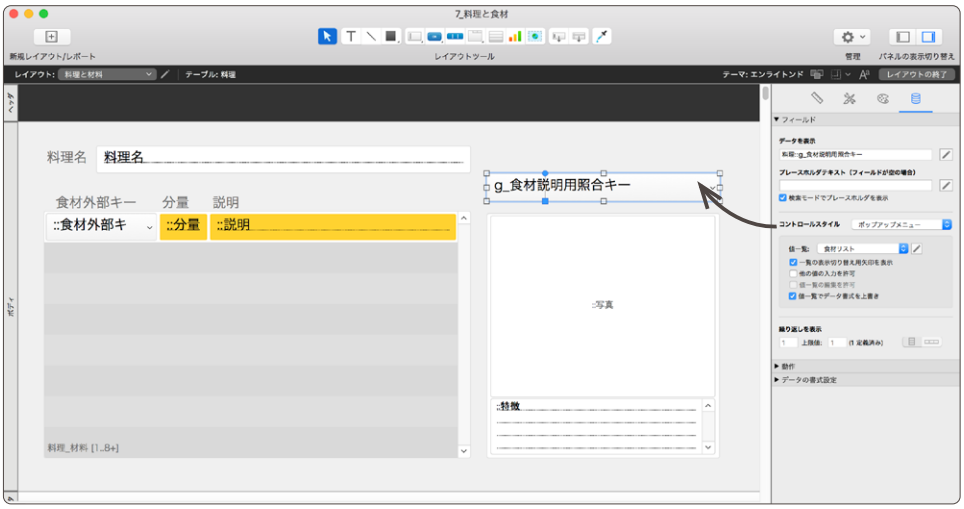


図 7-25　値一覧を割り当てる

(8)　レイアウトを保存してブラウズモードに切り替えます。

(9)　「g_ 食材説明用照合キー」のポップアップメニューで、任意の食材を選択すると、ポータルに情報が表示されます。



図 7-26 作業後の「料理と材料」レイアウト

(10)　レコードを前後に移動しても、表示中の食材の情報は変わらないことを確認します。

演習ここまで

左側のポータルの内容とは関係なく、任意の食材について写真や説明を確認できるようになりました。このような一時的な表示にはグローバル格納フィールドを使った関連が便利です。

グローバル格納でなかったら

もしも、演習 7-3、7-4 で作業した内容をグローバル格納ではないフィールドを照合フィールド指定するとどうなるでしょうか。一時的な照合のためのデータが 1 レコードごとに別のデータとして保存されます。共有されている場合、誰か（ユーザ）が表示のために入力したデータが各レコードに常に保存されている状態になります。

一時的な表示のためのデータは特に保存する必要はありません。それにグローバル格納フィールドであればどこでも「いま表示している食材」を確認できますし、ユーザごとに別のデータを持つことが可能です。

リレーションシップグラフの線があらわすもの

関連を定義した TOG をよくみると次図のような線が描画されています。

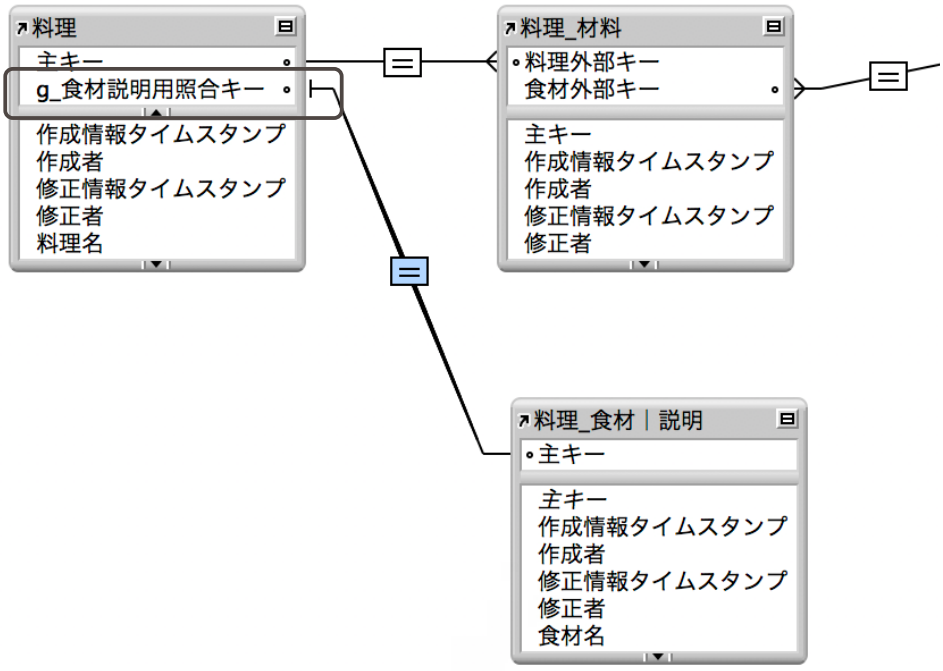


図 7-27 他と違う線が描画されている

「料理」TO の照合フィールドである「g_ 食材説明用照合キー」（グローバル格納フィールド）の関連を表す線の末端部分が縦線になっています。「料理 _ 食材｜説明」TO の「主キー」側は横線です。これは、

・右側の TO から左側の TO は関連が成立しない
・左側の TO から右側の TO の関連は成立する

ということです。どの演算子で照合しても同じです。グローバル格納フィールドには索引が作成されないため、参照先としては機能しないためです。

（「料理と食材 .fmp12」を閉じてください）

7.4 オブジェクトタイプフィールド

フィールドに、文字列ではなく PDF や画像、Microsoft Excel などのファイルを保存したい場合があります。

そのようなときは、「**オブジェクトタイプフィールド**」を使用します。

文字列ではない形式のデータのことを一般的に「**バイナリ**」データと呼びます。オブジェクトタイプフィールドは、このバイナリデータを格納するためのものです。データに視覚、聴覚的な情報を与えたり、関連情報を持つ他のアプリケーションで作成されたファイルが保存できることで、カスタム App に広がりを与えます。

7.4.1 格納できるもの

オブジェクトタイプフィールドに格納可能なファイル形式は大きく分けて次のようになります。

・画像（ピクチャ）
・映像（ビデオ）
・音（オーディオ）
・PDF
・その他のファイル形式

1 フィールドに、1 つのファイル（4GB まで）を保存できます。さらに、レイアウト上で表示、再生など「インタラクティブ」に取り扱うことが可能なファイル形式があります。主なものは次表を参照してください。さらに詳細な情報はヘルプを参照してください。

種類	タイプ
ピクチャ	GIF (.gif) HEIF/HEIC (.heic) (macOS, iOS, iPadOS) JPEG/JFIF (.jpg) PDF (.pdf) (macOS) PNG (.png) TIFF (.tif) Windows ビットマップ (.bmp) ……など。
オーディオ／ビデオ	AIFF オーディオファイル (aif、.aiff) AVI ムービー (.avi) FLAC オーディオファイル (.flac) (iOS, iPadOS) MP3 オーディオファイル (mp3) MPEG-4 オーディオファイル (.m4a) MPEG-4 ムービー (.mp4) MPEG ムービー (.mpg、.mpeg) MPEG-4 ビデオファイル (.m4v) Sun オーディオファイル (.au) ……など。
PDF	.pdf

オブジェクトタイプフィールドのフィールドオプションは次図です。

「見積作成管理 .fmp12」の［データベースの管理］ダイアログで「商品情報」テーブルの「商品画像」フィールドの［フィールドオプション］ダイアログの「データの格納」タブを表示してください。



図 7-28 オブジェクトタイプのオプション（なにも設定されていない状態）

これまでの「索引」部分が「オブジェクト」に変わっています。確認したら確認後、［キャンセル］をクリックして［データベースの管理］ダイアログを閉じてください。

「商品管理」レイアウトへ切り替えてください（表形式になっている場合は、フォーム形式に切り替えてください）。「商品情報」テーブルのいくつかのレコードに画像やファイルが保存されています。



図 7-29「商品情報」レイアウト

レコードを前後に移動すると、Microsoft Excel ファイルや PDF、写真があります。これらのファイルは、入力時に写真の縦横方向についての表示は指定はしていません。FileMaker Pro が「EXIF 情報」に基づいて表示をしています。

EXIF 情報
Exchangeble image file format です。簡単に説明すると、デジタルカメラで写真を撮ったときに画像と一緒に保存されている「メタデータ」の情報です。

メタデータ
簡単に説明すると、データについてのデータです。例えば、JPEG の写真の内容自体のことではなく、作成日時、写真を撮ったデジタルカメラの情報など、付加的な情報です。

入力方法
オブジェクトタイプフィールドへのバイナリデータの入力方法は、いくつかあります。

方法 1：メニュー
［挿入］メニューから入力します。

方法 2：ショートカットメニュー
右クリックで表示されるメニューから入力します。

方法 3：その他
ドラッグ＆ドロップで入力したり、スクリプトで入力したりすることが可能です。

やってみよう
「添付ファイル」フォルダにあるファイルをいろいろな方法で保存してみましょう（商品番号 20031 以降は画像が保存されていません）。

スクリプトのみで入力させたい場合
入力方法を制限したい場合があります。スクリプト（が割り当てられたボタンなど）だけでユーザが入力するようにコントロールするには、次のようにします。
(1)　レイアウト上でオブジェクトタイプフィールドへの入力をインスペクタで禁止する。
(2)　入力用スクリプトを作成する。
(3)　オブジェクトタイプフィールドへの入力用スクリプトを割り当てたボタン配置する。
そのオブジェクトタイプフィールドが置かれるレイアウトすべてで設定するようにします。さらに確実にするには「アクセス権セット」で制限し、専用のスクリプトを［完全アクセス権で実行］で作成します。

7.4.2　オブジェクトの保存種類

オブジェクトタイプフィールドのデータ保存は 2 種類から選択可能です。

・バイナリデータをフィールドに保存あるいは管理する (埋め込みまたは外部格納)
・バイナリデータの場所（参照パス）だけをフィールドに保存する

バイナリデータをフィールドに保存あるいは管理する (埋め込みまたは外部格納)
こちらを選択することをお勧めします。

カスタム App で「データの一元化」が可能です。例えば、複数のユーザがそれぞれ商品の写真を手元に保存しておく……などが不要です。常に「最新の写真」、あるいは「担当者が使って欲しいと考えている写真」の利用を促すことができます。本書では、このタイプについて解説します。

バイナリデータの場所（参照パス）だけを保存する
フィールドにパス「だけ」を保存し、その場所にあるバイナリデータを参照してレイアウト上に表示可能です。本書では解説をおこないません。詳細な情報はヘルプを参照してください。

7.4.3 バイナリデータの保存：埋め込み

バイナリデータをファイルの中に保存する方法です。「**埋め込み**」と呼びます。

どんどん保存されるに従ってファイルサイズが大きくなるので注意してください。ファイルサイズが大きくなりすぎると、問題が起きる場合があります。例えば、バックアップにとても時間がかかるようになります。

設定方法
オブジェクトタイプフィールドの初期設定は、埋め込みです。フィールドオプションの「データの格納」タブで、[オブジェクトデータを外部に保存] が　**オフ**　になっている状態です。



図 7-30 オブジェクトの格納設定（埋め込み）

埋め込みでは、表示についての次の特徴があります。

・原則としてバイナリデータは誰でも表示することができる
・FileMaker Go でネットワークがオフラインの状態でも表示ができる（モバイルデバイスにカスタム App ファイルが保存されている場合）

例えば、会社のロゴ画像などのカスタム App 全体で 1 つだけ保存されていれば良いものは、この設定で充分です。

7.4.4　バイナリデータの保存：外部格納

バイナリデータをファイルの外、FileMaker アプリケーションが管理する場所に保存することです。「**外部格納**」と呼びます。

オブジェクトタイプフィールドにバイナリデータを入力すると、オリジナルのファイルはそのままで、コピーされたファイルがレイアウトを通じて FileMaker アプリケーションが管理するフォルダ（以下「管理フォルダ」）に保存されます。そして管理フォルダに保存されているファイルのデータがレイアウト上に表示されます。次の特徴があります。

・原則としてバイナリデータは誰でも表示することができる
・FileMaker Go で、共有ファイルへ接続している場合、表示ができる
・グローバル格納フィールドでは設定できない

外部格納には、さらに 2 種類の保存方法が用意されています。フィールドオプションの「データの格納」タブで設定します。

設定方法
［オブジェクトデータを外部に保存］を　**オン**　にします。
・［保存先：］で基本ディレクトリを選択
・「セキュア格納」か「オープン格納」かを選択

セキュア格納
FileMaker アプリケーションの管理フォルダに保存する際に暗号化します。外部格納を選択すると、セキュア格納が初期値で選択されています。

オープン格納
FileMaker アプリケーションの管理フォルダに保存する際に暗号化しません。

7.4.5　バイナリデータの保存：外部格納のパス

セキュア、オープンどちらの外部格納を選択しても、ある「**パス**」が設定され、その場所に管理フォルダが自動作成されます。その場所を「**基本ディレクトリ**」と呼びます。

［オブジェクトデータを外部に保存］を　**オン**　にすると、［保存先：］に基本ディレクトリが表示されます。通常は、次の内容です。

基本ディレクトリ　→　**［データベースの場所］/（作成時のファイル名）/**

フォルダ名が作成時のファイル名になる
「(作成時のファイル名)」がフォルダ名となります。ファイル名を変更しても追従して変更されません。

パス
ファイルの場所を表す文字列のことです。ハードディスクのドライブやフォルダ名、ファイル名を「/」や「¥」で区切って記述します。例えば、macOS の書類フォルダへのパスは「/Macintosh HD/Users/xxxxx/Documents/」です（xxxxx は、現在のユーザ名）。

セキュア格納の基本ディレクトリ
［セキュア格納］が　**オン**　の場合は、通常は次の内容です。基本ディレクトリだけを設定します。

基本ディレクトリ　→　**［データベースの場所］/（作成時のファイル名）/**

オープン格納の基本ディレクトリとフォルダ名
［オープン格納］が　**オン**　の場合は、基本ディレクトリとさらにフォルダ名についても設定が可能です。通常は次の内容です。

基本ディレクトリ　→　**［データベースの場所］/（作成時のファイル名）/**
基本ディレクトリの中のフォルダ名　→　**（テーブル名）/（フィールド名）**

オープン格納のバイナリデータの保存先
オープン格納では、次のフォルダ構成でバイナリデータのファイルが管理されることになります。
［データベースの場所］/（作成時のファイル名）/（テーブル名）/（フィールド名）

オープン格納の保存ファイル名に接頭語をつける
保存されるファイル名に接頭語を付加することが可能です。例えば、次のように記述すると「abc-」という接頭語が付与されます。
［データベースの場所］/ 見積作成管理 / 商品情報 / 商品画像 /abc-:　→　最後が「：」コロンで終わる

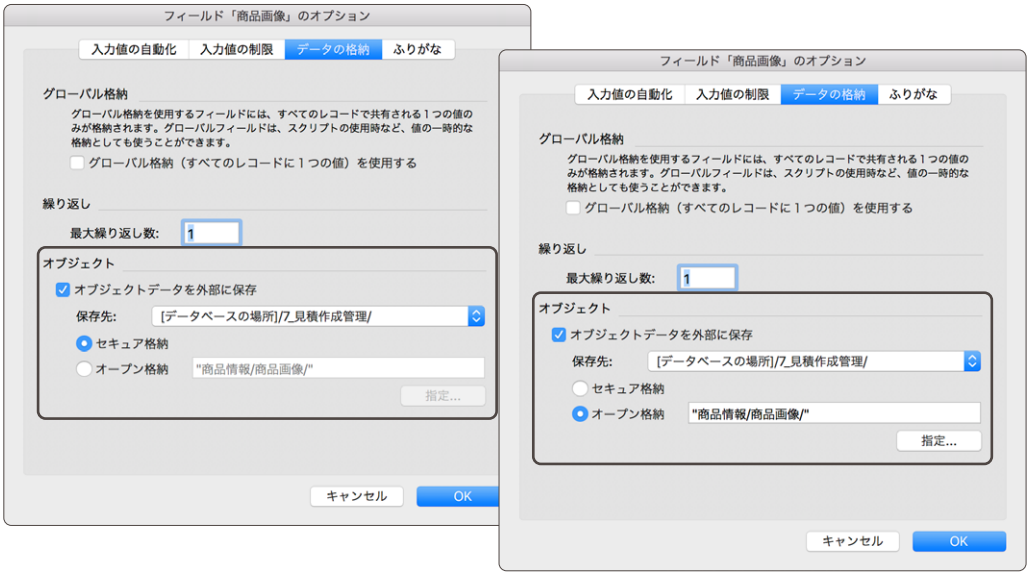


図 7-31 オブジェクトの格納設定（セキュア格納とオープン格納）

モバイルデバイスにカスタム App ファイルを保存し、オフライン状態で FileMaker Go から使用する場合「埋め込み」が必要です。

※ 自動作成される以外の基本ディレクトリや基本ディレクトリの中のフォルダ名を登録することが可能です
（次ページ以降を参照）。

7.4.6 ［オブジェクトの管理］ダイアログ

フィールドオプションの「データ格納」タブの外部格納を設定する場合に表示される［保存先：］のパスは、ファイルごとに設定が可能です。**［オブジェクトの管理］ダイアログ**でおこないます。

以下の方法で表示してください。「格納」タブの内容を確認します。

表示方法

［ファイル］–［管理］–［オブジェクト ...］メニュー

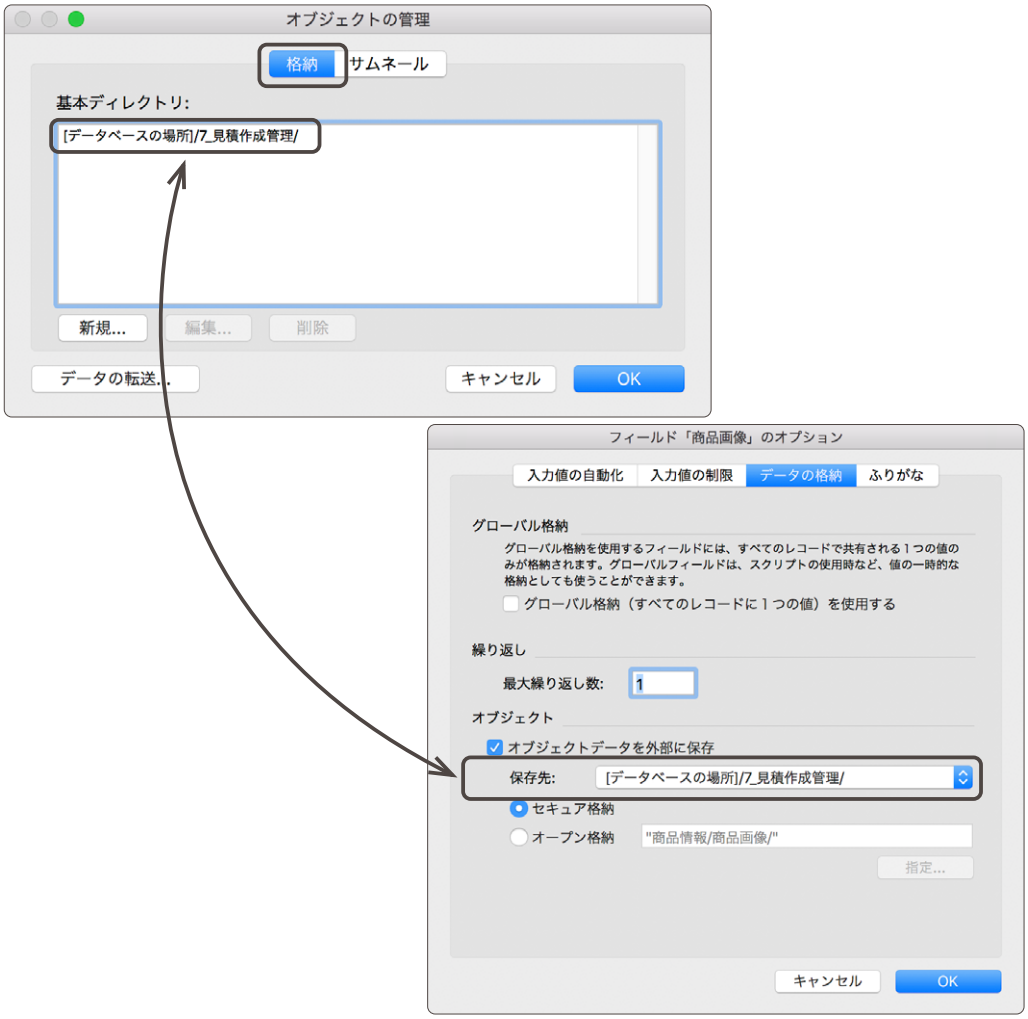


図 7-32［オブジェクトの管理］ダイアログ 「格納」タブ

「格納」タブ
外部格納を設定する場合に表示される［保存先：］で選択する「基本ディレクトリ」の登録と管理をおこなうことができます。

［データベースの場所］/（作成時のファイル名）/ のパスが登録されています。

このパスを変更することが可能です。記述方法は、次の 3 種類です。

・データベースディレクトリに対する相対パス
・完全パス
・ネットワークパス

いま初期登録されている ［データベースの場所］/7_ 見積作成管理 / が基本ディレクトリとして表示されています。他のパスを利用したい場合は、

・初期登録されているパスを変更する
・初期登録されているパスはそのままで、新しくパスを登録しフィールドオプションで設定を切り替える

などの方法があります。ファイルが保存されているパソコンではなく、他のボリュームや他のサーバー機へのパスも登録することが可能です。

「サムネール」タブ
画像のサムネールを作成してメモリに保存し、表示に使用する設定が可能です。

容量が大きい画像はレイアウトに表示するために時間がかかります。そこでサムネールを利用することで表示にかかる時間を少なくすることが可能です。ただしサムネールであっても、ハードディスクの容量を使います。その考慮をおこなって永続的にサムネールを格納するか、一時的にするか選択します。



図 7-33［オブジェクトの管理］ダイアログ 「サムネール」タブ

通常は、**イメージのサムネールを生成して保存が オン 、「一時的な格納」が オン** です。FileMaker Pro が、必要に応じレイアウト上で表示されるオブジェクトタイプフィールドのサイズに適したサイズのサムネールを自動的に作成します。

埋め込みと外部格納の違いを確認します。いま開いている「見積作成管理 .fmp12」を閉じてください。

「演習 7_5 セキュア格納」フォルダの中「7_ 見積作成管理セキュア格納 .fmp12」ファイルを使用してください。その前提で次の演習を記述しています。これからおこなういくつかの演習の結果、それぞれが違う基本ディレクトリとなるように［オブジェクトの管理］ダイアログで設定をおこないました。

演習 7-5　オブジェクトタイプフィールドの設定：外部格納（セキュア）

- **使用ファイル：7_ 見積作成管理セキュア格納 .fmp12**

(1)　ファイルを開きます。

(2)　「商品情報」レイアウトで「商品画像」フィールドにデータが入力されていることを確認してください。

(3)　［データベースの管理］ダイアログを開いて「商品情報」テーブルの「商品画像」フィールドオプションダイアログを開きます。現在は「埋め込み」になっていることを確認します。

(4)　「外部格納」に切り替えます。
- ［オブジェクトデータを外部に保存］を**オン**
- ［保存先：］は「［データベースの場所］/7_ 見積作成管理セキュア /」
- 「セキュア格納」を　**オン**

(5)　［OK］をクリックして［データベースの管理］ダイアログを閉じると［オブジェクトデータの転送］ダイアログが表示されます。



図 7-34［オブジェクトデータの転送］ダイアログ

(6)　［転送］をクリックすると少し時間をおいて［転送の概要］ダイアログが表示されます。

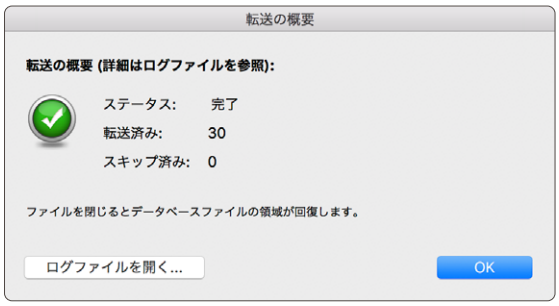


図 7-35［転送の概要］ダイアログ

(7)　［ログファイルを開く ...］をクリックすると転送のログが表示されます。
画像がファイルから管理フォルダに移動した記録を確認できます。

(8)　ログのダイアログを閉じます。

(9)　［OK］をクリックすると［転送の概要］ダイアログが閉じます。

(10)「商品情報」レイアウトには特に変化がないことを確認してください。

(11)「7_5 セキュア格納」フォルダに「見積作成管理セキュア」フォルダが作成されていることを確認してください。



図 7-36 作成されたフォルダ

演習ここまで

ログ
動作や、操作の記録を取ることです。その記録が書き込まれているファイルを「ログファイル」と呼びます。

演習で作成されたフォルダの中は次図のようになっています。

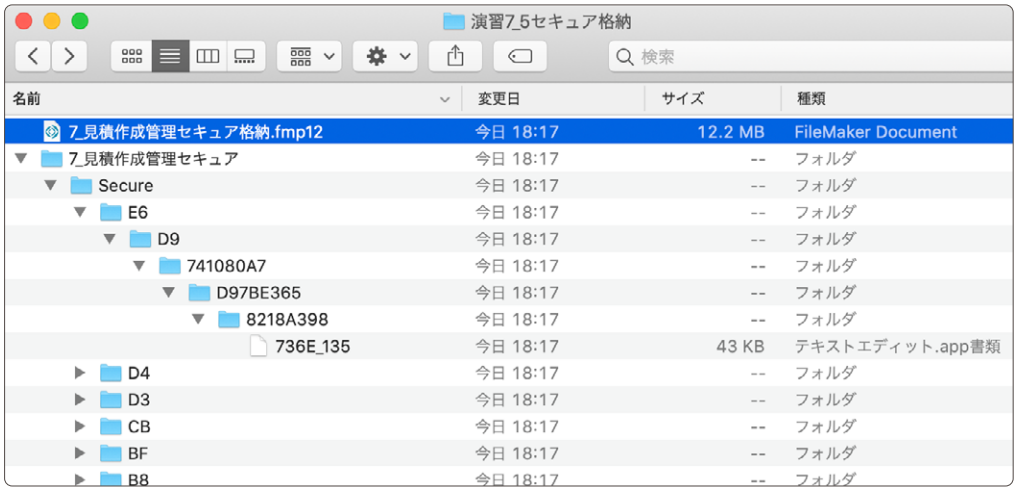


図 7-37 セキュア格納の場合のフォルダの中身

ファイルが暗号化されています。どのような種類のファイルが保存されているのかわかりません。

オブジェクトフィールドの保存設定を変更すると FileMaker Pro が適切にデータを転送してくれます。

基本ディレクトリにファイルが入っているとパスを編集できない
演習 7-5 で、基本ディレクトリを使ってセキュア格納をおこないました。[オブジェクトの管理] ダイアログのパスの管理では、次のルールがあります。

・フォルダにファイルが保存されているパスは編集できない

編集しようとすると次図のようなメッセージが表示されてできません。



図 7-38 パスが編集できない場合のメッセージ

やってみよう
［オブジェクトの管理］ダイアログでパスを編集してみましょう。パスのフォルダにファイルが空の場合は編集可能ですが、ファイルが保存されていると編集できません。

「7_ 見積作成管理セキュア格納 .fmp12」を閉じてください。

続いて、オープン格納を確認します。

「演習 7_6 オープン格納」フォルダの中「7_ 見積作成管理オープン格納 .fmp12」ファイルを使用してください。その前提で次の演習を記述しています。手順は演習 7-5 とほぼ同じです。

演習 7-6　オブジェクトタイプフィールドの設定：外部格納（オープン 1）

- **使用ファイル：7_ 見積作成管理オープン格納 .fmp12**

(1)　ファイルを開きます。

(2)　「商品情報」レイアウトで「商品画像」フィールドにデータが入力されていることを確認してください。

(3)　［データベースの管理］ダイアログを開いて「商品情報」テーブルの「商品画像」フィールドオプションダイアログを開きます。現在は「埋め込み」になっていることを確認します。

(4)　「外部格納」に切り替えます。
- ［オブジェクトデータを外部に保存］を　**オン**
- ［保存先：］は「［データベースの場所］/7_ 見積作成管理オープン /」
- 「オープン格納」を　**オン**
- パスは「" 商品情報 / 商品画像 /"」のまま

(5)　［OK］をクリックして［データベースの管理］ダイアログを閉じると［オブジェクトデータの転送］ダイアログが表示されます。

(6)　［転送］をクリックすると少し時間をおいて［転送の概要］ダイアログが表示されます。

(7)　［ログファイルを開く ...］をクリックすると転送のログが表示されます。
画像がファイルから管理フォルダに移動した記録を確認できます。演習 7-5 のセキュア格納のログに続いて、この演習のログが記録されています。

(8)　［OK］をクリックすると［転送の概要］ダイアログが閉じます。

(9)　「商品情報」レイアウトには特に変化がないことを確認してください。

(10)　「7_6 オープン格納」フォルダに「見積作成管理オープン」フォルダが作成されていることを確認してください。



図 7-39 作成されたフォルダ

演習ここまで

演習で作成されたフォルダの中は次図のようになっています。

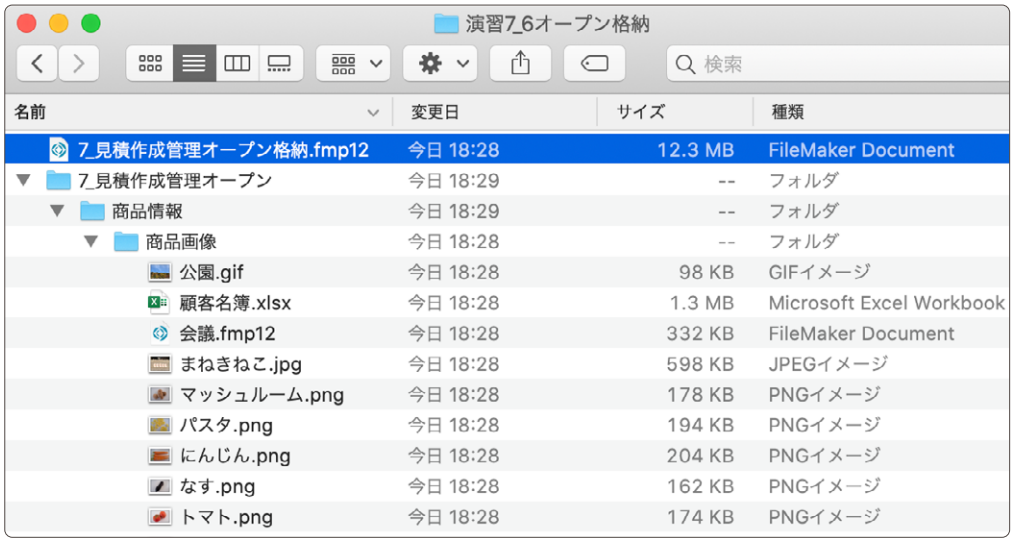


図 7-40 オープン格納の場合のフォルダの中身

ファイルは暗号化されていません。

次は、[オブジェクトの管理] ダイアログでパスを新しく登録して保存先のフォルダを切り替えます。

演習 7-7　オブジェクトタイプフィールドの設定：外部格納（オープン 2）

- **使用ファイル：7_ 見積作成管理オープン格納 .fmp12**

(1)　[ファイル]–[管理]–[オブジェクト ...] メニューを選択し、[オブジェクトの管理] ダイアログを表示します。

(2)　「格納」タブをクリックします。

(3)　[新規 ...] をクリックすると [新規基本ディレクトリ] ダイアログが表示されます。[基本ディレクトリ：] を登録します。次のように入力してください。

・ディレクトリ変更 /



図 7-41 パスを設定

(4)　[OK] を何度かクリックして［オブジェクトの管理］ダイアログを閉じます。

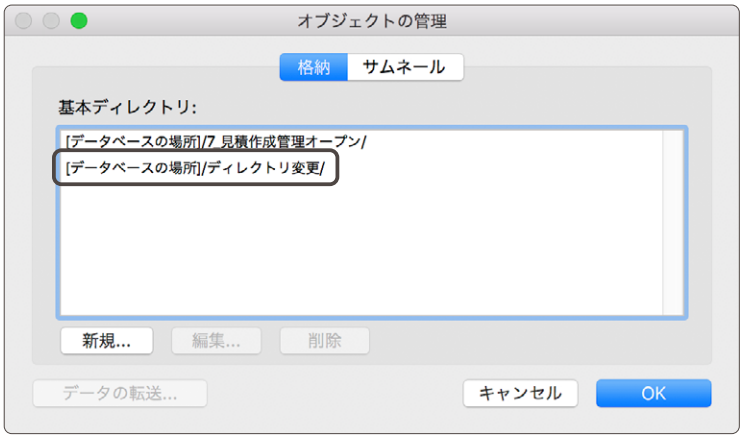


図 7-42［オブジェクトの管理］ダイアログ

(5)　[データベースの管理] ダイアログで「商品情報」テーブルの「商品画像」フィールドオプションダイアログを開いて、設定を変更します。

・[保存先：] は　　「[データベースの場所] / ディレクトリ変更 /」に**変更**
・「オープン格納」を　　**オン**
・パスは「" 商品情報 / 商品画像 /"」のまま



図 7-43 オブジェクトの格納設定（外部格納／オープン格納）

(6)　[OK] をクリックして［データベースの管理］ダイアログを閉じると［オブジェクトデータの転送］ダイアログが表示されます。

(7)　[転送] をクリックすると［転送の概要］ダイアログが表示されます。

(8)　[ログファイルを開く ...] をクリックすると転送のログが表示されます。
引き続き、この演習のログが記録されています。

(9)　［OK］をクリックすると［転送の概要］ダイアログが閉じます。

(10)　「商品情報」レイアウトには特に変化がないことを確認してください。

(11)　「7_6 オープン格納」フォルダの「見積作成管理オープン」フォルダがなくなって「ディレクトリ変更」フォルダが作成されていることを確認してください。



図 7-44　作成された「見積作成管理オープン」フォルダ

演習ここまで

同じファイル名で内容が異なるファイル

オープン格納が設定されている場合、同じファイル名で内容が異なるファイルが別々のレコードから入力されるとどうなるでしょうか。

FileMaker Pro が管理フォルダの中で重複するファイル名にシリアル番号を自動的に付与してユニークを保ちます。

例えば、「タスク管理票 .docx」というファイルが既に保存されていて、同じ名前の内容が異なるファイルが保存されると「タスク管理票 _1.docx」と名前を変更して保存します。

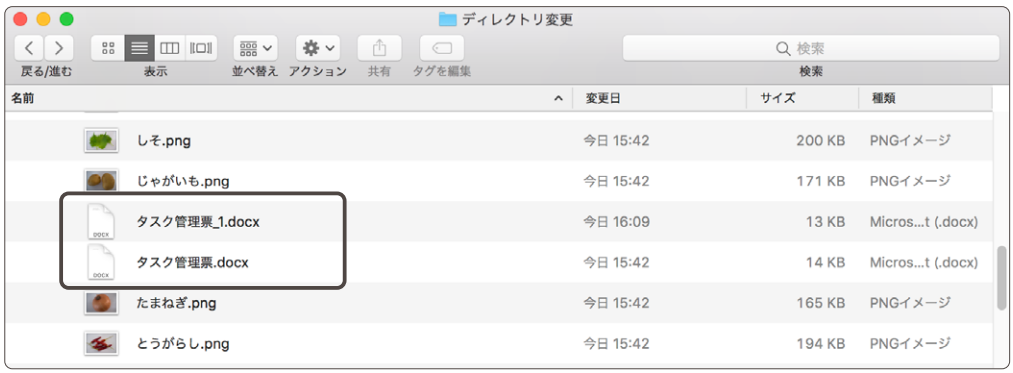


図 7-45　オープン格納の場合のフォルダの中身（重複したファイル名の場合）

やってみよう

同じファイル名で内容が異なるファイルをバイナリデータが入力されていないレコードから入力してみましょう。

オープン格納のパスは、一部を計算式で設定することが可能です。規則的に保存が可能になります。

オープン格納のパスに計算式を設定し、確認します。

演習 7-8　オブジェクトタイプフィールドの設定：外部格納（オープン 3）

- **使用ファイル：7_ 見積作成管理オープン格納 .fmp12**

(1)　[データベースの管理] ダイアログで「商品情報」テーブルの「商品画像」フィールドオプションダイアログを開いて、設定を変更します。

・[保存先：] は　　「[データベースの場所] / ディレクトリ変更 /」のまま
・「オープン格納」を　　**オン**
・パスを　以下の計算式で設定します。

" 商品情報 / 商品画像 / "　&　商品番号

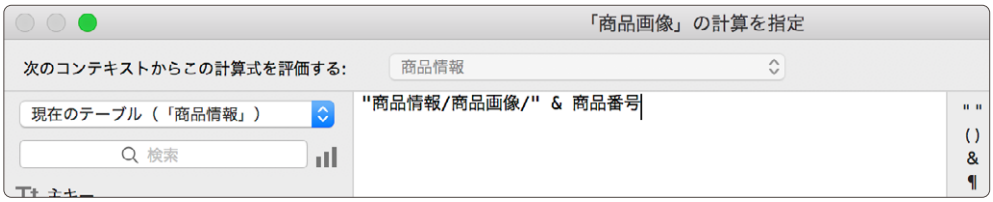


図 7-46 計算式を設定する

(2)　[OK] をクリックを何度かクリックして［データベースの管理］ダイアログを閉じると、［オブジェクトデータの転送］ダイアログが表示されます。

(3)　［転送］をクリックすると［転送の概要］ダイアログが表示されます。

(4)　[OK] をクリックすると［転送の概要］ダイアログが閉じます。

(5)　「商品情報」レイアウトには特に変化がないことを確認してください。

(6)　「7_6 オープン格納」フォルダの「ディレクトリ変更」フォルダ内容が変わっていることを確認してください。

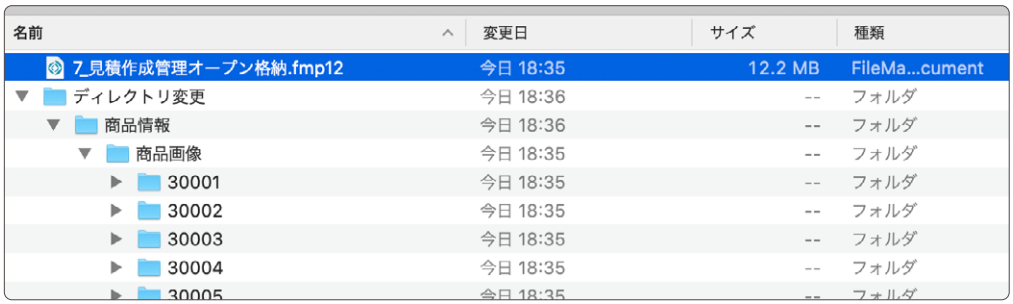


図 7-47 オープン格納の場合のフォルダの中身（計算式のパスを設定 ）

演習ここまで

計算式で「商品番号」フィールドを指定したので、商品番号がフォルダ名に付与されています。このフォルダの中にオブジェクトタイプフィールドに入力したデータが保存されています。このようなフォルダ構成ならば、同じ名前の内容が異なるファイルでも「_1」などが付与されることなく管理可能です。

「7_ 見積作成管理オープン格納 .fmp12」を閉じてください。

7.4.7　オブジェクトタイプフィールドのデータ書式

レイアウト上に配置されているオブジェクトタイプフィールドは、レイアウトモードの右パネルのインスペクタで、

・静的なコンテンツ（画像など）
・インタラクティブコンテンツ（映像、音、PDF）

の、どちらに最適化した表示をするかを指定することができます。フィールド定義とは関係ありません。格納方法が埋め込みでも外部格納でも同じようにレイアウト上での表示が変わります。

この設定の違いで、右クリックで表示されるショートカットメニューの内容も変わります。

設定による表示の違いを確認するためにデータ書式を設定します。

（「演習 7_9 から 11」フォルダの中の「7_ 見積作成管理 .fmp12」ファイルを使用してください。その前提で次の演習を記述しています）

演習 7-9　オブジェクトタイプフィールドのデータ書式を変更する　1

- **使用ファイル：7_ 見積作成管理 .fmp12**

(1)　ファイルを開きます。

(2)　「オブジェクト表示」レイアウトで 2 つの「商品画像」フィールドが並んで配置されていることを確認します。



図 7-48「オブジェクト表示」レイアウト

(3)　レイアウトモードに切り替えます。

(4)　データの書式を設定します。

a.　左側の「商品画像」フィールドをクリックして右パネルのインスペクタの「データ」タブの［データの書式設定］を確認します。

b.　［次の用途に最適化］は「イメージ」です（そのまま）。

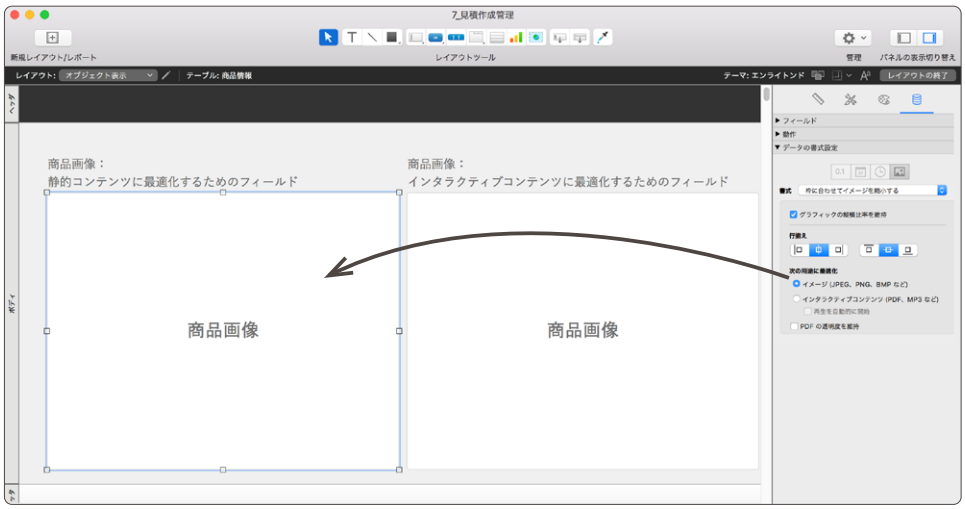


図 7-49 データの書式を設定

c.　右側の「商品画像」フィールドをクリックして右パネルのインスペクタの「データ」タブの［データの書式設定］を確認します。

d.　［次の用途に最適化］は「インタラクティブコンテンツ」に設定します。

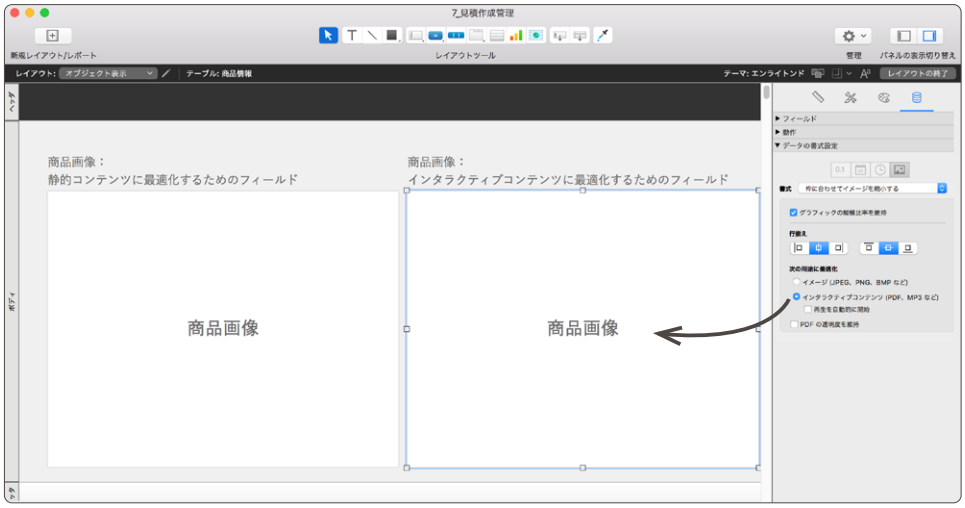


図 7-50 データの書式を設定

(5)　レイアウトを保存してブラウズモードに戻ります。

演習ここまで

データの書式設定による、表示の違いを確認します。

演習 7-10　オブジェクトタイプフィールドのデータ書式を変更する　2

- **使用ファイル：7_ 見積作成管理 .fmp12**

(1)　「オブジェクト表示」レイアウトの 7 レコード目に移動してください。

(2)　PDF ファイルが保存してあります。データの書式を設定後、表示が違うことを確認します。
右側はスクロールが可能になって 2 ページ目が表示されます。



図 7-51　2 つの「商品画像」フィールドの表示の違い

(3)　40 レコード目に移動してください。動画が保存されています。右側は動画を再生できます。



図 7-52　2 つの「商品画像」フィールドの表示の違い

(4)　レコードを前後して表示される内容の違いを確認します。
a.　1 から 4 番目のレコードは画像です。右と左の表示に違いはありません。
b.　6、9 番目は、Microsoft Excel と Word のファイルです。やはり違いがありません。

演習ここまで

書式設定による表示の違いを演習しました。格納するバイナリデータに合わせられるようになっています。表示されるメニューも変わります。**ショートカットメニューの違いを確認**します。

演習 7-11　オブジェクトタイプフィールドのデータ書式を変更する　3

- **使用ファイル：7_ 見積作成管理 .fmp12**

ショートカットメニューの違いを確認します。左右の「商品画像」フィールドで右クリックします。



図 7-53 2 つの「商品画像」フィールドのショートカットメニューの違い

演習ここまで

[次の用途に最適化] が「イメージ」に設定されていると、

・ピクチャを挿入 ...
・ファイルを挿入 ...

[次の用途に最適化] が「インタラクティブコンテンツ」に設定されていると

・ピクチャを挿入 ...
・オーディオ / ビデオを挿入 ...
・PDF を挿入 ...
・ファイルを挿入 ...

それぞれ表示が違います。[挿入]メニューも同じです。格納するバイナリデータに合ったメニューを選択して入力すると FileMaker Pro が適切に表示します。

やってみよう

いろいろな形式のインタラクティブコンテンツをショートカットメニューで入力して表示の違いを確認してみましょう。

インタラクティブコンテンツの受信

共有されているカスタム App について、クライアントがインタラクティブコンテンツを再生（ストリーミング）するためのデータの受信はポート 80 を使用します。

データ入力時の送信

共有されているカスタム App ファイル（または外部格納のフォルダ）へのオブジェクトフィールドへのデータ入力時のデータの送信はポート 443 を使用します。

データ転送時

埋め込みから外部格納に変更するときなど、データ転送にはポート 5003 を使用します。

ポート番号については、ヘルプやクラリス・ジャパン株式会社の Web サイトのナレッジベース「FileMaker Server によって使用されるポート番号」（https://support.claris.com/s/answerview?anum=000025632&language=ja）で、最新情報を参照してください。

7.4.8 オブジェクトの情報

オブジェクトタイプフィールドに入力（埋め込みでも外部格納でも）されているバイナリデータは、計算式や集計タイプフィールドから参照できます。しかし、データそのものを対象にして検索やソートをすることはできません。

別のフィールドを用意してメタデータを格納して利用する方法があります。メタ情報を取得するには次の関数を使用します。

```
GetContainerAttribute( ソースフィールド ; 属性名 ) 関数
```

下表は属性名に「General」を指定した時、オブジェクトタイプフィールドに入力されているファイルの種類に関わらず共通に使用できるものとその結果が表すものです。

属性名	返される値	戻り値のタイプ
filename	オブジェクトフィールドに挿入されたファイルの名前	テキスト
storageType	オブジェクトフィールドのデータの保存方法 Embedded: 埋め込み External (Secure): 外部保存 (セキュア格納) External (Open): 外部保存 (オープン格納) File Reference: ファイル参照 Text: テキスト	テキスト
MD5	ファイルに暗号化ハッシュ関数 MD5 を適用した結果	テキスト
fileSize	オブジェクトフィールドに挿入されたファイルのサイズ (バイト単位)	数字
internalSize	オブジェクトフィールドが占めるファイル内のサイズ (バイト単位)	数字
externalSize	外部に保存されるファイルサイズ (バイト単位) ファイル参照を使用するファイルのサイズまたはオブジェクトフィールド内のすべてのファイルのサイズ合計 (オープンまたはセキュア格納)	数字
externalFiles	オブジェクトフィールドに関連付けられた外部ファイルの一覧 (オープンまたはセキュア格納、またはファイル参照を使用するファイル)	テキスト

グループ属性は、他にも

・Image（画像）
・Photo（写真）
・Audio（音）
・Bar Codes（バーコード）
・Signatures（署名）
・All（各カテゴリの属性名がすべて）

があります。詳しくはヘルプを参照してください。

なお、メタデータは、すべて保持しているとは限りません。例えば画像編集のアプリケーションであらかじめ EXIF データが編集、削除されている場合もあります。

演習ファイルに入力されているデータのメタ情報を確認します。

演習 7-12　オブジェクトタイプフィールドの情報をみる

- **使用ファイル：7_ 見積作成管理 .fmp12**

(1)　「オブジェクト表示」レイアウトで 28 件目の、写真が入力されているレコードへ移動します。

(2)　左右どちらかの「商品画像」フィールドをクリックします。

(3)　[ツール]–[データビューア] メニューを選択して「データビューア」を表示します。

(4)　「監視」タブをクリックします。



図 7-54　データビューア

(5)　右下の「+」アイコンをクリックすると［式の編集］ダイアログが表示されます。
次の計算式を入力します（「自動的に評価する」は　**オン**　にしてください）。

```
GetContainerAttribute ( 商品情報::商品画像 ; "General" )
```

(6)　ダイアログの下に結果が表示されます。
（次図も参照してください）

Filename: マッシュルーム .png
Storage Type: Embedded
MD5: D513E9B8ECC0628ABBC6095058215155
File Size: 178249
Internal Size: 178301
External Size: 0
External Files: 0



図 7-55［式の編集］ダイアログ

(7)　［監視］をクリックして［式の編集］ダイアログを閉じます。

(8)　違う属性を試します。右下の「+」アイコンをクリックします。

(9)　［式の編集］ダイアログが表示されます。次の計算式を入力します。

```
GetContainerAttribute ( 商品情報::商品画像 ; "all" )
```

(10)　結果が表示されます。

[General]
Filename: マッシュルーム .png
Storage Type: Embedded
MD5: D513E9B8ECC0628ABBC6095058215155
File Size: 178249
Internal Size: 178301
External Size: 0
External Files: 0

[Image]
Width: 429
Height: 285
DPI Width: 72
DPI Height: 72
Transparency: 1 (True)

(11)　［監視］をクリックします。「データビューア」に 2 つの計算式が登録されました。
(ダイアログのサイズが小さい場合は広げて確認してください)



図 7-56 式が登録された

（12） 引き続き 29 レコード目へ移動します。

（13） GetContainerAttribute (商品情報 :: 商品画像 ; "all") の結果を確認します。
データビューアのままでは見づらいので、登録した式の 2 行目をダブルクリックして、もう一度［式の編集］ダイアログを開きます。

[General]
Filename: まねきねこ .jpg
Storage Type: Embedded
MD5: CC47BFA45C79A92A1712A20FF2B9ED15
File Size: 598178
Internal Size: 598228
External Size: 0
External Files: 0

[Image]
Width: 1008
Height: 756
DPI Width: 72
DPI Height: 72
Transparency: 0 (False)

[Photo]
Orientation: 1 (Normal)
Created:
Modified:
Latitude:
Longitude:
Make:
Model:

（14） ［監視］をクリックします。

（15） 別の属性を試します。右下の「+」アイコンをクリックして、計算式を入力します。

```
GetContainerAttribute ( 商品情報::商品画像 ; "filename" )
```

（16） 結果を確認します（2 件目の場合）。

まねきねこ .jpg

（17） ［監視］をクリックして［式の編集］ダイアログを閉じます。

演習ここまで

「商品情報」テーブルには、28 件目のレコードには PNG ファイル、29 件目は JPEG ファイルが入力されています。計算式の結果、表示される内容が異なっていました。GetContainerAttribute 関数では［Photo］属性は、JPEG ファイルと TIFF ファイルのみ結果が返ります。

※ この関数はグループ属性の名前だけではなく、個々の属性名も指定可能です。詳しくはヘルプを確認してください。

このような、計算式で得られるオブジェクトタイプフィールドのメタ情報を使って機能を作成することが可能です。

7.4.9 インポートとエクスポート

オブジェクトタイプフィールドもデータをインポートとエクスポートができます。

インポート

FileMaker のファイル同士であれば、オブジェクトタイプフィールドを含むテーブルからのインポートが可能です。

エクスポート

FileMaker のファイル形式を指定した場合のみ、オブジェクトタイプフィールドをエクスポート対象に指定することが可能です。

埋め込みや外部格納されているファイルそのものを開くには、内容をエクスポートしてから開く必要があります。

フィールドからバイナリデータをエクスポートする

［フィールド内容のエクスポート］を使います。

・オブジェクトタイプフィールドのショートカットメニュー
・［フィールド内容のエクスポート］スクリプトステップ

から実行可能です。

ショートカットメニューからのエクスポートを実行します。

演習 7-13　オブジェクトタイプフィールドの書き出し

- **使用ファイル：7_ 見積作成管理 .fmp12**

(1)　「オブジェクト表示」レイアウトの左右どちらかの「商品画像」フィールドをクリックします。

(2)　右クリックします。

(3)　「フィールド内容のエクスポート ...」を選択します。

(4)　デスクトップなど、わかりやすい場所を選択して保存します。

(5)　指定した場所にエクスポートされたファイルが保存されているか確かめます。

演習ここまで

エクスポートしても入力されているデータが消えたわけではありません。コピーがエクスポートされます。

7.4.10　その他

文字列以外のデータを取り扱うには、いくつか考えることがあります。

・格納方法（外部格納なら、保存先、セキュアかオープンか）
・レイアウト上のデータ書式設定（表示方法）
・入力方法

カスタム App にオブジェクトタイプフィールドが必要になったら、これらをどのように設定するかについて検討をおこなってください

表示速度
レイアウト上にオブジェクトタイプフィールドをたくさん配置すると、表示速度が遅くなります。前述したサムネールを設定したり、リスト表示でどの程度オブジェクトタイプフィールドを表示する必要があるか、などを考えます。

演習ファイルを閉じてください。

第 8 章 画面を設計しよう

ユーザに表示される画面遷移と画面を設計します。

8.1 画面の設計

画面について設計作業を行います。

・要件定義と構造の設計
・ファイル、テーブル、フィールドの定義

これらの情報を使っておこないます。これから設計するのは、構造ではなくて見た目の設計です。

・画面遷移
・画面内容
・画面の機能

以下を決定します。

(1) 必要な画面
(2) 項目やレイアウトオブジェクトの配置
(3) 画面の遷移
(4) 画面ごとの機能
(5) 画面の入出力項目
(6) 画面に共通のこと

ユーザがカスタム App と接するのは原則として「画面（＝レイアウト）」です。ユーザは、画面を通してデータを入力、出力します。とても重要です。

8.2 ユーザインターフェース

一般的に画面のことを「**UI（User Interface）**」と呼びます。ユーザが持つ接点（Interface）だからです。その決定についてはいくつかの手段があります。例えば、

UX
「UX（User Experience）」という考え方があります。これはユーザの体験（Experience）のことです。ユーザが体験して使いやすいなどと考えることに従って UI を改善していくことです。

HCD
UX の他に「HCD（Human Centered Design）」という言葉もあります。ユーザと限定せずに更に広く「人」がどう感じるかを中心に改善することです。

これらは、カスタム App の開発工程でプロトタイプモデルを実践すると取り入れられるものです。ユーザにデザイン案（プロトタイプ）を見せて（場合によっては使ってもらって）感想や意見を集めて改善のための参考意見とします。

UI はユーザにとってとてもわかりやすい部分です。画面案を提示すると、それまで要望されていなかった入出力の要件や機能を要望されることがあります。実際にカスタム App を使用するシチュエーションが想像しやすいためにアイデアがでてくるのです。それは素晴らしいことです。

なお、新しいアイデアはこれまで検討したことがない新しい作業に関するものかもしれません。その場合、これまでのことと照らしあわせて、

・企画で決定した目的に必要かどうか
・各種の設計にどのぐらいの変更点が発生するか

などを慎重に考えます。必要があれば改めてヒアリングや要件定義を実施して、設計内容を訂正します。もちろん全体のスケジュールに影響があるかどうかも確認します。

この方法の注意点としては『ユーザが良いと考えるから』の、「ユーザ」は誰なのかということです。

1 人だけの意見は偏りがあって、他の人にとって良くないことかもしれません。かといって、なるべく多くの意見を聞くと平均的な内容が採用されるので突出して良くもなく悪くもなくということになるかもしれません。ただ、業務で使用されるのであれば、その方が望ましいとも考えられます。

UI については、専門書が多数ありますので興味があれば詳細を確認してください。

8.3 画面の役割

画面の色やデザイン、各オブジェクトの配置について、ほとんどのユーザが興味を持ちます。ですから、決定に時間がかかることがあります。オブジェクト位置の微調整はもっと後でも可能なので、全体を決めることを優先させます。

画面には大きく分けて次の 2 つの役割があります。

・見た目の表示と操作を提供する
・機能を実行して結果を提供する

画面設計では、主に次のことを考えます。

・その画面では、なにを入出力するか、そのためにどのような機能が必要か

画面はいつ、どのようなシチュエーションで表示されるのでしょうか。実は、これまでの作業で概ねわかってきているはずです。

要件定義で、業務の流れのどこでカスタム App が必要になるかを考えました。フローを思い出してください。そして、設計した構造も考慮します。

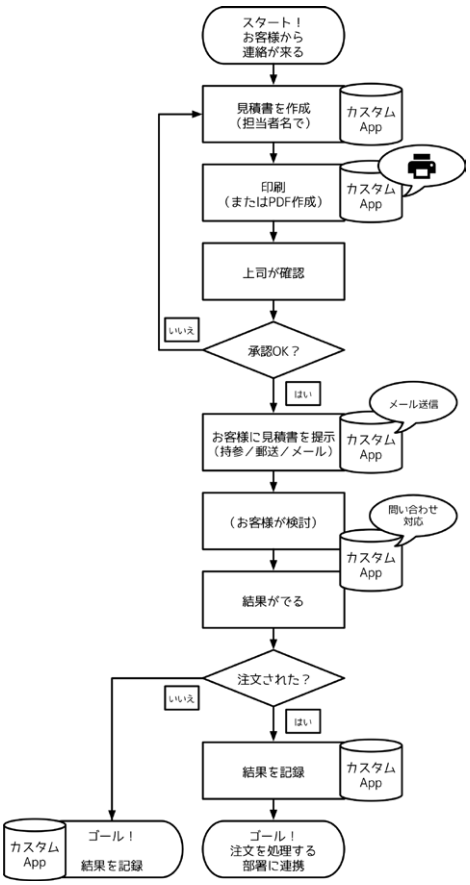


図 8-1 業務フロー図

クライアントの種類

画面の設計では、どのようなクライアントから使用されるのかもポイントです。

・パソコン
・iPad
・iPhone
・その他

画面の大きさ、処理能力が違います。ユーザの操作も違います（マウス、キーボード、指、タッチペンなど）。

ユーザに見せないレイアウト

ユーザに見せないレイアウトを作成することがあります。データを表示するためにレイアウト（TO）が必要なためです。次のようなものがあります。

・スクリプトの実行のために作成される
・開発者がデータの確認用に作成される

8.4　画面を考える

FileMaker Pro が得意とするところです。様々な工夫ができる腕の見せ所と考える方も多いのではないでしょうか。

本書では実際のレイアウト作業に入る前に画面について考えます。

8.4.1　画面一覧をつくる

全体からはじめます。最初に暫定的な画面の一覧を考えます。業務の流れのどこでカスタム App が登場し、どのような画面が必要か、です。

「見積作成管理システム」には、どのような画面が必要でしょうか。第 5 章で TOG を考えたときに、システムのメニューを考えました。これも画面作成に役立ちます。

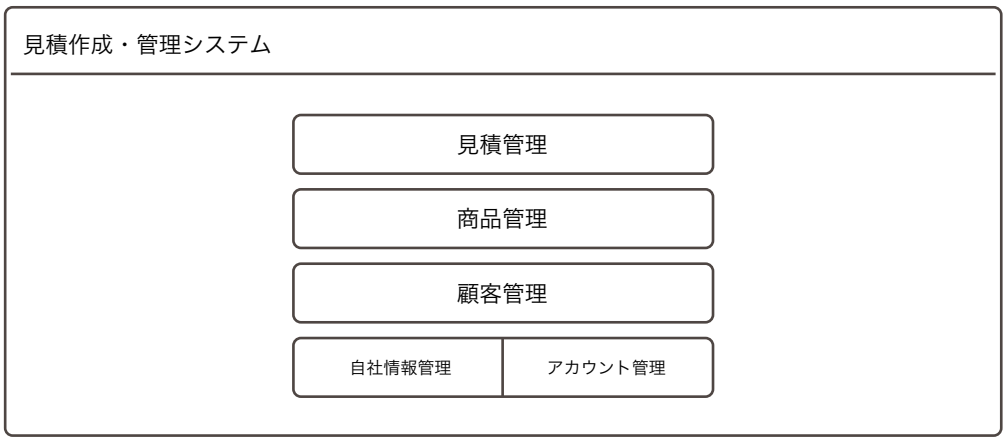


図 8-2 メニューイメージ

・見積管理
・商品情報管理
・顧客情報管理
・自社情報管理
・ログインのためのアカウント管理

メニューはこの 5 種類でした。では、これらの情報から

・画面名
・画面種類
・説明

を、項目として最低限考えます。

見積管理メニューは、次表の業務の流れに当てはめて考えてみます。画面名も仮で考えました。

見積業務（カスタム App 導入後）	作業名	カスタム App	主要な機能と要件
お客様から連絡がくる（開始）			
見積書を作成する （担当者名で）	見積作成	見積登録画面	・新しい見積データを作成する （見積の作成は月に 200 件程度） ・一部の見積項目を自動入力する （見積番号、発行日、期限日、担当名） （見積状況は　作成）
		見積検索画面 見積一覧画面 見積作業画面 見積登録画面	・既存の見積データを検索し指定の 1 件を複製し新しい見積データ作成する （検索は　見積検索　を参照）
印刷する （または PDF にする）	見積出力	見積作業画面 見積書（帳票）	・指定の 1 件の見積データを印刷 ・指定の 1 件の見積データを PDF 化 （用紙サイズは A4 縦）
上司が承認する	承認	×	
	状況記録 　見積検索 　見積管理	見積検索画面 見積一覧画面 見積作業画面	・項目に検索条件を入力して見積データを検索する （お客様電話番号、お客様名、見積管理番号など） ・指定の 1 件の見積データに状況を入力する （見積状況は　お客様検討中）
お客様に見積書を提示する （持っていく または　郵送（窓あき封筒） または　PDF をメール添付送信）	お客様に提示 　見積検索 　見積提示	（メールの場合） 見積検索画面 見積一覧画面 見積作業画面 見積書（PDF）	・項目に検索条件を入力して見積データを検索する ・指定した 1 件の見積データの PDF を添付しメールを送信する
お客様が検討する	注文有無伺い	×	
	問い合わせ対応 　見積検索 　見積訂正＆出力 　見積再出力	見積検索画面 見積一覧画面 見積作業画面 見積書（帳票） 見積書（PDF）	・項目に検索条件を入力して見積データを検索する ・指定の 1 件の見積データを訂正する （商品の増減、値引き） （新しいデータが必要） ・指定の 1 件の見積データ検索して出力する （出力は　見積出力を参照）
（結果がでる） 注文をもらう （FAX　または　電話　または　メール） または　注文されない		×	・注文の連絡は 1 日に 0-10 件程度
結果を記録する （注文されなければここで完了）	結果記録 　見積検索 　見積管理	見積検索画面 見積一覧画面 見積作業画面	・項目に検索条件を入力して見積データを検索する ・指定の 1 件の見積データに結果を入力する
注文をもらった場合、 注文を処理する部署に連携 （完了）	注文受付	×	

画面をまとめると次表の 6 個になります。

見積業務の画面			
	画面名	**レイアウト種類**	**機能**
1	見積登録画面	フォーム形式	新規見積作成 見積項目自動入力（一部） 見積項目入力 見積書印刷／メール添付送信
2	見積検索画面	フォーム形式	検索条件入力 検索実行 検索中止
3	見積一覧画面	リスト形式	見積情報表示（複数）
4	見積作業画面	フォーム形式	新規見積複製作成 見積項目入力 見積書印刷／メール添付送信
5	見積書（帳票）	フォーム形式	
6	見積書（PDF）	フォーム形式	

さて、業務の流れから考えた画面について、どうしてレイアウトの種類 (フォーム、リスト) までわかるのでしょうか。それは、

・その機能で一度に何件のデータを対象に作業をする想定か

を、基準に考えます。

・一度に 1 件ずつ　　→　フォーム形式
・一度に複数件ずつ　　→　リスト形式　または　フォーム形式

これらはあくまでも基準です。一度に複数件を対象に作業するなら多くの場合はリスト形式が適切だと思われますが、連続で入力するためであればフォーム形式が良いかもしれません。

他のメニューについても考えます。これらは同じような管理作業を行います。1 メニューあたり画面は 4 個です。

・顧客情報管理
・商品情報管理
・自社情報管理
・ログインのためのアカウント管理

顧客／商品／自社／アカウント管理			
	画面名	**レイアウト種類**	**機能**
1	登録画面	フォーム形式	新規作成 項目入力
2	検索画面	フォーム形式	検索条件入力 検索実行 検索中止
3	一覧画面	リスト形式	情報表示（複数）
4	作業画面	フォーム形式	項目入力

見積業務は 6 個、その他は 4 業務 ×4 個の 16 個、合わせて画面数が 22 個になりました。

これらが基本の画面一覧（必要な画面の基本）です。次は、画面のレイアウトオブジェクトの配置を考えます。この作業をしながらまた新たな画面が生まれてくるかもしれません。

8.4.2 画面の大きさを想定する

企画時にクライアントの種類も決めているはずです。概ね 4 種類のいずれかです。

・パソコン
・iPad
・iPhone
・その他

レイアウトモードには、パソコン、iPad、iPhone について標準的な画面の大きさの目安となる「画面ステンシル」機能が用意されています。次図のようにカスタムサイズも登録できます。



図 8-3 ステンシルのメニュー

クライアントが使用する画面の大きさを考えて画面サイズを決定します。

パソコンの考慮

パソコンでは、次の考慮が必要です。

・画面いっぱいに表示したいのかどうか
・他のアプリケーションも参照しながら作業したいのかどうか
・クライアントパソコンの画面（モニタ）サイズは」1 種類かどうか

デスクトップパソコンとノートパソコンが混在する場合、小さいサイズの画面で使用するユーザに配慮します。

iPad や iPhone の考慮

iPad と iPhone が混在している場合、共通の画面を作成するよりも、それぞれの画面を作成するようにします。また、縦、横、どちらで使うことを想定するかを決定しておきます。

8.4.3 画面の配置を考える

画面（レイアウト）上に実際になにをどこに置くかを考えます。一般的には画面レイアウトと呼びますが、レイアウトと紛らわしくなります。そのため、本書では「**配置**」と記述します。

レイアウトの種類とパート

これまでにでてきたレイアウトの種類によって概ね決まった配置になるものがあります。次図を参照してください。

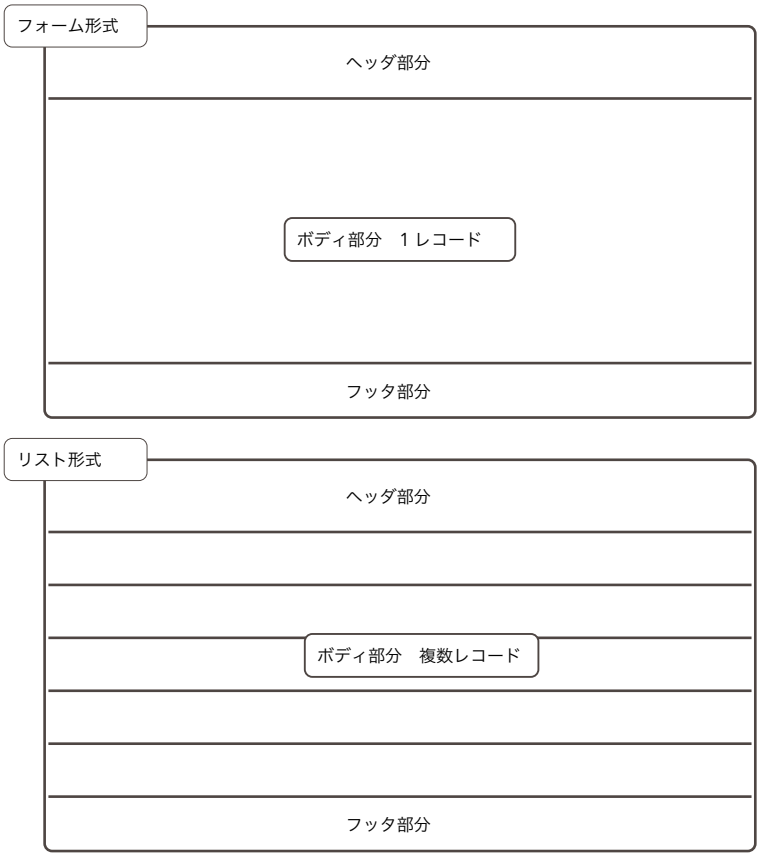


図 8-4 大まかな区分け

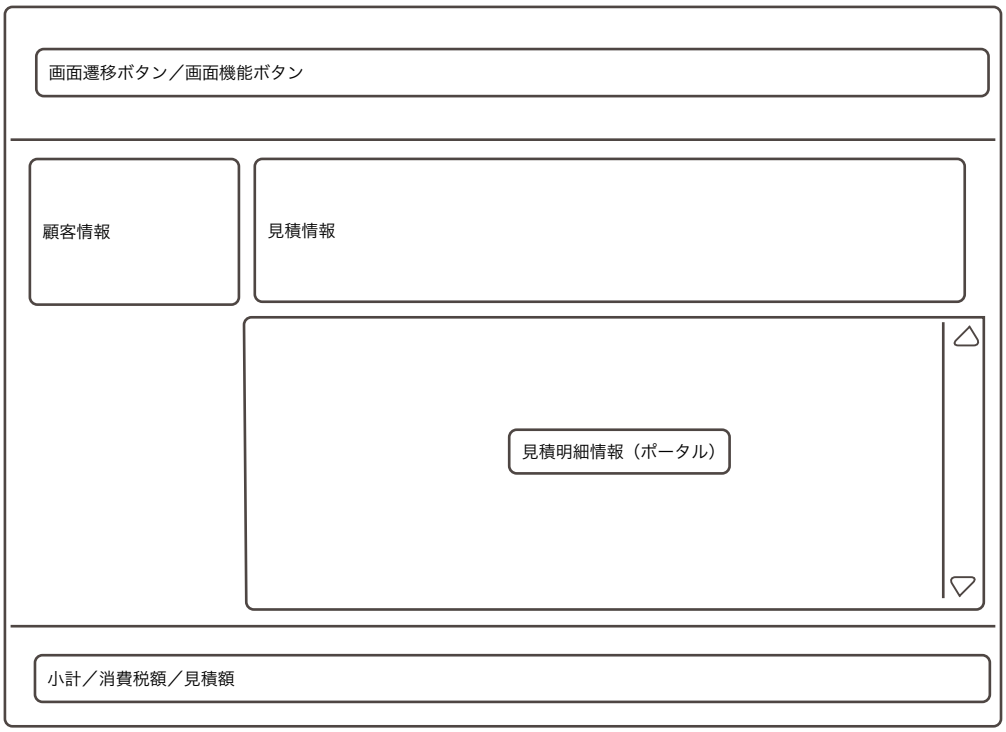
レイアウトには「**パート**」があり、レイアウト上で一番大きな項目配置の区切りになります。

パートを頭において「**ワイヤーフレーム**」を考えます。パートは積み重なるものなので、パートを縦に超えた項目の配置はできません。

例えば、リスト形式の「ボディ」パートの左側をナビゲーションパートにすることはできません。そのような画面を実現したい場合、フォーム形式にポータルを使用して実現します。

ワイヤーフレーム

なにをどこに配置するかだけを考えるときに使用します。手書きでも、図が描けるアプリケーションでも作成できます。



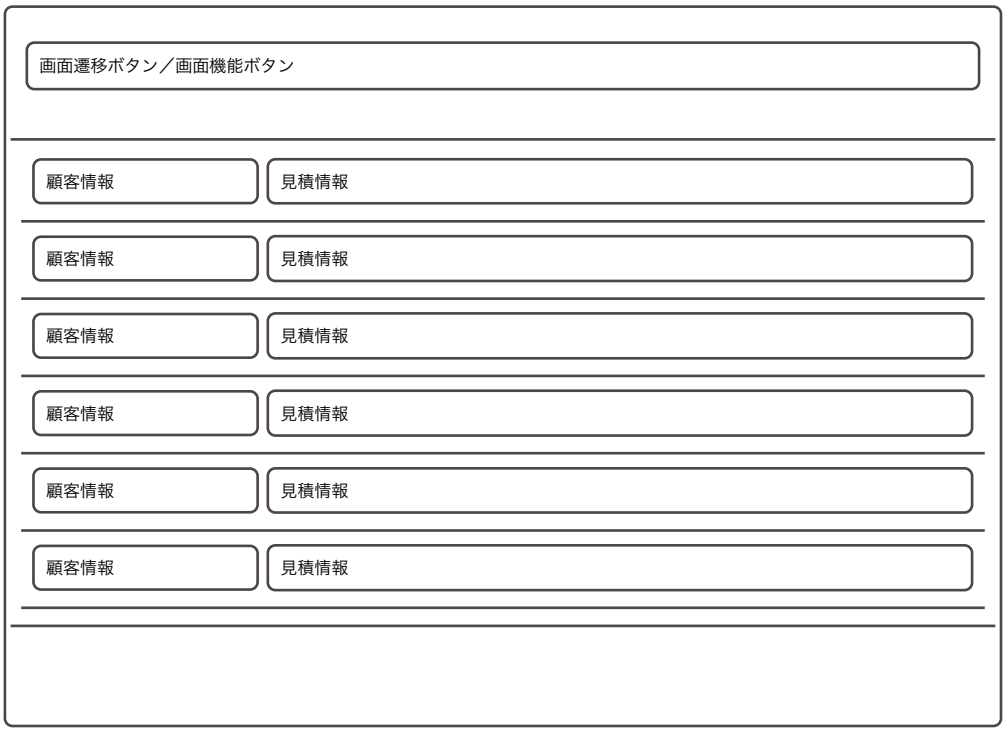


図 8-5 フォーム形式とリスト形式のワイヤーフレーム例

なお、フォーム形式で、フォームとリストを組み合わせたような配置も可能です。

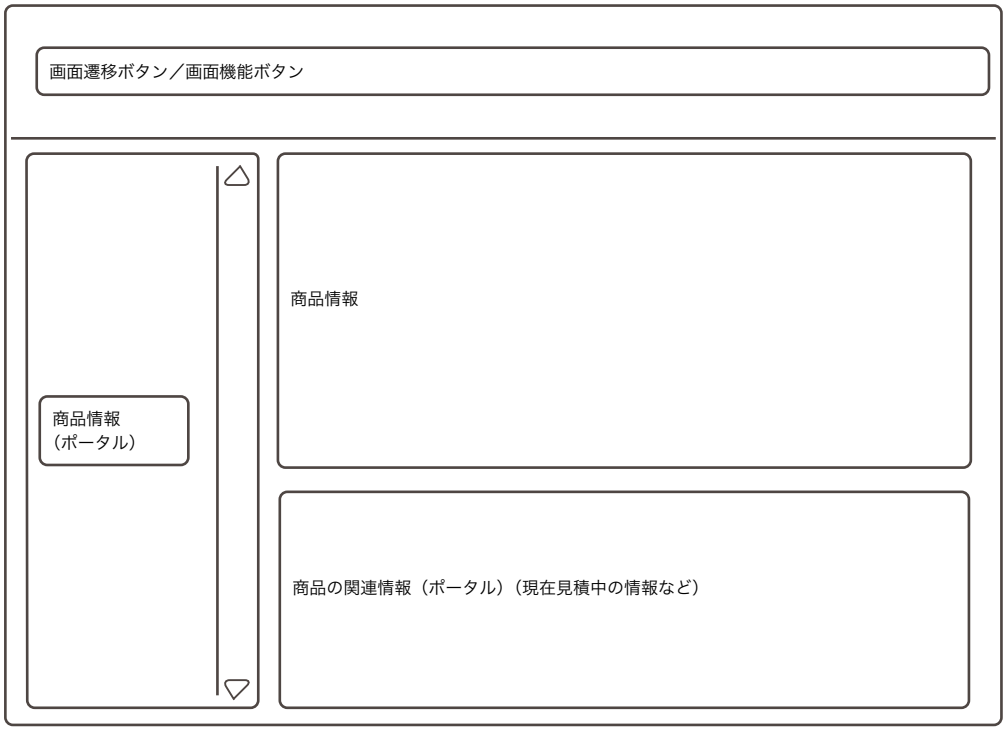


図 8-6 フォーム形式にリストが合わさったワイヤーフレーム例

フォーム形式のレイアウトに、データの一覧と一覧から選択した 1 件ずつのデータ詳細を一緒に表示することを想定した画面です。このような構成のレイアウトを「**マスタ / 詳細レイアウト**」と呼びます。

画面の構成を考える時、このように考えてください。

（1）フォーム形式かリスト形式か（パートで一番基本の画面構成が決まる）
（2）どこになにを配置するか
（3）レイアウトオブジェクトはなにを使うか

大まかで良いのです。どのあたりに、何の情報を配置するかをまず考えます。ざっくりと描きます。

打ち合わせ資料としてのワイヤーフレーム
ワイヤーフレームは A4 サイズの紙に手書きして、打ち合わせで紙芝居のように提示したり、Keynote や Microsoft PowerPoint のようなプレゼンテーションのためのアプリケーションで提示しても大丈夫です。

また、レイアウトオブジェクトの、どれを使って配置するのかもわかりやすいようにします。次図のような落書きのようなものでも、だいたい何のレイアウトオブジェクトかわかります。

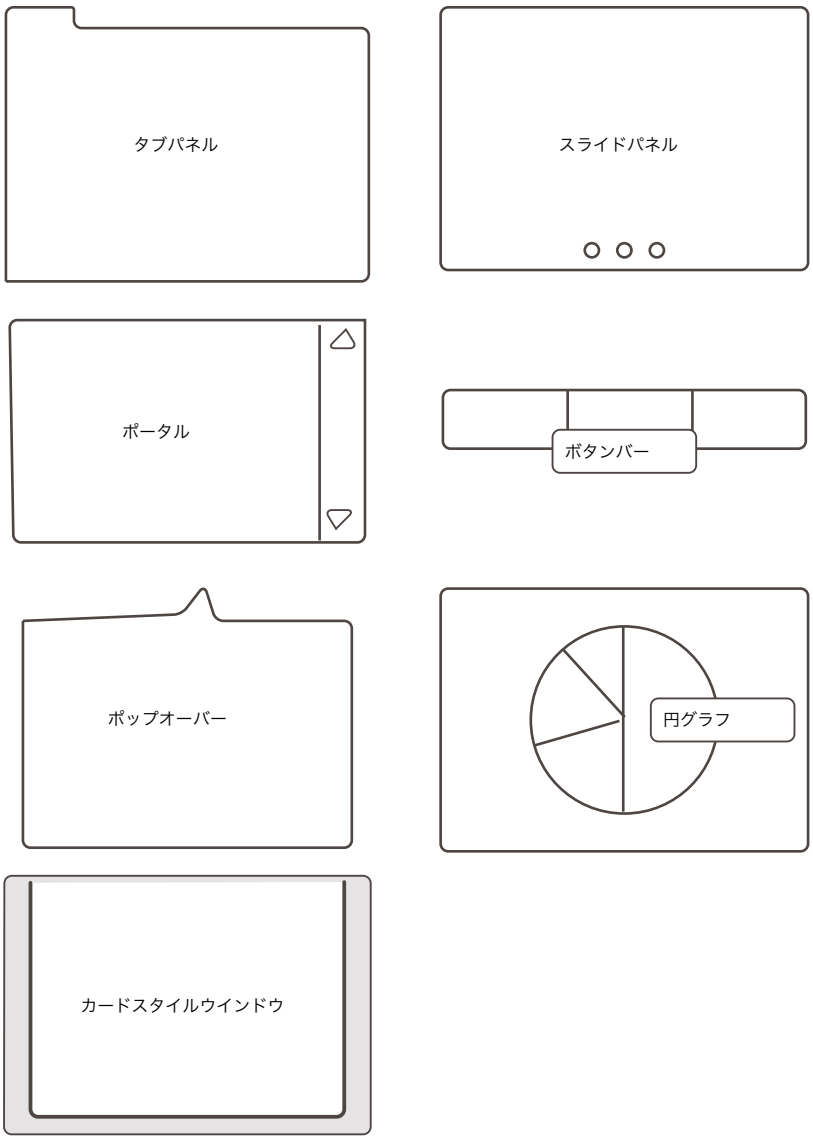


図 8-7 FileMaker Pro のレイアウトオブジェクトを簡略化した例

ワイヤーフレームの作成は、FileMaker Pro のレイアウトでおこなっても良いかもしれません。その場合もフィールドをつくるよりも、あくまで四角形や丸のレイアウトオブジェクトで表現してください。

ワイヤーフレームで、大まかな配置を決めたら、さらに詳細に考えます。「**モックアップ**」と呼ぶことがあります。

・項目
・機能（ボタン）

の配置です。例えば「見積作成管理システム」の「見積登録画面」の機能は次表でした。

見積業務の画面			
	画面名	**レイアウト種類**	**機能**
1	見積登録画面	フォーム形式	新規見積作成 見積項目自動入力（一部） 見積項目入力 見積書印刷／メール添付送信

2 つのテーブルの項目を配置する必要があります。「見積情報」と「見積明細」です。この関係は 1 対多でした。多くの場合、帳票で明細行として表示するものは、ポータルとして配置します。次の理由によるものです。

・見積情報レコード 1 件に対して、見積明細レコードの数が特に決まっていない

しかし、実際には業務上の理由から制限されることがあります。

・これまで 1 見積書に 10 行以上のデータはなかった

の、ような実績に基づくものや、

・1 見積書の合計額が 1,000 万円を超えてはならない

のような業務上のルールのためです。前者は、出力（印刷や PDF）の書式に影響するルールです。後者は、機能で制限するべきルールです。要件定義にルールが明記されていなければ、ヒアリングをして確認します。

打ち合わせ資料としてのモックアップ
モックアップもワイヤーフレームのように提示するとわかりやすいです。

モックアップを FileMaker Pro のレイアウトで作成するには、ダミーフィールドを作成して作業しても良いかもしれません。くれぐれも、これは見本であって本当に開発するファイルではないことを忘れないでください。

次ページ以降に「見積情報」と「見積明細」の設計内容の表と、そこから作成したモックアップ例があります。

やってみよう
「見積作成管理システム」の画面モックアップを考えてみましょう。

	見積情報	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	顧客外部キー	テキスト		
7	自社外部キー	テキスト		
8	発行日	日付	今日　→　作成日	西暦 4 桁の日付、空欄不可
9	見積番号	数字	10001 から自動採番 データ入力時の値変更禁止	空欄不可 ユニークな値
10	c_ 見積総額	計算（結果は　数字）		
11	件名	テキスト		最大文字数 30 カスタムメッセージ データの入力時上書き不可
12	期限日	日付	発行日＋ 1 ヶ月後	西暦 4 桁の日付、空欄不可
13	c_ 見積小計	計算（結果は　数字）		
14	c_ 消費税	計算（結果は　数字）		
15	備考	テキスト		
16	見積状況	テキスト	「作成」	空欄不可
17	見積結果	テキスト	「注文なし」	空欄不可
18	消費税	数字	発行日の消費税 8%だったら 0.08	数字、空欄不可

	見積明細	タイプ	入力値の自動化	入力値の制限
1	主キー	テキスト	計算値で Get(UUID) フィールドに既存の値が存在する場合は置き換えないオフ データ入力時の値変更禁止	空欄不可 ユニークな値
2	作成情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	空欄不可
3	作成者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	空欄不可
4	修正情報タイムスタンプ	タイムスタンプ	作成情報（タイムスタンプ） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
5	修正者	テキスト	修正情報（アカウント名） データ入力時の値変更禁止	西暦 4 桁の日付、空欄不可
6	見積外部キー	テキスト		
7	商品外部キー	テキスト		
8	数量	数字	「0」	数字
9	c_ 金額	計算（結果は　数字）		
10	備考	テキスト		
11	商品番号	数字	商品テーブルから同名フィールドを計算値自動入力	
12	商品名	テキスト	商品テーブルから同名フィールドを計算値自動入力	
13	単価	数字	商品テーブルから同名フィールドを計算値自動入力	

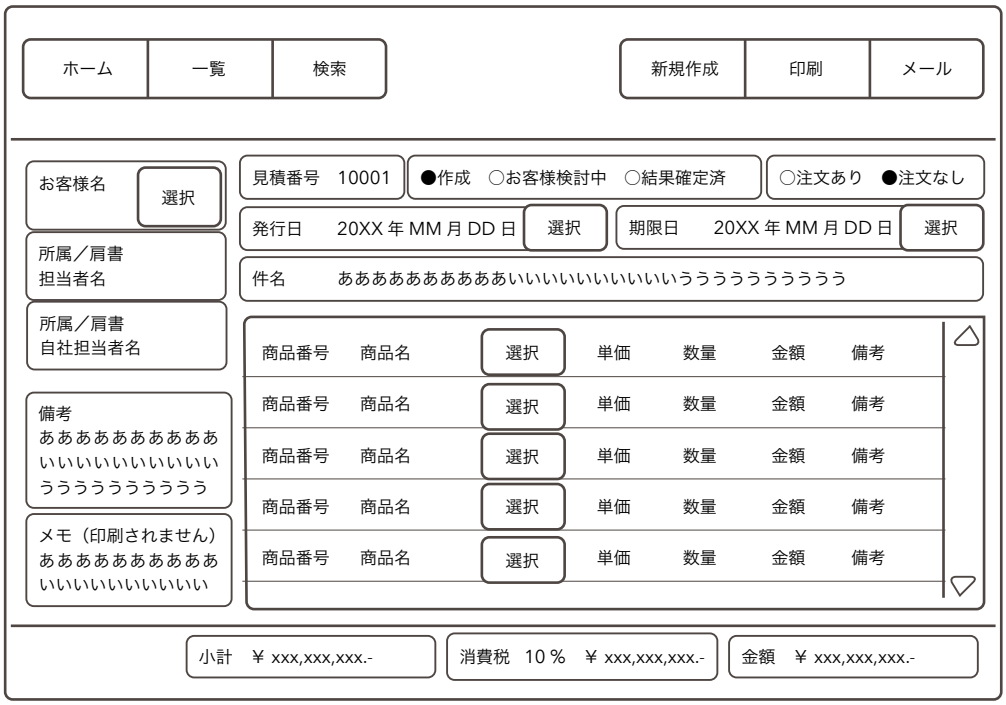


図 8-8 モックアップ例

項目は全部入っているでしょうか。次図のようなことも大事です。機能や、表示方法も考えます。

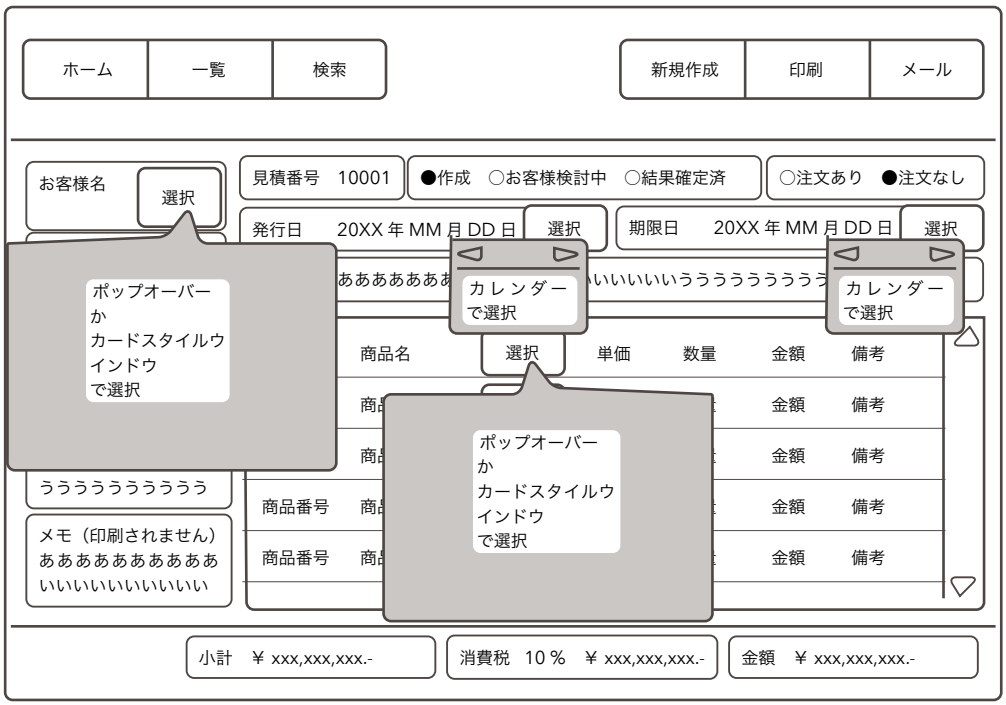


図 8-9 モックアップ例 2　どうやって選択するかを提示

8.4.4 画面遷移

打ち合わせで紙芝居のように提示をおこなうには、どのような順番が良いのでしょうか。画面がユーザに表示される順番を決めるのが「**画面遷移**」です。

画面は、流れ（フロー）でもありますが、起動時に表示される画面を中心として枝が伸びて交差することもある木構造（ツリー構造）としても考えられます。これまでに考えた画面を、流れやツリー構造の側面から見てユーザがどのように行き来するのかも、また図にするのです。

画面を四角形で描いて、つながりを矢印で表現します。

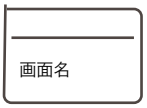


図 8-10 画面遷移の画面を表す記号例

	画面名	レイアウト種類
1	見積登録画面	フォーム形式
2	見積検索画面	フォーム形式
3	見積一覧画面	リスト形式
4	見積作業画面	フォーム形式
5	見積書（帳票）	フォーム形式
6	見積書（PDF）	フォーム形式
7	顧客登録画面	フォーム形式
8	顧客検索画面	フォーム形式
9	顧客一覧画面	リスト形式
10	顧客作業画面	フォーム形式
11	商品登録画面	フォーム形式
12	商品検索画面	フォーム形式
13	商品一覧画面	リスト形式
14	商品作業画面	フォーム形式
15	自社情報登録画面	フォーム形式
16	自社情報検索画面	フォーム形式
17	自社情報一覧画面	リスト形式
18	自社情報作業画面	フォーム形式
19	アカウント情報登録画面	フォーム形式
20	アカウント情報検索画面	フォーム形式
21	アカウント情報一覧画面	リスト形式
22	アカウント情報作業画面	フォーム形式

いま「見積作成管理システム」では、上表の 22 個の画面を考えていますが、画面遷移を考えながらさらに検討していくと……。

やってみよう

「見積作成管理システム」の画面遷移を考えてみましょう。

ホーム画面が必要です。

そして、ユーザは自社担当者でもあることに気付きました。アカウント管理画面と自社担当者情報は一緒にできます。自社担当者の管理画面は不要です。

しかし、自社情報（会社名や住所など）は全体で 1 つ必要です。従って自社情報を管理するためのフォーム形式の画面が 1 つ必要です。

○○登録　と　○○作業　は、同じ内容で充分と考えられます。そして、見積書は、印刷でも PDF でも同じ内容です（画面遷移的には別でもレイアウトとしては同じ画面があるということです。）。

また、システムの規模を思い出します。

・ユーザは 10 人程度
・1 ユーザが扱う見積書は 10 ～ 20 件／月（0 ～ 2 件／ 1 日、120 ～ 240 件／ 1 年間）
・作業が集中する日は特に無し、期間としては年度末には集中しがち

ユーザ数に注目します。ユーザ数と自社担当者数は同じだとすると、10 人程度なら検索する必要はありません。「マスタ / 詳細レイアウト」で良さそうです。

	画面名	レイアウト種類
1	見積登録・作業画面	フォーム形式
2	見積検索画面	フォーム形式
3	見積一覧画面	リスト形式
~~4~~	~~見積作業画面~~	~~フォーム形式~~
5	見積書（帳票）（印刷／ PDF）	フォーム形式
~~6~~	~~見積書（PDF）~~	~~フォーム形式~~
7	顧客登録・作業画面	フォーム形式
8	顧客検索画面	フォーム形式
9	顧客一覧画面	リスト形式
~~10~~	~~顧客作業画面~~	~~フォーム形式~~
11	商品登録・作業画面	フォーム形式
12	商品検索画面	フォーム形式
13	商品一覧画面	リスト形式
~~14~~	~~商品作業画面~~	~~フォーム形式~~
~~15~~	~~自社情報登録画面~~	~~フォーム形式~~
~~16~~	~~自社情報検索画面~~	~~フォーム形式~~
~~17~~	~~自社情報一覧画面~~	~~リスト形式~~
~~18~~	~~自社情報作業画面~~	~~フォーム形式~~
19	自社担当者・アカウント情報登録・作業画面	フォーム形式（マスタ / 詳細）
~~20~~	~~自社担当者・アカウント情報検索画面~~	~~フォーム形式~~
~~21~~	~~自社担当者・アカウント情報一覧画面~~	~~リスト形式~~
~~22~~	~~自社担当者・アカウント情報作業画面~~	~~フォーム形式~~
23	自社情報管理画面	フォーム形式
25	ホーム画面	フォーム形式

だいぶ少なくなりました。次表と次図を確認してください。

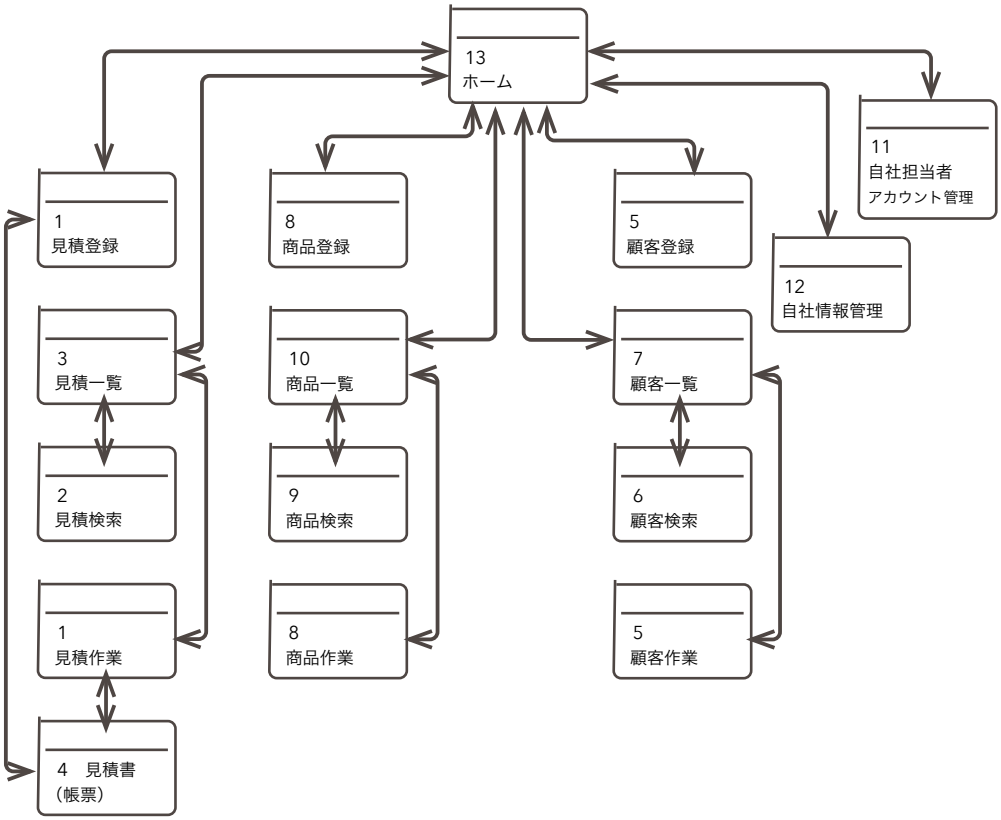


図 8-11 画面遷移図の例

		画面名	レイアウト種類
1	~~1~~	見積登録・作業画面	フォーム形式
2	~~2~~	見積検索画面	フォーム形式
3	~~3~~	見積一覧画面	リスト形式
4	~~5~~	見積書（帳票）（印刷／ PDF）	フォーム形式
5	~~7~~	顧客登録・作業画面	フォーム形式
6	~~8~~	顧客検索画面	フォーム形式
7	~~9~~	顧客一覧画面	リスト形式
8	~~11~~	商品登録・作業画面	フォーム形式
9	~~12~~	商品検索画面	フォーム形式
10	~~13~~	商品一覧画面	リスト形式
11	~~19~~	自社担当者・アカウント情報登録・作業画面	フォーム形式（マスタ / 詳細）
12	~~23~~	自社情報管理画面	フォーム形式
13	~~25~~	ホーム画面	フォーム形式

これで、紙芝居の順番ができました。画面の管理番号も整理しました。

打ち合わせでは、現在検討している画面がどれかをわかりやすくします。画面に管理番号を付与するとシンプルに管理できます。レイアウト番号ではなく、あくまでも設計上の管理番号です。

実際に『○○作業をするには』のシチュエーションを提示して、画面を順番に提示します。ユーザがイメージしやすくなります。画面遷移図、ワイヤーフレーム、モックアップを確認しながら、業務の流れの中で、どの画面を使用するのかを検討すると、

・画面ごとに必要な機能
・画面ごとに必要な入出力項目

が、わかってきます。

8.4.5 メニューやホーム画面

画面遷移図で、一番上に「メニュー」がありました。この画面は『起動時に表示される画面』です。「ホーム」、「トップ」画面とも呼びます。画面遷移のことを考えた結果、メニュー項目が次図のようになりました。

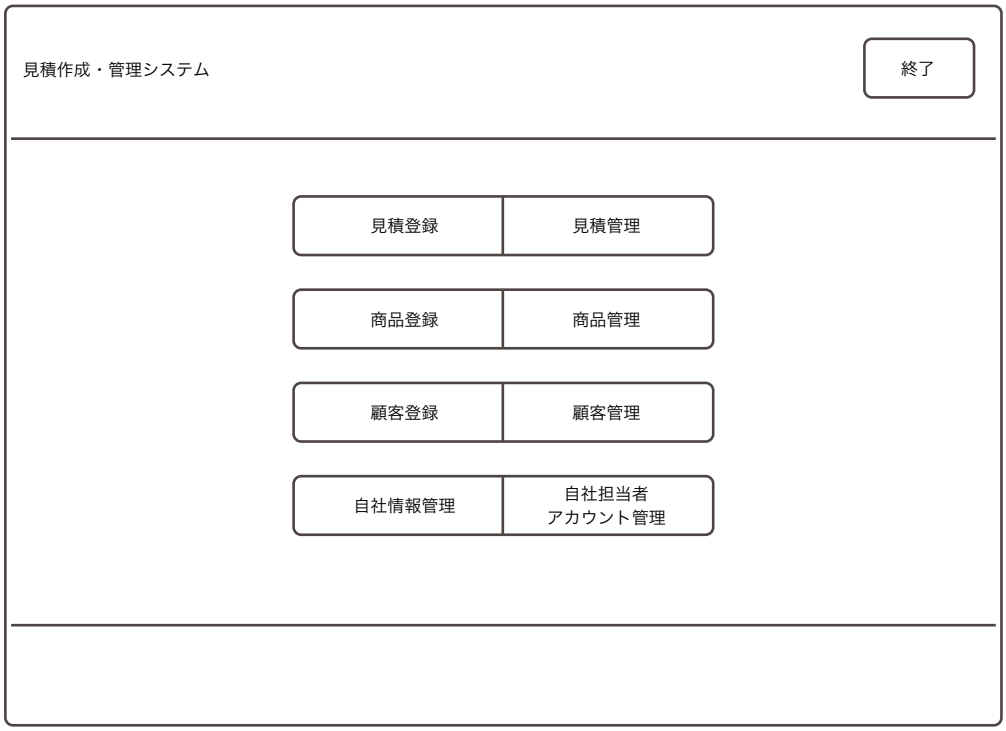


図 8-12 画面の例（メニューの役割）

メニューボタンだけを置いても良いですが、その他にログインしたアカウントに従って、

・アカウントの担当者が作成した見積データの「見積状況」フィールドが「お客様検討中」のデータだけをポータルに表示して、そこから「見積作業」画面へ遷移するようにする

なども可能です。ダッシュボード的な利用方法です。

ホームとなる画面もユーザにわかりやすいため要望がたくさんでることがあります。例えば、作業の効率化のため、この画面を起点として様々なリスト、フォームへの遷移を要望されることがあります。その場合、遷移するだけではなく

・遷移したときのレイアウト（TO）の対象レコードがどうなるか

を考慮します。作業後にホーム画面に戻るならばあまり気になりませんが、リスト画面へ遷移したらどうなるでしょうか。

なお、画面にポータルを多数配置すると表示遅くなることがあります。1 つの画面に多様な情報が表示されることはユーザにとって有益ですが、一方で表示されるまでに時間がかかることはユーザにとってはストレスになります。

例えば、次図のようにグラフや件数をわかりやすく表示してユーザの状況把握と作業を行うきっかけにするなら、そのグラフや件数がリアルタイムの必要があるかどうかを確認します。

・前営業日の営業終了時間時点のグラフや件数

で充分かもしれません。それならば FileMaker Cloud あるいは FileMaker Server で夜間にスクリプトをスケジュール実行し集計しておけば、ユーザがホーム画面を表示するたびに処理が実行されることがありません。

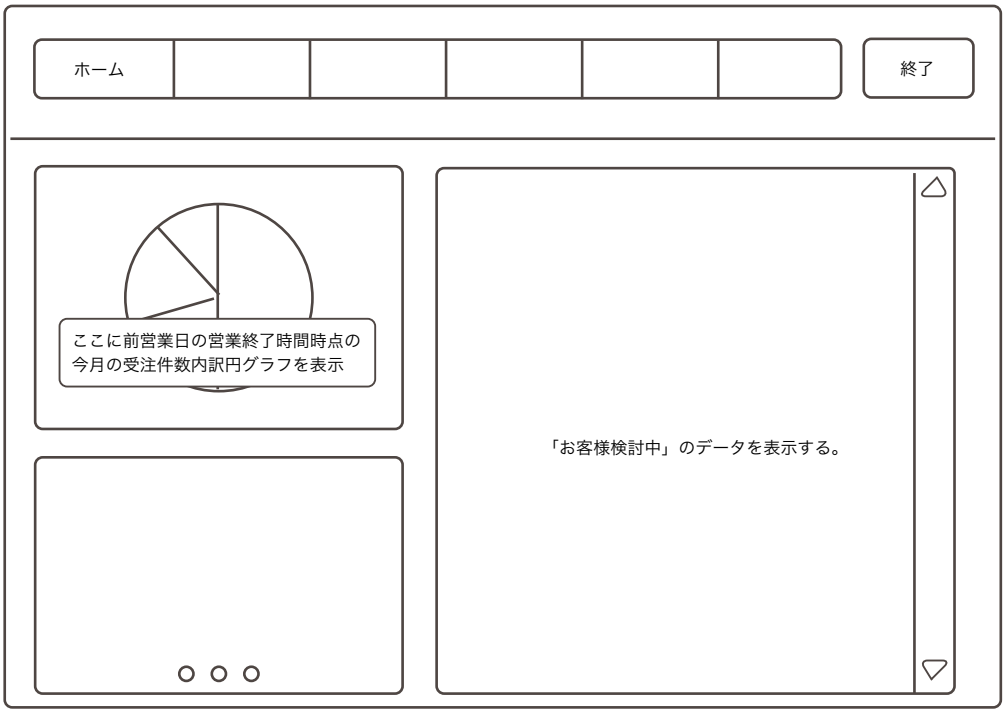


図 8-13 ダッシュボード的ホーム画面の例（メニューとメニュー以外の役割）

やってみよう

「見積作成管理システム」のホーム画面を考えてみましょう。

8.5 モバイルデバイスの画面

iPad や iPhone などのモバイルデバイスの FileMaker Go からの使用を想定する場合、どのように考えられるでしょうか。

8.5.1 どのような作業をおこなうか

『モバイルデバイスでおこなう作業はなにか』を決定します。「見積作成管理システム」のメニューのうち、

・見積

に関する作業は、外出先でもできると便利です。

見積業務（カスタム App 導入後）	作業名	カスタム App	
		パソコン	モバイル
お客様から連絡がくる（開始）		×	×
見積書を作成する （担当者名で）	見積作成	見積登録画面	○
		見積検索画面 見積一覧画面 見積作業画面 見積登録画面	○
印刷する （または PDF にする）	見積出力	見積作業画面 見積書（帳票）	○
上司が承認する	承認	×	×
	状況記録 見積検索 見積管理	見積検索画面 見積一覧画面 見積作業画面	○
お客様に見積書を提示する （持っていく または　郵送（窓あき封筒） または　PDF をメール添付送信）	お客様に提示 見積検索 見積提示	（メールの場合） 見積検索画面 見積一覧画面 見積作業画面 見積書（PDF）	○
お客様が検討する	注文有無伺い	×	×
	問い合わせ対応 見積検索 見積訂正＆出力 見積再出力	見積検索画面 見積一覧画面 見積作業画面 見積書（帳票） 見積書（PDF）	○
（結果がでる） 注文をもらう （FAX　または　電話　または　メール） または　注文されない		×	×
結果を記録する （注文されなければここで完了）	結果記録 見積検索 見積管理	見積検索画面 見積一覧画面 見積作業画面	○
注文をもらった場合、 注文を処理する部署に連携 （完了）	注文受付	×	×

他のメニューはどうでしょうか。

・顧客
・商品
・自社情報
・自社担当者とアカウント管理

新しい顧客の登録は、ありそうです。しかし、他の作業は外出先から作業できなくても良いと考えられます。

もうすこし考えると、商品については管理作業ができなくても商品情報について確認出来た方が良さそうです。自社情報についても同じです。

8.5.2 クライアントの種類で画面を分ける

モバイルデバイスで使いたい画面については、パソコンとモバイルデバイスを共通の内容にするよりも、それぞれの画面の大きさ、使い方を考慮して別々のレイアウトを作成することをお勧めします。

次表になりました（薄く塗った部分が増えた画面です）。

	画面名	レイアウト種類	パソコン	モバイル
1	見積登録・作業画面	フォーム形式	○	×
2	見積検索画面	フォーム形式	○	×
3	見積一覧画面	リスト形式	○	×
4	見積書（帳票）（印刷／ PDF）	フォーム形式	○	×
5	顧客登録・作業画面	フォーム形式	○	×
6	顧客検索画面	フォーム形式	○	×
7	顧客一覧画面	リスト形式	○	×
8	商品登録・作業画面	フォーム形式	○	×
9	商品検索画面	フォーム形式	○	×
10	商品一覧画面	リスト形式	○	×
11	自社担当者・アカウント情報登録・作業画面	フォーム形式 (マスタ / 詳細)	○	×
12	自社情報管理画面	フォーム形式	○	×
13	ホーム画面	フォーム形式	○	×
14	モバイル　見積登録・作業画面	フォーム形式	×	○
15	モバイル　見積検索画面	フォーム形式	×	○
16	モバイル　見積一覧画面	リスト形式	×	○
17	モバイル　顧客登録・作業画面	フォーム形式	×	○
18	モバイル　顧客検索画面	フォーム形式	×	○
19	モバイル　顧客一覧画面	リスト形式	×	○
20	モバイル　商品閲覧画面	フォーム形式 (マスタ / 詳細)	×	○
21	モバイル　ホーム画面	フォーム形式	×	○

8.6　画面に共通のこと

少し違う角度から考えます。前述しましたが画面は、大きく分けて、

・フォーム
・リスト

になります。これまでに例示したワイヤーフレーム、モックアップには、

・上部にヘッダ
・間にボディ
・下部にフッタ

があります。どの画面もこの 3 つの要素を基準に構成を考えます。要素の中は、さらにいくつかのパートを指定することが可能です。

ヘッダ要素
画面遷移や画面の機能ボタンを配置するための要素です。さらに細分化して、画面遷移のための「ナビゲーション」と「機能や画面名､ユーザの目に入って欲しい文言（○○を ×× してください)」の 2 系統のパートにより配置できます。つまり、上部ナビゲーションパートとヘッダパートを使い分けることが可能です。なにも配置しないこともありますし、ヘッダを使用しないこともあります。

上部ナビゲーションパート
画面の一番上に表示され（順番は変えられません)、レイアウトのナビゲーションに使用するボタンなどを配置するパートとして使用されます。1 つのレイアウトに 1 つだけ設定できます。このパートは画面をスクロールしても位置が変わりません。ズームコントロールの拡大と縮小の影響を受けません。印刷もされません。

ボディ要素
主にレコードの表示場所となります。必要に応じてヘッダやフッタが無いボディだけの画面、逆にボディが無い画面をつくることもあります。

フッタ要素
ヘッダ要素に配置するものをあえてフッタ要素に配置することがあります。例えば、普段はユーザが意識する必要がないハウスキーピングフィールドの情報を小さい文字で表示したりします。なにも配置しないこともありますしフッタを使用しないこともあります。ヘッダ要素と同じように下部ナビゲーションパートとフッタパートを使い分けることができます。

下部ナビゲーション
画面の一番下に表示され（順番は変えられません）ます。上部ナビゲーションと同じ特徴があります。

どの画面も同じように構成されていることで画面に統一感がでます。ユーザは『同じカスタム App を使用している』と意識します。ボタンの大きさや色の傾向などもそうです。

FileMaker Pro に用意されているデザインテンプレートでも色の付与の仕方などで、画面の統一感を出す工夫がされています。

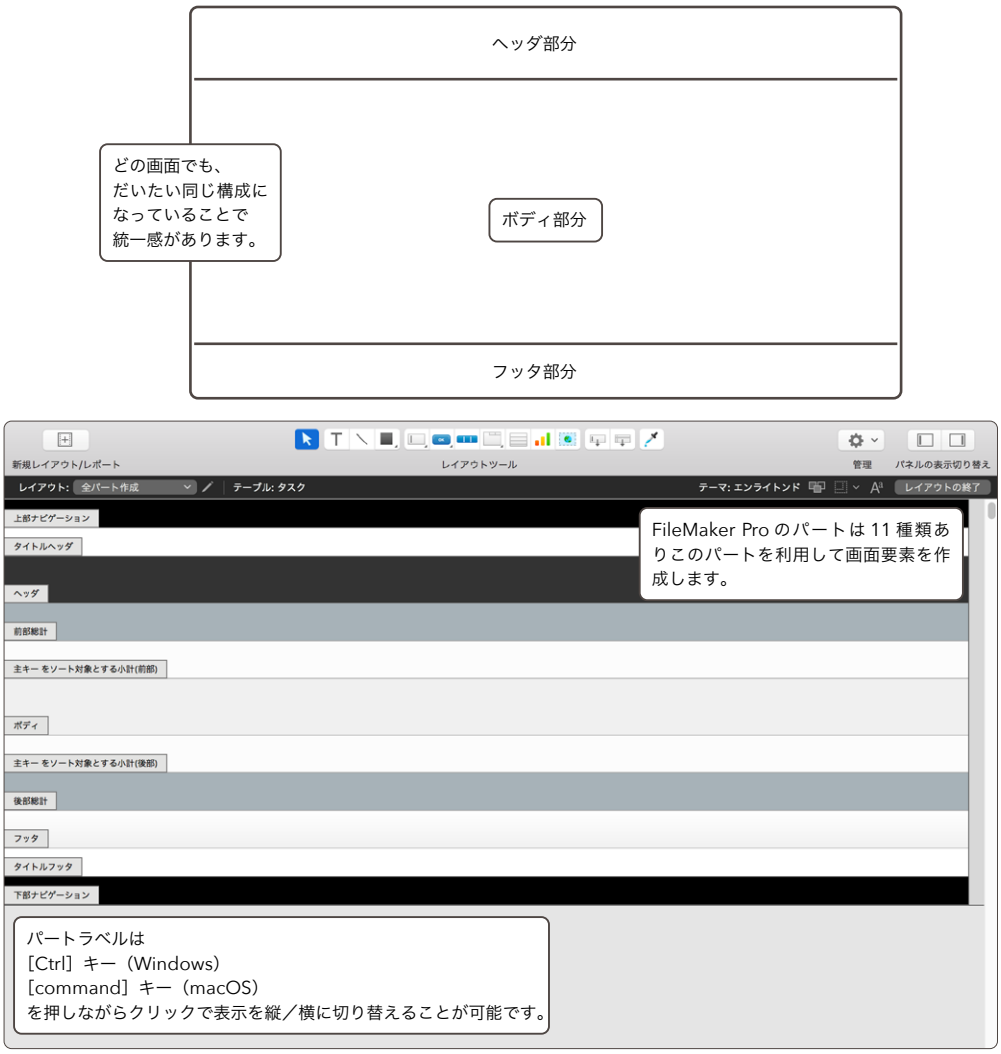


図 8-14 画面の要素とレイアウトパート

8.7 ナビゲーション

画面の遷移（ナビゲーション）には、いくつかの方法があります。一番シンプルなのはボタンを押すことです。とてもユーザにわかりやすい方法です。いくつかのナビゲーション例を記述します。これらは、ナビゲーションパートを使用して配置することをお勧めします。

ボタン

画面の左上に配置することが多いものです。ボタンバーを使用します。ボタンバーの左から順にホーム、移動可能な画面 1、移動可能な画面 2……のように並んでいます。クリックすると遷移します。

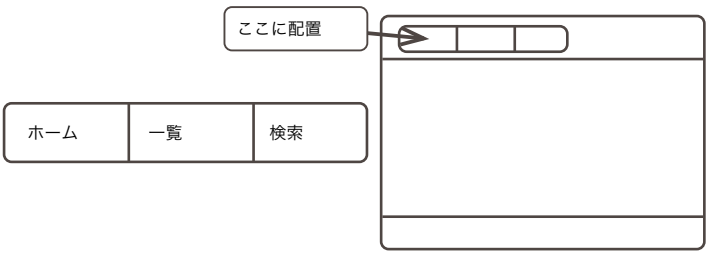


図 8-15 ボタンバーの画面遷移

ポップアップメニュー

左上または右上にあって、ボタンのようなオブジェクトをクリックするとポップアップメニューが表示されて画面名やメニュー名を選択すると遷移します。上から順にホーム、移動可能な画面、あるいはメニュー名のように並びます

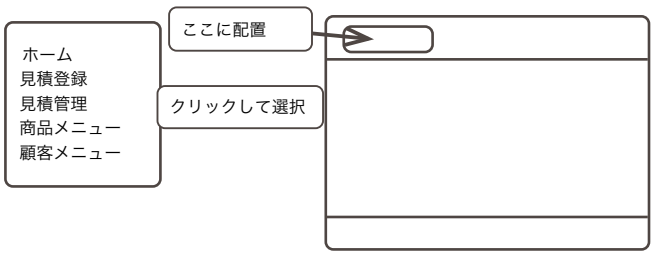


図 8-16 ポップアップメニューの画面遷移

ポップオーバー

左上または右上にあってボタンをクリックするとポップオーバーが表示されてそこに画面遷移のボタンが表示されます。クリックすると遷移します。

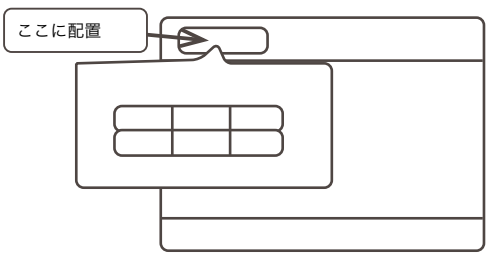


図 8-17 ポップオーバーの画面遷移

その他

カスタムメニュー、ポップオーバー上にポータルを置いてポータルフィルタを使用、カードスタイルウインドウを利用など、工夫できることが色々あります。

いずれも、遷移先の画面をなんらかのわかりやすい形でユーザに選択してもらうことが目的です。

もう 1 つ考えることがあります。ホーム画面に戻るかどうかです。

(1) メニューへ移動する
(2) 作業を実施する
(3) そのメニューでの作業が終わったらホームへ戻る
(4) 次のメニューへ……

このような流れで作業をおこなうので、必ずホームへ戻ります。

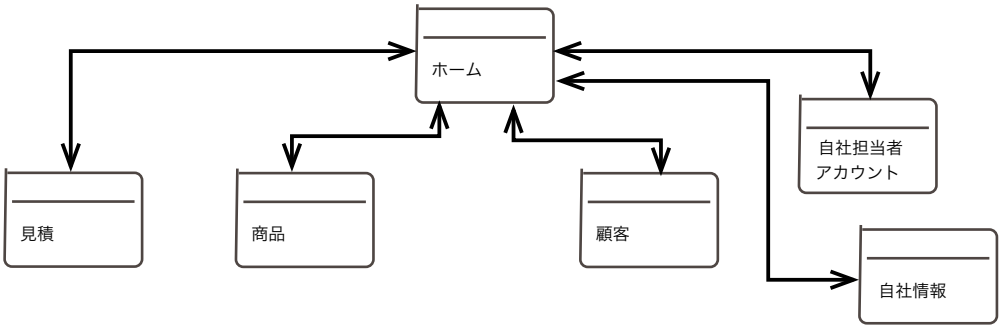


図 8-18 メニュー同士は行き来せずホームを中心にする

または、良く使う 3 つのメニューを横断して移動できるようになると

(1) メニューへ移動する
(2) 作業を実施する
(3) そのメニューでの作業が終わったら次のメニューへ……

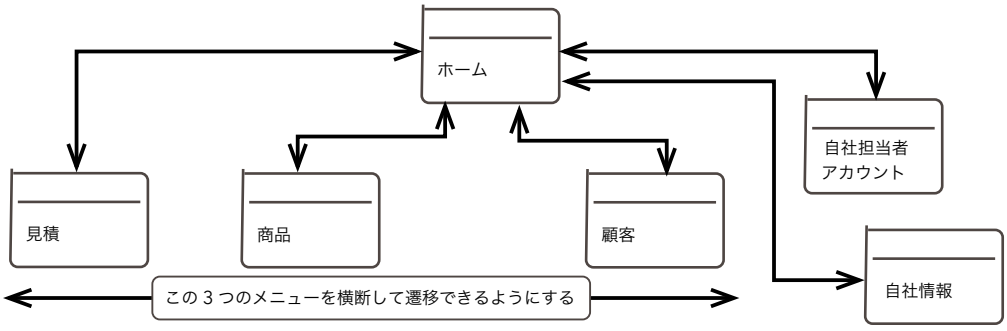


図 8-19 メニュー同士も行き来できるようにする

ユーザの手間を 1 つ短縮できるので行き来が楽になります。しかし、ユーザがいまどこにいるかがわからなくなるかもしれません。まずはホームを中心に各メニューを行き来するほうがシンプルです。その後にメニューの横断による遷移を考えてください。

メンテナンスの観点から考えても、まず、常にホーム画面に戻る方法が無難でお勧めです。

メンテナンス
画面遷移のためのスクリプトを作成する場合どのようなナビゲーション方法を選択しているかを考慮してください。

ホーム画面に戻る方法では、メニューごとに意識する必要があるのは原則としてそのメニューの中での画面遷移だけです。また、メニューが増えたときに別のメニューのことをあまり考えなくてもよくなります。それだけ画面遷移のためのスクリプトがシンプルで済みます。

メニュー間を横断する方法では、常に「どの画面からどこへ遷移できるようにするか」を意識していなくてはいけません。メニューが増えたときに別のメニューから遷移されたときのことを考える必要があります。スクリプトは複雑になるかもしれません。

「見積作成管理システム」の画面は 21 個、画面遷移はクライアントの種類によって 2 種類です。

	画面名	レイアウト種類	パソコン	モバイル
1	見積登録・作業画面	フォーム形式	○	×
2	見積検索画面	フォーム形式	○	×
3	見積一覧画面	リスト形式	○	×
4	見積書（帳票）（印刷／ PDF）	フォーム形式	○	×
5	顧客登録・作業画面	フォーム形式	○	×
6	顧客検索画面	フォーム形式	○	×
7	顧客一覧画面	リスト形式	○	×
8	商品登録・作業画面	フォーム形式	○	×
9	商品検索画面	フォーム形式	○	×
10	商品一覧画面	リスト形式	○	×
11	自社担当者・アカウント情報登録・作業画面	フォーム形式 （マスタ / 詳細）	○	×
12	自社情報管理画面	フォーム形式	○	×
13	ホーム画面	フォーム形式	○	×
14	モバイル　見積登録・作業画面	フォーム形式	×	○
15	モバイル　見積検索画面	フォーム形式	×	○
16	モバイル　見積一覧画面	リスト形式	×	○
17	モバイル　顧客登録・作業画面	フォーム形式	×	○
18	モバイル　顧客検索画面	フォーム形式	×	○
19	モバイル　顧客一覧画面	リスト形式	×	○
20	モバイル　商品閲覧画面	フォーム形式 （マスタ / 詳細）	×	○
21	モバイル　ホーム画面	フォーム形式	×	○

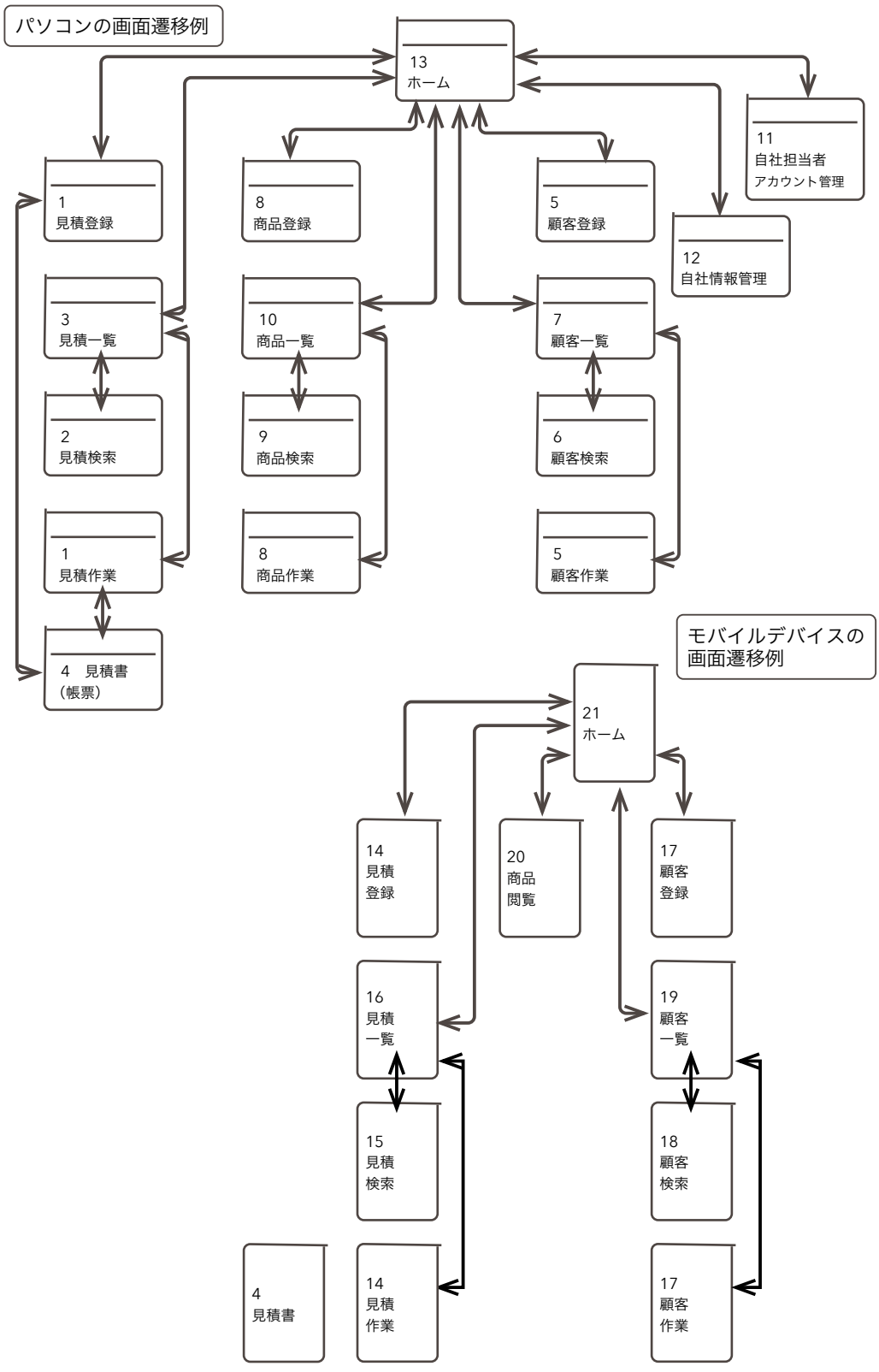

パソコンの画面遷移例
13
ホーム
1
見積登録
8
商品登録
5
顧客登録
11
自社担当者
アカウント管理
12
自社情報管理
3
見積一覧
10
商品一覧
7
顧客一覧
2
見積検索
9
商品検索
6
顧客検索
1
見積作業
8
商品作業
5
顧客作業
4　見積書
（帳票）
モバイルデバイスの
画面遷移例
21
ホーム
14
見積
登録
20
商品
閲覧
17
顧客
登録
16
見積
一覧
19
顧客
一覧
15
見積
検索
18
顧客
検索
4
見積書
14
見積
作業
17
顧客
作業

図 8-20 画面遷移図

運用開始後と画面

共有され、複数のユーザが使用する前提のカスタム App では、ユーザに予告せずにレイアウトの大きな変更を何度も実施することはお勧めできません。微調整は構いませんが、表示される画面の順番や項目の位置が、ある日大きく変わっていたらユーザが戸惑うからです。

絶対にそのようなことをしてはいけないのではありません。必要があれば作業します。計画してユーザへ告知するタイミングなどを検討してから実施します。

画面案を作成し、実際にカスタム App を使用するユーザに見せておくと見慣れた画面になるので運用開始がスムーズになるかもしれません。

第 9 章　レイアウトツール

画面の設計では自由な発想が大事ですが、FileMaker Pro を使ってどのようなレイアウトがつくれるか、がわかっていると、より確実なデザインが可能です。

「**環境設定**」の［**新規フィールドを現レイアウトに加える**］チェックを　**オフ**　にしてください。

9.1　画面について

画面構成の基本を確認します。画面構成は次図です。



図 9-1　画面の各部と名前（ブラウズモード）

ステータスツールバー
画面上部にあり、よく使われる「コマンド（命令）」を簡単に実行できるようにアイコンのボタンが配置されています。モードによって表示される内容が異なります。

レイアウトバー
ステータスツールバーの下にあります。表示中の「レイアウト名」や表示形式の切り替えなどのボタンが並んでいます。モードによって表示される内容が異なります。

コンテンツエリア
レイアウトバーの下のエリアでレイアウトが表示されます。モードにより表示内容が異なります。

画面の表示には「表示形式」と「モード」があります。

3 つの表示形式
コンテンツエリアのデータ表示は、3 つの表示形式があります。

- **フォーム形式** → 1 件ごとにレコードを表示する
- **リスト形式** → 複数件のレコードを表示する
- **表形式** → 複数件のレコードをスプレッドシートのように表示する

次のいずれかの方法で表示形式を切り替えます。

方法 1：メニュー
［表示］メニューで切り替えたい形式を選択します。

方法 2：レイアウトバー
切り替えたい形式のアイコンをクリックします。

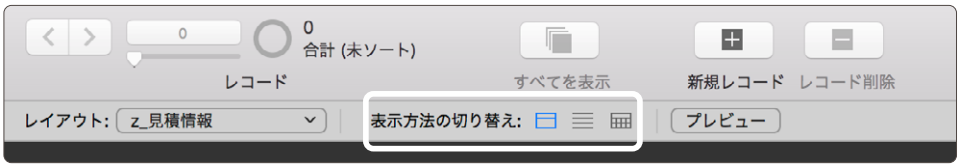


図 9-2 レイアウトバーのアイコン（フォーム／リスト／表　の順で並んでいる）

4 つのモード
作業内容に適した画面表示の「**モード**」が用意されています。

- **ブラウズモード** → データの表示、作成、編集および削除を実行する
- **検索モード** → データを検索する
- **レイアウトモード** → レイアウトを作成したり編集したりする
- **プレビューモード** → 印刷のイメージを確認する

次のいずれかの方法でモードを切り替えます。

方法：メニュー
［表示］メニューで切り替えたい形式を選択します。

方法：ショートカット
ブラウズモード → ［Ctrl］＋［B］（Windows）あるいは［command］＋［B］（macOS）
検索モード → ［Ctrl］＋［F］（Windows）あるいは［command］＋［F］（macOS）
レイアウトモード → ［Ctrl］＋［L］（Windows）あるいは［command］＋［L］（macOS）
プレビューモード → ［Ctrl］＋［U］（Windows）あるいは［command］＋［U］（macOS）

※ カスタム App の作成作業では、レイアウトモードで修正、ブラウズモードで確認と、何度も 2 つのモードを行き来します。ショートカットなど、手早くできる方法を覚えると効率が良くなります。

9.2 レイアウトモード

画面に関する作業を主に行うモードです。レイアウトモードの画面構成は次図です。



図 9-3 レイアウトモードの画面

ステータスツールバーやレイアウトバーの内容が変わり、パネルが利用できるようになります。フィールドやレイアウトオブジェクトを配置、操作、デザインするためのツールが用意されています。

左右のパネル

左パネルに「フィールド」「オブジェクト」「アドオン」、右パネルには「インスペクタ」が表示されます。

パネルの表示／非表示

レイアウトのパネルは、ステータスツールバー右の［パネルの表示切り替え］をクリックすると表示／非表示が切り替ります（次図を参照）。

ステータスツールバーとレイアウトバー

各種コマンドと、テーマ、画面ステンシルなどが用意されています。

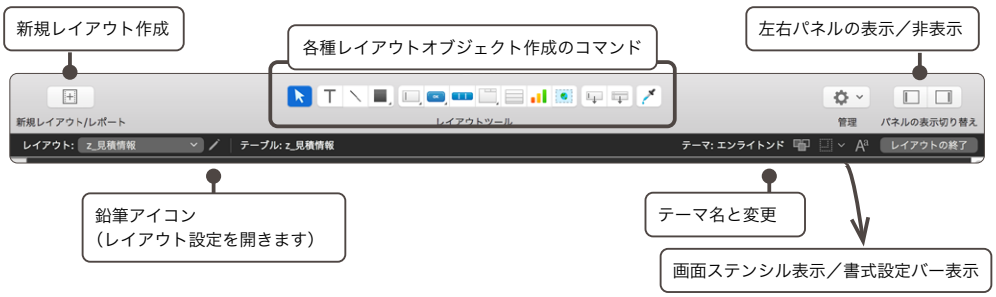


図 9-4 レイアウトモードのステータスツールバーとレイアウトバー

レイアウト保存の確認

レイアウトモードでレイアウトに関する変更を行って、保存せずにレイアウトモードを終了すると確認ダイアログが表示されます。この保存はデータへの影響は一切ありません。

（ここから、本書に付属している演習ファイルの「9 章開始時」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

「見積作成管理 .fmp12」ファイルを開いて「z_ 見積管理」レイアウトでレイアウトモードに切り替えてください。

9.2.1 ［レイアウトの管理］ダイアログ

ファイルに作成されているレイアウトを一覧で確認、管理するためには、**［レイアウトの管理］ダイアログ**を使用します。次のいずれかの方法でダイアログを表示します。

方法 1：メニュー
［ファイル］–［管理］–［レイアウト ...］メニューを選択します。

方法 2：ショートカット
ショートカットキー［Ctrl］＋［Shift］＋［L］（Windows）あるいは［command］＋［Shift］＋［L］（macOS）を使います。

方法 3：その他
レイアウトポップアップメニュー（レイアウトバーのレイアウト名をクリックすると表示されるポップアップ）で「レイアウトの管理 ...」を選択します。

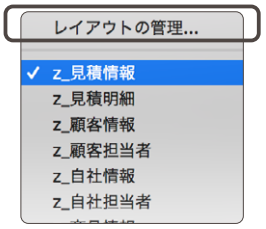


図 9-5 レイアウトポップアップメニュー

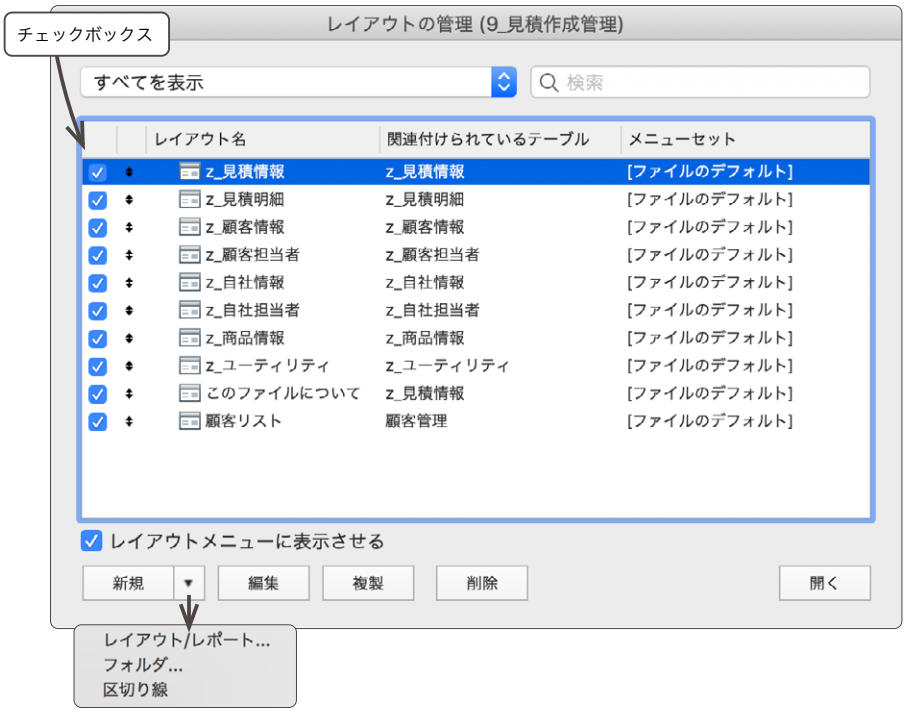


図 9-6［レイアウトの管理］ダイアログ

フォルダや区切り線も使いながら、作業しやすいようにレイアウトを整理できます。レイアウトの作成、複製、削除も可能です。

チェックボックス
レイアウト名の左にあるチェックボックスを　**オン**　にするとレイアウトポップアップメニューに表示されます。

レイアウト切り替えの方法を限定する
ユーザがレイアウトポップアップメニューで自由にレイアウトを切り替えると困る場合があります。想定している画面遷移以外のルートでは、ユーザの期待しているレコード以外が表示されるかもしれません。レイアウトポップアップメニューでのレイアウト切り替えをさせないように利用します。

レイアウトの順番
行をドラッグすることでレイアウト順を変更できます。変更した順番は**レイアウト番号**やレイアウトポップアップメニューの表示に反映されます。

レイアウト番号
Get (レイアウト番号) 関数で取得できます。レイアウトモードのステータスツールバーにブックアイコンが表示されている場合は、そこにも表示されます。

ダイアログの下には、いくつかのボタンがあります。

[新規]
新しいレイアウトの作成を開始します。他にもフォルダや区切り線を作成できます。

フォルダ
レイアウトをフォルダに入れて整理できます。ポップアップメニューで「フォルダ ...」を選択してフォルダ名を付与すると作成されます。

区切り線
レイアウトとレイアウトの間に、区切り線を入れて整理できます。ポップアップメニューで「区切り線」を選択すると挿入されます。レイアウトポップアップメニューにも表示されます。

[編集]
選択しているレイアウトの**［レイアウト設定］ダイアログ**を表示します。

[複製]
選択しているレイアウトを複製して新しいレイアウトを作成します。ウィザードは表示されません。

[削除]
選択しているレイアウトを削除します。

[開く]
選択しているレイアウトを開きます。新しいウインドウで表示されます。

9.3 レイアウトの作成

カスタム App にとってレイアウト（画面）は、ブラウズモードやプレビューモードでユーザにデータを表示する「接点（インターフェース）」です。

作成できるレイアウト数に制限はありません。入力、表示、検索、文書や宛先ラベルの印刷、グラフ、Web ページ表示など、用途に応じて作成、デザインを行います。

「見積作成管理 .fmp12」の［レイアウトの管理］ダイアログで確認できるレイアウトのうち、「このファイルについて」レイアウトを除いた 8 個は、ファイルにテーブルを作成したときに一緒に作成されたものです。

テーブルの作成とレイアウト
新しいテーブルが作成されると、自動で次の作業が実行されます。

・リレーションシップグラフに TO をつくる
・上記の TO を使ったレイアウトをつくる

本章からは設計に従って作成した TOG を使ったレイアウトを作成します。

9.3.1 新規レイアウト作成のウイザード

新しいレイアウトの作成にはウィザードを使用します。次のいずれかの方法で新規レイアウト作成のウィザードが開始します。

方法 1：メニュー
［レイアウト］–［新規レイアウト / レポート］メニューを選択します。

方法 2：ショートカット
ショートカットキー［Ctrl］+［N］（Windows）あるいは［command］+［N］（macOS）を使います。

方法 3：レイアウトバー
ステータスツールバーの［新規レイアウト / レポート］ボタンをクリックします。

方法 4：［レイアウトの管理］ダイアログ
［新規］をクリックしてポップアップメニューで「レイアウト / レポート ...」を選択します。

どの方法でも［新規レイアウト / レポート］ダイアログが表示されて、ウィザードが開始します。3 つのことを指定して進めます（「レポート」では、さらに詳細なことを指定します）。

・どこからレコードを表示するか TO を設定する
・レイアウト名をつける
・レイアウトタイプを選択する（コンピュータ／タッチデバイス／プリンタから選択）

ウィザート完了後、選択したタイプにあわせて基本構成（テーマ、パート、フィールドなど）が用意されたレイアウトが作成、表示されます。その後、各種ツールでレイアウトを仕上げます。

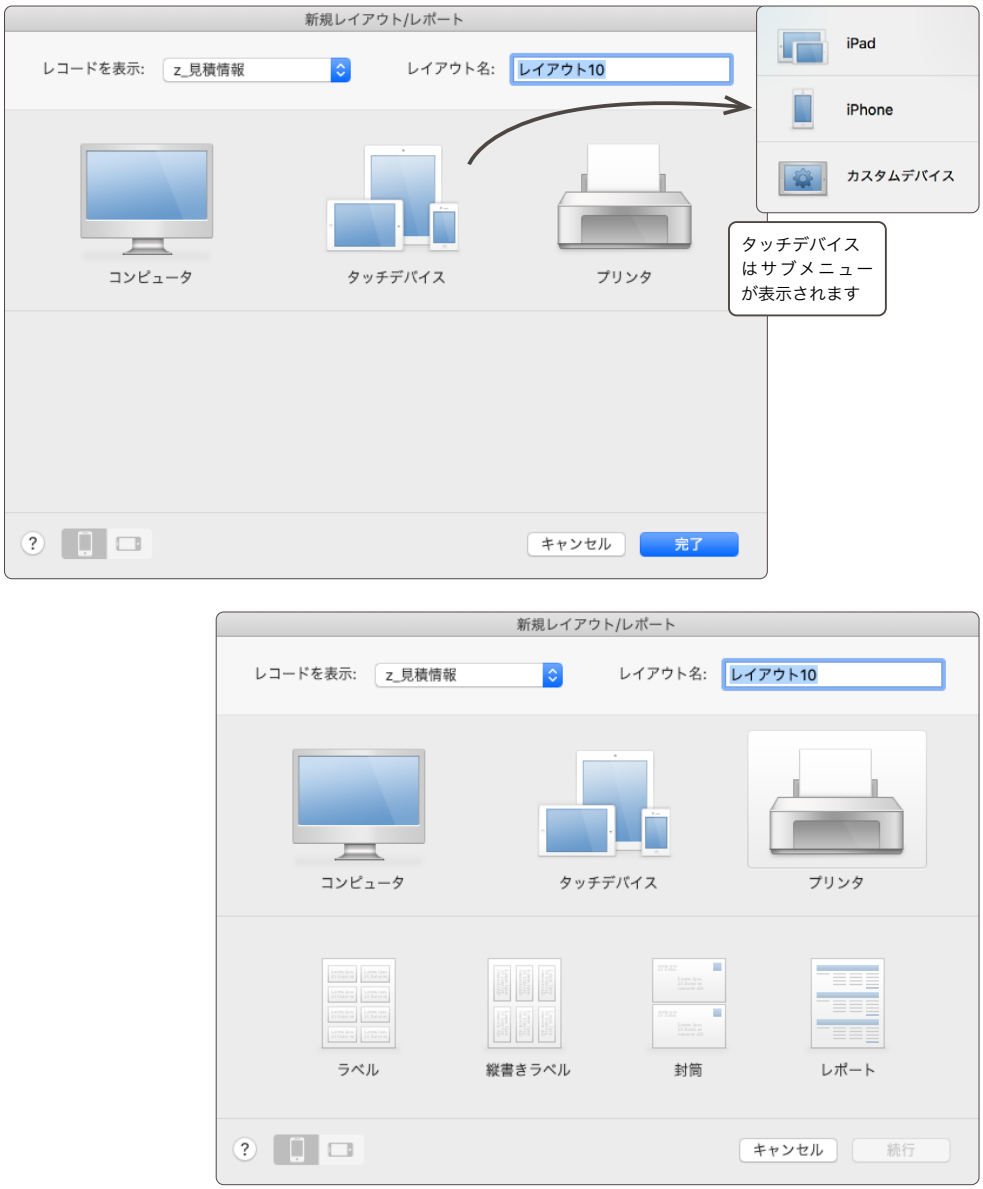


図 9-7 レイアウト作成ウィザードの開始ダイアログ

9.4 フィールドの配置

レイアウトは、指定した TO と、その TO に関連するデータの入出力を行う場所です。データを表示するには、適切なフィールドを配置する必要があります。配置方法は大きく分けて 2 種類用意されています。

・レイアウトの左パネルの「フィールド」タブ
・ステータスツールバーの「フィールド／フィールドコントロール」ツール

※ [挿入]–[フィールド ...] メニューも用意されていますが、上記の 2 種類をお勧めします。

9.4.1 「フィールド」タブ（左パネル）

フィールドの一覧が表示されています。コンテンツエリアにドラッグ & ドロップすると配置されます。一度に 1 つだけではなく複数のフィールドを配置できます。

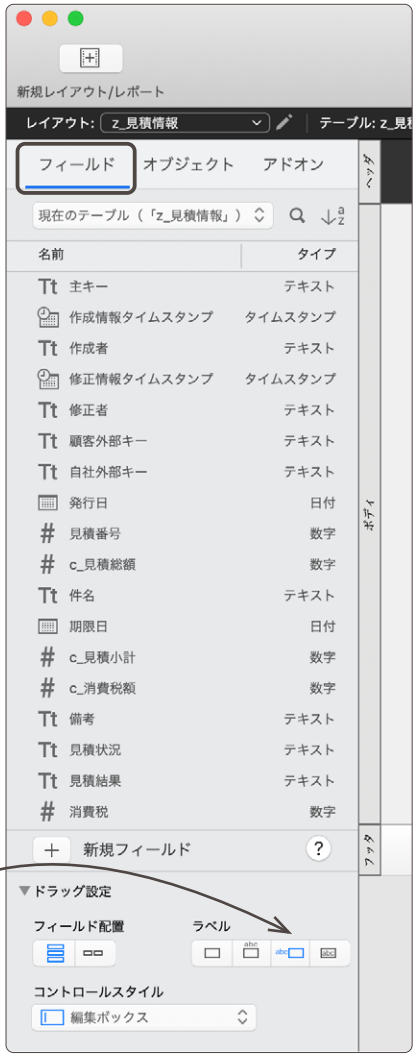


図 9-9「フィールド」タブ

[ドラッグ] 設定

次の設定をあらかじめ選択できます。

・複数のフィールドの並び方（縦／横）
・フィールドラベルの位置（ラベルなしも選択可能）
・フィールドのコントロールタイプ

配置後、インスペクタ（右パネル）で任意のスタイルを設定したり外観の変更ができます。

フィールドラベル

レイアウト上でフィールドの内容がわかりやすいように配置する文字列です。配置時はフィールド名で作成されます。



図 9-8 フィールドラベル

配置されるラベルはテキストオブジェクトなので、書き換え可能です。

プレースホルダテキストにフィールド名を使う

「ラベル」の一番右を選択すると、フィールド名がプレースホルダテキストとして設定されます。

その他、フィールドの新規作成、定義の変更（タイプ、オプションなど）、削除が可能です。

9.4.2 ステータスツールバーのツール

ステータスツールバーからのフィールド配置はレイアウトツール内の「フィールドツール」と「フィールド / コントロールツール」の 2 種類のツールで可能です。

フィールドツール

フィールドツールを長押しして、そのままドラッグします。その後、配置したいフィールド名を選択してラベルの有無を指定します。配置されたフィールドのコントロールスタイルの変更などはインスペクタを使います。



図 9-10 ステータスツールバーの「フィールドツール」

フィールド／コントロールツール

フィールド / コントールツールを長押しして、表示されたポップアップメニューからコントロールスタイルを選択します。そのままレイアウト上で任意の四角形を描いてドラッグします。その後、配置したいフィールド名を選択してラベルの有無を指定します。配置されたフィールドのコントロールスタイルの変更などはインスペクタを使います。



図 9-11 ステータスツールバーの「フィールド／コントロールツール」

フィールドのコントロールスタイル

フィールドのレイアウト上の表示種類です。7 種類です。

・編集ボックス
・ドロップダウンリスト
・ポップアップメニュー
・チェックボックスセット
・ラジオボタンセット
・ドロップダウンカレンダー
・マスク付き編集ボックス

コントロールスタイルは、配置時だけではなく配置後も変更が可能です。インスペクタの「データ」タブで変更します。



図 9-12 インスペクタのコントロールスタイル設定

「見積作成管理 .fmp12」には、9 個のレイアウトがあります。これまでの演習では、テーブル作成時の TO とレイアウトを利用してきました。本章からは設計に従って定義された TOG を使います。

レイアウトを作成してフィールドを配置します。まず、演習ファイルの TO と TOG を確認します。

「9 章開始時」フォルダ

演習 9-1　演習ファイルの TOG を確認する

- **使用ファイル：9_ 演習ファイル .fmp12**

(1)　［データベースの管理］ダイアログの「リレーションシップ」タブを表示します。

(2)　5 つの TOG（TO が 1 つだけのものもあります）があります。次ページ図も参照してください。

演習ここまで

5 つの TOG は第 8 章で考えたメニューと同じです。これを使用してレイアウトを作成していきます。

・見積管理
・商品情報管理
・顧客情報管理
・自社情報管理
・ログインのためのアカウント管理（自社担当者管理）

テーブルを作成したときにできた TO は残して TOG を改めて作成しました。同じようにテーブルを作成したときにできたレイアウトはそのまま残して新しいレイアウトを作成します。

これまでの演習で利用していた TO は、リレーションシップグラフの下の方に配置して、今後の TO 選択時に紛らわしいので名前に「z_」をつけておきました。

TO 選択のためのポップアップメニューは TO 名でソートされるので、間違って選択しないようにするには命名ルールをこのように工夫します。

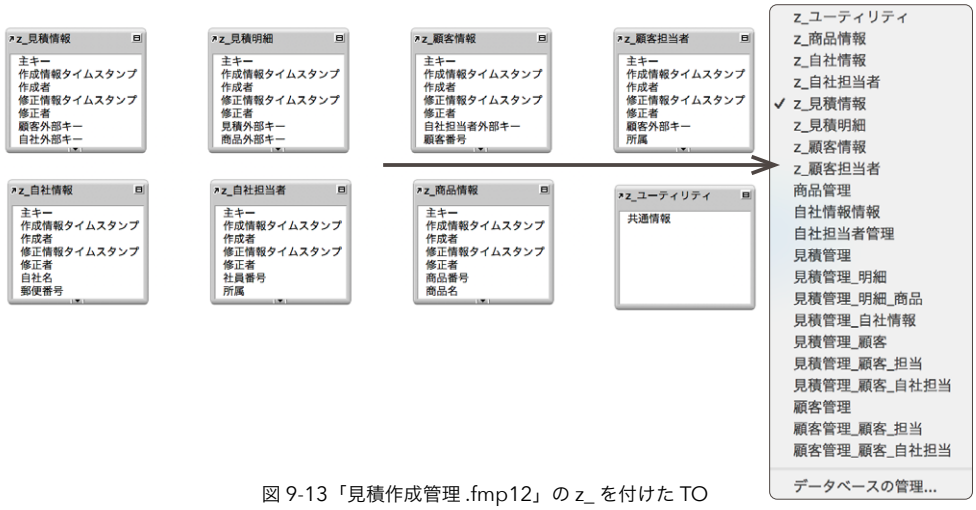


図 9-13「見積作成管理 .fmp12」の z_ を付けた TO

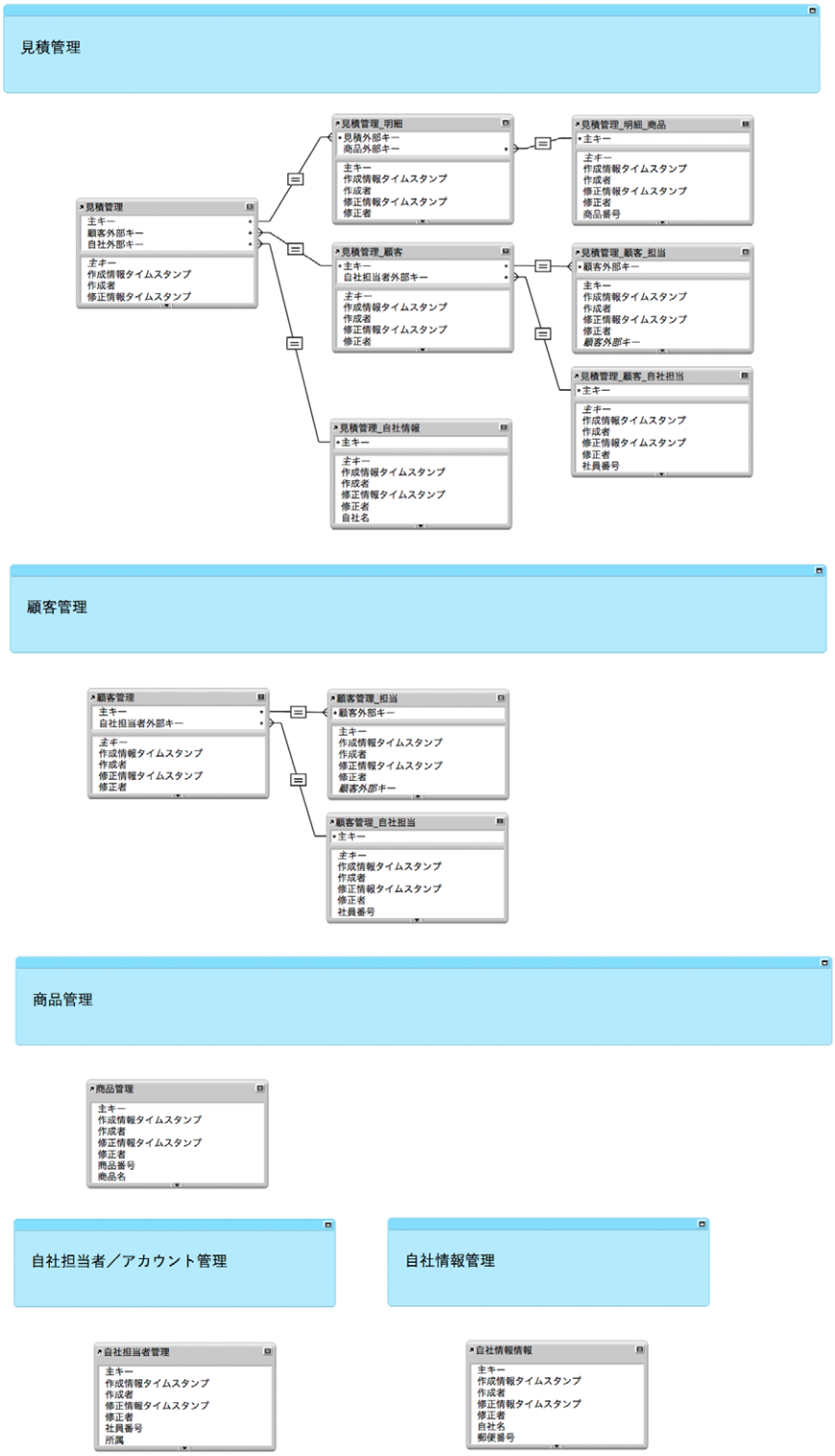


図 9-14「見積作成管理 .fmp12」の TOG

第 8 章の画面リストから、まず「顧客一覧」画面となる、パソコン用の「顧客リスト」レイアウトを作成します。

	画面名	レイアウト種類	パソコン	モバイル
1	見積登録・作業画面	フォーム形式	○	×
2	見積検索画面	フォーム形式	○	×
3	見積一覧画面	リスト形式	○	×
4	見積書（帳票）（印刷／ PDF）	フォーム形式	○	×
5	顧客登録・作業画面	フォーム形式	○	×
6	顧客検索画面	フォーム形式	○	×
7	**顧客一覧画面　→　顧客リスト**	**リスト形式**	**○**	**×**
8	商品登録・作業画面	フォーム形式	○	×
9	商品検索画面	フォーム形式	○	×
10	商品一覧画面	リスト形式	○	×
11	自社担当者・アカウント情報登録・作業画面	フォーム形式 （マスタ / 詳細）	○	×
12	自社情報管理画面	フォーム形式	○	×
13	ホーム画面	フォーム形式	○	×
14	モバイル　見積登録・作業画面	フォーム形式	×	○
15	モバイル　見積検索画面	フォーム形式	×	○
16	モバイル　見積一覧画面	リスト形式	×	○
17	モバイル　顧客登録・作業画面	フォーム形式	×	○
18	モバイル　顧客検索画面	フォーム形式	×	○
19	モバイル　顧客一覧画面	リスト形式	×	○
20	モバイル　商品閲覧画面	フォーム形式 （マスタ / 詳細）	×	○
21	モバイル　ホーム画面	フォーム形式	×	○

※ 本章では、次回からモバイルの画面は表から省略します。

演習 9-2　顧客リストレイアウト作成

- **使用ファイル：9_ 見積作成管理** .fmp12

(1)　［キャンセル］をクリックして［データベースの管理］ダイアログを閉じます。

(2)　レイアウトモードでステータスツールバーの［新規レイアウト / レポート］ボタンをクリックします。

(3)　［新規レイアウト / レポート］ダイアログが表示されます。
次の設定をします。
・［レコードを表示］は　「顧客管理」TO を選択
・［レイアウト名］は　「顧客リスト」と入力
・［コンピュータ］は　［リスト］を選択



図 9-15 ウィザードの設定

(4) ［完了］をクリックします。

(5) 「顧客リスト」レイアウトができました。なにも配置されていません。

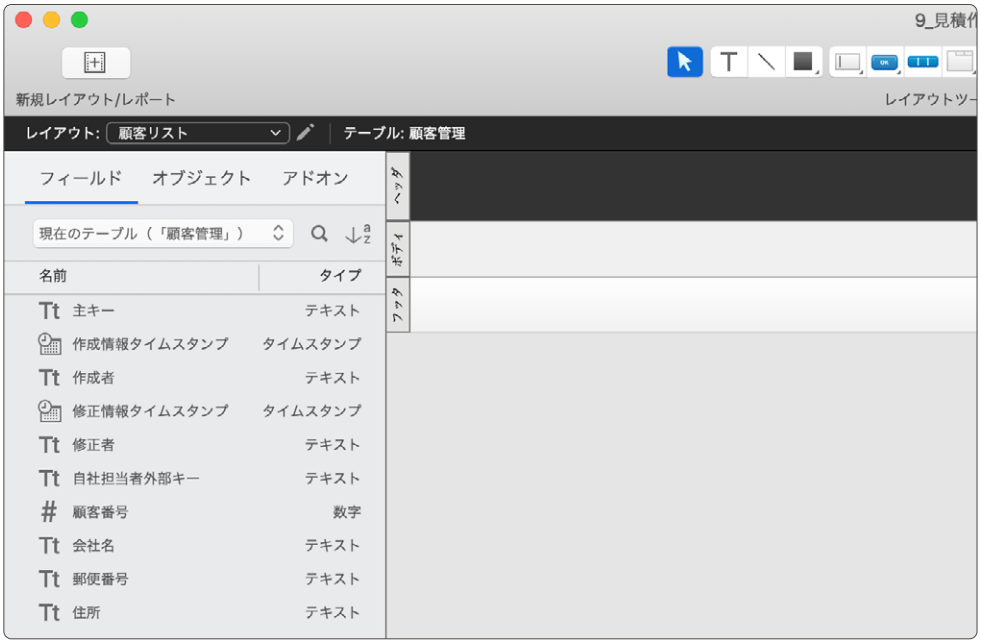


図 9-16 リストレイアウト

演習ここまで

左パネルの「フィールド」タブからフィールドを選択して配置します。

演習 9-3　顧客リストレイアウトにフィールドを配置する 1

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　左パネルを表示して「フィールド」タブをクリックし、配置するフィールドを選択します。

a.　［Ctrl］キー（Windows）［command］キー（macOS）を押し続けながら

・顧客番号
・会社名
・郵便番号
・住所

をクリックします。
（現在のテーブルは「顧客管理」、フィールドは位置は「横」、ラベル位置は「上」、コントロールスタイルは「編集ボックス」を選択します）

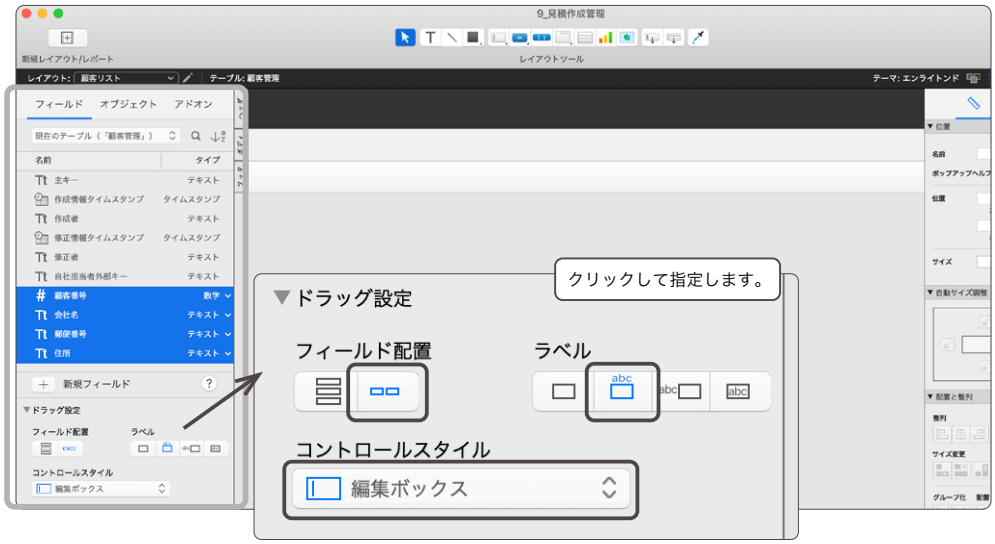


図 9-17「フィールド」タブを使用したフィールド配置

(2)　フィールドをドラッグして「ボディ」パートへ配置します。

(3)　フィールドが配置されました。



図 9-18 横並びで、ラベルが上に配置された

(4)　位置の微調整をします。
ブラウズモードとレイアウトモードを切り替えつつ、フィールド幅を調整します。

・「郵便番号」フィールドはこのように長い必要はありません。
・リスト形式では、フィールドはボディの上下の中心に配置されていると見やすくなります。



図 9-19 微調整の例

演習ここまで

フィールドの複数選択
[Ctrl] キー（Windows）または［command］キー（macOS）を押しながらクリックすると、複数選択が可能です。活用してください。

別の方法で関連フィールドを配置します。ステータスツールバーのフィールド／コントロールツールを使います。自社担当者のフィールドを配置します。

演習 9-4　顧客リストレイアウトにフィールドを配置する 2

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　ステータスツールバーの「フィールド／コントロール」ツールをクリックして、レイアウトの住所の右側でドラッグして四角形を描きます。

(2)　［フィールド指定］ダイアログが表示されます。
a.　「顧客管理 _ 自社担当」TO を選択します。
b.　「氏名」フィールドを指定します。

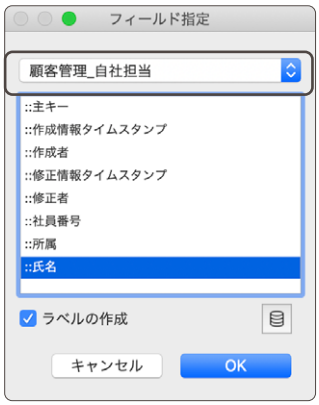


図 9-20［フィールド指定］ダイアログ

(3)　［OK］をクリックするとフィールドが配置されます。

(4)　ラベル位置を微調整します。

・ラベルを他と同じヘッダに配置
・ラベルを「自社担当」に変更

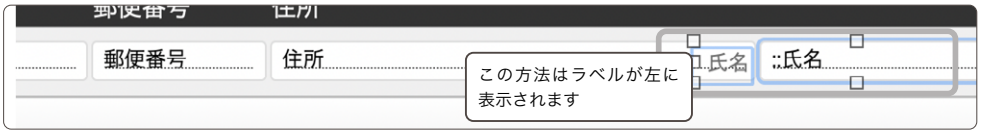


図 9-21　ラベルを調整

(5)　「自社担当」ラベルを選択しインスペクタの「スタイル」タブで、スタイルを割り当てます。
そのままだと、ヘッダの色と合いません。他のラベルと同じスタイルで揃えます。

・スタイル名は「ヘッダテキスト」

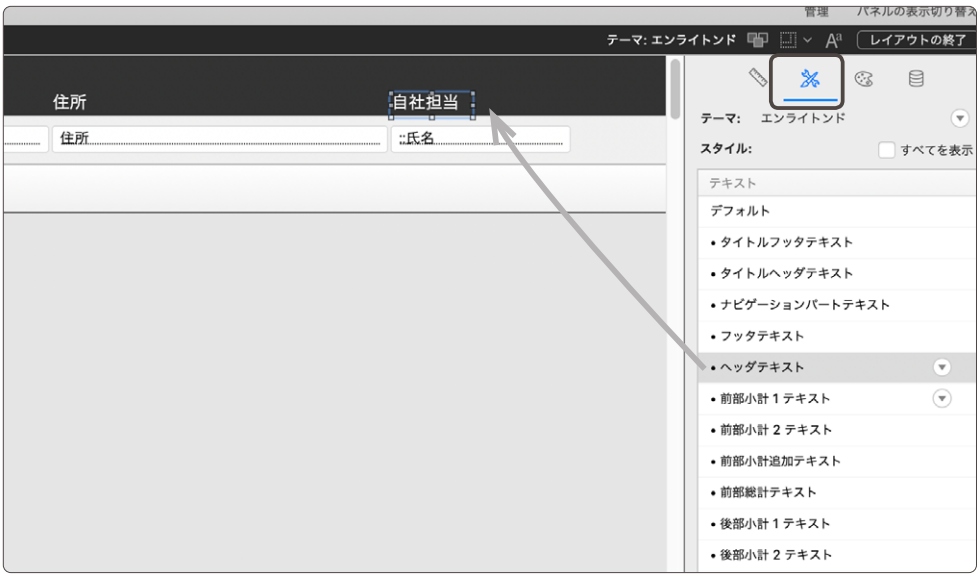


図 9-22　スタイルを設定

(6)　レイアウトを保存してブラウズモードに戻ります。「顧客リスト」レイアウトが完成しました。



図 9-23「顧客リスト」画面

演習ここまで

（ここから、本書に付属している演習ファイルの「演習 9_5」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

9.5　レイアウトの設定

「顧客管理」TO のリスト形式レイアウト「顧客リスト」の設定を確認します。レイアウトモードで次のいずれかの方法でダイアログを表示します。

方法 1：メニュー

［レイアウト］–［レイアウト設定 ...］メニューを選択します。

方法 2：その他

・レイアウトバーのレイアウト名の右にある鉛筆アイコンをクリックします。
・［レイアウトの管理］ダイアログで［編集］ボタンをクリックします。

9.5.1　［レイアウト設定］ダイアログ

このダイアログでは、いくつかの設定が可能です。新規レイアウト作成時にウィザードで設定した情報を変更できます。初めは「一般」タブが表示されます。

「一般」タブ

レイアウト全体についての設定が可能です。

・レイアウト名
・レイアウトポップアップメニューの表示／非表示（［レイアウトの管理］ダイアログのチェックと同じ）
・レイアウトの TO 名
・レコードの自動保存の有効／無効
・レコードの表示について
・レイアウトに切り替えた時にインストールするメニューセットの指定
・クイック検索の有効／無効

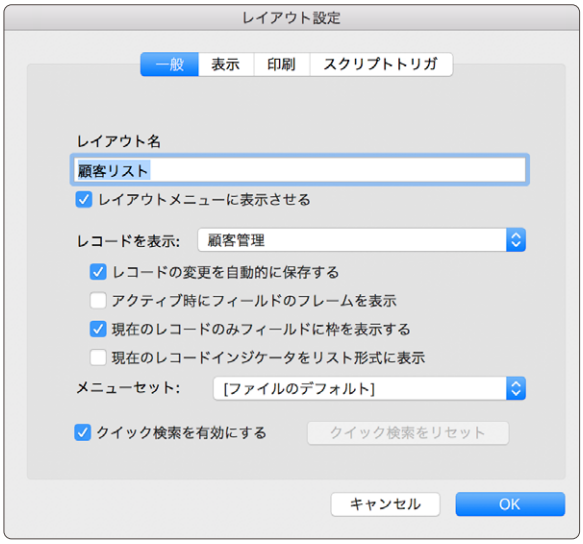


図 9-24［レイアウト設定］ダイアログ（「一般」タブ）

［レコードの変更を自動的に保存する］
FileMaker Pro は、作業中の『あるタイミング』でレコードを保存します。「レコードの確定」と呼びます。レコードは、次の操作をおこなった時点で確定されます。
・他のレコードへ移動する
・現在のフィールドの外側をクリックする
・［Enter］キーを押下する
・他のレイアウトやモードへ切り替える
・ウインドウを閉じる
チェックが　**オフ**　の場合、上記のタイミングで、レコードへの変更を保存するかどうかを尋ねる確認ダイアログが表示されるようになります。



図 9-25 確認ダイアログ

［アクティブ時にフィールドのフレームを表示］と［現在のレコードのみフィールドに枠を表示する］
任意のレコードのフィールドをクリックしてカーソルがそこに入っている状態にすると、
・**「現在のレコード」**
が移動します。そして、次のように呼びます。
・現在カーソルがあるフィールドを**「アクティブなフィールド」**
・アクティブなフィールドがあるレコードを**「アクティブなレコード」**
チェックが　**オン**　の場合、アクティブなフィールドの枠や、現在のレコードのフィールドの枠が表示されます（アクティブでないフィールドの枠は表示されません）。

［現在のレコードインジケータをリスト形式に表示］
チェックが　**オン**　で、リスト形式の場合に、左端に『現在のレコード』がわかるようにインジケータが表示されます。



図 9-26 レコードインジケータ

［クイック検索を有効にする］
チェック　**オン**　の場合、レイアウトでクイック検索が可能です。インスペクタの「データ」タブの［動作］でレイアウト上のフィールドを対象とするかどうか指定します。なお、**［クイック検索をリセット］ボタン**はレイアウト上のすべてのフィールドがクイック検索の対象にリセットされます。

「表示」タブ

レイアウトのデフォルトの表示形式と、レイアウトバーの「表示形式の切り替え」の有効／無効について設定が可能です。

ウィザードを使用して作成したレイアウトは、ウィザードで指定した表示形式以外には切り替えできないように設定されています。他の形式に切り替えできるようにするには、このタブで設定が必要です。

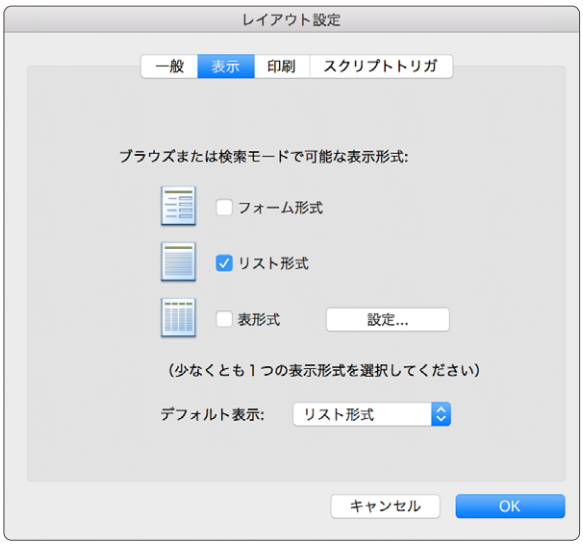


図 9-27［レイアウト設定］ダイアログ（「一般」タブ）

「印刷」タブ

印刷に関する設定を行います。

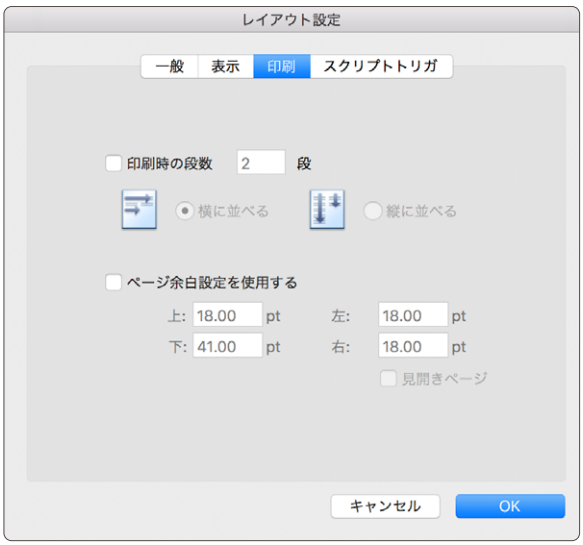


図 9-28［レイアウト設定］ダイアログ（「印刷」タブ）

「スクリプト」トリガタブ

レイアウトを切り替える前や後などのタイミングでスクリプトを実行するよう設定します（スクリプトトリガについては後述します）。



図 9-29［レイアウト設定］ダイアログ（「スクリプトトリガ」タブ）

9.5.2 レイアウト設定の変更

レイアウト設定で指定した TO を変更する場合は注意してください。コンテキストが変わると、レイアウト上に配置したフィールドが表示できなくなったり、スクリプトトリガで指定したスクリプトが動作しなくなることなどがあります。

やってみよう

「顧客リスト」レイアウトの［レイアウト設定］ダイアログを表示して設定を確認しましょう。

演習 9-4 で作成した「顧客リスト」レイアウトをこれまでのレイアウトと区別してわかりやすくするために［レイアウトの管理］ダイアログで**「見積作成管理 .fmp12」のレイアウトを整理**します。

「演習 9_5」フォルダ

演習 9-5　レイアウトを整理する

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　［レイアウトの管理］ダイアログを表示します。
（［ファイル］–［管理］–［レイアウト ...］メニューを選択します）

(2)　新しい区切り線を挿入します。
a.　［レイアウトの管理］ダイアログの一番上の行（「z_ 見積情報」）をクリックします。
b.　「新規」ボタンの　▼　をクリックして「区切り線」を選択します。
c.　区切り線が挿入されます。
d.　一番上に区切り線を移動します。

(3)　新しいレイアウトフォルダを作成します。
a.　「新規」ボタンの　▼　をクリックして「フォルダ ...」を選択します。
b.　［フォルダ名：］に「開発者用」と名前を入力します。
c.　［OK］をクリックするとフォルダが作成されます。

(4)　「z_」のつくレイアウトを「開発者用」フォルダにドラッグして移動します。

(5)　区切り線の上に「このファイルについて」レイアウトを移動します。（次図を参照）

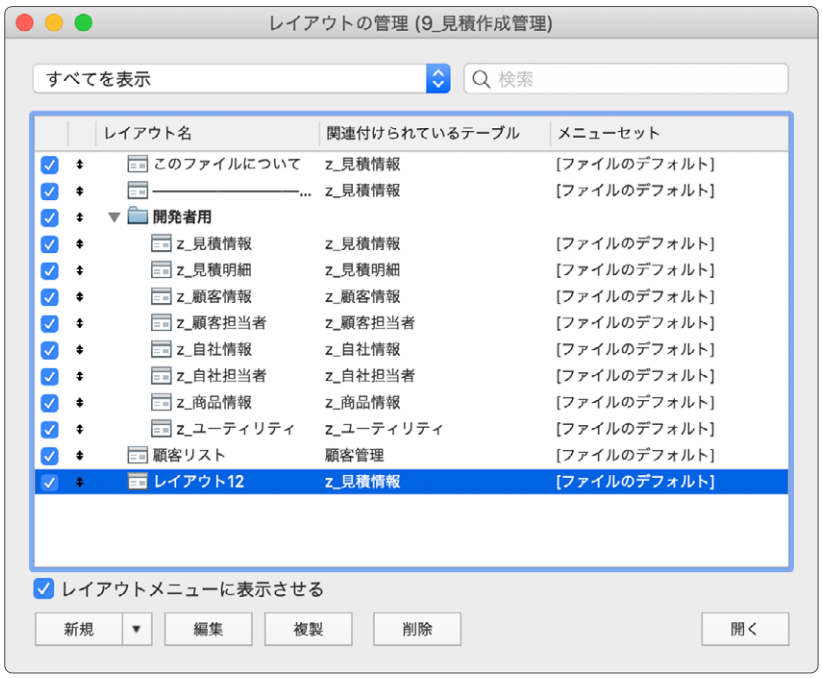


図 9-30　レイアウトの整理後

演習ここまで

9.6 レイアウトパート

レイアウトを作成すると「ヘッダ」「ボディ」「フッタ」の 3 つのパートが自動で作成されます。11 のパートが用意されていますが、必要に応じて使い分けることができます。



図 9-31 パート

拡大と縮小

レイアウトの表示を拡大／縮小する機能です。

・[表示]–[拡大表示] メニューを選択で拡大
・[表示]–[縮小表示] メニューを選択で縮小
・[表示]–[実際のサイズ] メニューを選択で 100%

9.6.1 パートの種類

特徴を確認します。

上部ナビゲーション

画面の一番上に表示され（順番は変えられません）、レイアウトのナビゲーションに使用するボタンなどを配置するパートとして使用されます。1 つのレイアウトに 1 つだけ設定できます。このパートは、ブラウズモードで画面をスクロールしても常に画面の上部位置に固定し表示されます。表示の拡大と縮小の影響を受けません。印刷もされません。

タイトルヘッダ

上部ナビゲーションの下に表示されます（順番は変えられません）。印刷では先頭ページに一度だけ表示されます。1 つのレイアウトに 1 つだけ設定できます。
「タイトルヘッダ」と「ヘッダ」が両方が設定されているレイアウトを印刷すると、最初のページは「タイトルヘッダ」が印刷され「ヘッダ」は印刷されません。2 ページ目以降は「タイトルヘッダ」は印刷されず「ヘッダ」が印刷されます。

「タイトルヘッダ」に配置されているフィールドは、ブラウズモードではアクティブなレコードのデータが表示されますが、印刷時には対象レコードの最初のレコードのデータが印刷されます。

ヘッダ
「上部ナビゲーション」、「タイトルヘッダ」の下に表示されます（順番は変えられません）。1 つのレイアウトに 1 つだけ設定できます。
印刷時は、前述の「タイトルヘッダ」を参照してください。主に、画面名や表形式でフィールド名を配置するためのパートとして使用されます。
「ヘッダ」に配置されているフィールドは、ブラウズモードではアクティブなレコードのデータが表示されます。印刷時は各ページに印刷されるレコードの最初のレコードデータが印刷されます。

ボディ
対象レコードの各レコードごとのデータを表示するパートとして使用されます。1 つのレイアウトに 1 つだけ設定できます。

フッタ
「下部ナビゲーション」、「タイトルフッタ」の上に表示されます（順番は変えられません）。
印刷時は「ヘッダ」を参照してください。1 つのレイアウトに 1 つだけ設定できます。
「フッタ」に配置されているフィールドは、ブラウズモードではアクティブなレコードのデータが表示されます。印刷時は各ページに印刷されるレコードの最後のレコードデータが印刷されます。

タイトルフッタ
「下部ナビゲーション」の上に表示されます（順番は変えられません）。「タイトルヘッダ」と同じような特徴があります。
「タイトルフッタ」にフィールドが配置されていると、ブラウズモードではアクティブなレコードのデータが表示されますが、印刷時には最初のページに印刷されるレコードの最後のレコードのデータが印刷されます。

下部ナビゲーション
画面の一番下に表示されます（順番は変えられません）。「上部ナビゲーション」と同じ特徴があります。

次からは、レイアウト上に集計結果のデータを表示するためのパートです。

総計（前部または後部）
このパートに集計タイプのフィールドを配置すると、すべての対象レコードの集計値が表示されます。1 つのレイアウトに前部総計と後部総計を 1 つずつ設定できます。
- 前部総計の場合、表示位置はヘッダの下で小計パートとボディを下に配置できます。
- 後部総計の場合、表示位置はフッタの上で小計パートとボディを上に配置できます。

小計（前部または後部）
このパートに集計タイプのフィールドを配置すると、パートの定義で設定されているフィールドでソートされたレコードの各グループの集計値（小計）が表示されます。1 つのレイアウトに複数の小計パートを設定できますが、パートの定義で設定できるのは 1 レイアウトごとに 1 フィールド 1 回だけです。

「小計」パートは「ボディ」パートの前または後に表示されます。

9.6.2 パートの作成、変更、削除

必要なパートを任意に作成、削除、変更をおこなうことができます。

パートの作成

［パート設定］ダイアログ、あるいはステータスツールバーのレイアウトツールのパートツールアイコンからおこないます。



図 9-32 パートツールアイコン

パートの変更

「ヘッダ」を、「上部ナビゲーション」パートにするなどパートの種類を変更することが可能です。**［パートの定義］ダイアログ**からおこないます。パートの種類を変更後、パート上に配置されているレイアウトオブジェクトの表示などについて確認をおこなってください。

パートの削除

［パート設定］ダイアログ、あるいはパートラベルをクリックして［Backspace］キーか［Delete］キーを押します。パートを削除すると、パート上に配置されていたレイアウトオブジェクトも削除されるので慎重に作業をおこなってください。

9.6.3 ［パート設定］ダイアログ

レイアウトにパートを追加／変更／削除するためのダイアログです。次の操作で表示できます。

方法 1：メニュー

［レイアウト］–［パート設定 ...］メニューを選択します。

方法 2：その他

レイアウトの任意の位置で右クリックしてショートカットメニューから［パート設定 ...］を選択します。

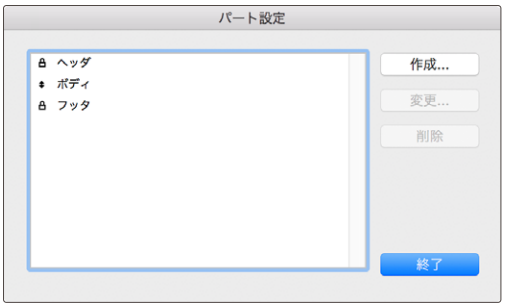


図 9-33［パート設定］ダイアログ

9.6.4 ［パートの定義］ダイアログ

パートごとの設定は、［パート設定］ダイアログを使います。

方法 1

［パート設定］ダイアログの［変更 ...］ボタンをクリックします。

方法 2

パートラベルで右クリックしショートカットメニューから［パートの定義 ...］を選択します。

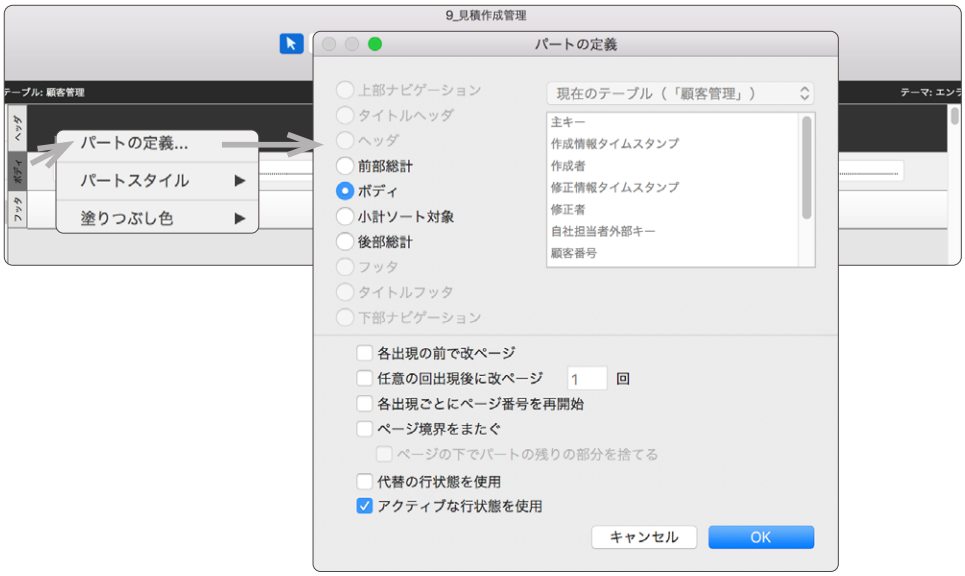


図 9-34 ショートカットメニューと［パートの定義］ダイアログ

9.7 レイアウトエリア

ブラウズモードでデータが表示されるコンテンツエリアとなるのが「**レイアウトエリア**」です。

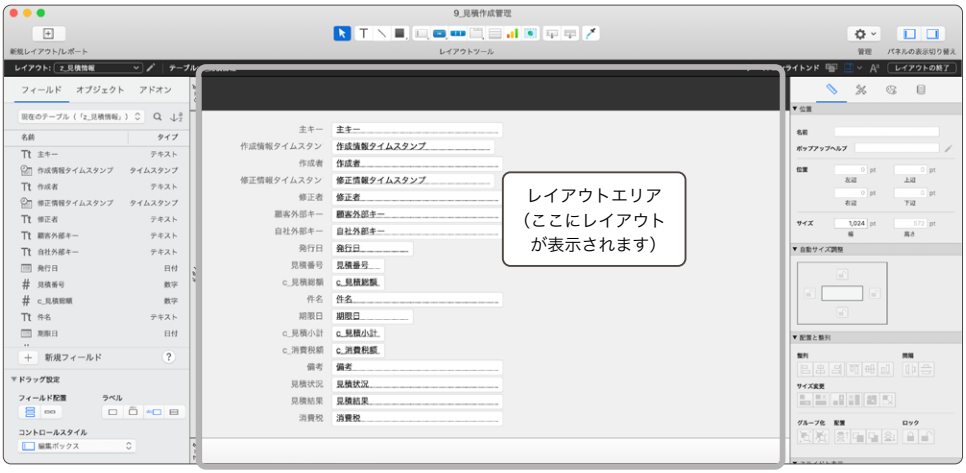


図 9-35 レイアウトエリア

9.7.1 レイアウトエリアのサイズ

レイアウトエリアのサイズは、インスペクタの「位置」タブの［サイズ］に表示されています。この数値を変更することで任意に設定できます。



図 9-36 インスペクタの位置タブ

・幅は、全パートに共通して設定（パートごとに別の幅を設定することはできない）
・高さは、パートごとに設定（レイアウトエリアの高さは全パートの高さの合計）

ブラウズモードでは、ウインドウ内のコンテンツエリアよりもレイアウトエリアが大きい場合、スクロールバーが表示されます。小さい場合は余白が表示されますが「アンカー」の設定によりコンテンツエリア一杯にレイアウトが表示されるような設定ができます。

インスペクタで数値を指定する以外に、パートの境界線などをドラッグして任意の大きさに変更することが可能です。

パートのサイズをドラッグで変更：高さ
パートの境界線を上下にドラッグして高さを変更可能です。パートになにかレイアウトオブジェクトが配置されていると、それ以上は縮まりません。しかし、

・[Alt] キー（Windows）あるいは［Option］キー（macOS）を押しながらパートの境界線のドラッグ

この操作でレイアウトオブジェクトとは関係なく上下サイズを変更できます。その後、どちらかのパートにレイアウトオブジェクトが配置されるように調整します。

パートのサイズをドラッグで変更：幅
右のレイアウトエリアの境界線を左右にドラッグしてサイズを変更できます。配置されているレイアウトオブジェクトは関係ありません。

やってみよう
「z_ 見積情報」レイアウトで、上記のドラッグをやってみましょう。

9.7.2 エリア外のレイアウトオブジェクト

レイアウトエリアの「外」にレイアウトオブジェクトを配置できます。その場合、ブラウズモードのフォーム形式、リスト形式ではコンテンツエリアに表示されません。レイアウトエリアよりもウインドウサイズを大きくしても同じです。なお表形式では表示されます。

他にもいくつかの特徴があります。タブ順を設定したり、オブジェクト名を付加してスクリプトで利用したりできます。また、レイアウト上のフィールド名を取得する関数（FieldNames 関数）の結果にレイアウトエリア外に配置されているフィールドも含まれます。

機能の作成時に役立てることもできます。例えば、レイアウト上に表示されていないフィールドをクイック検索の対象にしたい場合、そのレイアウトのエリア外にフィールドを配置します。

9.7.3 画面ステンシル

レイアウトエリアのサイズは想定するウインドウサイズと合わせます。サイズが大きいと、ユーザは常に縦横のスクロール操作をする必要があるかもしれません。それは面倒です。レイアウトエリアのサイズを簡単に設定できるガイドが「画面ステンシル」です。

主要なパソコンやモバイルデバイスの画面サイズのガイドを表示する機能です。いくつかのサイズが用意されています。さらに「カスタムサイズ」を登録することが可能です。



図 9-37 レイアウトバーの画面ステンシル表示／表示切り替え

9.7.4 レイアウトオブジェクトの識別バッジ

レイアウトモードで［表示]–[オブジェクト]–[ボタン］メニューを選択すると、ボタンオブジェクトの右下に小さい水色の四角が表示されます。これは「識別バッジ（アイコン)」です。

バッジの上にマウスを持っていくと、その設定内容がポップアップヘルプで表示されます。この他にも条件付き書式、ポップアップヘルプなどが設定されていることが一目で解るようにいくつか用意されています。またクイック検索の対象かどうかもわかるようになっています。

これらのバッジを表示するには、レイアウトモードで、[表示]–[オブジェクト］メニューで表示するバッジのタイプを選択します。主なバッジと機能は次表のとおりです。

オブジェクトの識別バッジの例

バッジ	説明
	条件付き書式が設定されています
	スクリプトトリガが設定されています
	緑色の虫眼鏡バッジ クイック検索の対象になるフィールドであることを示しています。
	黄色の虫眼鏡バッジ クイック検索の対象ですが非保存の計算式や関連先のフィールドで検索時間が長くかかる可能性があることを示しています。
	ポップアップヘルプが設定されています
	ボタンが設定されています
	ポップオーバーボタンが設定されています。
	条件によりオブジェクトを隠す機能が設定されています。
	プレースホルダテキストが設定されています。
	灰色の虫眼鏡バッジ レイアウト設定でクイック検索が無効にされていることを表しています。 ※インスペクタで個別のフィールドに「クイック検索に含めない」と設定したときは虫眼鏡バッジは表示されません

9.8 ステータスツールバーとレイアウトバー

フィールドをはじめとしてレイアウトに置かれるものは様々ですが、これらを総称して「**レイアウトオブジェクト**」と呼びます。レイアウトツールを使用して設定します。

9.8.1 ステータスツールバー

ステータスツールバーには、これまでにも使用してきたツールが配置されています。これらを使ってレイアウトオブジェクトを作成します。

ツールのボタンにはいくつかのルールがあります。レイアウトモードに切り替えると「選択ツール（矢印アイコン）」が選択されています。



図 9-38「選択ツール（矢印アイコン）」

他のツールを 1 回クリックすると、そのボタンがグレーになります。



図 9-39 グレーになっている状態

レイアウトオブジェクトを作成するとグレーが解除されて選択ツールに戻ります。

ツールのロック

連続してツールを使用する場合は、ボタンをダブルクリックします。ボタンが青になりレイアウトオブジェクトを追加しても解除されなくなります。解除には、［Esc］キーや［Enter］キーを押します。または、他のツールをクリックしても解除されます。



図 9-40 ボタンバーツールをロックした例

常にロックする

［環境設定］ダイアログの「レイアウト」タブで［レイアウトのツールをロックする］のチェックを **オン** にするとクリックしたツールが常にロックされた状態になります。

レイアウトオブジェクトの選択

レイアウトの作業は、常にレイアウトオブジェクトを選択して設定します。選択は 1 つ、連続した複数、任意の複数、などです。この選択を素早くできるようになると作業の効率が上がります。ぜひ覚えてください。

ステータスツールバーで選択ツール（矢印アイコン）が選択されていると、レイアウトオブジェクトの選択が可能です。

1 つの選択
普通にクリックするとひとつずつの選択になります。

連続した複数の選択
レイアウト上でドラッグして選択したいレイアウトオブジェクトを囲むように四角形を描くと、その中に少しでも含まれるレイアウトオブジェクトが選択されます。ドラッグ時に完全に囲まれたものだけを選択するには、次の操作です。

・［Ctrl］キー（Windows）あるいは［command］キー（macOS）を押しながらドラッグ

任意の複数の選択
［Shift］キーを押しながら、レイアウトオブジェクトをクリックする、そのレイアウトオブジェクトが選択されます。選択されているレイアウトオブジェクトを、もう一度クリックすると解除されます。

レイアウトオブジェクトの複製
いくつか方法があります。

・レイアウトオブジェクトを選択後、［Ctrl］キー（Windows）あるいは［Option］キー（macOS）を押しながらドラッグ
・レイアウトオブジェクトを選択後、[Ctrl]＋[D] キー（Windows）あるいは [command]＋[D] キー（macOS）を押す

複製されたものがフィールドの場合、［フィールド指定］ダイアログが表示されます。

9.8.2 レイアウトバー

レイアウトバーにもツールが配置されています。

・テーマの変更
・画面ステンシルの表示／非表示
・書式設定バーの表示／非表示

ステータスツールバーやレイアウトバーは、本章の後半以降で徐々に解説します。これまでの演習ファイルを閉じて次の演習ファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 9_6」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

9.9　インスペクタ（右パネル）

レイアウトオブジェクトの設定に使用します。次の順番に 4 つのタブにツールが用意されています。

・位置
・スタイル
・外観
・データ



「位置」タブ
位置に関する設定を行うタブです。

[位置]
レイアウトオブジェクトの位置や名前付与などに使用します。[ポップアップヘルプ] の設定も行います。
[名前] でレイアウトオブジェクトに名前を付与します。オブジェクト名は［オブジェクトへ移動］スクリプトステップなどで使用できます。

[自動サイズ調整]
「アンカー」を使って、ウインドウサイズの変更に合わせてレイアウトオブジェクトのサイズを自動変更したり、水平または垂直に移動する設定ができます。上下左右の位置の調整が錠のマークに固定されるイメージです。

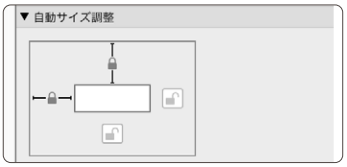


図 9-41 左と上位置を調整する場合の設定

[配置と整列]
複数のレイアウトオブジェクトの整列、サイズを合わせる、重ね順、グループ、ロックに使用します。整列のショートカットを覚えると便利です。
・[Ctrl]＋[Alt] ＋上下左右矢印キー（Windows）
・[command]＋[Option]] ＋上下左右矢印キー（macOS）

[グループ化]
複数のレイアウトオブジェクトのグループ化と解除が可能です。グループ化後に設定したボタン設定などは、グループを解除すると消去されるため注意します。

[スライドと表示]
印刷時のレイアウトやフィールドの空き部分を狭める設定を行います。[オブジェクトの表示] では印刷時にレイアウトオブジェクトを非表示にする設定ができます。設定の結果はプレビューモードで確認できます。

［グリッド］
レイアウトにグリッドを表示できます。間隔やステップは数値を入力すると変更できます。

自動サイズ調整について演習をします。［自動サイズ調整］は、1 つのレイアウトオブジェクトだけではなく、その周りにどのような影響があるかを確認して設定する必要があります。

「演習 9_6」フォルダ

演習 9-6　自動サイズ調整

- **使用ファイル：9_ 見積作成管理 .fmp12**

（1）「自動サイズ調整確認」レイアウト（1 件目のレコード）でレイアウトモードに切り替えます。

（2）右パネル（インスペクタ）を表示します。

（3）［自動サイズ調整］を確認します。はじめは、左と上の錠が選択されています。
・「商品説明」フィールド　　左と上
・「商品画像」フィールド　　左と上

（4）ブラウズモードに切り替えて、ウインドウサイズをレイアウトエリアより大きくしたり小さくしたります。配置されているレイアウトオブジェクトは何も変化がありません。

（5）レイアウトモードに切り替えます。［自動サイズ調整］の設定を変更します。
・「商品説明」フィールド　**左と上と右**
・「商品画像」フィールド　左と上（そのまま）

（6）レイアウトを保存してブラウズモードに切り替えてウインドウサイズをレイアウトエリアより大きくしたり小さくしたります。

「商品説明」フィールドだけが、ウインドウサイズの横幅に合わせて横幅が広がります。

（7）レイアウトモードに切り替えます。［自動サイズ調整］の設定を変更します。
・「商品説明」フィールド　**左と上と右と下**
・「商品画像」フィールド　左と上（そのまま）

（8）レイアウトを保存してブラウズモードに切り替えてウインドウサイズをレイアウトエリアより大きくしたり小さくしたります。
「商品説明」フィールドだけが、ウインドウサイズに合わせて縦横に広がります。「商品画像」フィールドにも重なって拡大します。

（9）レイアウトモードに切り替えます。［自動サイズ調整］の設定を変更します。
・「商品説明」フィールド　**左と上と右**
・「商品画像」フィールド　**左と上と右と下**

（10）レイアウトを保存してブラウズモードに切り替えてウインドウサイズをレイアウトエリアより大きくしたり小さくしたります。
「商品画像」フィールドがウインドウサイズに合わせて広がり、重なりません。

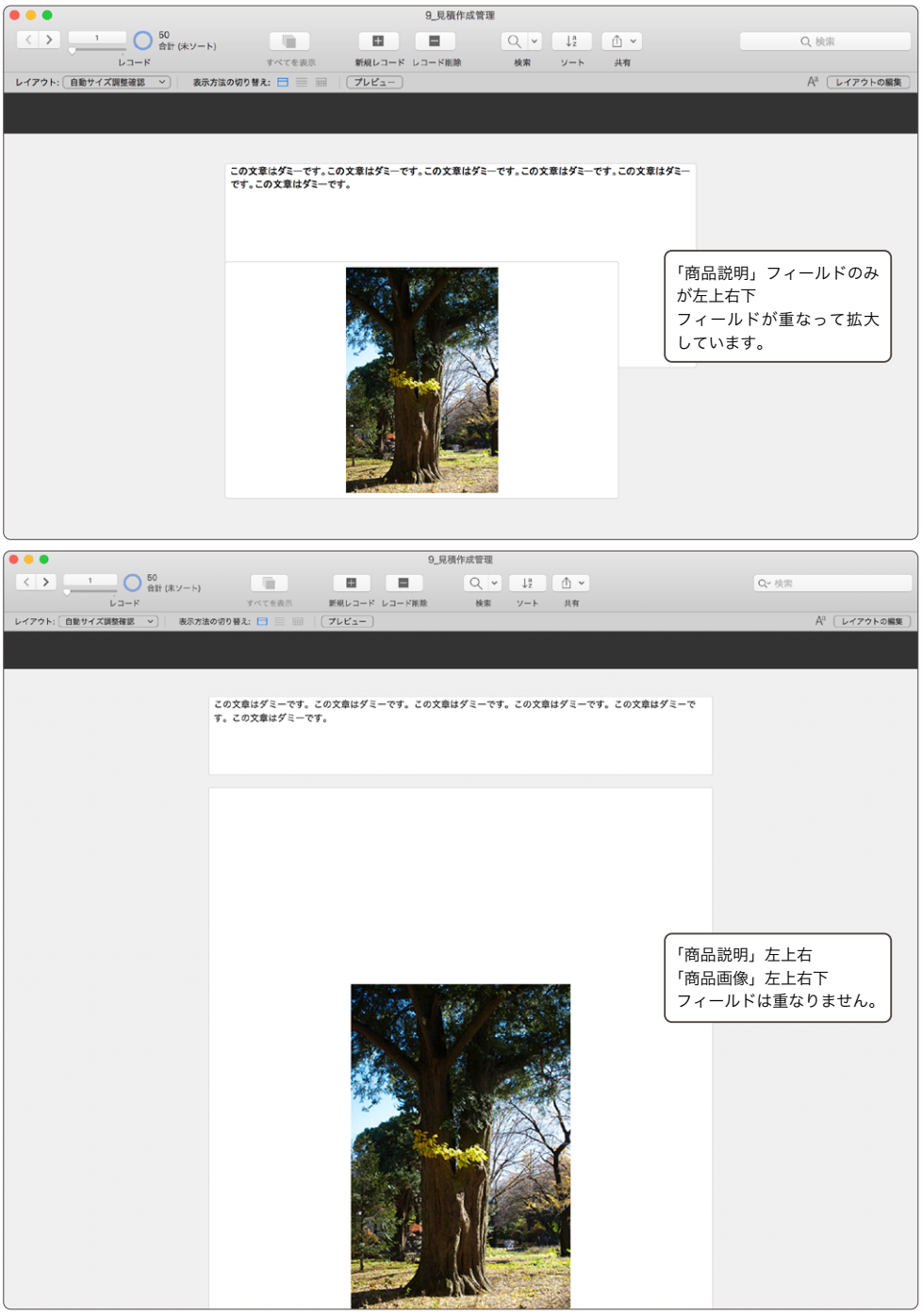


図 9-42 フィールドが重なるとき、重ならないときの例

演習ここまで

「スタイル」タブ
スタイルを設定するタブです。

「外観」タブ
見た目の設定を行うタブです。テーマやスタイルについても設定が可能です。

「データ」タブ
レイアウトオブジェクトに表示されるデータについて設定するタブです。



［フィールド］
配置するフィールドの指定、コントロールスタイルの指定などを設定します。

［動作］
条件による非表示や、レイアウト上でフィールドへ入力できるかどうかなどを設定します。

［次のオブジェクトへの移動に使用するキー］
レイアウトオブジェクトへの移動（タブ順）に使用できるキーボードのキーを設定します。

［インプットメソッドの設定］［タッチキーボードタイプ］
日本語入力のモード、iPad や iPhone で表示するキーボードを設定します。

［データの書式設定］
オブジェクトタイプフィールドの演習で使用しましたが、フィールドの表示に関する書式を設定します。

インスペクタの設定は、本章の後半以降で徐々に解説します。

9.10　オブジェクトタブ（左パネル）

左パネルのタブの 1 つです。

レイアウトには、たくさんのレイアウトオブジェクトが配置されます。レイアウトオブジェクトを重ねて配置することもあります。「オブジェクト」タブは、レイアウトに配置されているすべてのレイアウトオブジェクトを上から下に重なり順で表示します。

例えば、レイアウトの作業時に、

・目的のレイアウトオブジェクトを探す
・作業をおこなうために目的のレイアウトオブジェクトを選択する
・目的のレイアウトオブジェクトの作業のために他のレイアウトオブジェクトを非表示にする

などが簡単に可能です。一度に 1 つだけではなく複数のレイアウトオブジェクトを選択することも可能です。

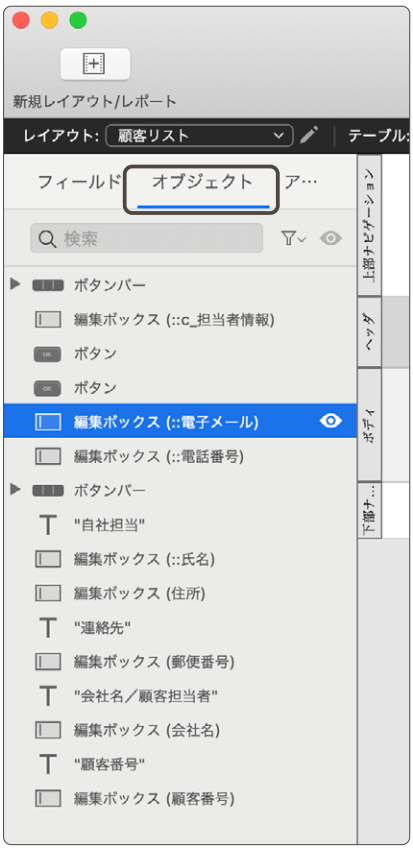


図 9-43「オブジェクト」タブ

目のアイコン

レイアウトオブジェクトをクリックすると、目のアイコンが表示されます。これは、選択されていることを表しています。クリックすると非表示になります。

注意点

非表示は、レイアウトモードでのみ機能します。FileMaker Pro を終了して再度開くと表示が解除されます。

重ね順の変更

行をドラッグして順を変更すると、レイアウトエリア上のオブジェクトの重ね順が変わります。

グループ化と「オブジェクト」タブ

フィールドなどのレイアウトオブジェクトにボタンを設定したり、レイアウトオブジェクトをグループ化すると、右図のような表示になります。

▶をクリックすると、グループ化された内容が表示されます。そしてグループ化されたレイアウトオブジェクトを選択できます。

また、ロックされているレイアウトオブジェクトのロック解除をしなくても一部の設定が可能です。



図 9-44　グループ化例

ショートカットメニューで出来ること

「オブジェクト」タブでは、グループの解除をしなくても、グループに含まれるレイアウトオブジェクトを選択して各種の設定が可能です。

右クリックで表示されるショートカットメニューを使用すると便利です（右図を参照）。

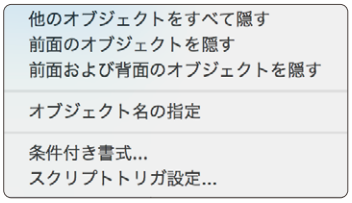


図 9-45 右クリックで表示されるメニュー

オブジェクトのタイプでフィルタ

タイプごとに表示をフィルタすることが可能です。

例えば、ボタンだけを表示するには、フィルタアイコンをクリックすると表示されるタイプの中から「ボタン」を選択すると表示がフィルタされます。

レイアウトオブジェクトの検索

上部の検索ボックスに任意の文言を入力して、その結果に合致するレイアウトオブジェクトを表示することが可能です。

例えば、「顧客」という文言を入力すると、右図のようにボタン名や、フィールド名、レイアウトオブジェクト名など、その文言が含まれるものだけが表示されます。



図 9-46 フィルタ結果例と検索結果例

フィルタと検索を掛け合わせる

検索後のフィルタ、フィルタ後に検索、と掛け合わせて表示内容を絞り込むことが可能です。

非表示と検索とフィルタ

非表示のレイアウトオブジェクトは、フィルタや検索後に表示されますが、グレー表示になります。

これまでの演習ファイルを閉じて次の演習ファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 9_7 から 9」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

「見積一覧」画面となる、「見積リスト」レイアウトの作業をおこないます。演習 9-2 で作成した「顧客リスト」レイアウトを加工しました。演習用の「見積リスト」レイアウトも用意しました。今後、「見積作成管理システム」では、以下を前提とします。

・パソコン用画面は「1024×768」のサイズを想定する
・各レイアウトに「上部ナビゲーション」パートを「ヘッダ」パートの上に配置する
・テーマは「エンライトンド」を変更し「エンライトンド演習」として利用

これまでに作成されたのは下表の薄く色をつけた 2 つのレイアウトです。

	画面名	レイアウト種類	パソコン	モバイル
1	見積登録・作業画面	フォーム形式	○	×
2	見積検索画面	フォーム形式	○	×
3	**見積一覧画面　→　見積リスト**	**リスト形式**	○	**×**
4	見積書（帳票）（印刷／ PDF）	フォーム形式	○	×
5	顧客登録・作業画面	フォーム形式	○	×
6	顧客検索画面	フォーム形式	○	×
7	**顧客一覧画面　→　顧客リスト**	**リスト形式**	○	**×**
8	商品登録・作業画面	フォーム形式	○	×
9	商品検索画面	フォーム形式	○	×
10	商品一覧画面	リスト形式	○	×
11	自社担当者・アカウント情報登録・作業画面	フォーム形式（マスタ / 詳細）	○	×
12	自社情報管理画面	フォーム形式	○	×
13	ホーム画面	フォーム形式	○	×
（モバイル画面は省略）				

演習ファイルの「顧客リスト」レイアウトを確認します。

「演習 9_7 から 9」フォルダ

演習 9-7　リスト表示レイアウトの作業：見本の「顧客リスト」を確認

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「顧客リスト」レイアウトでレイアウトモードに切り替えます。
今後の画面の確認のために、いくつかの識別バッジが表示されるようにします。

a.　[表示]–[オブジェクト]–[ボタン] メニューを選択してください。
b.　[表示]–[オブジェクト]–[条件を隠す] メニューを選択してください。
c.　[表示]–[オブジェクト]–[条件付き書式] メニューを選択してください。
d.　[表示]–[オブジェクト]–[スクリプトトリガ] メニューを選択してください。
e.　レイアウトバーのステンシルの「デスクトップ 1024×768」を選択してください。
f.　レイアウトバーで画面ステンシルを　**オン**　にしてください。

(2)　レイアウトの内容を確認します。

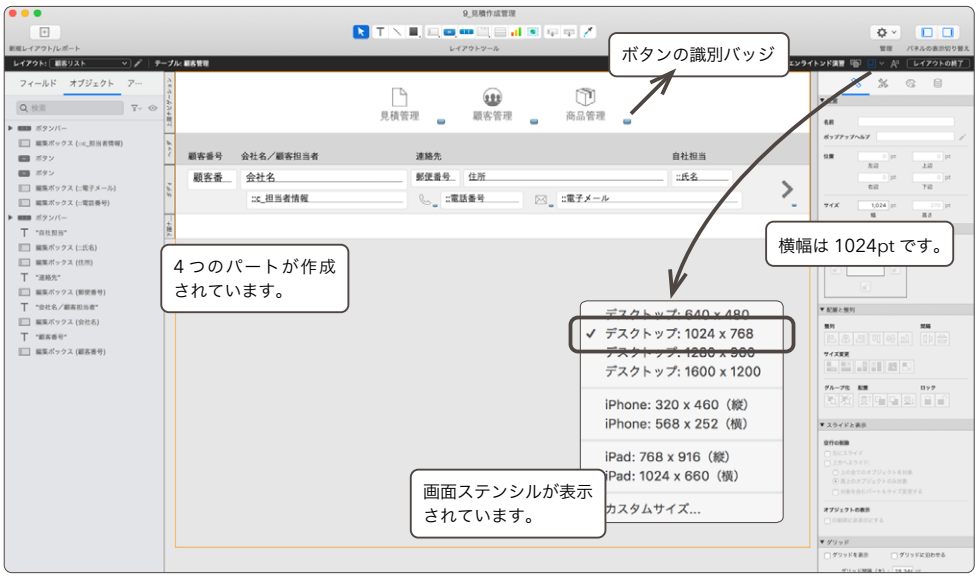


図 9-47　レイアウトモードの「顧客リスト」レイアウト

a.　画面ステンシルがオレンジの線で表示されているはずです。
b.　レイアウトエリアの横幅は「1024」pt です。「ボディ」パートのどこかをクリックしてインスペクタで確認してください。
c.　パートは 4 つ、「上部ナビゲーション」「ヘッダ」「ボディ」「下部ナビゲーション」です。
d.　「上部ナビゲーション」パートには、ボタンバーが配置されています。
e.　「ボディ」パートには、フィールドと、いくつかのボタンが配置されています。
f.　これらのボタンには、識別バッジが表示されているはずです。

(3)　確認したらブラウズモードに切り替えます。

演習ここまで

レイアウトサイズの高さとリスト形式

リスト形式の場合は、ヘッダやフッタなどのパートの高さの他に『「ボディ」パートの高さ＝画面に何件のレコードが表示されるか』を考えてレイアウトの高さを想定します。

例えば「見積作成管理システム」ではボディの高さを 80pt としています。768pt の高さにおおよそ 8 件を想定しています。

また、リスト形式の「ボディ」パートには、そのレイアウトの TO あるいは関連の TO の全てのフィールドを配置することはあまりしません。さらに詳細な情報が表示されるレイアウトへ遷移するかどうかを判断するために必要な項目だけにとどめます。

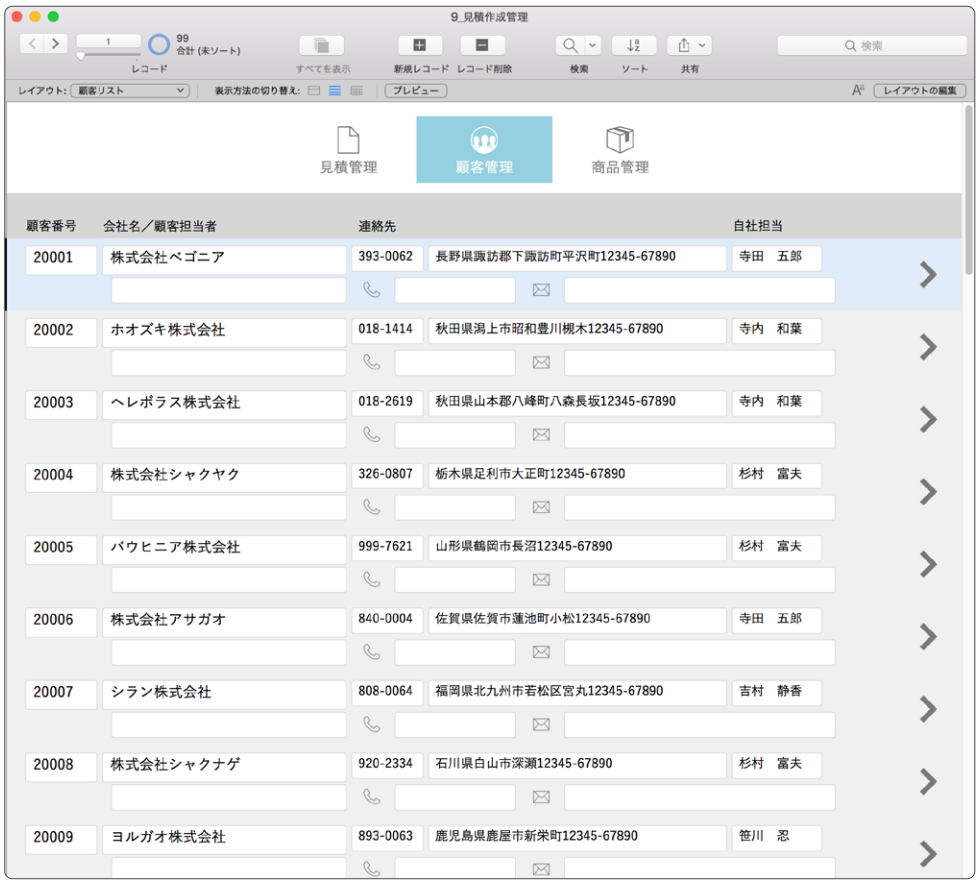


図 9-48 ブラウズモードの「顧客リスト」レイアウト

「顧客リスト」レイアウトでは、「上部ナビゲーション」パートのボタンバーは上だけ、「ボディ」パートの「>」ボタンは右だけの［自動サイズ調整］が設定されています。ブラウズモードでウインドウサイズの横幅を大きくすると、

・「上部ナビゲーション」パートのボタンは、常にウインドウの真ん中に表示
・「ボディ」パートの「>」ボタンは、常にウインドウの右側に表示

となります。

「顧客リスト」レイアウトのように**「見積リスト」レイアウトを整えます。**

演習 9-8　リスト表示レイアウトの作業：「見積リスト」の作業 1

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「見積リスト」レイアウトに切り替えて、レイアウトモードに切り替えます。

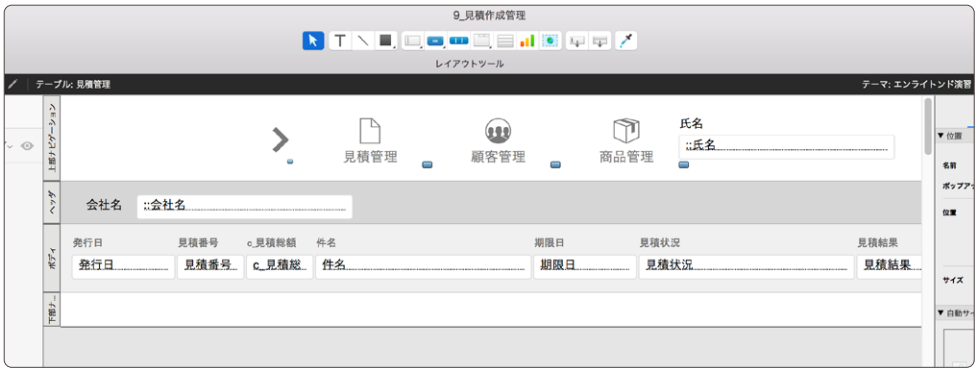


図 9-49 作業前の「見積リスト」レイアウト

(2)　内容を確認します。

a.　2 つのボタンがあります。「>」ボタンはこの位置から移動が必要です。

b.　「ボディ」パートに配置する必要があるフィールドがいくつかのパートに置かれています。

(3)　画面ステンシルを非表示にします。
（グリッドが見えやすいように）

(4)　グリッドを表示します。

a.　右パネルのインスペクタの「位置」タブの［グリッドを表示］を　**オン**　にします。

b.　グリッドの間隔をわかりやすい数値に変更します。

・［グリッド間隔（大）］　100

・［グリッド間隔（小）］　5



図 9-50 グリッドの設定

(5)　[表示]–[ダイナミックガイド] メニューが　**オン**　であることを確認します。

(6)　「ダイナミックガイド」について確認します。
a.　「上部ナビゲーション」パートに配置されている「>」ボタンをクリックしてドラッグします。
b.　ボタンを動かすと青い線のガイドが、縦横の中心、他のオブジェクトとの合致などとして都度に表示されます。これがダイナミックガイドです。

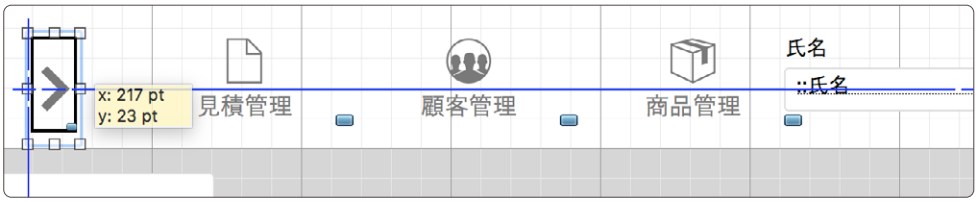


図 9-51 ダイナミックガイド

(7)　フィールドの幅や高さを適切に設定できるようにサンプルデータを表示します。
a.　[表示]–[オブジェクト]–[サンプルデータ] メニューを選択します。
b.　サンプルデータが表示されます。

(8)　フィールドを次図のように配置します。フィールドラベルは「ヘッダ」パートに置いてください（それぞれのフィールドの上になるように）。
ダイナミックガイドも利用してサイズ（サンプルデータが表示されるように）を変更したり位置を変更します。
・「見積状況」と「見積結果」フィールドは同じ幅
・「発行日」と「期限日」フィールドが同じ幅

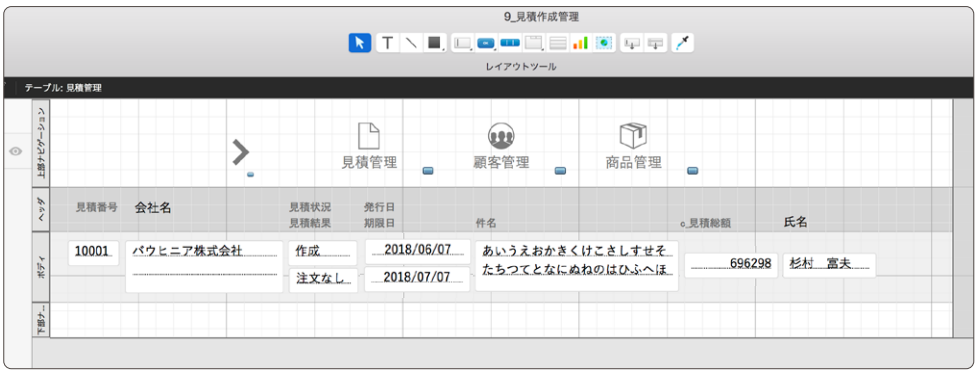


図 9-52 サイズや位置を調整

(9)　作業時間に余裕がある場合は、インスペクタの「外観」タブの［段落設定］でフィールドの行揃えをします。
・「見積番号」「発行日」「期限日」は　中央揃え
・「c_ 見積総額」は　右揃え

(10)　フィールドを整列させます。まず、フィールドを複数選択します。
「オブジェクト」タブで、［Ctrl］キー（Windows）を押しながらクリック
または
［command］キーまたは［Shift］キー（macOS）を押しながらクリック

・「見積番号」フィールド
・「見積管理 _ 顧客 :: 会社名」フィールド
・「見積状況」フィールド
・「発行日」フィールド
・「件名」フィールド
・「c_ 見積総額」フィールド
・「見積管理 _ 顧客 _ 自社担当 :: 氏名」フィールド

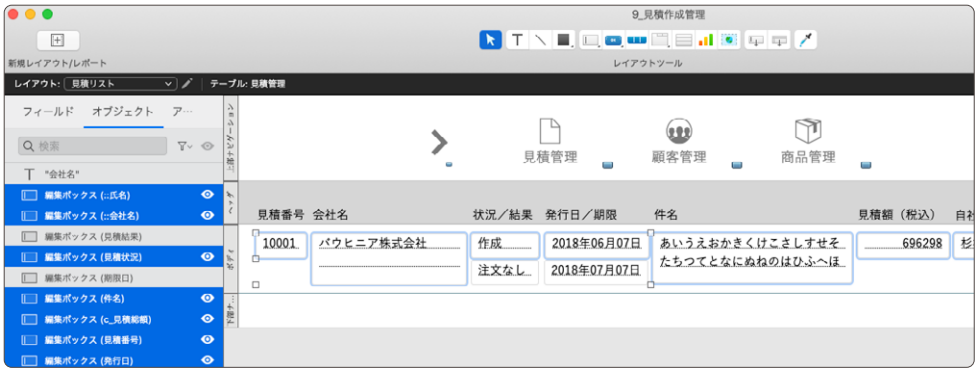


図 9-53 複数のオブジェクトを選択

a. インスペクタの「位置」タブの［配置と整列］で整列を実行します。
・上
・横に等間隔

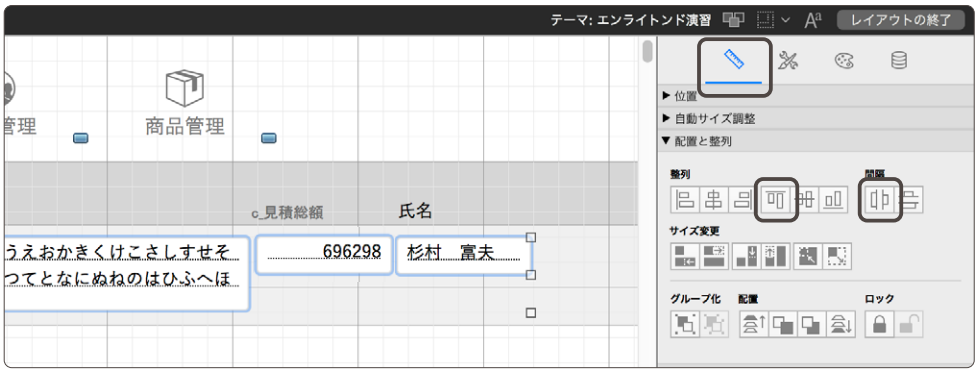


図 9-54 インスペクタの上と横に等間隔の整列

(11) 他のレイアウトオブジェクトを調整します。
a. 「見積結果」と「期限日」フィールドを上のフィールドに合わせて位置を調整します。
b. 「オブジェクト」タブで「フィールド」でフィルタして、「ボディ」パートに配置されているすべてのフィールド選択します、
c. b で選択したフィールドをパートの高さの中心に配置します。
d. 「>」ボタンを「ボディ」パートの右辺 1004pt に配置します。
e. d のボタンをパートの高さの中心に配置します。

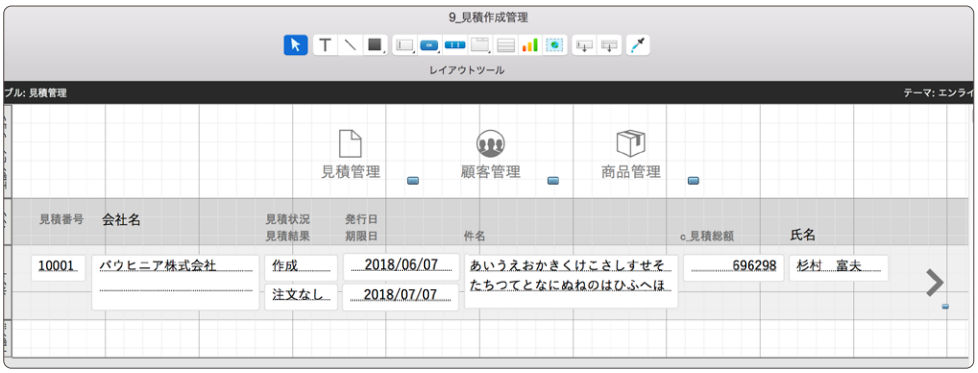


図 9-55 調整後

(12) 「ヘッダ」パートのフィールドラベルを変更します。

a. 左から次のように並べます。必要があれば、テキストオブジェクトの大きさや内容を変更したり、削除します。

・見積番号
・会社名
・状況／結果
・発行日／期限日
・件名
・見積額（税込）
・自社担当

b. フィールドラベルのスタイルを共通にします。
インスペクタの「スタイル」タブの「ヘッダテキスト」を適用します。

c. フィールドラベルを各フィールドの上に配置します。

d. インスペクタの「位置」タブの［配置と整列］で「上」の整列を実行します。

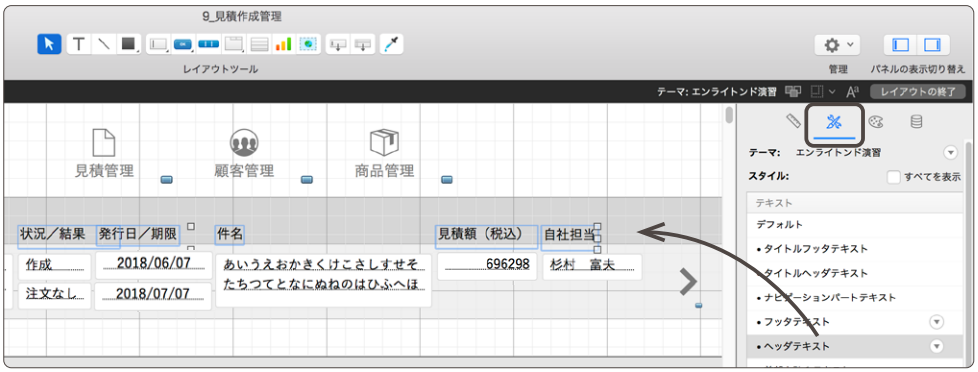


図 9-56 インスペクタのスタイル設定

(13) 「ボディ」パートの［>］ボタンのアンカーを設定します。
インスペクタの「位置」タブで［自動サイズ調整］を「右」だけにします。

(14) 「上部ナビゲーション」パートのボタンバーについて確認します。
インスペクタの「位置」タブで［自動サイズ調整］が「上」だけになっているはずです。

(15) ボタンバーのアクティブセグメントを設定します。

a. ボタンバーをダブルクリックすると、[ボタンバー設定] ダイアログが表示されます。

b. [アクティブセグメント：] のポップアップメニューを表示して「見積管理」を選択します。

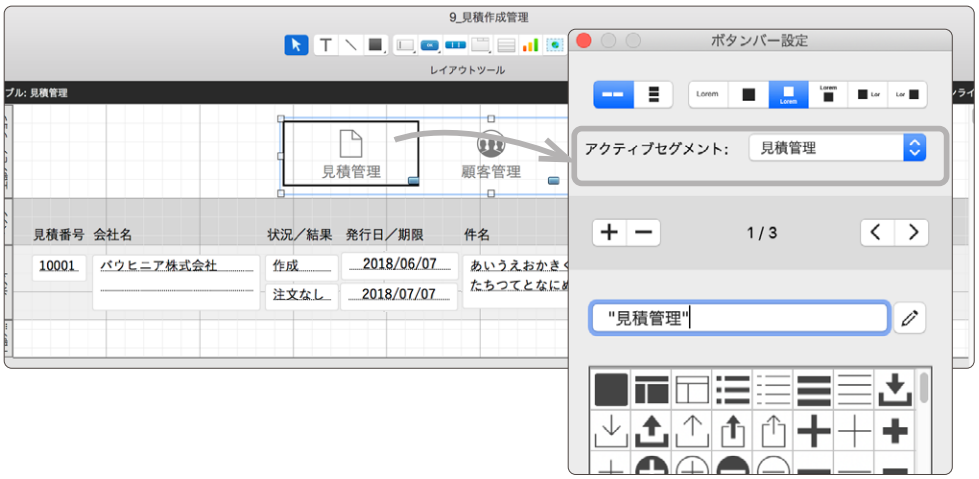


図 9-57［ボタンバー設定］ダイアログ

(16) レイアウトを保存してブラウズモードへ切り替えます。

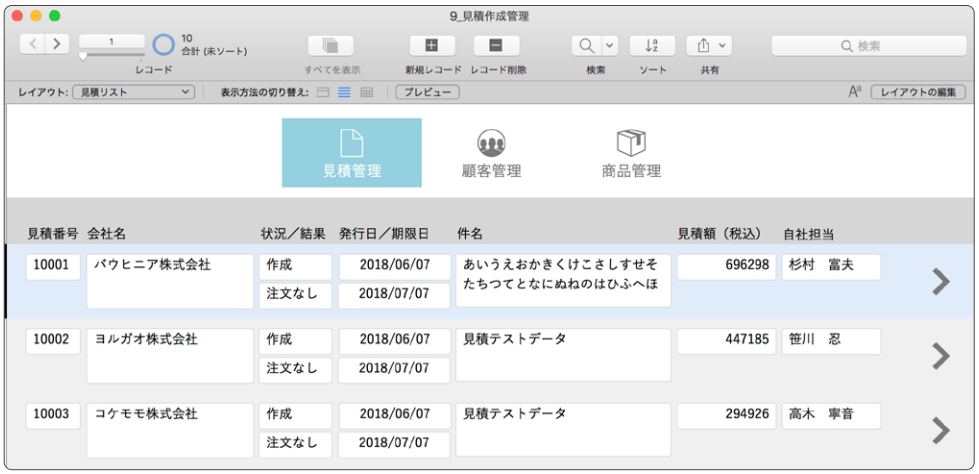


図 9-58 演習後の「見積リスト」レイアウト

演習ここまで

レイアウトの作成には、様々な作業が必要です。画面の「**モックアップ**」があると便利です。

・配置するフィールド
・各レイアウトオブジェクトの位置
・他の画面との統一感

これらのことを行き当りばったりで考えるよりも、あらかじめ設計しておくことでユーザにとって見やすく使いやすい画面を作成することが可能です。

もう少し作業が必要です。**スタイルや、フィールドの書式についても設定します。**

演習 9-9　リスト表示レイアウトの作業：「見積リスト」の作業 2

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「見積リスト」レイアウトでレイアウトモードに切り替えます。

(2)　「発行日」と「期限日」フィールドの書式を設定します。
a.　両方のフィールドを選択します。
b.　インスペクタの「データ」タブで次の設定をします。
・[データの書式設定] の [書式]「20xx 年 xx 月 xx 日」を選択

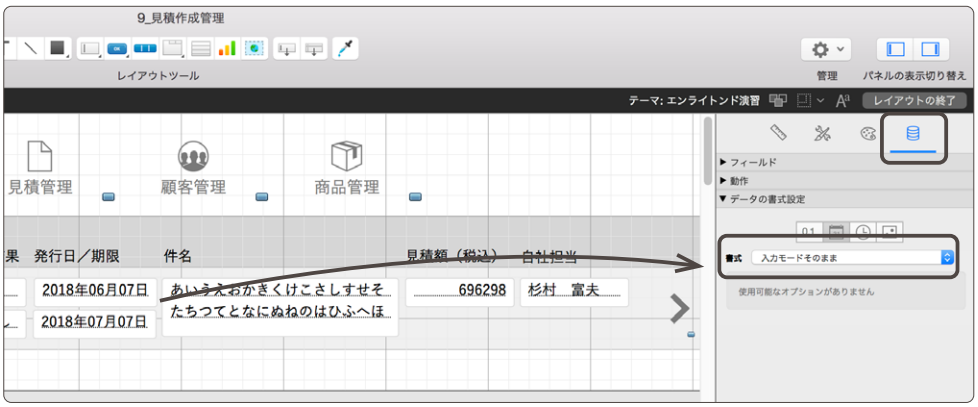


図 9-59 インスペクタの設定

(3)　「c_ 見積総額」フィールドの書式を設定します。
a.　フィールドを選択します。
b.　インスペクタの「データ」タブの [データの書式設定] で次の設定をします。
・[書式] は　「通貨」を選択
・[セパレータ] は 「3 桁区切りを使用：」を　**オン**

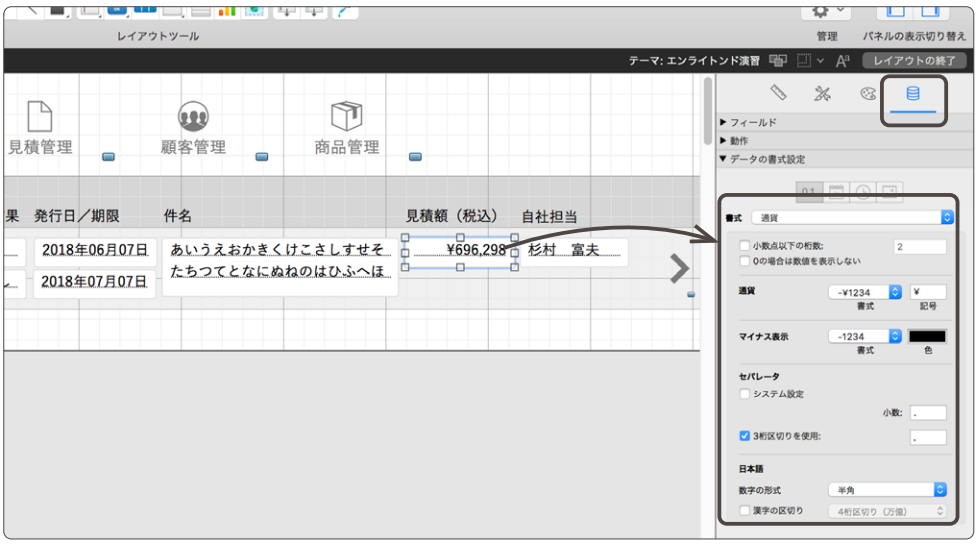


図 9-60 インスペクタの設定

(4)　このレイアウトでは、フィールドに入力できないようにします。

a. 「ボディ」パートに配置されているフィールドをすべて選択して設定します。
b. インスペクタの「データ」タブの［フィールド入力］で「ブラウズモード」を **オフ** にします。

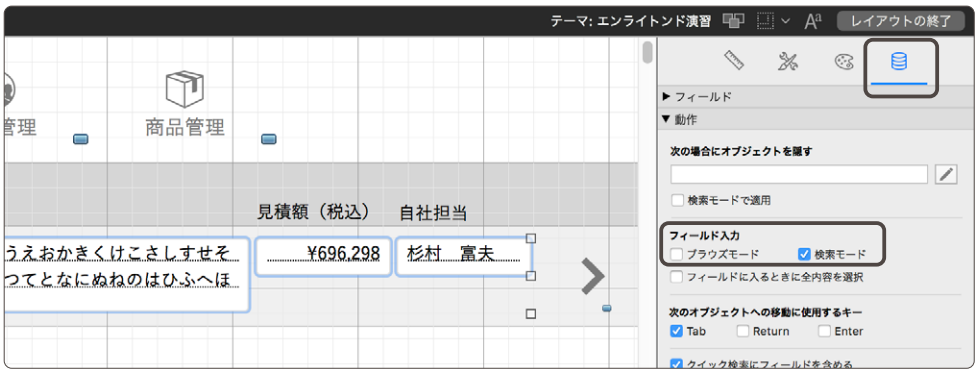


図 9-61 インスペクタの「フィールド入力」設定

(5)　パートにスタイルを設定します。

a. 「上部ナビゲーション」パートのパートラベルを選択します。
b. インスペクタの「スタイル」タブで「ナビゲーションパート」を選択します。
c. 「ヘッダ」パートのパートラベルを選択します。
d. インスペクタの「スタイル」タブで「ヘッダパート」を選択します。
e. 「ボディ」パートのパートラベルを選択します。
f. インスペクタの「スタイル」タブで「リストボディ」を選択します。

(6)　レイアウトを保存してブラウズモードへ切り替えます。

演習ここまで

見本とした「顧客リスト」レイアウトと同じスタイルやデザインの「見積リスト」レイアウトができあがりました。

フィールドの書式設定

書式設定は、フィールドに格納されているデータを変更することなくレイアウト上で適切に表示するために利用します。

日付は、書式を設定後に配置場所やサイズを決定するほうが良いことがわかります。「20xx/xx/xx」と「20xx 年 xx 月 xx 日」では、後者の方が横幅が広い必要があります。

数字は、3 桁区切りがある方がユーザが桁を判別しやすくなります。￥を数値の前に付加する場合も必要な横幅が広くなります。

リスト形式のフィールド入力

別途に 1 件ずつの入力用画面を用意するのであれば、あえてリスト形式で入力できなくても良いはずです。

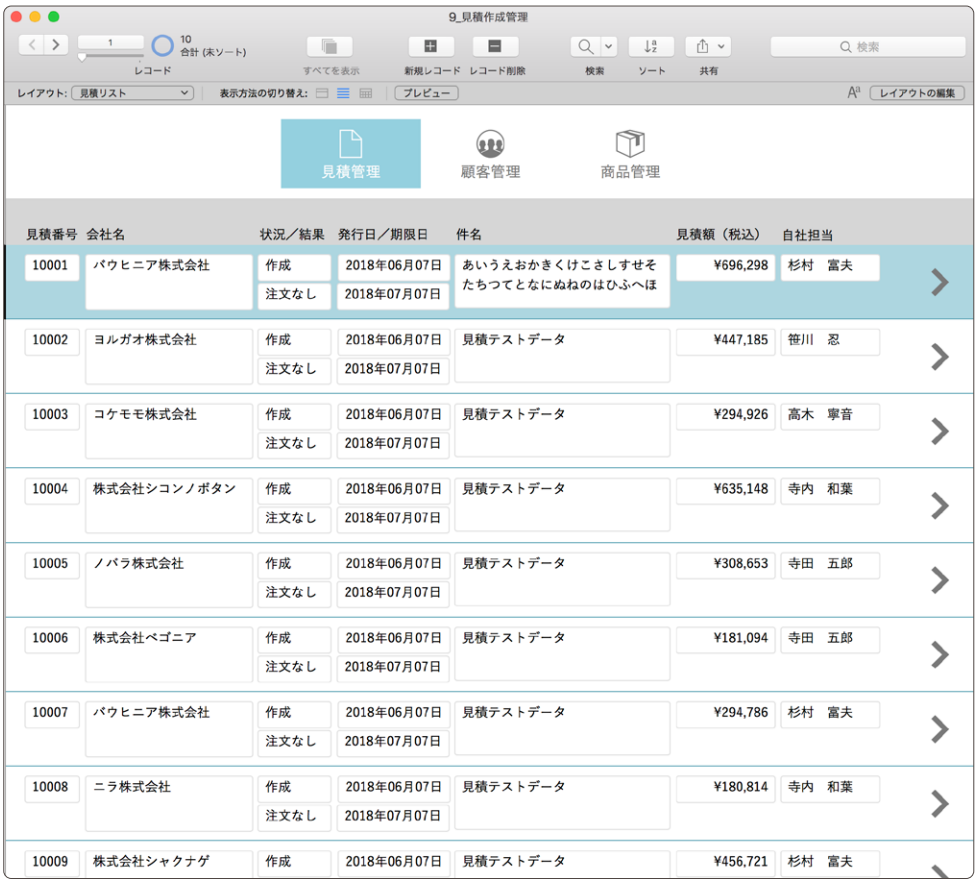


図 9-62 ブラウズモードの「見積リスト」レイアウト

やってみよう

レイアウトモードで各レイアウトオブジェクトのスタイルを確認します。ボタンバーやパート、各種レイアウトオブジェクトをクリックしてインスペクタの「スタイル」タブでスタイル名を確認してください。

これまでの演習ファイルを閉じて次の演習ファイルを開いてください。3 つめの画面「見積詳細」レイアウトといくつかの値一覧を用意しました。

（ここから、本書に付属している演習ファイルの「演習 9_10 から 13」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

9.11　フィールドオブジェクトについて

ほとんどの場合、レイアウトにフィールドが配置されます。ですから様々な設定が用意されています。「見積詳細」レイアウトで確認と演習を行います。

9.11.1　フィールドのコントロールスタイル

フィールドのレイアウト上の表示スタイルは 7 種類です。

・編集ボックス
・ドロップダウンリスト
・ポップアップメニュー
・チェックボックスセット
・ラジオボタンセット
・ドロップダウンカレンダー
・マスク付き編集ボックス

編集ボックス
単純にフィールドにデータを表示したり入力するには「編集ボックス」を使用します。

入力の補助が必要な場合には他のスタイルを選択します。すでに演習ファイルにいくつか設定されています。

ドロップダウンカレンダー
日付タイプフィールドの入力補助に最適です。カレンダーで日付をクリックすると西暦 4 桁の日付が入力されます。データのレイアウト上の表示はインスペクタの［データの書式設定］で設定します。

「発行日」フィールドに設定されています。

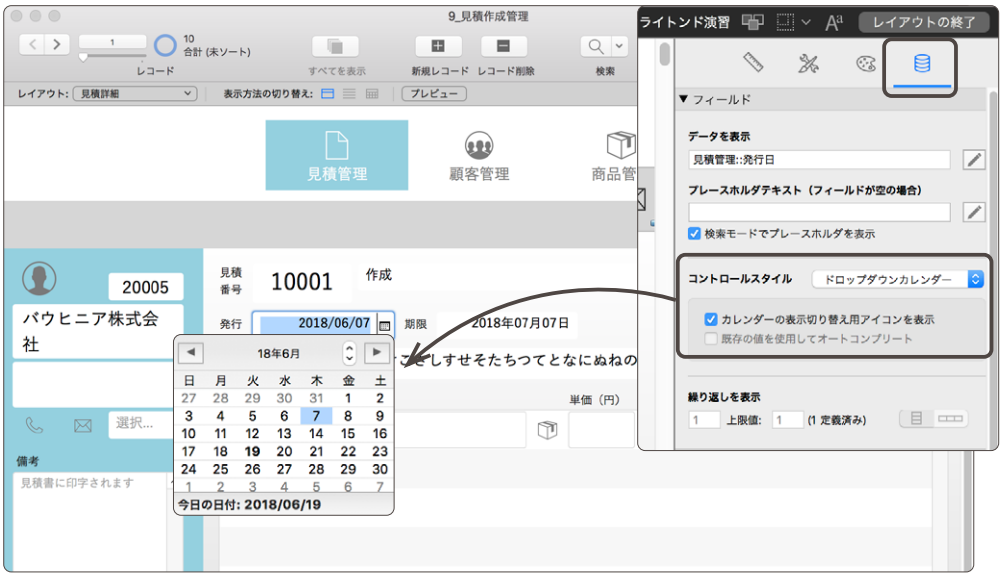


図 9-63 ドロップダウンカレンダー

マスク付き編集ボックス
見た目は編集ボックスと同様ですが、入力するとデータが「●」で表現されます。

入力されている文字数に関わらず一定数の●になります。注意するのは、これは「表示だけ」で格納されるデータが暗号化されているわけではないことです。また、レイアウトごとに指定する必要があります。

やってみよう
「見積詳細」レイアウトのフィールドにマスク付き編集ボックスを設定し、動作を確認しましょう。

9.11.2 値一覧

残りのコントロールスタイルは値一覧に関するものです。ここで、値一覧について確認します。

フィールドに割り当てて入力を補助するものです。1 つの値一覧は、複数のフィールド、ファイルで使用できます。次の 3 種類を設定できます。

・フィールドの値を使用
・他のファイルの値一覧を使用
・カスタム値を使用

フィールドの値を使用
TO（テーブル）とフィールド（2 つまで）を指定します。指定したフィールドに入力されているデータによって値一覧の内容が変化します。**入力されるのは「最初のフィールド」のデータ**です。

そして、表示されるデータはユニークな内容にまとめられて、ソートされます。例えば、7 件のレコードに次のようなデータが入力されていると、

あいうえお
かきくけこ
あいうえお
さしすせそ
かきくけこ
たちつてと
あいうえお

値一覧に表示される内容は次のようになります。

あいうえお
かきくけこ
さしすせそ
たちつてと

値一覧の元になるデータの範囲
指定した TO のフィールドの全てのデータを元にユニークな内容を表示するか、指定した関連レコードのデータのみを元にするかを選択できます。

[2 番目のフィールドの値のみを表示]
フィールド値の値一覧では、2 つまでのフィールドを指定できます。
「最初のフィールド 2 番目のフィールド」
の並び順でデータが表示されます。しかし[2 番目のフィールドの値のみを表示]を　**オン**　にすると、2 番目のフィールドのデータだけが表示されるようになります。

この特徴を利用して、実際に入力される内容とは別のデータを見せてユーザに選択させることが可能です。例えば、次のようなフィールド（「商品」TO）が設定されていて、

・最初のフィールド　　主キー
・2 番目のフィールド　商品名

[全ての値を含める]と、[2 番目のフィールドの値のみを表示]が　**オン**　の値一覧を作成します。

あるレイアウトで「商品外部キー」フィールドにこの値一覧を設定する商品名のみのリストが表示されます。そして、ユーザは商品名を選択することで「商品外部キー」フィールドに「商品」TO の主キーを入力させることができます。

インスペクタの［値一覧でデータ書式を上書き］
ポップアップメニューでは［値一覧でデータ書式を上書き］を　**オン**　にすることで実際に入力されるデータではなく、ユーザが選択した見た目のデータをレイアウト上で表示できます。

他のファイルの値一覧を使用
複数のファイルの値一覧で入力されるほうが良い場合に使用します。別の見方をすれば、値一覧はどこかのファイルにまとめて作成されていても良いのです。

カスタム値の値一覧
設定されたデータ（改行区切り）が常に表示されます。

インスペクタの［値一覧の編集を許可］
インスペクタで［値一覧の編集を許可］を　**オン**　にするとデータをレイアウト上から編集できます。ただし、誰かがデータを変えてしまったら、それまでと違う内容（選択肢）になってしまうかもしれません。

9.11.3　値一覧を使用するコントロールスタイル

値一覧では 4 つのコントロールスタイルのいずれかを選択して使うことができます。

・ドロップダウンリスト
・ポップアップメニュー
・チェックボックスセット
・ラジオボタンセット

これは、

・あらかじめ決まった中からデータを選択して入力する

ということです。ちょっとした文言の違いや表記揺れを防ぎ、精度の高いデータをフィールドに入力させることが目的です。ユーザは選択するだけなので入力作業が楽になります。

ドロップダウンリストとポップアップメニュー

一覧が表示され選択して入力する同じような動作ですが、次の違いがあります。

・フィールドにカーソルを入れて値一覧以外のデータを入力できるのはドロップダウンリスト

・ポップアップメニューでは値一覧以外のデータを簡単に入力することはできない

インスペクタで設定できる内容も異なります。ドロップダウンリストでは［値一覧でデータ書式を上書き］は設定できません。

チェックボックスとラジオボタン

チェックボックスは複数選択できます。複数選択されたデータは、改行区切りでフィールドに格納されます。常に 1 つを選択するのがラジオボタンです。この原則で使い分けをします。

演習で確認します。現在の「見積詳細」レイアウトは次図です。まだ値一覧がフィールドに設定されていません。

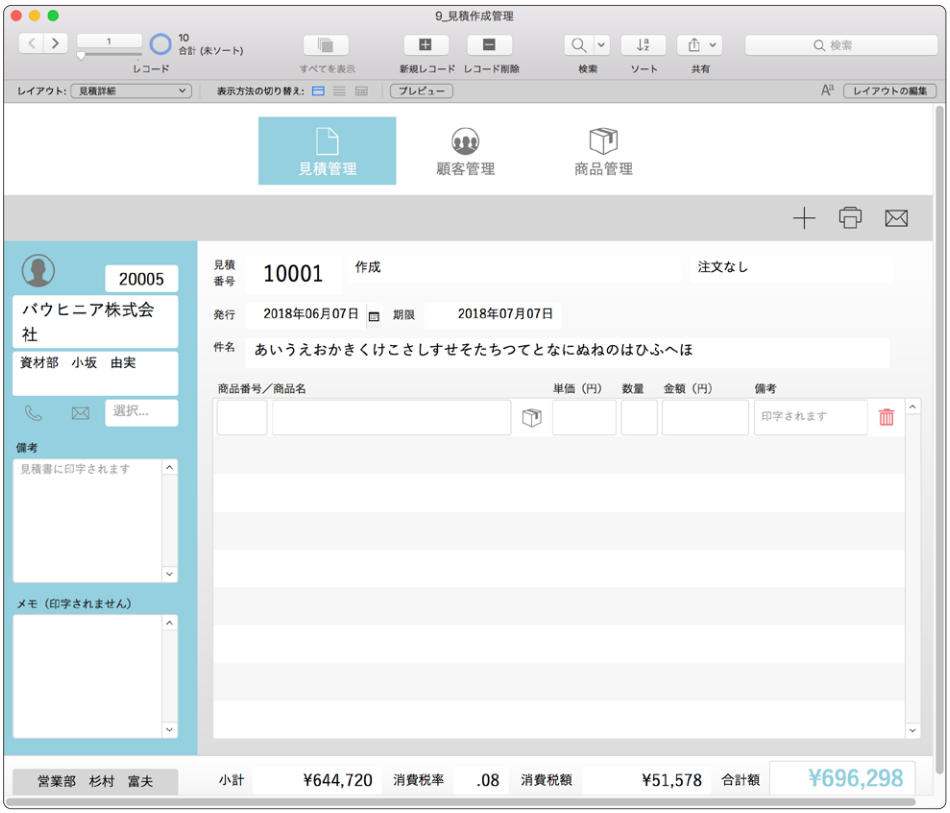


図 9-64 ブラウズモードの「見積詳細」レイアウト

「演習 9_10 から 13」フォルダ

演習 9-10　フォーム表示レイアウトの作業：「見積詳細」の作業 1（値一覧）

- **使用ファイル：9_ 見積作成管理** .fmp12

(1)　登録済みの値一覧を確認します。3 つの値一覧（カスタム値）登録されています。

a.　［ファイル］–［管理］–［値一覧 ...］メニューを選択します。

b.　［値一覧の管理］ダイアログが表示されます。

c.　登録を確認します。

・敬称

・見積状況

・見積結果

(2)　「見積詳細」レイアウトでレイアウトモードに切り替えます。

(3)　コントロールスタイルを設定します。インスペクタの「データ」タブを使います。

a.　画面左の「敬称」フィールド

・［コントロールスタイル］は　ドロップダウンリスト

・［値一覧：］は　「敬称」

・［一覧の表示切り替え用矢印を表示］のチェックを　**オン**

b.　「見積状況」フィールド

・［コントロールスタイル］は　ラジオボタンセット

・［値一覧：］は　「見積状況」

c.　「見積結果」フィールド

・［コントロールスタイル］は　ラジオボタンセット

・［値一覧：］は　「見積結果」



図 9-65「敬称」と「見積状況」フィールドを設定時のインスペクタ

d.　「期限日」フィールドの［コントロールスタイル］でドロップダウンカレンダーを割り当てます。

(4)　レイアウトを保存してブラウズモードへ切り替えます。



図 9-66 値一覧設定後

演習ここまで

「カスタム値を使用」の値一覧をフィールドに設定しました。次は「フィールドの値を使用」の値一覧を作成します。

明細行で商品を選択できるようにします。

演習 9-11　フォーム表示レイアウトの作業：「見積詳細」の作業 2（値一覧）

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　フィールドのデータを使った値一覧を作成します。

a.　［ファイル］–［管理］–［値一覧 ...］メニューを選択します。

b.　［値一覧の管理］ダイアログが表示されます。

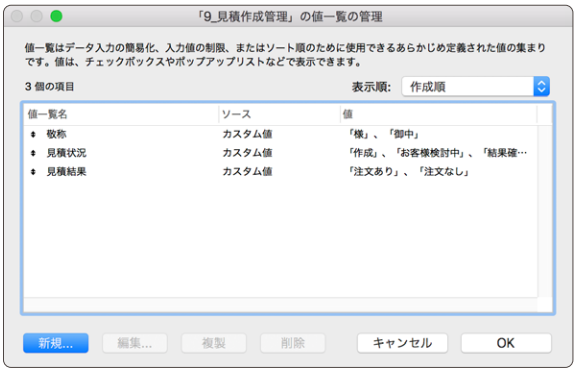


図 9-67［値一覧の管理］ダイアログ

c.　左下の［新規］をクリックします。

d.　［値一覧の編集］ダイアログが表示されます。

e.　［値一覧名］に「商品リスト」と入力します。

f.　［フィールドの値を使用］ラジオボタンをクリックします。

g.　［値一覧に使用するフィールドの指定］ダイアログが表示されます。

(2)　次の設定をします。

a.　［最初のフィールドの値を使用］は、「商品管理」TO を選択し「主キー」フィールドを指定します。

［2 番目のフィールドの値も表示］は、なにを選択するか……。
「商品名」だとユーザが選択しやすそうですが「商品番号」も表示されていると良さそうです。
そこで値一覧の表示用のフィールドを作成します。

(3)　値一覧用のフィールドを用意します。

a.　［2 番目のフィールドの値も表示］のポップアップメニューで［データベースの管理 ...］を選択します。

b.　［データベースの管理］ダイアログが表示されます。

c.　「商品情報」テーブルにフィールドを作成します。

・フィールド名　は　　「c_ 商品番号と商品名」

・フィールドタイプ　は　　計算

d.　次の計算式を設定します。［計算結果：］は「テキスト」を設定します。

```
商品番号 & " " & 商品名
```

(4)　［OK］を何度かクリックして設定に戻ります。

a. 「c_ 商品番号と商品名」フィールドを［2 番目のフィールドの値も表示］で選択します。
b. その他の設定はそのままにします。

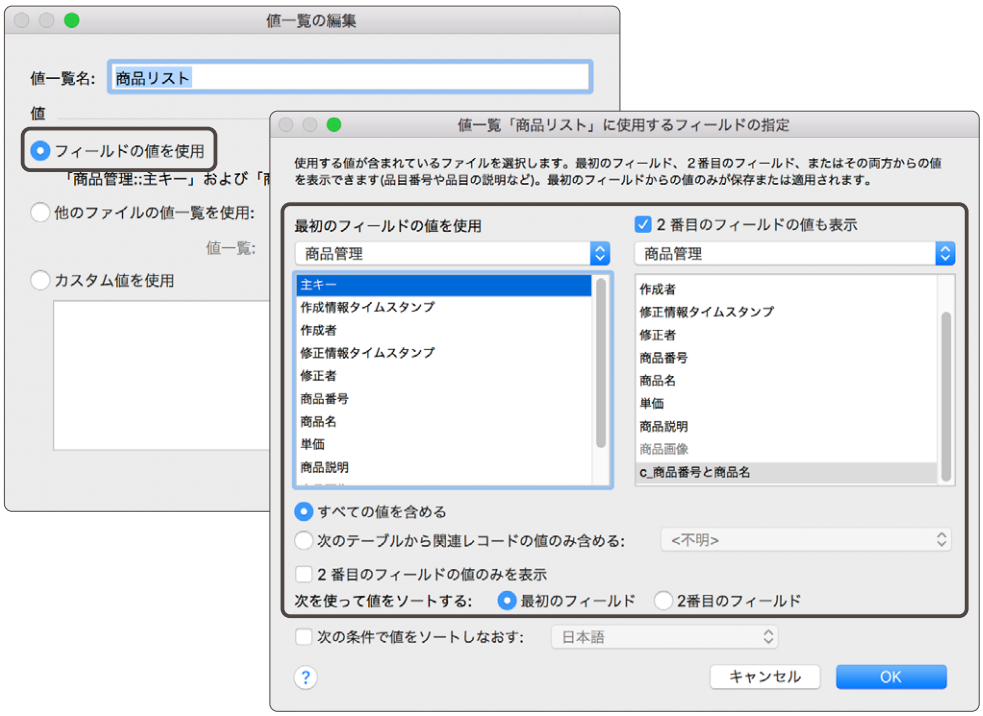


図 9-68［値一覧に使用するフィールドの指定］ダイアログ

c. ［OK］を何度かクリックして値一覧の作成を終了します。

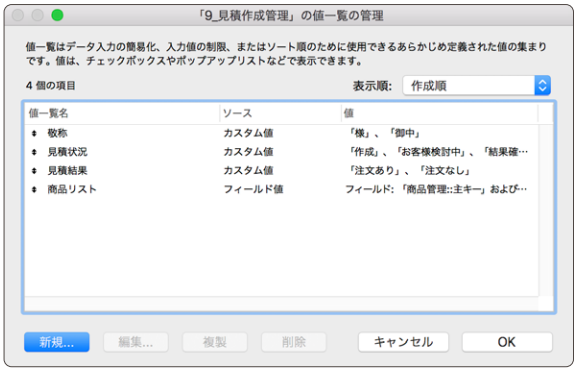


図 9-69［値一覧の管理］ダイアログ（「商品リスト」値一覧作成後）

演習ここまで

作成した値一覧をポータルのフィールドに割り当てます。

演習 9-12　フォーム表示レイアウトの作業：「見積詳細」の作業 3（値一覧）

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「見積詳細」レイアウトでレイアウトモードに切り替えます。

(2)　新しい値一覧を割り当てます。

a.　ポータル上の左の「商品番号」フィールドを「見積管理 _ 明細」TO の「商品外部キー」フィールドに変更します。

b.　インスペクタの「データ」タブの［動作］で［フィールド入力］のブラウズモードが **オン**　になっていることを確認します。

c.　インスペクタの「データ」タブの［コントロールスタイル］で値一覧を割り当てます。

・ポップアップメニュー

・値一覧は　「商品リスト」

・［一覧の表示切り替え用矢印を表示］は　**オン**

・［値一覧でデータ書式を上書き］は　**オフ**



図 9-70 値一覧を設定

(3)　レイアウトを保存してブラウズモードへ切り替えます。1 件目のレコードに移動します。

(4)　動作を確認します。ポータルの「商品外部キー」フィールドをクリックします。

この設定では、主キーが丸見えです！特に UUID を主キーとしている場合、長い文字列なのでユーザが戸惑います。何か選択して入力しても商品がわかりません。



図 9-71 値一覧に「商品情報」テーブルの「主キー」が表示されている

(5)　表示の設定を変更します。レイアウトモードに切り替えます。

(6)　値一覧を変更します。
a.　「商品外部キー」フィールドをクリックします。
b.　インスペクタの「データ」タブの［コントロールスタイル］で値一覧名の右の鉛筆アイコンをクリックします。
c.　［値一覧の管理］ダイアログが表示されます。
d.　「商品リスト」をクリックして［編集 ...］をクリックします。
e.　［値一覧の編集］ダイアログが表示されます。
f.　［フィールド指定 ...］をクリックします。
g.　［値一覧に使用するフィールドの指定］ダイアログが表示されます。
h.　［2 番目のフィールドの値のみを表示］を　**オン**　にします。

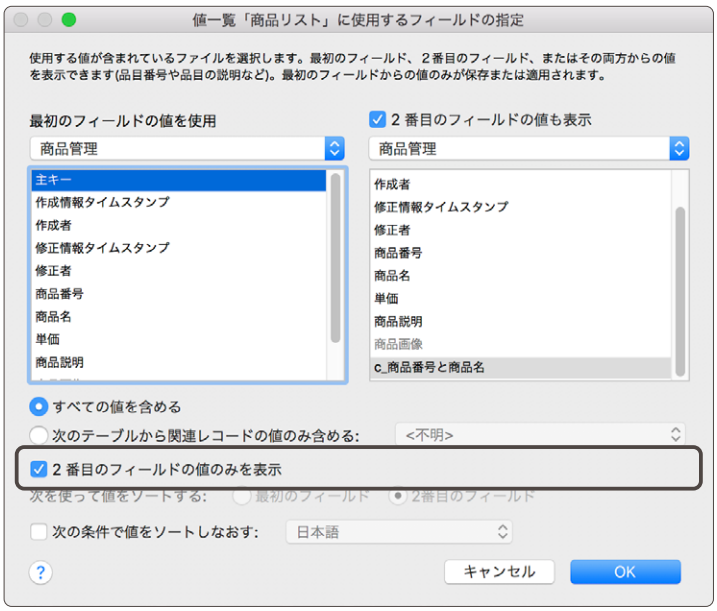


図 9-72［値一覧に使用するフィールドの指定］ダイアログ（[2 番目のフィールドの値のみを表示］をオン）

(7)　インスペクタの設定を変更します。
・［値一覧でデータ書式を上書き］は　**オン**

(8)　レイアウトを保存してブラウズモードへ切り替えます。同じように動作を確認してください。

(9)　商品番号と商品名で選択できるようになりました。

演習ここまで

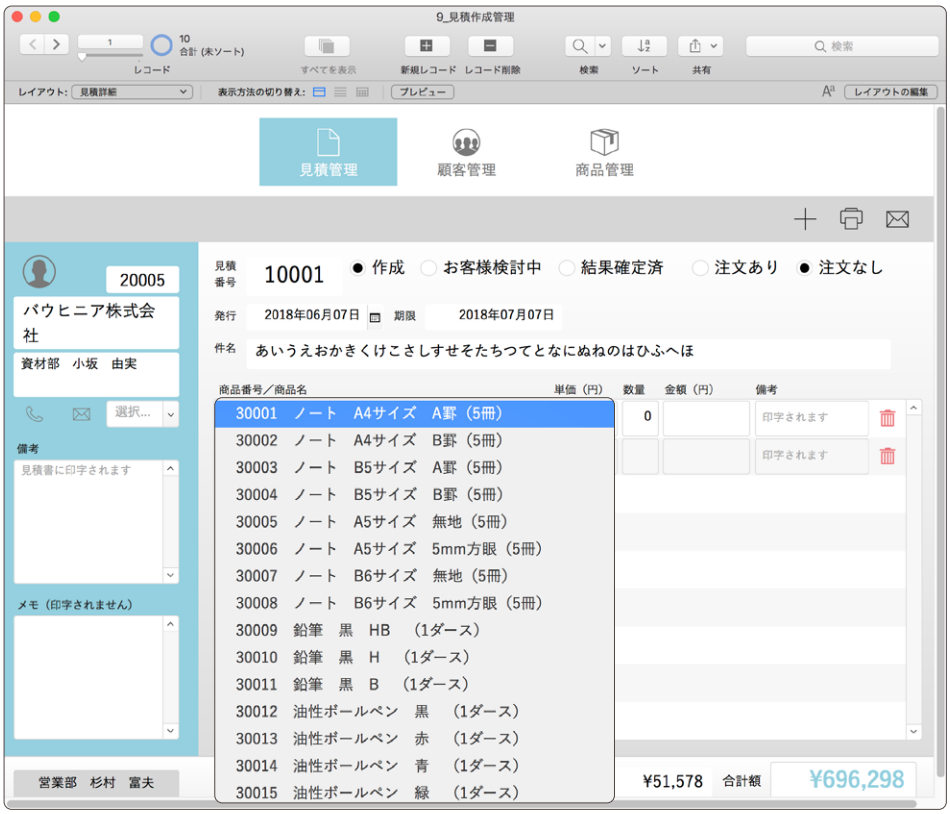


図 9-73 値一覧が選択しやすくなった

何か商品を選択してみます。「商品外部キー」フィールドが商品番号と商品名を表示するには狭すぎるようです。「商品名」フィールドを一旦レイアウトエリアの外に置きます。

値一覧で選択した情報が表示されるようにします。

演習 9-13　フォーム表示レイアウトの作業：「見積詳細」の作業 4（値一覧）

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「見積詳細」レイアウトでレイアウトモードに切り替えます。

(2)　ポータル上の「商品名」フィールドをレイアウトエリアの外に置きます。

(3)　「商品外部キー」フィールドを横に延ばし、インスペクタの「外観」タブの［段落設定］でフィールドの行揃えを「左揃え」にします。

(4)　レイアウトを保存してブラウズモードへ切り替えます。動作を確認してください。

演習ここまで

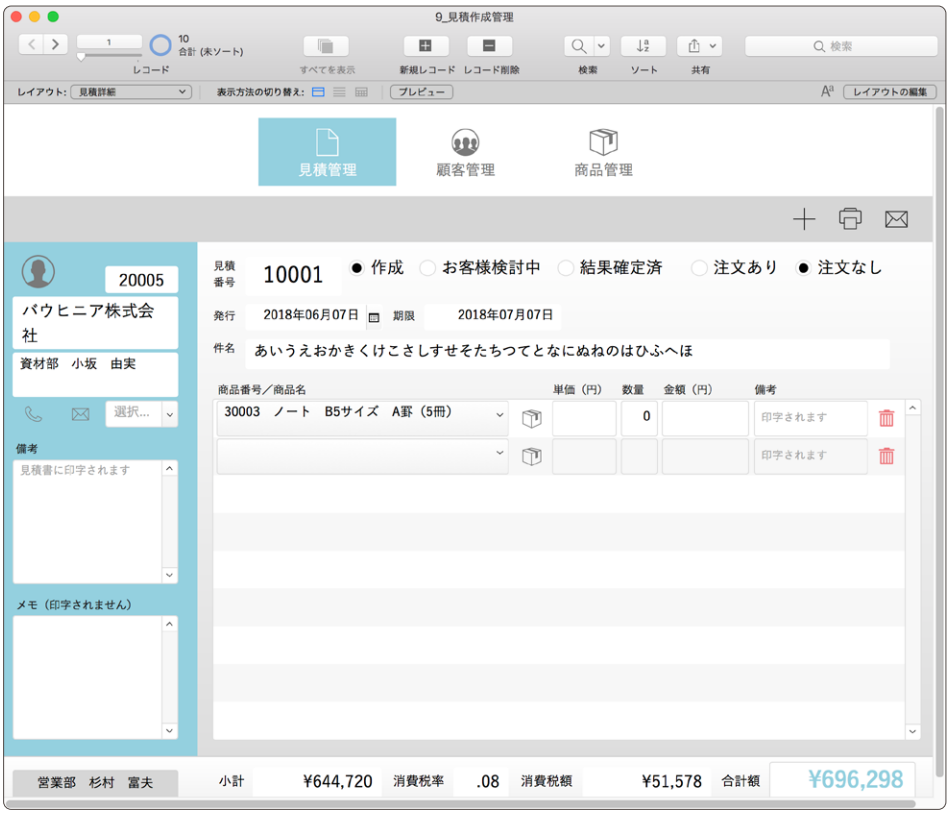


図 9-74 演習後の「見積詳細」レイアウト

値一覧は、

・入力したいデータ
・ユーザに表示したいデータ
・値一覧以外の入力の可否

などを考え合わせてコントロールスタイルを設定します。

これまでの演習ファイルを閉じて次の演習ファイルを開いてください。「見積詳細」レイアウトといくつかの値一覧を用意しました。

（ここから、本書に付属している演習ファイルの「演習 9_14 から 16」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

9.11.4 関連を利用した値一覧

さらに値一覧について確認します。マスタテーブルのデータは「顧客」や「商品」のような静的なデータを扱うものですが、

・もう使わないデータ

が、登場することがあります。何らかの理由で顧客との取引が終了したり、商品であれば販売終了になることもあります。しかし、これらのデータは**削除してはいけません！**関連が成立しなくなってしまいます。過去のデータで販売が終了した商品が表示されなくなってしまいます。

販売が終了した商品のデータは残しつつ、値一覧では販売中の商品だけが表示されていればユーザは困りません。

関連を使用して、「いま販売している」商品だけが値一覧に表示されるようにします。

どのような照合フィールドが必要でしょうか。

状況を表すフィールド（論理値を使う）
「販売状態」フィールド（参照先）として、数値を入力して使う方法があります。例えば、

・販売中の商品　　＝　1（あるいは 0 以外）
・販売終了の商品　　＝　0（あるいは 1 以外）

とするルールです。

前の演習で利用したような「1」と常に入力されている参照元フィールドと関連させます。この方法では 1 つの参照元フィールドを共通で使って複数の参照先との関連を作成することができます。例えば、

・販売中の商品、取引中の顧客　　＝　1：有効なもの
・販売終了の商品、取引終了の顧客　　＝　0：無効なもの

など、です。ある程度の規模のカスタム App では「販売状態」フィールドのようなデータの状況を表すフィールドが必要になることがあるので便利です。

なお、数値で状況を表す場合、ユーザにはルールを明示する必要があります。

販売停止の商品が値一覧に表示されないように、関連を利用した値一覧を作成します。データ（参照先フィールド）を準備します。

後から追加したフィールドにはなにも入力されていません。データに一貫性を持たせ、ユーザが戸惑わないようにデータを入力しておく必要があります。

「演習 9_14 から 16」フォルダ

演習 9-14　フォーム表示レイアウトの作業：「見積詳細」の作業 6

• **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　データの準備をします。

(2)　「商品情報」テーブルに「販売状態」フィールドを作成します。手順は省略します。

・フィールド名　は　　「販売状態」
・タイプ　は　　「数字」
・フィールドオプション　は［入力値の自動化］で　データに「1」

a.　「z_ 商品情報」レイアウトに切り替えます（表形式になっているはずです）。
b.　「販売状態」フィールドを「商品番号」フィールドの右側に追加します。

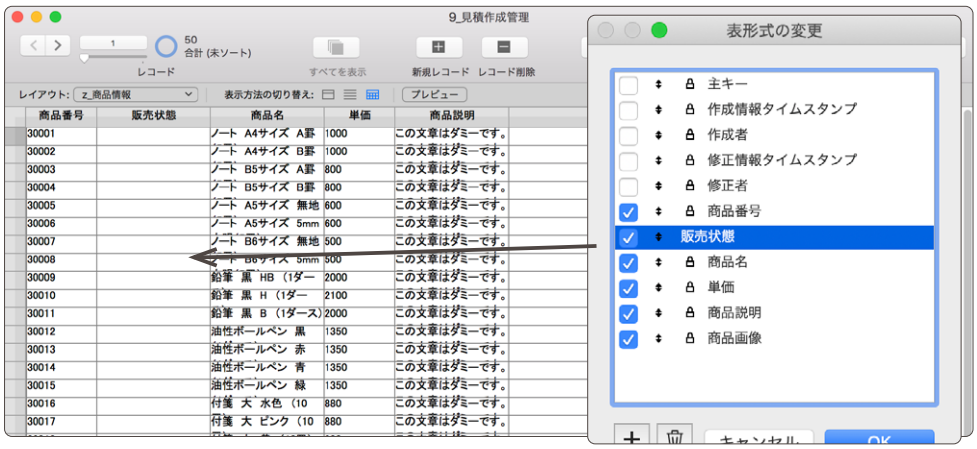


図 9-75 表形式にフィールドを追加

(3)　「販売状態」フィールドに［フィールド内容の全置換］で「1」を入力します。手順は省略します。

(4)　テストのために商品番号　30002 から 30005 のレコードの「販売状態」フィールドに「0」を入力します。



図 9-76「0」を入力

(5)　準備ができました。

演習ここまで

次に関連のための参照元フィールドの準備と構造の作業を行います。

演習 9-15　フォーム表示レイアウトの作業：「見積詳細」の作業 5（値一覧）

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　［データベースの管理］ダイアログを開きます。

(2)　「ユーティリティ」テーブルに「_1」フィールドを作成しておきました
「フィールド」タブで作成されていることを確認してください。
「z_ ユーティリティ」レイアウトを表示して、1 が入力されていることを確認してください。

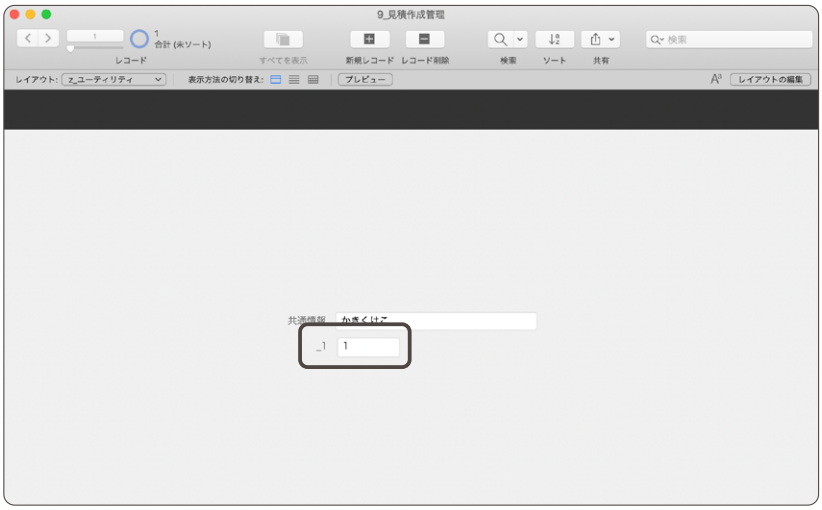


図 9-77「_1」を確認

(3)　リレーションシップグラフに TOG を新しく作成します。
次図も参照してください。
a.　「z_ ユーティリティ」TO を複製します。
b.　「ユーティリティ値一覧」に名前を変更します。
c.　「z_ 商品情報」TO を複製します。
d.　「ユーティリティ値一覧 _ 商品」に名前を変更します。
e.　TO 名が長いので、横幅を広くします。
f.　これらの TO を空いている場所に配置します。
g.　「ユーティリティ値一覧」TO の「_1」フィールドと「ユーティリティ値一覧 _ 商品」TO の「販売状態」フィールドを関連させます。
h.　照合条件は「=」、オプションはなにも設定しません。

(4)　［OK］を何度かクリックして［データベースの管理］ダイアログを閉じます。

演習ここまで

TOG を作成しました。

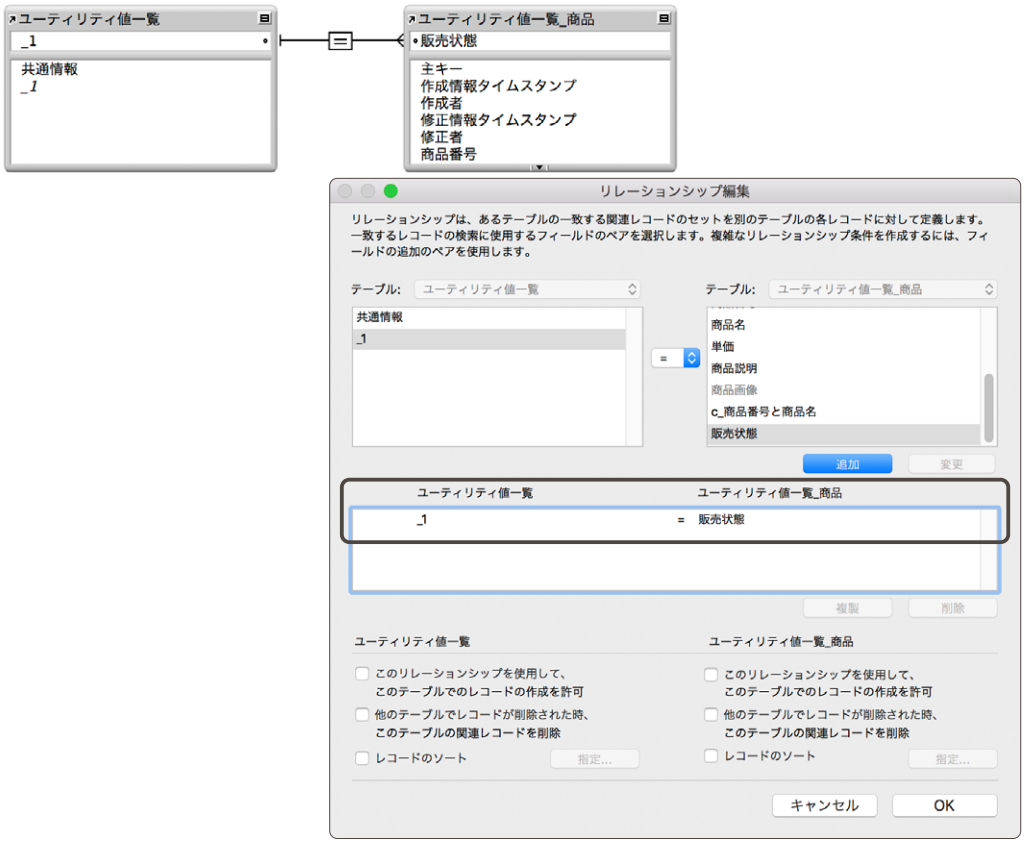


図 9-78 値一覧用の TOG を作成

演習ではやりませんが、商品の販売状態を分かりやすくレイアウト上で表示するには、次のような方法があります。

文言で状況表示する

インスペクタの「位置」タブの［データの書式設定］で「論理値」を選択すると、数値を文言としてレイアウト上に表示することが可能です。これはレイアウトオブジェクトごとに設定する必要があります。また、もう一つの方法として数値の 0/1 を元に、別の計算タイプフィールドを作成してユーザにわかりやすい文言として表示することも可能です。

2 つ以上の状況

データの状況は、2 つ以上になることもあります。その場合、論理値ではなくなります。「1、2、3、4、5…」とルールを決めることが可能です。参照元のフィールドも状況に合わせて複数作成します。

空欄をどう考えるか

データの状況について設計する場合に、次の 2 つも考慮します。

・空欄
・空欄でない

ただし、「空欄である」という条件は関連には使えません。「なにかデータが入力されている」という条件は関連として使えます。

続いて、販売中商品の値一覧を作成します（比較できるように値一覧を新しく作成します）。

演習 9-16　フォーム表示レイアウトの作業：「見積詳細」の作業 7

- **使用ファイル：9_ 見積作成管理** .fmp12

(1)　値一覧を作成します。手順は省略します。

a.　［値一覧の編集］ダイアログでは、

・［値一覧名］は　　「商品リスト（販売中）」と入力

・［フィールドの値を使用］は　　ラジオボタンをクリック

b.　［値一覧に使用するフィールドの指定］ダイアログでは、

・［最初のフィールドの値を使用］は　「ユーティリティ値一覧 _ 商品」TO を選択し「主キー」フィールドを指定

・［2 番目のフィールドの値も表示］は　同じ TO の「c_ 商品番号と商品名」フィールドを指定

・［次のテーブルから関連レコードの値のみ含める］は　「ユーティリティ値一覧」TO を選択

・［2 番目のフィールドの値のみを表示］を　**オン**

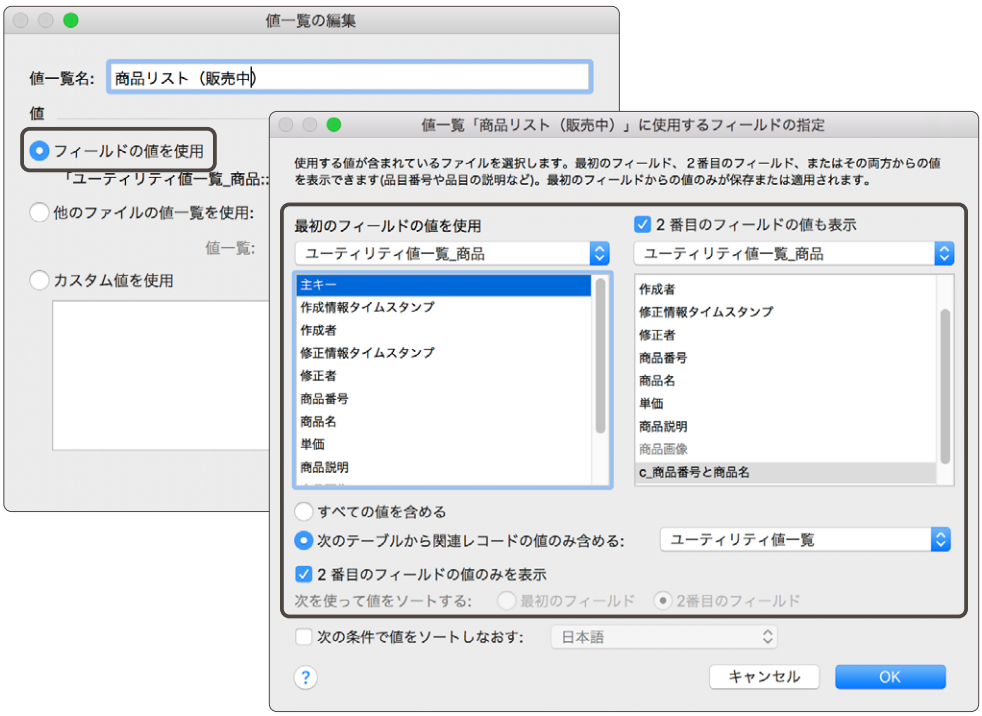


図 9-79［値一覧に使用するフィールドの指定］ダイアログ

(2)　［OK］を何度かクリックして値一覧の作成を終了します。

(3)　「見積詳細」レイアウトでレイアウトモードに切り替えます。

(4)　新しく作成した「商品リスト（販売中）」値一覧をポータルの「商品外部キー」フィールドに割り当てます。インスペクタの「データ」タブの［コントロールスタイル］で値一覧を割り当てます。

・ポップアップメニュー

・値一覧は　　「商品リスト（販売中）」

・［一覧の表示切り替え用矢印を表示］を　**オン**
・［値一覧でデータ書式を上書き］を　**オン**

(5)　レイアウトを保存してブラウズモードへ切り替えます。動作を確認してください。
商品番号　30002 と 30005　が表示されません。



図 9-80 販売していない商品が表示されなくなった

(6)　販売されていない商品が選べなくなりました。

演習ここまで

しかし、気になることがでてきます。［値一覧でデータ書式を上書き］は、値一覧のデータを表示することなので関連が成立しなくなって値一覧の対象で無くなったら「値一覧の内容」が表示されなくなります。

4 件目のレコードに移動してください。ポータルの 4、5 件目の商品情報が表示されていません。

販売停止になった商品を選択できなくなりましたが、一方、過去に販売中だった商品情報は表示されなくなりました。本来そのフィールドに保存されているデータ（外部キー）が表示されるようになったためです。



図 9-81 商品情報が表示されなくなった

販売停止後も情報を表示するには、選択した商品の商品名を格納するためのフィールドを配置して、商品の選択と、商品名の表示を別途に考える必要があります。ここでは、そのままにして先に進みます。

9.11.5 フィールド入力

レイアウトに配置されるフィールドには、フィールドオプションで設定されている内容とは異なる様々なコントロールを行うことができます。

フィールドは、状況によって必ずしもユーザが入力するためのものではありません。

・『このレイアウトでは入力して欲しくない』けれど『別のレイアウトは入力できて良い』

という場合、入力して欲しくないレイアウトで、

・インスペクタの「データ」タブの［動作］で設定

これまでに「顧客リスト」「見積リスト」レイアウトでそのように設定をしました。「見積詳細」レイアウトにもいくつか設定されています。

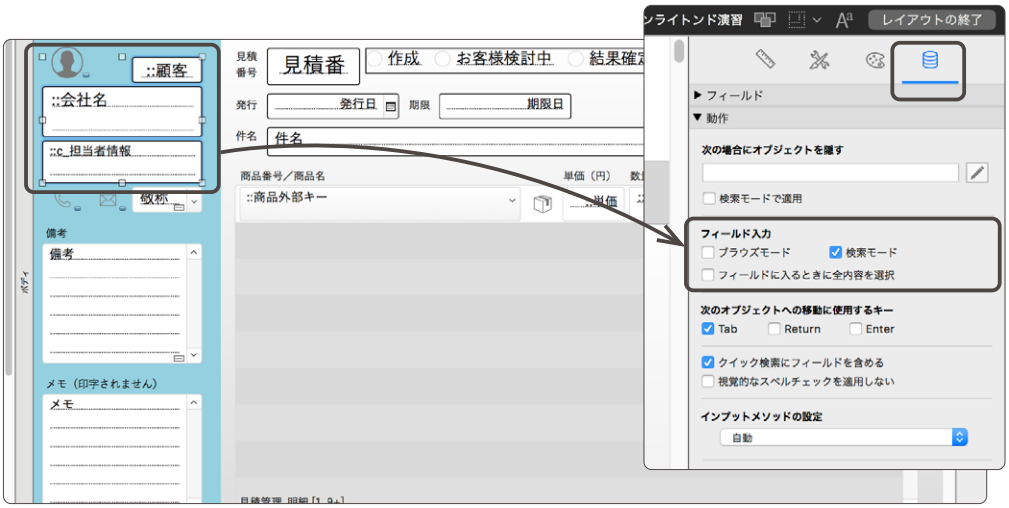


図 9-82 フィールド入力の設定

入力できるモードも指定できます。

・『このレイアウトではブラウズモードでは入力して欲しくない』けれど『検索モードでは検索条件を入力できる必要がある』

のような状況で活用できます。

やってみよう

クリックしても入力できないことを確認しましょう。

プレースホルダテキスト

プレースホルダテキストはフィールドになにもデータが入力されていないときに表示されます。編集ボックス、ドロップダウンリスト、ポップアップメニューに有効です。

ユーザに「ここになにを入力するべきなのかを伝える」ために利用します。入力後は表示されなくなります。

・インスペクタの「データ」タブの［フィールド］で設定

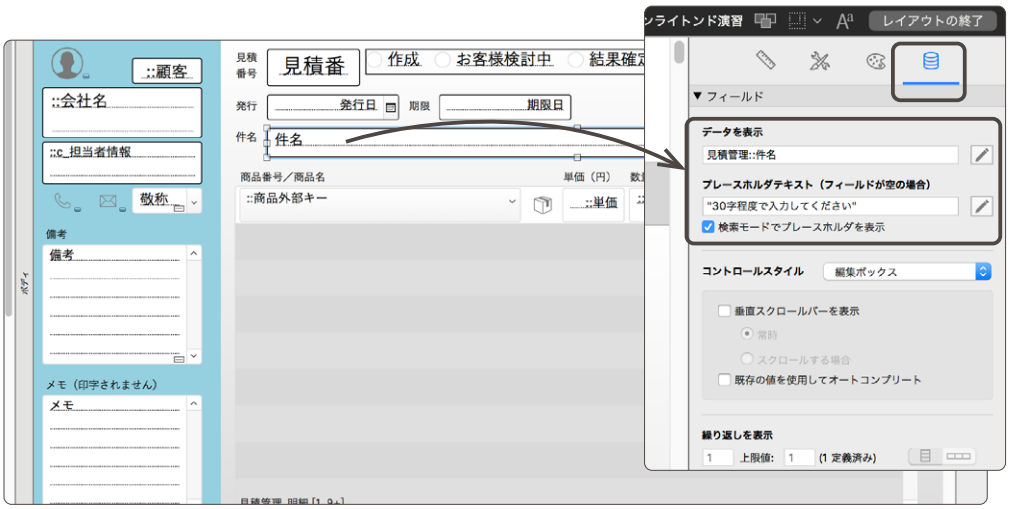


図 9-83 プレースホルダテキストの設定

「見積詳細」レイアウトでは、件名や備考などに設定されています。

やってみよう

入力されているデータを消去したり入力したりして確認しましょう。

9.11.6 レイアウトオブジェクトの表示／非表示

レイアウト単位でも、モードでもなく、

・「○○」フィールドが□□だったら「△△」フィールドに入力して欲しい
・「○○」フィールドが ×× だったら「△△」フィールドに入力しないで欲しい

という場合は、「**次の場合にオブジェクトを隠す**」を使用すると便利です。

・ある条件に合致したらフィールドが「表示／非表示」になる

という機能です。これは、次のように設定します。

・インスペクタの「データ」タブの［動作］の「次の場合にオブジェクトを隠す」で設定
・条件は、計算式で指定

表示形式とレイアウトオブジェクトの表示／非表示
表形式では、この設定が有効になりません。また、リスト形式表示では、表示されている各レコードに対して評価が起こります。そのために、たくさんの非表示設定をすると表示が遅くなることがあります。

レイアウトオブジェクトの表示／非表示の注意点
ポータルそのものやポータル行中のフィールドなどに適用できますが、ポータル行の表示／非表示には使えません。その場合はポータルフィルタを使います。そして、タブパネル、スライドコントロールパネルの 1 つのパネルだけに適用することはできません。

例えば、「見積作成管理 .fmp12」では「見積状況」フィールドが「結果確定済」になったら「見積結果」フィールドに『注文があったのかなかったのか』を入力してもらう必要があります。しかし、それ以外の状況では入力は不要です。次のように考えます。

・「見積状況」フィールドが「結果確定済」以外では「見積結果」フィールドを表示しない

［次の場合にオブジェクトを隠す］を演習します。

演習 9-17 フォーム表示レイアウトの作業：「見積詳細」の作業 8

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1) 「見積詳細」レイアウトでレイアウトモードに切り替えます。

(2) 「見積結果」フィールドをクリックします。

(3) ［次の場合にオブジェクトを隠す］を設定します。
インスペクタの「データ」タブの動作］の「次の場合にオブジェクトを隠す」に以下の計算式を設定します。

見積管理 **::** 見積状況 ≠ " 結果確定済 "



図 9-84「次の場合にオブジェクトを隠す」の設定

(4)　レイアウトを保存してブラウズモードへ切り替えます。動作を確認してください。

演習ここまで

「見積状況」フィールドが「結果確定済」以外であれば「見積結果」フィールドは表示されません。非表示のレイアウトオブジェクトはユーザは選択したり入力することはできません。フィールドの背景を透明にして見えないようにすることとは違います。

この設定は「論理値」を使っています。次のように評価しています。

・「見積状況」フィールドが「結果確定済」　＝　0　→　表示
・「見積状況」フィールドが「結果確定済」ではない＝　1　→　非表示

検索モードと非表示
評価を検索モードでも有効にするには［検索モードで適用］を**オン**にします。これによりブラウズモードと検索モードの両方で適用されます。

例えば、検索モードでだけ適用したいがブラウズモードでは適用したくない場合、以下の計算式を設定します。

```
Get( ウインドウモード ) = 1 and 見積管理::見積状況 ≠ "結果確定済"
```

9.11.7　クイック検索対象の指定

レイアウト設定で［クイック検索を有効にする］のチェックが　**オン**　の場合、クイック検索が有効です。さらに、レイアウト上に置かれているフィールドごとに「クイック検索の対象／非対象」を設定できます。

・インスペクタの「データ」タブの [動作] の「クイック検索にフィールドを含める」のチェックのオン／オフ



図 9-85「クイック検索にフィールドを含める」の設定

そして、

・集計タイプフィールド
・オブジェクトタイプフィールド
・グローバル格納フィールド

これら以外は、レイアウトに配置されると自動的にクイック検索の対象に設定されます（クイック検索も索引を使用しています）。[表示]–[オブジェクト]–[クイック検索] メニューを選択するとバッジ（虫眼鏡マーク）が表示されます。検索に時間がかかるフィールドがバッジの色でわかります。

・緑はクイック検索対象
・黄はクイック検索対象であるが、検索に時間が掛かるもの
・グレーはクイック検索対象外



図 9-86 虫眼鏡バッジ（緑、黄、グレー）

黄は、索引が作成されないフィールド、または、関連テーブルのフィールド（レイアウト設定で指定されている TO のフィールドではない）ということです。

クイック検索は、対象のフィールドすべてに検索が実行されるためフィールド数や非保存の計算タイプフィールドなどにより結果表示に時間がかかることがあります。検索する必要がないフィールドは対象外にしておく方法もあります。

9.11.8 フィールドへの移動

フィールドに関する設定は、他にもあります。どのような順番で入力するか、です。

・インスペクタの「データ」タブの［動作］の「次のオブジェクトへの移動に使用するキー」のチェックのオン／オフで設定

3 種類のキーでレイアウトオブジェクトを移動することが可能です。

・[Tab]
・[Enter]
・[return]



図 9-87「次のオブジェクトへの移動に使用するキー」の設定

マウスでフィールドをクリックしなくてもフィールド間を移動できると、ユーザは、気持ち良く作業できるかもしれません。テンキーで数字を入力しながら次々とフィールドを移動するには［Enter］キーでの移動が便利です。パソコンで連続して入力することが想定されるレイアウトでは、ユーザと一緒にフィールド移動を設定すると効果的かもしれません。

設定の注意点

フィールドの移動に［Enter］キーを使用すると、
・レコード確定
・スクリプトの再開
・検索実行
が［Enter］キーでできません。また、フィールドに改行を入力できないようにするためにも使用できます。

また、フィールドの移動に［Retern］キーを使用すると、
・フィールドの中に改行を入力する
ができなくなります。

フィールド移動に順番を設定することを「**タブ順**」と呼びます。

タブ順
レイアウトオブジェクトが配置されると FileMaker Pro が自動的にタブ順を設定します。原則、

・左から右、そして、上から下

の順になります。配置のタイミングや配置後に移動することもありますので、結果として思ったようなタブ順が設定されていないことがあります。なお、非表示になっているレイアウトオブジェクトは移動することができません。

[レイアウト]–[タブ順設定 ...] メニューで［タブ順設定］ダイアログから現在の設定を確認したり順番を設定できます。

タブパネルやスライドコントロールパネルとタブ順
1 つのパネルごとにタブ順を設定することはできません。パネル全体に移動することになるので、左右の矢印キーでパネル間を移動して目的のパネルを［Space］バーキーを押して表示します。

ボタン、ボタンバーとタブ順
ボタンの場合、［Space］バーキーを押すと、ボタンを押したことになります。ボタンバーはタブパネルなどと同じ様に左右の矢印キーでボタン間を移動して目的のボタンを［Space］バーキーを押します。

コントロールスタイルとタブ順
ドロップダウンリストとドロップダウンカレンダーをタブで移動すると表示させることができます。ただし、以下の設定が必要です。
・ドロップダウンリストは［一覧の表示切り替え用矢印を表示］が　**オフ**
・ドロップダウンカレンダーは［カレンダーの表示切り替え用アイコンを表示］が　**オフ**
ポップアップメニューは表示できません。タブで移動後に［Space］バーキーを押して表示することになります。

やってみよう
タブ順を入力しやすいように設定してみましょう。

これまでの演習ファイルを閉じて次の演習ファイルを開いてください。「商品管理」レイアウトを用意しました。

（ここから、本書に付属している演習ファイルの「演習 9_18 から 25」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

9.12　様々なレイアウトオブジェクト

フィールド以外に、レイアウト上で特別な役目を持ったレイアウトオブジェクトがあります。これらは、ユーザが使いやすく、わかりやすい画面にするために使用します。

・テキスト／線／図形
・ボタン／ボタンバー／ポップオーバーボタン
・タブコントロール／スライドコントロール
・ポータル
・Web ビューア
・グラフ

9.12.1　テキスト／線／図形

任意の位置に「テキスト」「線」「図形（長方形、角丸長方形、楕円）」をレイアウトオブジェクトとして配置することが可能です。配置後、インスペクタでサイズ、色、線の太さ、塗りつぶしの有無などを設定します。

例えば、フィールドラベルは「テキストオブジェクト」です。ユーザにフィールドの内容をわかりやすくするためなどに使用します。



図 9-88 テキスト、線、図形のツール

9.12.2　ボタン／ボタンバー／ポップオーバーボタン

ボタンは、ユーザが押すことでスクリプトや FileMaker Pro のコマンドを実行するものです。ボタンには次のものがあります。

・ボタンは、1 つのボタン
・ボタンバーは、1 つまたは複数のボタン
・ポップオーバーボタンは、ボタンを押すと「吹き出し」を表示

また、テキストオブジェクト、線、図形、フィールドなど様々なレイアウトオブジェクトをボタンとして設定することが可能です。ボタンにできるかどうかは右クリックでショートカットメニューに［ボタン設定］があるかどうかで確認できます。本書の演習の中でも適宜確認していきます。



図 9-89 ボタンとボタンバーのツール

設定は、専用のダイアログを使用します。

・ボタンは［ボタン設定］ダイアログ
・ボタンバーは［ボタンバー設定］ダイアログ

ボタンの名前（ラベル）
オブジェクト名ではなく、ボタンのラベルについて設定します。6 種類の表示があります。文字列のラベルと記号のアイコンの組み合わせです。

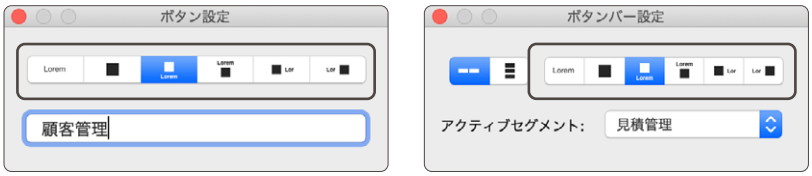


図 9-90 ラベルの表示設定

文字列のラベル
ボタンに文字列を表示できます。
・ボタンは　テキストのみが使用可能
・ボタンバーは　テキスト、計算式、変数が使用可能

ボタンバーの文字列ラベルに計算式を使用するとレイアウトモードでのラベル名は「計算」となります。[表示]–[オブジェクト]–[サンプルデータ］のチェックを**オン**にするとブラウズモードと同じく計算結果が表示されます。

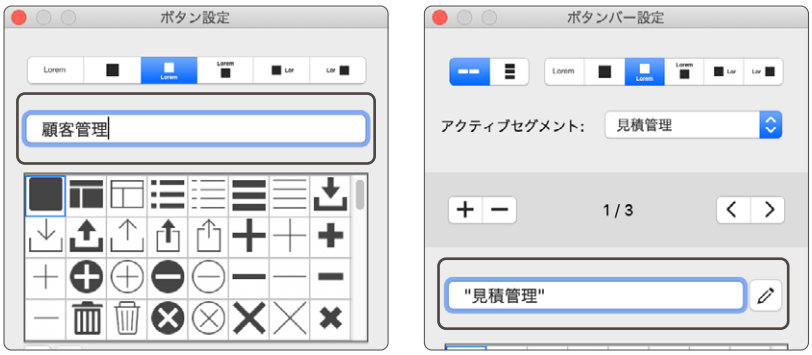


図 9-91 文字列ラベルの設定

ボタンと書式設定
ボタンは詳細な書式設定が可能です。インスペクタの「外観」タブで設定できます。これらの書式設定をワンセットにして「スタイル」として保存できます。レイアウトのテーマごとに、あらかじめいくつかのボタンスタイルが設定されています。

ボタンは「ボタン」と「アイコン」について 4 種類の状態を設定できます
・通常
・ポイントしたときに表示
・押したとき
・フォーカス
ボタンバーは「ボタン（ボタンバー)」、「区切り」、「セグメント」、「アイコン」について 5 種類の状態を設定できます。

・アクティブ
・非アクティブ
・ポイントしたときに表示
・押したとき
・フォーカス

ボタンと条件付き書式
条件付き書式が使用できます。ボタンバーの場合、全体ではなくセグメントごとの設定になります。

ボタンと非表示
ボタンを押せなくするには、インスペクタの［次の場合にオブジェクトを隠す］も使用できます。ボタンバーの場合、全体あるいはセグメントごとに設定が可能です。セグメントの非表示は、非表示になったセグメントの幅の分、他のセグメントが広くなります。

ボタンバーのセグメント
ボタンバーは、1 つ以上の複数のボタンを取り扱うことが可能です。ボタンバーに含まれているボタンをセグメントと呼びます。



図 9-92 セグメント

アクティブなセグメント
ボタンバーにはセグメントに「アクティブ」という概念があります。

「見積作成管理 .fmp12」の「顧客リスト」「見積リスト」「見積詳細」レイアウトではレイアウトごとに塗りつぶされるセグメントの色が違います。同じボタンバーですが、アクティブセグメントの設定が違うためです。



図 9-93 アクティブセグメントの設定例

設定は、レイアウトごとにセグメント名を指定するかセグメントに付与したレイアウトオブジェクト名を結果とする計算式を使用します。

セグメントは、ポップアップメニューで指定します。ボタンに文字列のラベルが設定されていると、ポップアップメニューに表示されます。文字列のラベルが設定されていない場合「<セグメント 1>」などと、ポップアップメニューに表示されます。



図 9-94 アクティブなセグメントの設定

ポップオーバーボタン

ボタンをクリックするとポップオーバー（吹き出し）が表示されるボタンです。



図 9-95 ポップオーバーボタン

ステータスツールバーで、「ボタン」のツールアイコンを長押しクリックすると選択できます。

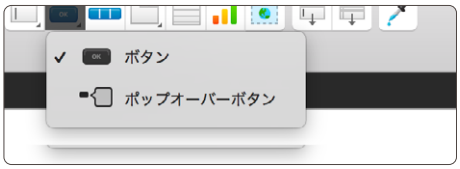


図 9-96 ポップオーバーボタンのツール

配置の注意点があります。ポータル上のポップオーバーボタンのポップオーバーの中に、ポータルやタブコントロール、スライドコントロールを配置することはできません。ポータルの中にポータルが配置できないことと同じです。

ボタンバーでは、セグメントごとにポップオーバーボタンにするかどうかを設定可能です。

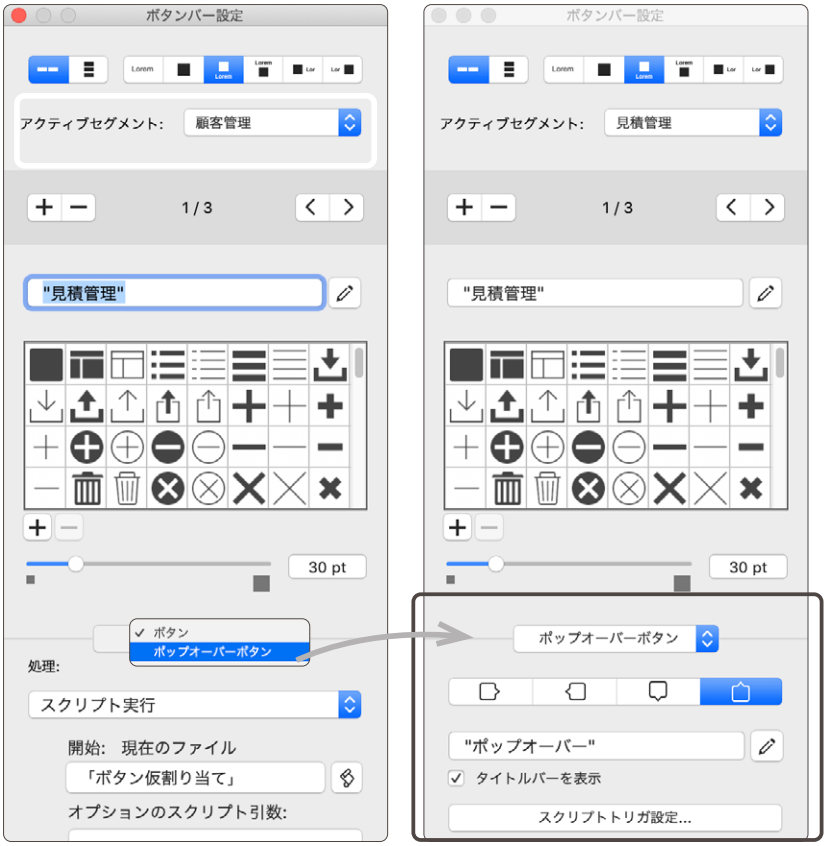


図 9-97 ポップオーバーボタンの設定

ポップオーバーとウインドウ

別のウインドウを開く機能を作成する前に、ポップオーバーが使えないかを考えてください（あるいはカードスタイルのダイアログでも）。

・現在のレイアウトに表示されていない詳細な情報の表示（例えば、オブジェクトタイプフィールドの内容を拡大して表示する）
・関連レコードの表示（ポータル上ではダメ）
・Web ビューア

などです。または画面に配置するボタンを減らしたい場合にも使用できます。

ポップオーバーのとじ方

ポップオーバー以外のどこかをクリックすると閉じます。ボタンなどで閉じるようにするには［ポップオーバーを閉じる］スクリプトステップが用意されています。

ポップオーバーの位置

上下左右を指定できます。レイアウトの状況によって FileMaker Pro が表示を調整するので、必ずしも設定した方向に表示されない場合があります。

ボタンとボタンバーは切り替えられない

ボタンはセグメントを増やせません。しかしボタンバーは 1 つのセグメントにできます。

9.12.3 タブコントロール／スライドコントロール

レイアウトに配置するレイアウトオブジェクトを整理するために使用するものです。本書の演習の中で適宜使っていきます。



図 9-98 タブコントロール／スライドコントロールのツール

どちらもパネルが重なっているレイアウトオブジェクトです。このパネルの上に更にレイアウトオブジェクトを配置します。配置されているレイアウトオブジェクトは、タブコントロールやスライドコントロールを移動すると一緒に動きます。

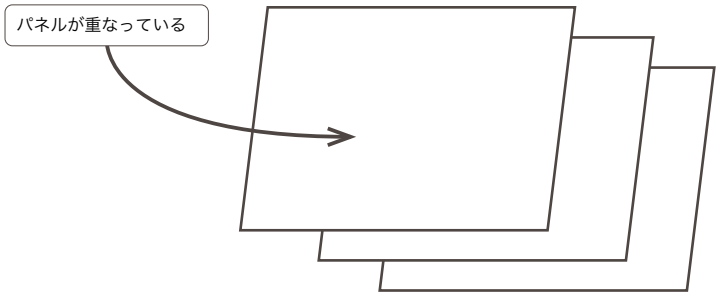


図 9-99 パネルのイメージ

タブコントロールはタブパネル名を上に表示することにより、整理されている内容がわかりやすくなります。スライドコントロールはパネルの名前はありません。代わりにパネルの下にドットがあります。

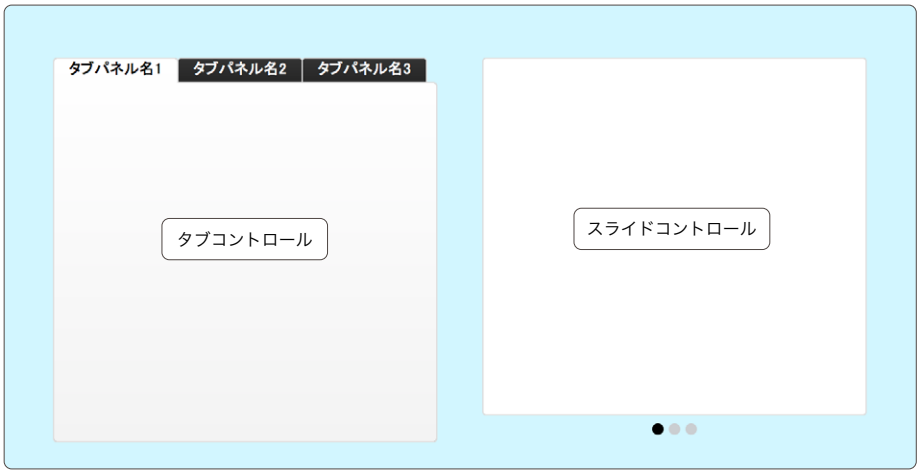


図 9-100 見た目の違い

タブコントロールのタブパネル名
テキスト、計算式、変数が使用可能です。［タブコントロール設定］ダイアログを使用します。

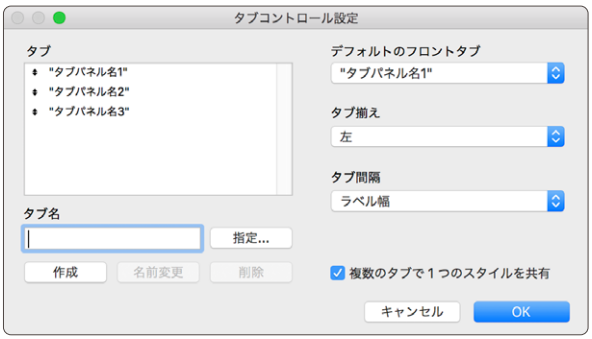


図 9-101［タブコントロール設定］ダイアログ

スライドコントロールとスワイプジェスチャ
[スライドコントロール設定]ダイアログの[スワイプジェスチャを有効にする]が　**オン**　になっているとタッチ対応の機器でパネルをスワイプで切り替えることができます。

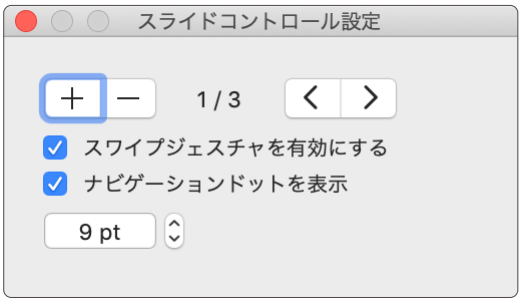


図 9-102［スライドコントロール設定］ダイアログ

スワイプ
タッチ対応のモニタ画面に触っている状態で指を左右に滑らせる操作です。

パネルと条件付き書式
条件付き書式が使用できます。全体ではなくパネルごとの設定になります。

非表示
インスペクタの［次の場合にオブジェクトを隠す］を使用して非表示設定できます。

パネルとアクティブ
現在表示されているパネルがアクティブとなります。印刷や［レコードの保存／送信 ...］の Excel や PDF 形式では、アクティブなパネルの内容が出力されます。

配置
レイアウトオブジェクトがパネルからはみ出していると、正しく表示されない場合があります。パネルの中に、さらにタブコントロールやスライドコントロールを配置できます。

スタイル
詳細なスタイルを持っていて、インスペクタで設定できます。テーマごとにあらかじめスタイルが設定されています。

9.12.4　ポータル

レイアウトに指定されている TO の関連レコードを複数表示するために使用するものです。本書の演習の中で適宜使っていきます。



図 9-103 ポータルのツール

ポータルの中の配置

ポータルに表示するレイアウトオブジェクトは、1 行目に配置します。配置できるのは、

・フィールド
・テキスト
・線／図形
・画像
・ボタン／ボタンバー／ポップオーバーボタン

これらです。次のものは配置できません。

・タブコントロール／スライドコントロール
・グラフ
・Web ビューア
・ポータル



図 9-104 ポータル

9.12.5 Web ビューア

レイアウトに Web サイトを表示するために使用するレイアウトオブジェクトです。入力されているデータを URL の引数として使用することも可能です。



図 9-105 Web ビューアのツール

Web ビューアをレイアウト上に配置すると次図のダイアログが表示され、Web ビューアの設定を行うことができます。

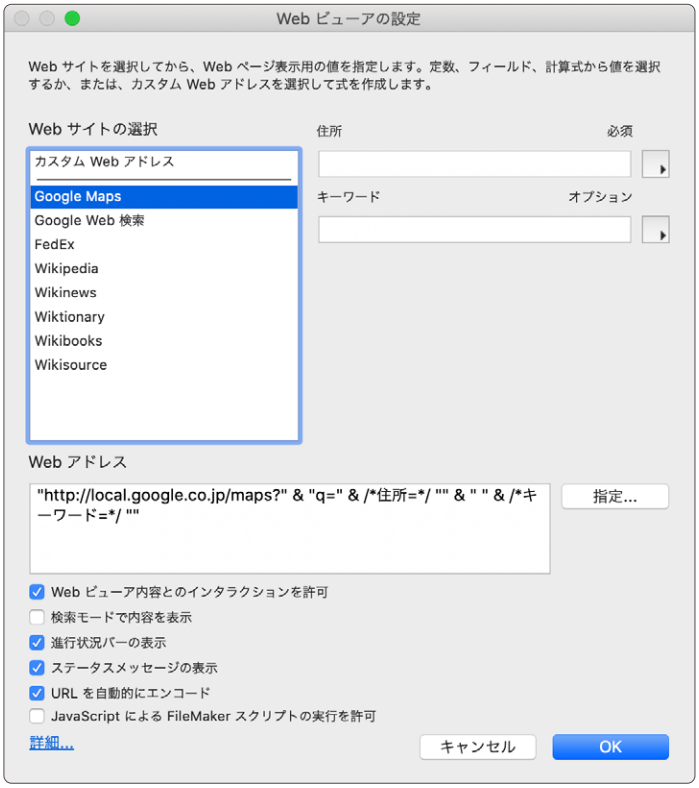


図 9-106［Web ビューアの設定］ダイアログ

Web ビューアは、レイアウト上で Web ブラウザの機能を利用するものです。地図を表示したり Web 検索などが可能です。計算式で「カスタム Web アドレス」を作成することも可能です。なお、インターネット上の Web サイトの表示を前提とした Web ビューアを使用する場合、インターネットに接続できるネットワーク環境が必要です。

データ URL

Web ビューアはインターネット以外の情報の表示が可能です。「data:text/html」という接頭語からはじまる「データ URL」を計算式などで作成しカスタム Web アドレスとして使用します。例えば、HTML や JavaScript をフィールドにテキストデータとして格納して、Web ビューアで表示することが可能です。

URL では次の注意をします。

・URL スキーマが正しいこと　例えば「https://」、「ftp://」、または「file://」
・［URL を自動的にエンコード］が選択されていて 正しくエンコードされていること

Web ビューアと Web ブラウザ
Web ビューアがレイアウト上で動作するのは FileMaker Pro や FileMaker Go をインストールしているパソコンや iPad、iPhone の OS の Web ブラウザを使用しているからです。

そして Web ブラウザの設定も Web ビューアにも反映されます。例えば、JavaScript の有効／無効です。Web ブラウザの設定は、企業や団体としてルールが決まっている場合がありますので、要件定義時に前もってルールの確認をしておきます。

Web ビューアの演習を行います。フィールドに HTML を入力して Web ビューアに表示します。注意点として**「"」、「:」、「/」、「;」など、記号の前後にスペースを入れずに入力**します。

「演習 9_18 から 25」フォルダ

演習 9-18　Web ビューアの動作 1：準備

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「商品管理」レイアウトに切り替えます。スライドパネルが配置されています。

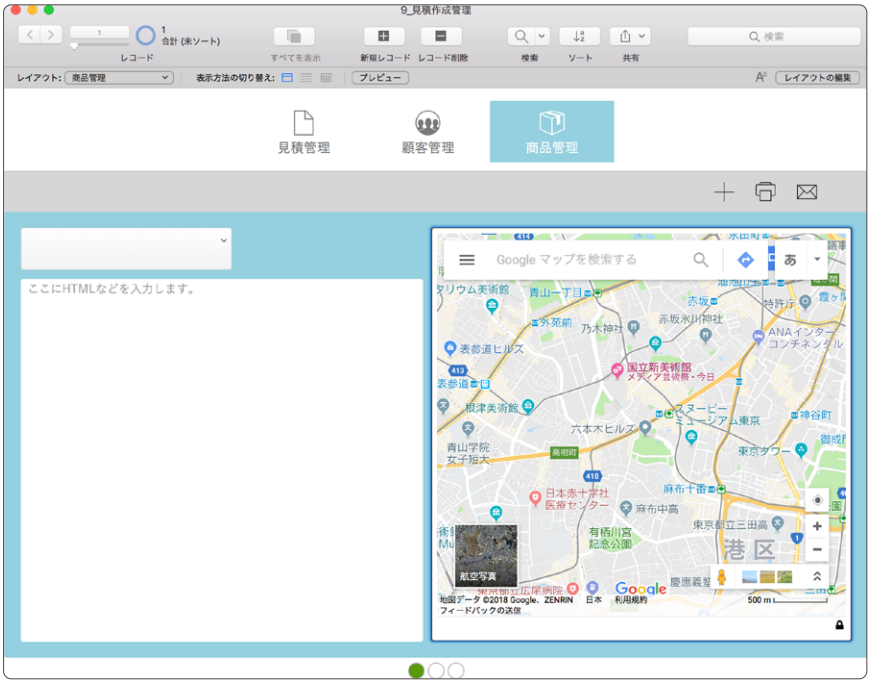


図 9-107「商品管理」レイアウト

(2)　左側のグローバル格納フィールドの HTML が右の Web ビューアに表示されるように設定します。

(3)　レイアウトモードに切り替えて Web ビューアをダブルクリックし［Web ビューアの設定］ダイアログを表示して設定を行います。

「Web アドレス」の右の［指定 ...］をクリックして、以下の式を入力します。

```
"data:text/html," & ユーティリティ管理::g_Webビューアテスト
```

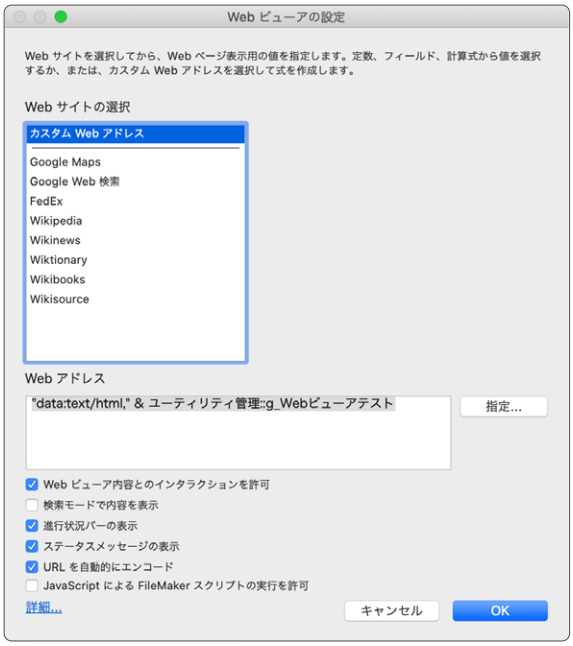


図 9-108［Web ビューアの設定］ダイアログ (data:text/html を使用)

(4)　レイアウトを保存してブラウズモードへ切り替えます。

演習ここまで

動作を確認します。

演習 9-19　Web ビューアの動作 2：表示の確認

- **使用ファイル：9_ 見積作成管理 .fmp12**

左側のグローバル格納フィールドのデータを変えます。フィールドの外でクリックすると、ビューアの内容も変わることを確かめてください。例えば次を入力してみてください。

```
あいうえお
```

そのまま表示されます。次は、

```
<h1>あいうえお</h1>
```

と、入力すると、文字が大きくなります。また次のように入力すると、文字が赤くなります。

```
<p style="color:#ff0000">あいうえおかきくけこ</p>
```

演習ここまで

違う設定してみます（記号の前後にスペースを入れずに入力します）。

演習 9-20　Web ビューアの動作 3：設定

- **使用ファイル：9_ 見積作成管理 .fmp12**

［Web ビューアの設定］ダイアログで設定します。
「Web アドレス」の右の［指定 ...］をクリックして変更します。

ユーティリティ管理 **::** g_Web ビューアテスト

演習ここまで

動作を確認します。次の演習にはインターネットの接続環境が必要です。

演習 9-21　Web ビューアの動作 4：表示の確認

- **使用ファイル：9_ 見積作成管理 .fmp12**

左側のフィールドのデータを変えてフィールドの外でクリックすると、右側のビューアの内容も変わることを確かめてください。例えば次の入力では Web サイトが表示されます。

```
https://www.claris.com/ja
```

演習ここまで

また違う設定をします。フィールドの画像を表示します。

演習 9-22　Web ビューアの動作 5：設定

- **使用ファイル：9_ 見積作成管理 .fmp12**

［Web ビューアの設定］ダイアログで「Web アドレス」の右の［指定 ...］をクリックして変更します。

```
"" & Base64Encode
  (  ユーティリティ管理_商品｜選択 :: 商品画像  )
```

演習ここまで

動作を確認します。

演習 9-23　Web ビューアの動作 6：表示の確認

- **使用ファイル：9_ 見積作成管理 .fmp12**

左側のポップアップメニューで、「30002　ノート　A4 サイズ　B 罫（5 冊）」を選択します。他も選択して画像が Web ビューアに表示されるのを確認します。

演習ここまで

やってみよう

［Web ビューアの設定］ダイアログの現在の設定をすべて消して、以下の HTML を直接記述したらどうなるでしょうか。演習の最初の表示と同じになります。

```
"<html><body><p>あいうえお</p></body></html>"
```

さらに違う設定をします。次の演習にはインターネットの接続環境が必要です。

演習 9-24　Web ビューアの動作 7：設定

- **使用ファイル：9_ 見積作成管理 .fmp12**

［Web ビューアの設定］ダイアログで設定します。

・［Web サイトの選択］は、「Google Web 検索」を選択します。
・［クエリー］は「ユーティリティ管理 _ 商品｜選択」TO の「商品名」フィールドを選択



図 9-109［Web ビューアの設定］ダイアログ (Google Web を検索を使用)

［Web アドレス］に次の URL が入力されます。

```
"https://www.google.co.jp/search?" & "q="
& ユーティリティ管理_商品｜選択::商品名
```

［OK］をクリックして設定を完了し、レイアウトを保存してブラウズモードへ切り替えます。

演習ここまで

動作を確認します。

演習 9-25　Web ビューアの動作 8：表示の確認

- **使用ファイル：9_ 見積作成管理 .fmp12**

左上のポップアップメニューで商品を切り替えると検索される内容も変わることを確かめてください。

演習ここまで

グラフオブジェクトについては第 12 章で解説します。

9.13 テーマとスタイル

新しいファイル､新しいレイアウトを作成すると「**テーマ**」と「**スタイル**」が割り当てられています。これらは、レイアウトモードで設定、変更します。

・レイアウトバーでテーマ名
・インスペクタの「スタイル」タブで、名とスタイル名
・インスペクタの「外観」タブで、スタイルの詳細な書式

9.13.1 テーマを設定する

レイアウトバーの［テーマ：］をクリックすると、［テーマの変更］ダイアログが表示されます。



図 9-110 レイアウトバー

新しいファイルを作成すると「エンライトンド」が割り当てられます。必要に応じて変更できます。印刷に使用するレイアウトには「ミニマリスト」が便利です。テーマ名に「タッチ」と付いているテーマは、モバイルデバイスを想定しています。

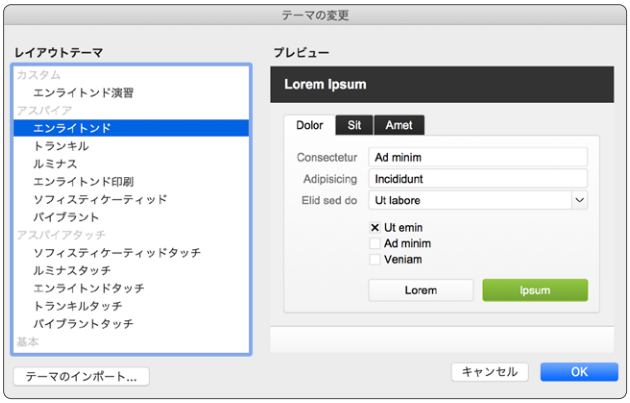


図 9-111［テーマの変更］ダイアログ

テーマを管理する

［ファイル］–［管理］–［テーマ ...］メニューで［テーマの管理］ダイアログが表示されます。

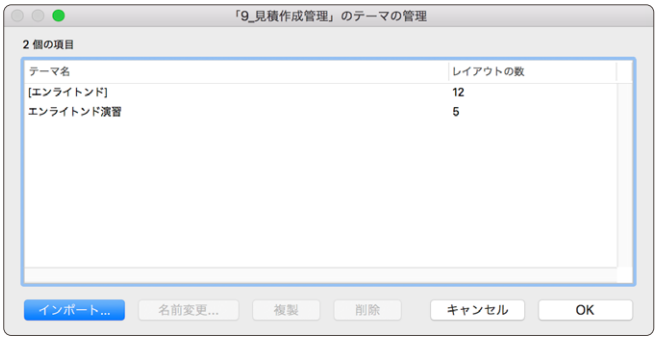


図 9-112［テーマの管理］ダイアログ

テーマの名前の変更、インポートなどが可能です。1 つのファイルでたくさんのテーマを使うと管理が煩雑になる場合がありますので注意してください。

9.13.2 スタイルを設定する

各レイアウトオブジェクトには「デフォルト」スタイルがあり、バリエーションとして他のスタイルもあらかじめ用意されています。テーマのすべてのスタイルを確認するには、

・インスペクタの「スタイル」タブで［すべてを表示］のチェックを **オン** にすると確認できます。



スタイルを割り当てる
レイアウトオブジェクトが選択された状態で、スタイル名をクリックすると設定されます。

スタイルの書式変更
現在割り当てられているスタイルの詳細な書式は、「外観」タブで変更することが可能です。変更があると○に▼のボタンが赤くなります。

9.13.3 スタイルとテーマの保存

内容の変更後、○に▼のボタンをクリックして表示されるメニューから実施します。なお、テーマやスタイルを、どのように保存するかにより、その反映範囲が決まります。

・あるレイアウトオブジェクトだけの固有スタイル

この場合、テーマにもスタイルにも保存しません。

・同じスタイルが割り当てられている同じレイアウト上の他のレイアウトオブジェクトにも反映させたい

は、スタイルに保存することで可能です。なお、既存のスタイルから新しいスタイルを作成することも可能です。右図を参照してください。

・スタイルの変更を別のレイアウト上でも反映させたい

は、スタイルとテーマの両方に変更を保存します。次の順番でないといけません。

(1) スタイルの変更を保存する
(2) テーマに変更を保存する

テーマを変更すると用意しているテーマがコピーされて、そのコピーに変更が保存されます。「エンライトンド」という名前は同じですが「カスタムテーマ」になります。本書では「エンライトンド演習」という名前で保存しました。

9.13.4 コピーと貼り付け（ペースト）

スタイルや状態のコピーと貼り付け（ペースト）は、レイアウトで一貫性を持った表示がされることで、ユーザが見やすくわかりやすい画面を作成するために用意されています。

スタイルのコピーと貼り付け（ペースト）

スタイルは、コピーと貼り付けも可能です。「外観」タブの［スタイル:］の右下にある上段スポイトアイコンや［編集］メニューから可能です。

方法

(1) 左のスポイトアイコンで、レイアウトオブジェクトのスタイルをコピー
(2) 右のスポイトアイコンで、レイアウトオブジェクトのスタイルを貼り付け

・同じレイアウトの同じタイプのレイアウトオブジェクトに貼り付けすると、スタイルが変更されます。
・違うタイプのレイアウトオブジェクトに貼り付けすると、スタイルは同じで内容のみが変更されます。
・違うレイアウトのレイアウトオブジェクトに貼り付けすると、タイプが一緒でも「デフォルト」スタイルが割り当てられ、変更された状態になります。

状態のコピーと貼り付け（ペースト）

スタイルの状態についてもコピーと貼り付けが可能です。「外観」タブの［スタイル：］の右下にある下段スポイトアイコンや［編集］メニューから可能です。

方法

(1) 左のスポイトアイコンで、レイアウトオブジェクトの状態をコピー
(2) 右のスポイトアイコンで、レイアウトオブジェクトの状態を貼り付け

別のレイアウトオブジェクトタイプに状態を反映させたいときや、同じレイアウトオブジェクトタイプで別のスタイルに状態を反映させたい場合に使用します。



9.13.5 レイアウトオブジェクトの表示の優先順位

スタイル以外に、レイアウトオブジェクトの色や塗りつぶし、線などを設定するものには条件付き書式があります。これらの表示の優先順位を覚えていると便利です。

条件付き書式は、常に優先順位の一番上です。ただし状態の「フォーカス」時には、フォーカスに設定されている内容が優先されます。

(1) 条件付き書式
(2) ローカル書式
(3) レイアウトに保存されているスタイル（変更されたもの＝カスタムスタイル）
(4) テーマに保存されているスタイル（変更されたもの＝カスタムテーマ）
(5) レイアウトのデフォルトスタイル
(6) テーマのデフォルトスタイル

通常は、この優先順位です。上の順位ほど、ブラウズモードでのレイアウト表示に優先されます。

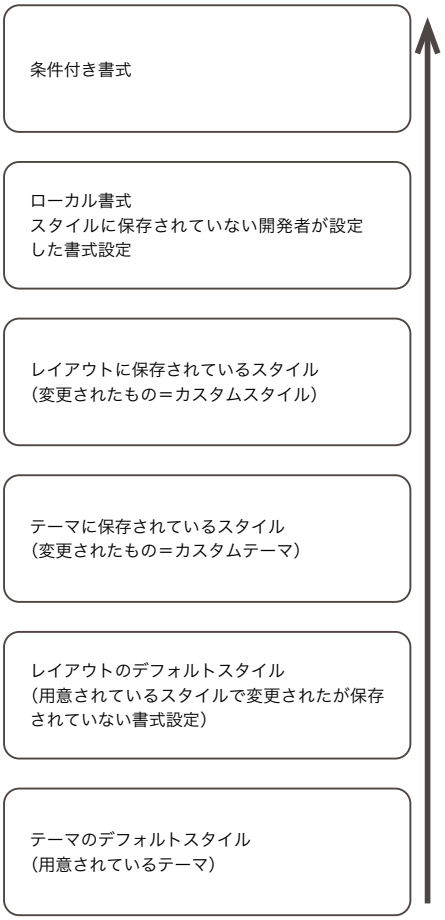


図 9-113 優先順位（上の方が高い）

やってみよう

レイアウトで設定を色々と試してみましょう。

これまでの演習ファイルを閉じて次の演習ファイルを開いてください。レイアウトを用意しました。

（ここから、本書に付属している演習ファイルの「演習 9_26 から 28」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

9.14　画面の構成：マスタ／詳細レイアウト

これまで、ある 1 つのテーブルデータについて「リスト形式の一覧画面」と「フォーム形式の詳細画面」の 2 画面を行き来することで取り扱う前提のレイアウトを作成しました。

（1）複数の対象レコードを確認する（リスト形式）
（2）必要であれば任意の 1 件のレコードを選択して詳細な情報を確認する（フォーム形式）
（3）また、複数の対象レコードを確認する（リスト形式）
（4）必要であれば任意の ...（フォーム形式）

この行き来を省略するために、レイアウトオブジェクトをうまく使って 1 画面に一覧と詳細を配置することが可能です。「**マスタ／詳細**」レイアウトと呼びます。次図のようなレイアウトです。

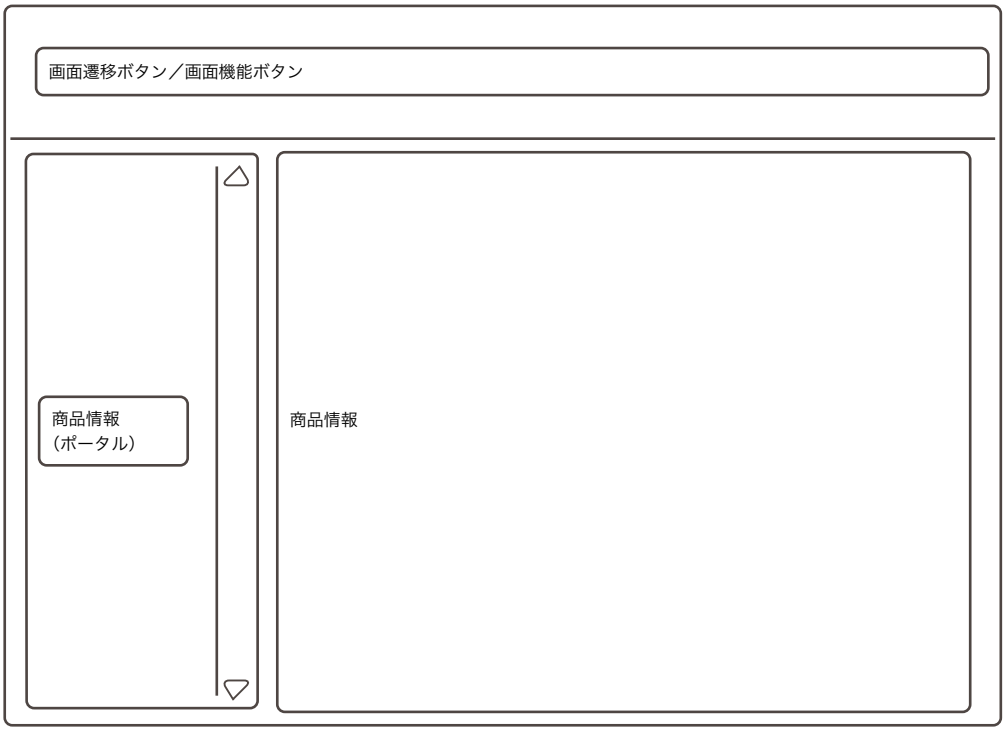


図 9-114 マスタ／詳細レイアウトのイメージ

レイアウトの「ボディ」パート上の左あるいは右にポータルが配置されていて、ポータル行をクリックするとそのデータの詳細が横に表示されるものです。

現在、商品登録は 50 件です。このぐらいの件数だったら、ポータルで上下にスクロールすることでどのような商品が登録されているか、などを確認することが可能です。

マスタ／詳細を「商品管理｜マスタ詳細」レイアウトに作成します。まず関連などを確認します。

「演習 9_26 から 28」フォルダ

演習 9-26　フォーム表示レイアウトの作業：「商品管理」の作業 1（マスタ／詳細）

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「商品管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。

(2)　レイアウトには、左側にポータルを配置するための空間があります。
右側のフィールドはすべて「商品管理」TO のものです。

(3)　このレイアウトに指定されている TO は「商品管理」です。



図 9-115「商品管理｜マスタ詳細」レイアウトに指定されている TO 名

(4)　［データベースの管理］ダイアログを開いて「リレーションシップ」タブをクリックします。
下にスクロールすると「商品管理」TO が確認できます。なにも関連は設定されていません。

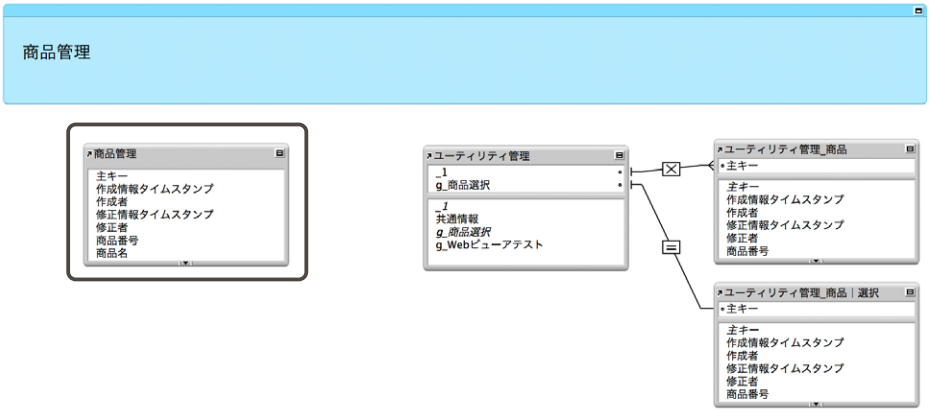


図 9-116「商品管理」TO

(5)　確認後、［キャンセル］をクリックしてダイアログを閉じます。

演習ここまで

ポータルを作成します。

演習 9-27　フォーム表示レイアウトの作業：「商品管理」の作業 2（マスタ／詳細）

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　「商品管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。

(2)　ポータルを作成します。手順は省略します。

a.　「ボディ」パートの左側に縦長の長方形でポータルを作成します。

b.　［ポータル設定］ダイアログが表示されます。次の設定をします。

・［レコードを表示：］は　　現在のテーブル「商品管理」

・［行数］は　　「10」

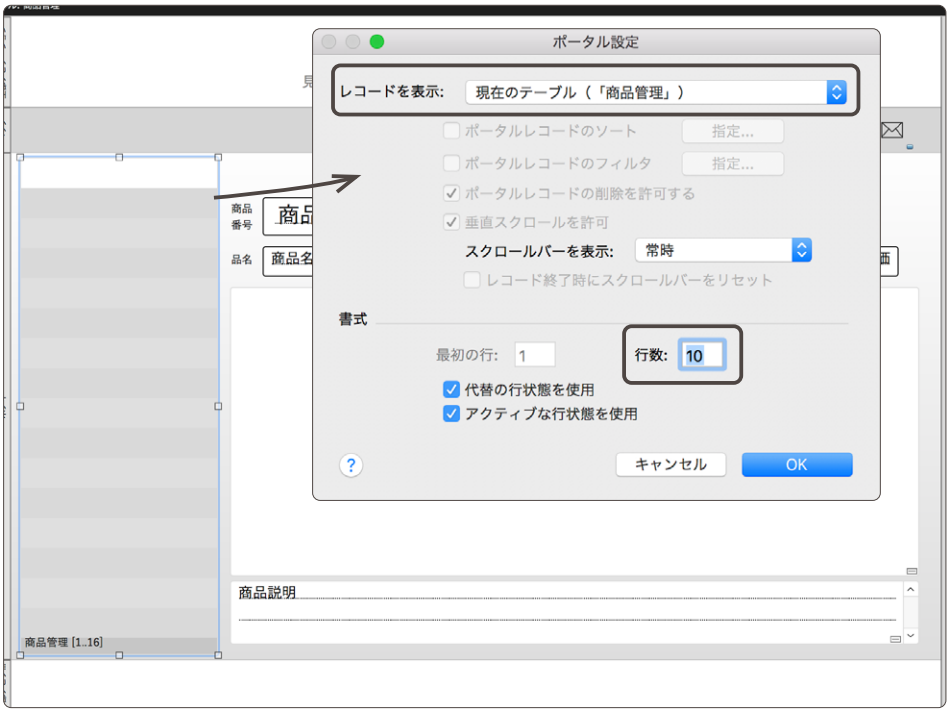


図 9-117 ポータルを作成

c.　［OK］をクリックして、次のフィールドを選択します。

・「c_ 商品番号と商品名」

d.　［OK］をクリックします。

(3)　商品名をすべて表示するためにポータルを調整します。1 行目のハンドラをドラッグして 1 行目を高くします。

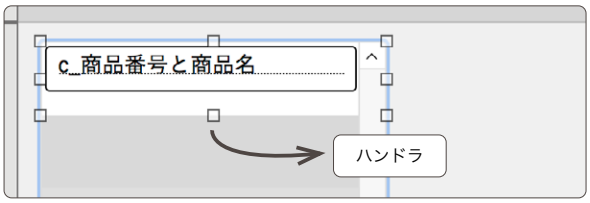


図 9-118 ハンドラ

(4)　フィールドを調整します。

a.　行の高さを調整します。



図 9-119 調整後（レイアウトモード）

b.　ポータルをクリックします。

c.　右パネルのインスペクタの「位置」タブの[自動サイズ調整]の「下」をチェックします。

(5)　レイアウトを保存してブラウズモードへ切り替えます。



図 9-120 調整後（ブラウズモード）

(6)　動作を確認します。

左側のポータルの商品名のどれかをクリックしてください。右側の表示が変わります。

演習ここまで

関連などはどうなったでしょうか。確認します。

演習 9-28　フォーム表示レイアウトの作業：「商品管理」の作業 3（マスタ／詳細）

- **使用ファイル：9_ 見積作成管理 .fmp12**

(1)　[データベースの管理] ダイアログを開いて「リレーションシップ」タブをクリックします。下にスクロールすると「商品管理」TO が確認できます。**なにも関連が設定されていないまま**です。

(2)　確認後、[キャンセル] をクリックしてダイアログを閉じます。

(3)　[スクリプト]–[スクリプトワークスペース ...] メニューを選択して、[スクリプトワークスペース] ダイアログを開きます。「ボタン仮割り当て」以外には**ポータルに関連した新しいスクリプトは作成されていません。**

(4)　ダイアログを閉じます。

(5)　レイアウトモードに切り替えます。演習 9-27 で作成した**ポータルには、ボタンはありません。**

演習ここまで

簡単にマスタ／詳細レイアウトが作成できました。レイアウトを行き来せずにユーザが作業することができる点で便利です。モバイルデバイスでは環境（通信速度など）によっては画面を切り替える遷移が少ないほうが良いこともあります。

9.14.1　ポータルで現在のテーブルを選択

演習 9-27 の作業後、特に機能を作成していませんがポータルの行をクリックするだけで右側に詳細情報が表示されるようになりました。詳しく確認します。

(1) 1 件目のレコードに移動します（商品番号 30001 の情報が表示されています）。
(2) 左側のポータルで、4 行目をクリックします。
(3) 右側に商品番号 30004 の情報が表示されます。
(4) ステータスツールバーを確認してください。4 件目のレコードに移動しています。

他のポータル行をクリックしても同様にレコード移動が行われます。つまり、次の機能ができました。

・ポータル行をクリックすると、クリックしたレコードに移動する

この機能は、ポータルの作成時に [レコードを表示：] で「現在のテーブル」を選択した場合のみ動作します。マスタ／詳細レイアウトの作成にとても便利です。なお、操作にいくつかの特徴があります。

9.14.2 設定によるポータルの動作の違い

［レコードを表示：］で「現在のテーブル」を選択すると、次のポータル設定は変更できません。

・ポータルレコードのソート **なし**
・ポータルレコードのフィルタ **なし**
・ポータルレコードの削除の許可 **オン**
・垂直スクロールを許可 **オン**
・レコード終了時にスクロールバーをリセット **オフ**

これらの設定は、別の方法で実現されます。次表は設定による各種動作の違いです。

レイアウト設定 →	**「現在のテーブル」のポータル**	**「関連テーブル」のポータル**
レコードをソート	現在のソート順でポータル内の行を表示	現在のソート順とは異なる順にポータル内の行をソートできる
レコードのフィルタ	現在の対象レコード内のすべてのレコードを表示	現在の対象レコードとは異なる内容でポータル内の行をフィルタできる
行をクリック	クリックしたポータル行をアクティブにしてから対応するレコードに移動し、そのレコードが現在のレコードになる	クリックしたポータル行をアクティブにする （別のレコードに移動するには、ポータル内のオブジェクトにボタンで機能を追加する必要がある）
行を追加または削除	現在の対象レコード内のレコードが追加または削除される	関連テーブル内のレコードが追加または削除される
検索条件を表示	各検索条件が作成順にポータル内の行に表示	常に現在の検索条件（関連テーブル内のレコード）としてポータル内の行が表示
変更を確定	別のポータル行を選択したときに現在のレコード内のすべての変更が確定される	レイアウト上の他のオブジェクトに移動する前に、現在のポータル行でのみ操作が実行される
Tab キーでフィールド間を移動 **データを検索および置換** **スペルチェック** **レコードをコピー** **データを Excel ファイルとして保存** **……など**	レイアウト上の他のオブジェクトに移動する前に、現在のポータル行でのみ操作が実行される	レイアウト上の他のオブジェクトに移動する前に、すべてのポータル行で順次操作が実行される

いくつかの動作を確認します。

ソートと現在のテーブルのポータル
現在のソート順がポータルに適用されます。

（1）「c_ 商品番号と商品名」フィールドの降順でソートします。
（2）ポータルの表示を確認します。
（3）選択しているポータル行は変わりませんが、表示順は「c_ 商品番号と商品名」フィールドの降順に変わっています。

検索モードと現在のテーブルのポータル
ポータル内に表示されるレコードを変更するには、検索をします。

（1）検索条件に切り替えます。
（2）右側の「商品番号」フィールドに「30001...30002」と入力します。
（3）ステータスツールバーで「検索実行」をクリックします。
（4）検索結果は、2 件です。ポータル行も対象レコードと同じく 2 行です。

ポータルに表示されるのは、現在の対象レコードであることがわかります。

レコード削除と現在のテーブルのポータル
レイアウト上でレコードを削除すると、当然ですがポータル上の当該行もなくなります。

スクリプトトリガなど
ポータル行をクリックするとレコード移動が起きるため、レコード確定なども起きます。OnRecortCommit や OnRecordRoad などのスクリプトトリガが実行されます。

これまでの演習ファイルを閉じて次の演習ファイルを開いてください。

（ここから、本書に付属している演習ファイルの「9 章完了時」フォルダの中の「9_ 見積作成管理 .fmp12」ファイルをベースに進めます）

9.14.3 アクティブな行

ポータルで、現在選択されている行の番号が結果となる関数が用意されています。

「商品管理｜マスタ詳細」レイアウトで確認します。ポータルに ⓘ のアイコンのボタンが表示されています。別のポータル行をクリックするとそのポータル行に表示されます（どの行がアクティブかわかりやすいようにスタイルも少し変更しています）。

ボタンには、右パネルのインスペクタの「データ」タブの［動作］の［次の場合にオブジェクトを隠す］に計算式が設定されています。

```
Get ( レコード番号 ) ≠ Get ( アクティブレコード番号 )
```

```
Get ( アクティブレコード番号 ) 関数
```

Get (アクティブレコード番号) 関数は、現在の対象レコード内の現在のレコードを表す番号が結果になります。対して、Get (レコード番号) は、現在の対象レコード内のレコードの位置（何番目か）を表す番号が結果となります。

9.14.4 関連テーブルのポータルのマスタ／詳細

別の方法でマスタ／詳細レイアウトを作成している例を確認します。「商品管理｜ X 関連」レイアウトに切り替えてください。次図の内容が表示されます。

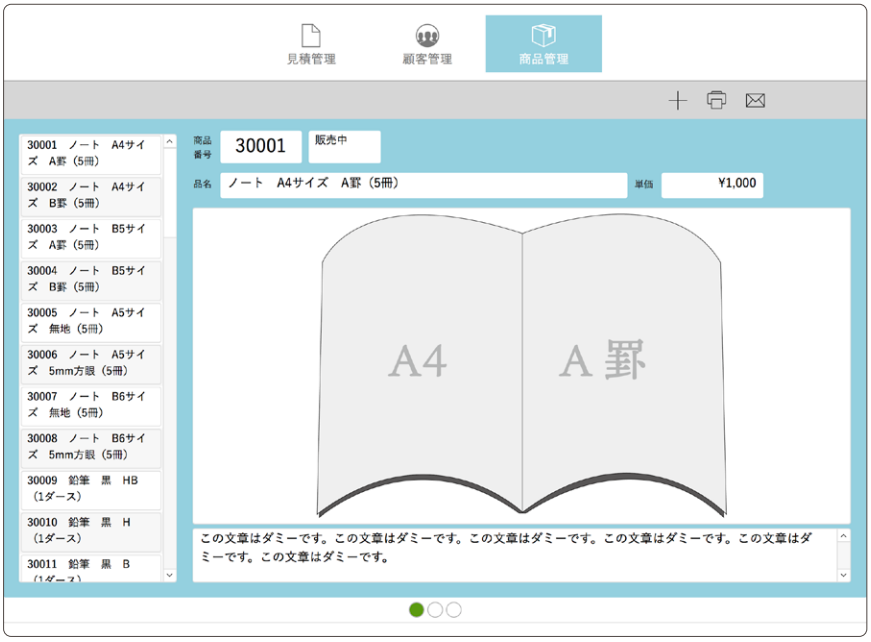


図 9-121 商品登録画面例 1

左側のポータル行をクリックすると、右側の詳細表示が変わります。一見、「現在のテーブル」のポータルと動作は変わりません。しかし、仕組みが違います。

「商品管理｜ X 関連」レイアウトに指定されているのは「ユーティリティ管理」TO です。

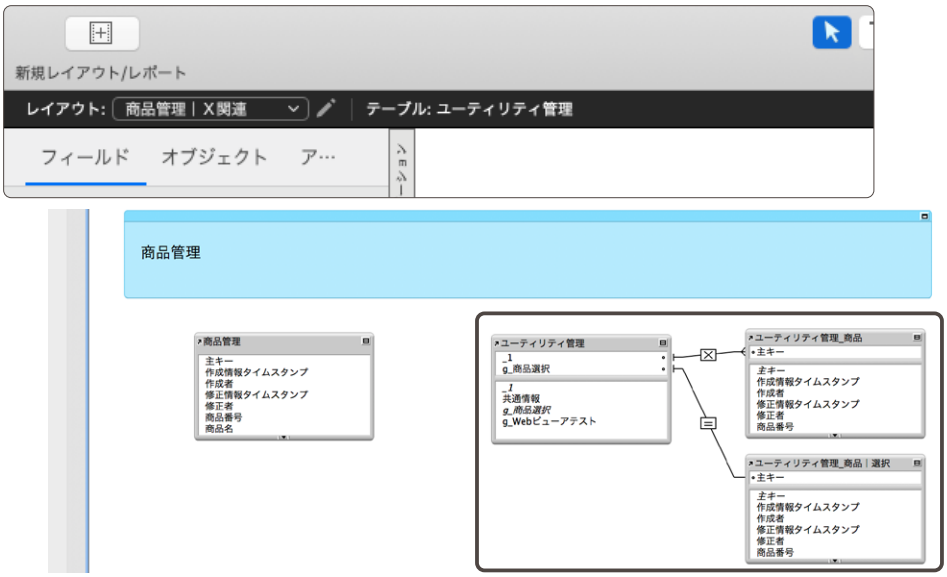


図 9-122「商品管理｜ X 関連」レイアウトの TO とリレーションシップグラフの TOG

このレイアウトには、グローバル格納フィールドを照合フィールドとしている関連が使われています。2 種類あります。

・左側の商品のポータルを表示するための関連　　照合条件が「X」
・右側に選択された 1 商品の詳細を表示するための関連　照合条件が「＝」

「ユーティリティ管理」TO と「ユーティリティ管理 _ 商品」TO の［リレーションシップ編集］ダイアログを開くと「X」の記号を使った関連が定義されています。左側のポータルに全商品を表示してユーザが選べるようにするためです。

X の関連（リレーションシップ）
「X（デカルト積）」の演算子を使用すると照合フィールドのデータに関係なく左側の TO のすべてのレコードを右側の TO のすべてのレコードに一致させる」という『なんでも OK』な関連を定義したことになります。

「ユーティリティ管理」TO と「ユーティリティ管理 _ 商品｜選択」TO の［リレーションシップ編集］ダイアログを開くと「＝」の関連が定義されています。

ポータルに配置されているフィールドにはボタンが設定されています。ユーザが任意のポータル行をクリックすると「商品情報」テーブルの「主キー」フィールドのデータが「g_ 商品選択」フィールドへ入力されることで画面右側に商品の情報が表示される、という仕組みです。

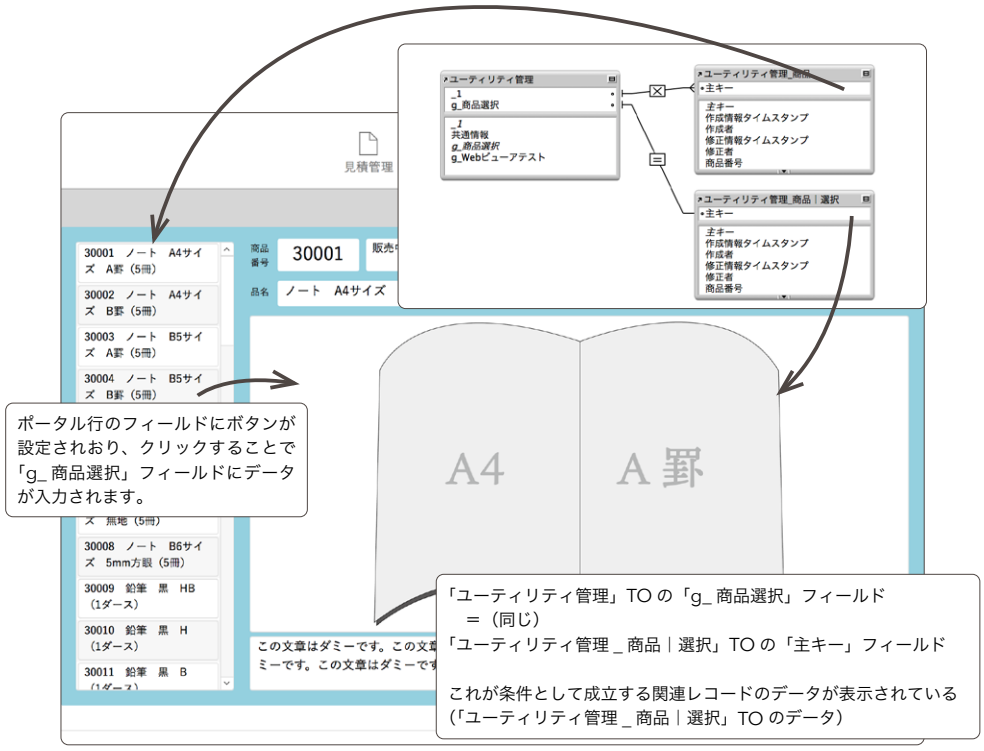


図 9-123「商品管理｜ X 関連」レイアウトの仕組み

現在の登録件数（50 件）程度であれば、ポータルではなくてポップアップメニューで商品の選択をすることも可能です。

やってみよう

どのようなスクリプトステップがポータルのフィールドボタンに設定されているか確認してみましょう。

「商品管理｜X 関連」レイアウトのスライドパネルの真ん中の○をクリックすると、次図の内容が表示されます。

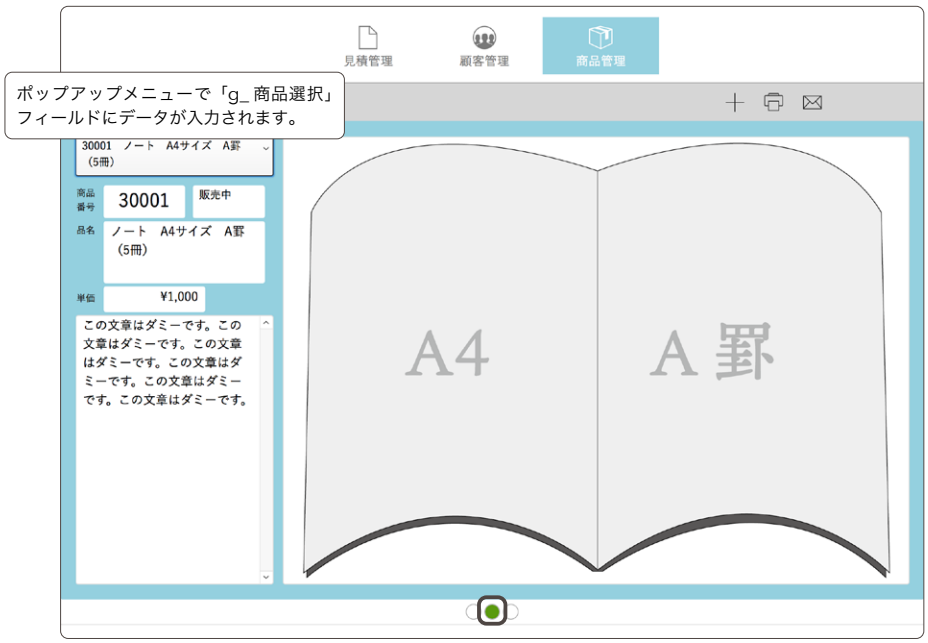


図 9-124 商品登録画面例 2

左上のポップアップメニューで商品名のどれかを選択してください。表示が変わります。同じ仕組みを使っています。

2 つのレイアウト例を確認しました。

スライドパネルとレイアウト

スライドパネルは、1 つのレイアウト上にも関わらず、別のレイアウトに切り替えたかのようにレイアウトオブジェクトを配置仕分けることが可能です。

「見積作成管理システム」では、商品登録に必要なフィールドは、6 個です。右側の詳細表示部分には余裕があるので項目を増やすことが可能です。マスタ／詳細レイアウトを活用できそうです。

「リスト形式の一覧画面」と「フォーム形式の詳細画面」の 2 画面は、基本の画面遷移です。間違っているわけではありません。『ユーザにとってどのような画面が使いやすいか』を考えるとき、用意されているたくさんのレイアウトオブジェクトを知っていれば、上手に利用することで使いやすくてわかりやすい画面構成を考える助けになります。

演習ファイルを閉じてください。

第 10 章 関数と計算式

カスタム App の作業では、様々な場所で関数と計算式を使います。

10.1 計算式は論理

これまで、あらかじめ要件定義と設計を実施しておく利点を何度か述べました。計算式も合意した要件の論理に従って記述し、計算結果を利用します。計算式を活用するとカスタム App の開発作業が一気にステップアップします。

10.1.1 計算式の基本

フィールド、レイアウト、スクリプトなどで計算式を使います。

・「**関数**」や「**演算子**」を使って記述された式

式は、「**評価**」されて「**計算結果**」を返します。結果は、そのまま利用されたり、結果を判断（分岐など）して機能に利用されます。式の内容は、主に次のようなもので、組み合わせた記述も可能です。

・単純な四則演算（+ − ÷ ×）
・演算子を使った式
・関数を使った式

FileMaker Pro には多数の関数が用意されています。次のような情報も計算結果としてわかります。

・『現在』の日付や時刻
・『現在』のユーザのクライアントの種類
・『現在』のユーザのクライアントの OS

例えば、『現在のデータ』や、ユーザが入力したデータを使った計算式を要件の論理に従って記述し、その結果を、

・レイアウト上に表示する内容を変化させる
・機能における分岐を判断する

などの、動的（ダイナミック）な処理に利用できます。

演習ファイルを開いてください。これまでの計算式がどのような内容だったか確認します。

（ここから、本書に付属している演習ファイルの「10 章開始時」フォルダの中の「10_ 見積作成管理 .fmp12」ファイルをベースに進めます）

```
商品番号 & "  " & 商品名
```
→「商品番号」と「商品名」フィールドのデータを結合する式（計算タイプフィールド）

単純に&の結合演算子でフィールドのデータを並べて1つの値にするためです。ユーザの見やすさを考えたものです。商品を選択するための「商品リスト」値一覧に使用しました。

```
Date ( Month ( 発行日 )+1 ; Day ( 発行日 ) ; Year ( 発行日 ) )
```
→「期限日」フィールドの入力値の自動化オプションの内容。
ある日付の1ヶ月後の日付が計算値自動入力される（日付タイプフィールド）

Date関数を使っています。「見積作成管理システム」が対象にしている業務のルールです。
「見積書では、原則として発行日の1ヶ月後が有効期限である」

```
見積管理::見積状況 ≠ "結果確定済"
```
→ あるフィールドのデータが別のフィールドのデータと等しいかどうかを得る
1（真 = True ）か0（偽 = False）の論理式（[オブジェクトを隠す]）

「見積詳細」レイアウトの「見積結果」の表示／非表示を[オブジェクトを隠す]で判断しています。「結果確定後に注文有無を入力する」というユーザのデータ入力のサポートを考えたものです。

他にもいくつか設定されています。このように計算式は様々な場所で使われます。

計算式とフィールドタイプ
すべてのタイプが計算式に使用できます。ただし、関数によっては使用できないタイプがあります。

計算結果タイプ
計算結果のデータタイプを指定できます。あるいは自動的にタイプが決まります。[計算式の指定]ダイアログの下部に、計算結果のタイプを指定する場所があります。適切に指定されていないと意図しない結果になる可能性があるため気をつけます。次のいずれかです。

- テキスト
- 数字
- 日付
- 時刻
- タイムスタンプ
- オブジェクト

必要な計算結果のタイプ
計算式の指定箇所によりタイプが自動で設定され変更できません。例えば、プレースホルダテキストを計算式で設定する場合、必要な計算結果のタイプは「テキスト」になり[オブジェクトを隠す]の計算式では、「論理値」（後述します）となります。

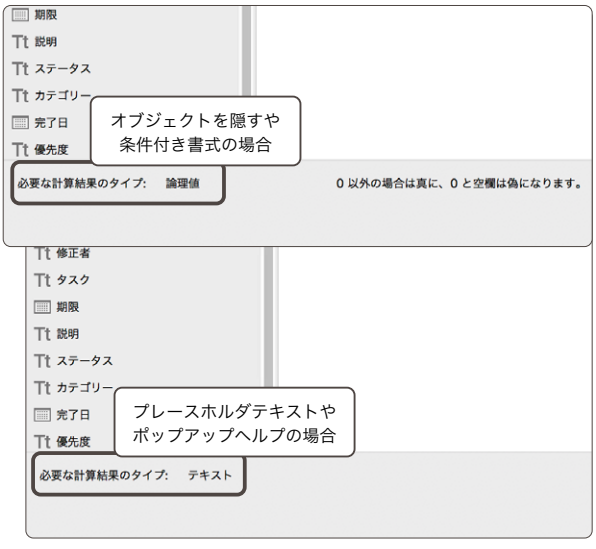


図 10-1 必要な計算結果タイプが決まっている例

10.1.2 評価

計算式が「**評価**」されることで計算結果を得られます。一体いつ評価されるのでしょうか？

評価の優先順位

計算式は左から右に評価され、加算と減算よりも乗算と除算が先に実行されます。ただし「( )」を使用すると、カッコ内の式が先に評価されます。

※ さらに詳しい演算子の評価順位などはヘルプを参照してください。

計算タイプフィールドに設定された計算式の評価

索引が保存か非保存かで評価のタイミングが異なります。

索引が非保存

計算した結果を保持しないため、そのフィールドのデータを参照（レイアウト上に表示する、検索するなど、そのフィールドを使用すること）するたびに毎回計算式を評価し直します。つまり結果の値が評価されるタイミングは「**参照時に常に**」ということになります。

索引が保存

一度評価が行われたら、その結果を格納しておき、そのフィールドのデータが参照された時に格納している値を表示します。つまり表示するだけでは再評価（再計算）されません。

索引の保存と再評価（再計算）

では、索引が保存の場合、いつ再計算されるのでしょうか。再評価のタイミングは原則として、

・計算式で使用されているフィールド（「被計算フィールド」とも呼びます）のデータが変更された時

です。ただし、次のルールがあります。

・被計算フィールドが『同じテーブルのフィールドで索引を持つ』場合だけ、再計算される

さらに注意することがあります。

・「Get (日付)」や「Get (タイムスタンプ)」などの取得関数は、日付や時刻の変化に従って再評価されない

と、いうことです。

他のタイプのフィールドの計算式の評価

計算タイプのフィールド以外の計算式の評価のタイミングは、『それが必要とされるとき』です。計算式の評価のタイミングを最小限にする最適化をおこなうことでパフォーマンス全体に影響を及ばさないように配慮されています。

式を評価できないと「?」が結果として返ります。原因を探るためにエラーコードを知りたい場合、次の関数を使います。詳しくはヘルプを参照してください。

```
EvaluationError ( 式 ) 関数
```

10.2 ［計算式の指定］ダイアログ

計算式の作成は、主に［計算式の指定］ダイアログで実施します。

10.2.1 計算式の作成

計算式を指定できる箇所では、鉛筆のアイコン、fx マーク、［指定…］などの、ボタンやメニューをクリックすることなどにより表示されます。下図は、計算タイプフィールドに計算式を指定する場合のダイアログです。

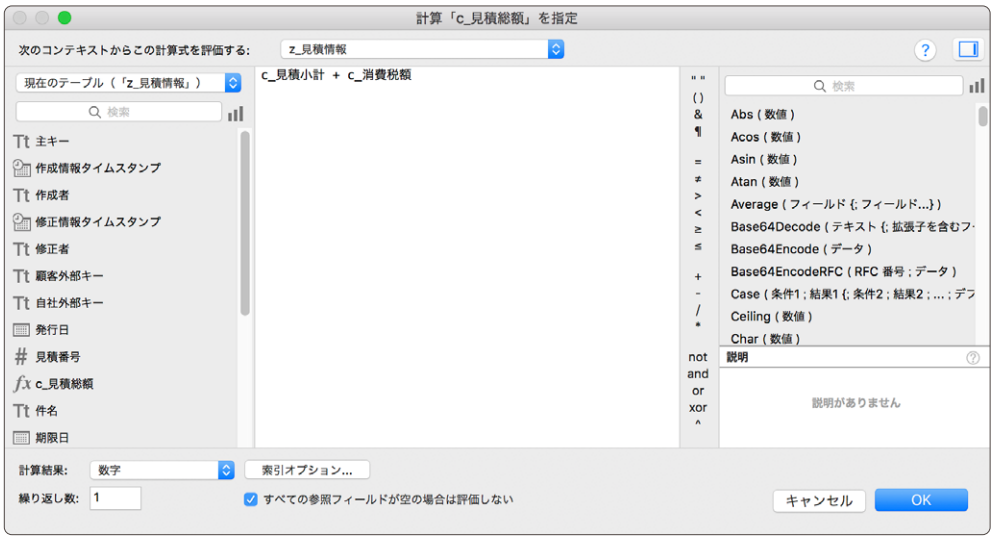


図 10-2［計算式の指定］ダイアログ

コンテキストの指定

主に、計算タイプフィールドやフィールドオプションで計算式を記述する場合に、［計算式の指定］ダイアログに［次のコンテキストからこの計算式を評価する］が、表示されます。ポップアップメニューで TO 名（コンテキスト）を指定します。

これは、実体テーブルの計算式をリレーションシップグラフのどの TO（仮想テーブル／コンテキスト）からの論理で評価するのか、を指定する必要があるためです。

計算式に、そのテーブルのフィールドのみ、あるいは、なにもフィールドを使わないのであれば、どのコンテキストを選んでも問題ありません。しかし関連テーブルのフィールドが計算式に使われる場合、コンテキストが正しく指定されていないと正しく評価できません。

計算式の記述が終わって［OK］をクリックすると、計算式の「**構文**」にエラーがないかチェックされます。エラーがあるとメッセージが表示されます。

構文

あらかじめ決まっている文法や書式です。適切でないと計算式の結果が出せません。スクリプトの記述も同様に構文があります。

10.2.2 フィールドパネル

計算式で使用するフィールドを指定します。

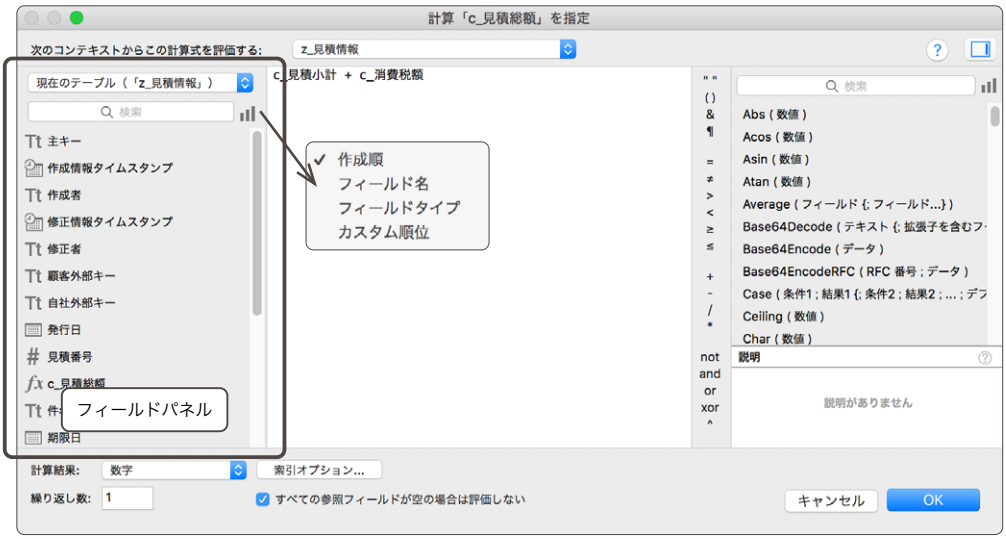


図 10-3［計算式の指定］ダイアログ ; フィールドパネル

TO の選択

上部に TO の切り替え用プルダウンリストがあります。

注意が必要です。［次のコンテキストからこの計算式を評価する］で指定されている TO 名とフィールドパネルで指定した TO 名に関連がない場合、計算式がエラーになることがあります。

※ グローバル格納フィールドは関連がなくても計算式中で参照できます。

フィールドの検索

検索ボックスに文字列を入力すると表示されているフィールド名から検索します。検索ボックス右の［X］ボタンをクリックすると検索が解除され、すべてのフィールドが表示されます。

フィールドの表示

各フィールド名の先頭にはフィールドタイプがアイコンで示されています。ここから計算式の中に指定するフィールドを「フィールド参照」と呼びます。

検索ボックスの右のアイコンをクリックすると、表示順を切り替えることができます。

［データベースの管理］ダイアログを開く

計算式の記述途中でリレーションシップやテーブル、フィールドが必要だと気付いたらプルダウンリストの最後の「データベースの管理 ...」を選択してフィールドを定義してから戻ってきて式の記述を続けることもできます。

10.2.3 計算式パネル

計算式を実際に記述する場所です。

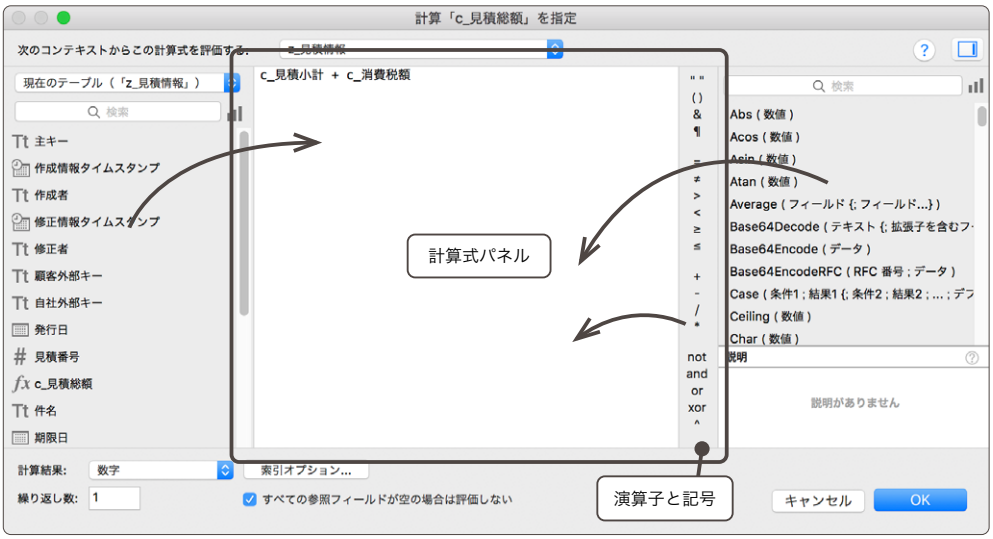


図 10-4［計算式の指定］ダイアログ ; 計算式パネル

・左のフィールドパネル
・計算式パネルのすぐ右の演算子や記号ボタン
・一番右端の関数パネル

これらから、目的の項目をダブルクリックあるいはシングルクリックすると、入力カーソルの位置に挿入されます。

直接入力のサポート

直接文字列を入力して補助を利用しながら作成することができます。文字列を入力しはじめると候補のフィールド名や関数名のリストが補足情報と一緒にその場で表示され入力を補完してくれます。

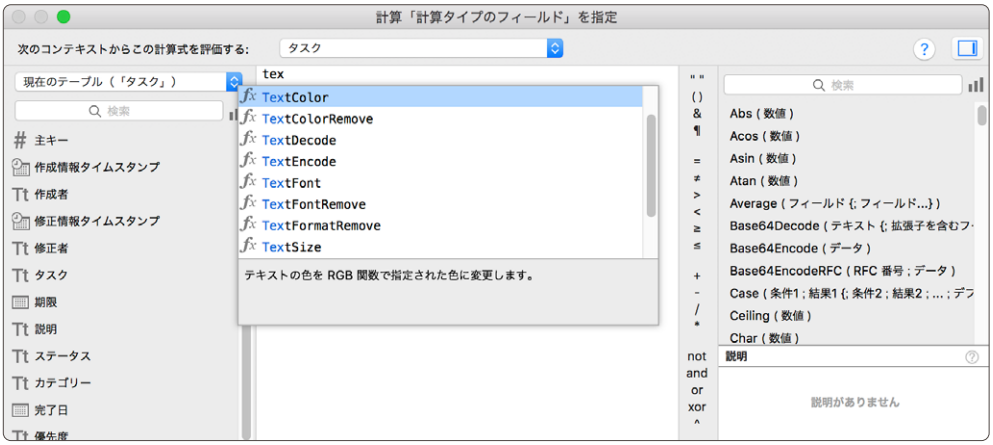


図 10-5 候補表示

演算子と記号ボタン
縦に配置された演算子ボタンです。

・文字列演算子と記号
・比較演算子
・算術演算子
・論理演算子

の 4 つのグループに分かれています。

文字列演算子と記号
テキストタイプのフィールド値や文字列テキストを扱うために使用する演算子に、「&」「" "」「¶」などが用意されています。
「&」は、テキストを連結するときに使います。また文字列としてテキストを記述するときは「"（ダブルクォーテーションマーク）」で括ります。「¶」は改行を意味します。

比較演算子
数字や日付タイプのフィールド値や文字列テキストを扱うために使用する演算子で、「＝」「≠」「＞」「＜」「≧」「≦」があります。「≠」は等しくないという意味です。また「＝」「≠」はテキストの比較にも使用できます。

算術演算子
「＋（加算）」「−（減算）」「*（乗算）」「/（除算）」「^（累乗）」があります。
( ) は優先順位を指定します。通常、計算式は左から右に評価され、加算と減算よりも乗算と除算が先に実行されますが、( ) を使用すると、( ) の中の式が先に評価されます。

論理演算子
「not」「and」「or」「xor」があります。次表を参照してください。

not	値を偽から真、または真から偽に変更する	**1 つの式に使用**
and	右側と左側の両方の値が真である場合のみ真	**2 つの式の組み合わせに使用**
or	左側と右側のどちらかの値が真である場合は真	**2 つの式の組み合わせに使用**
xor	左側と右側のどちらか一方の値だけが真の場合は真	**2 つの式の組み合わせに使用**

10.2.4　関数パネル

計算式に使用する関数を指定するパネルです。

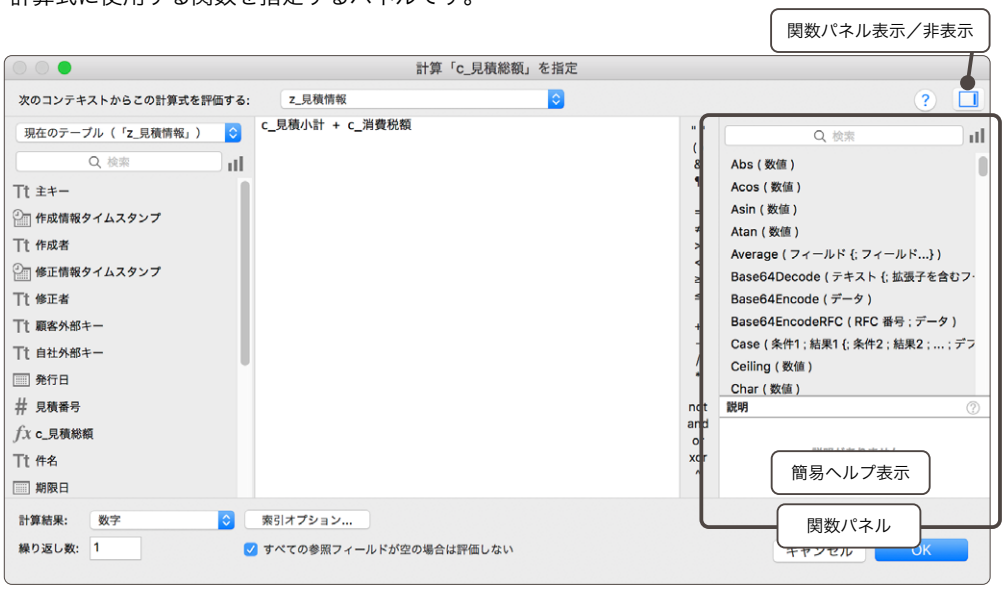


図 10-6［計算式の指定］ダイアログ ; 関数パネル

関数は、名前や種類の順に表示されます。検索ボックスの右のアイコンをクリックすると、表示順を切り替えたり、関数の種類を指定できます。

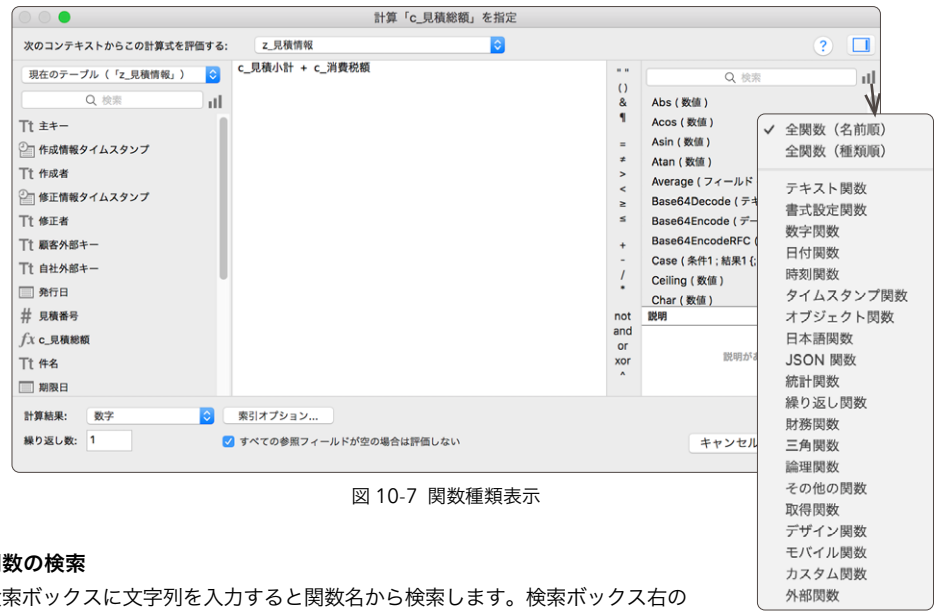


図 10-7　関数種類表示

関数の検索

検索ボックスに文字列を入力すると関数名から検索します。検索ボックス右の［X］ボタンをクリックすると検索が解除されます。

関数パネル表示 / 非表示のボタン

関数パネルを非表示にし、計算式パネルの領域を広げて作業できます。

10.3　フィールド定義と計算式

フィールド定義の計算式について確認します。第 6 章で「見積作成管理 .fmp12」の、計算タイプのフィールドの設定について後回しにしました。本章の演習ファイルには、計算タイプフィールドの計算式が記述されています。

［データベースの管理］ダイアログの「見積情報」テーブルのフィールドタブを表示してください。「c_」がついている 3 つの計算タイプフィールドがあります。

・c_ 見積総額
・c_ 見積小計
・c_ 消費税額

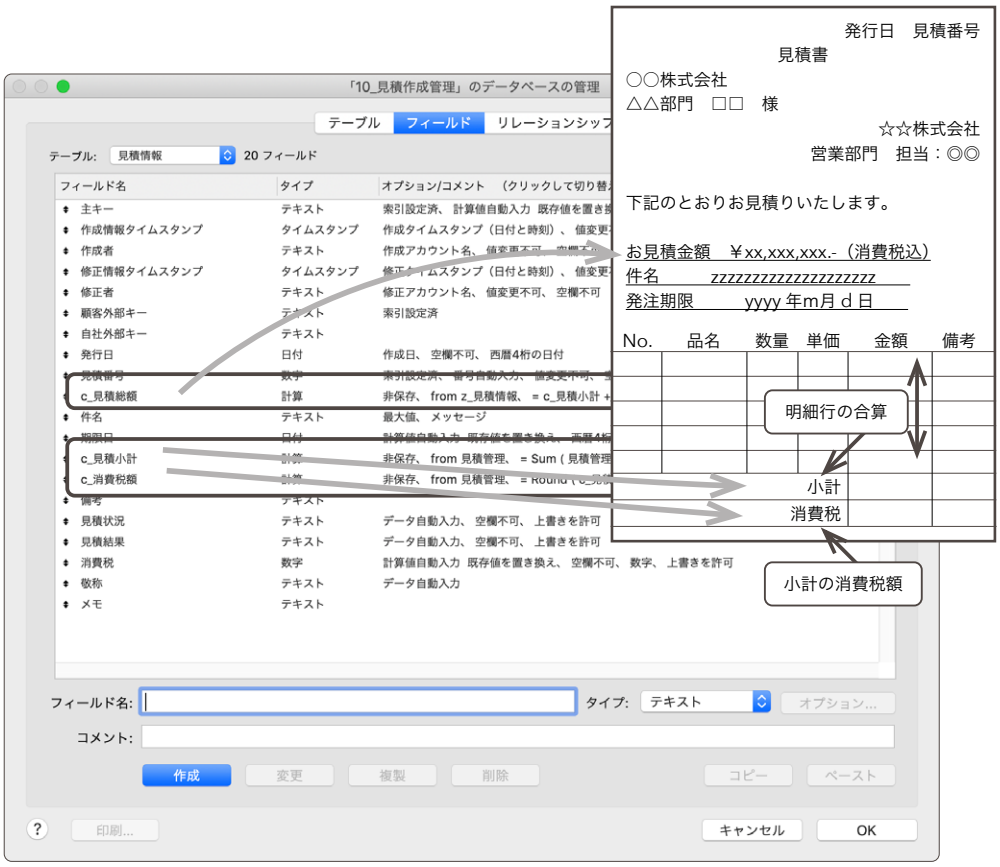


図 10-8「見積情報」テーブルの計算タイプフィールド

見積書の金額に関するものです。その関係は次の内容です。

c_見積小計＋c_消費税額＝c_見積総額
→　明細行の金額の合算が、見積の小計となり、その消費税額を合算すると見積の総額となる

10.3.1　関連テーブルのフィールドを使った計算

「c_ 見積小計」フィールドは明細行（「見積明細」テーブルの「金額」フィールド）の合計額です。次の計算式が指定されています。

```
Sum （ 見積管理_明細::c_金額 ）
```

```
Sum （フィールド { ; フィールド...}）関数
```

「統計関数」の 1 つです。同じテーブル、関連 TO の 1 つのフィールド、複数のフィールドの合計を結果として返します。

この計算式は単純な計算式ですが論理が実際に構造として設定されていないと正しい計算結果が得られません。第 4 章までに設計した内容を元に作成された次図の TOG のコンテキストを使用しています。

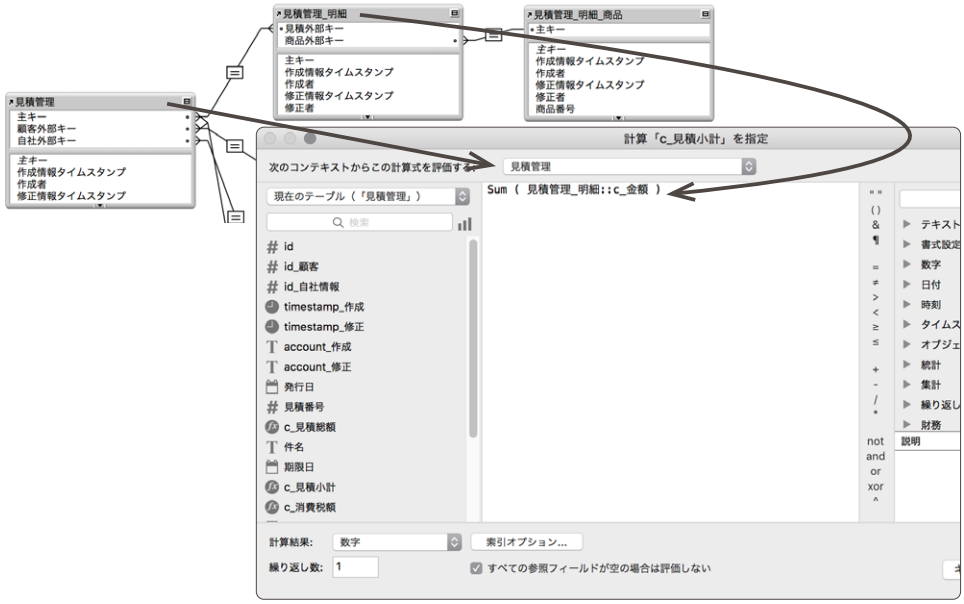


図 10-9　c_ 見積小計の計算式

計算式の中で関連テーブルのフィールドを使用しているため、索引は自動的に「非保存」が設定されています。この設定は解除できません。



図 10-10　解除できない場合のエラーメッセージ

10.3.2 同じテーブルのフィールドを使った計算

残りの 2 つのフィールドは関連 TO のフィールドは使っていません。

「c_ 消費税額」フィールド

「c_ 見積小計」フィールドに消費税率（「消費税」フィールド）を掛け合わせて四捨五入した額です。関連 TO のフィールドは使用していません。

```
Round ( c_見積小計 * 消費税 ; 0 )
```

しかし計算式に使われている「c_ 見積小計」フィールドの索引が非保存なので、やはり非保存が自動的に設定されています（解除できません）。

「消費税」フィールド

消費税率は**「消費税」フィールド**に計算値自動入力が設定されています。「発行日」フィールドと、消費税率の切り替え施行日を比較しています。関連テーブルのフィールドは使用していません。

```
Case ( 発行日 < Date ( 10 ; 1 ; 2019 ) ; .08 ; .1 )
```

同じテーブルのフィールドを使っていますが索引は必要に応じて保存／非保存を設定できます。

消費税の要件定義

業務のルールを確認してユーザの合意を得た要件でフィールド定義や計算式を設定するべきです。金額に影響があり、業務にとってとても重要な事柄です。

まず、計算方法と端数の処理です。
・消費税率の適用は発行日時点にするか、任意の日付（受注日や請求日）の時点にするか
・消費税額の端数処理は、切り上げ、切り捨て、四捨五入のどれか

さらに『消費税をどこで計算するか』も決定します。
・明細の合計額に対して消費税を計算する
・明細行ごとに消費税を計算する

「見積作成管理システム」では、次の内容とします。
・消費税率の適用は発行日時点とする
・消費税額の端数処理は、四捨五入とする
・明細の合計額に対して消費税を計算する

「見積総額」フィールド

「c_ 消費税額」と「c_ 見積小計」フィールドの合算です。関連 TO のフィールドは使用しません。

```
c_見積小計 + c_消費税額
```

この式は、以下も結果は同じです。

```
Sum ( c_見積小計 ; c_消費税額 )
```

確認後、［キャンセル］をクリックしてダイアログを閉じてください。

10.4　関数

FileMaker Pro には、たくさんの関数が用意されています。関数とは、あらかじめ定義されている名前がついた特定の演算です。そして結果を 1 つ返すものです。

［計算式の指定］ダイアログの関数パネルで、次図のように表示されています。

・関数名
・1 対のカッコ（引数を使う場合）
・必要とする引数

（複数の引数が必要な場合、「 ; (セミコロン)」で区切ります。オプションの引数は、「{ }(中カッコ)」で囲まれています。「...（省略記号)」は、セミコロンで区切った引数をさらに追加できることを表します。

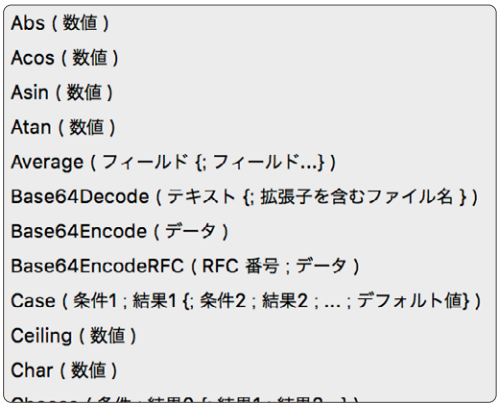


図 10-11 関数の表示

10.4.1　引数

関数には引数が必要なものがあります。必須、あるいはオプションとしてなど様々です。引数には、

・定数（任意の文字列やフィールド名など）
・フィールドのデータ
・式
・変数
・他の関数の結果

などを使います。

テキストの引数はダブルクォーテーションマーク「"」で囲む必要があります。また、フィールド名を引数として指定する場合は、原則としてその計算式が評価されるコンテキストにおいて有効な TO が指定されていないと正しい結果が返りません。

引数は、次のように入力されます。「日付関数」を使って確認します。

Year (日付) 関数

引数は1つです。指定した「日付」の4桁の年を返します。

Year ("2018/08/25")
のようにテキストの引数で日付を入力すると「2018」と結果を返します。

Year (発行日)
「発行日」フィールド（日付タイプ）があり、そのフィールド名を引数として入力すると、データが2018/8/25だった場合、「2018」を返します。
Year (Get(日付))
Get(日付)関数を引数として入力すると、今日が2018/8/25だった場合、「2018」を返します。

Date (月 ; 日 ; 年) 関数

引数は3つです。指定した「月」「日」「年」のカレンダー上の日付を返します。

Date (8 ; 25 ; 2018)
のように数字で引数を入力すると「2018/8/25」と結果を返します。数字は「"」で囲む必要はありません。

Date (8+2 ; 25 ; 2018)
のように式も含んだ引数を入力しても「2018/10/25」と結果を返します。

Date (1 ; 2 ; Year (発行日))
「発行日」フィールド（日付タイプ）があり、そのフィールド名を引数として入力すると、データが2018/8/25だった場合、「2018/1/2」を返します。

データビューアと引数
『この式、ちゃんと動くかな……』と思ったらデータビューアで確認できます。そして、注意があります。データビューアの計算式ではコンテキストの指定が必須です。

例えば、「見積情報」テーブルに「発行日」と「発行年」フィールドがあり、「発行年」フィールドに Year (発行日) の計算値自動入力が設定されています。評価を実行したい場合、Year (発行日) をそのままデータビューアで確認すると評価できずにエラーになります。データビューアで計算式を評価するためには、式を変更し、Year (見積管理 :: 発行日) のような、現在表示されているレイアウトの TO を付加した完全修飾名の表記にするとエラーは起こらなくなります。
※ 高度なツールを **オン** にしているとデータビューアを使用可能です（方法は 9 ページを参照）。

10.4.2 関数の種類

関数は、用途によりいくつかの種類に分かれています。次表が主な種類と特徴です。

テキスト関数	テキストの情報、分析、並べ替え、抽出、作成に使用することができます。
書式設定関数	指定されたテキストの色、フォント、サイズ、およびスタイルを変更することができます。
数字関数	数値データを操作します。
日付関数	日付を計算したり日付情報を操作したりします。
時刻関数	時刻を計算したり時刻情報を操作したりします。
タイムスタンプ関数	タイムスタンプは、特定のイベントが発生した正確な日付と時刻を記録するなど、様々な同期処理に使用されます。
オブジェクト関数	オブジェクトタイプフィールドに格納されているデータの計算、操作、情報の取得を行います。
日本語関数	日付部分を日本語で返します。また、日付、数字、およびテキストの値を他のデータタイプや日本語の仮名に変換します。
JSON 関数	JSON 形式のデータを解析、作成、変更、および書式設定します。
統計関数	フィールドに含まれる数値 (関数によっては日付や時刻) の統計分析を実行します。
財務関数	財務情報の計算を行います。例えば PMT 関数を使用すると、ある利率のローンで車を購入する際の月々の支払い額を計算することができます。
三角関数	度、角度、およびその他の幾何情報の計算を行います。
論理関数	指定した条件が真か偽かを評価します。 FileMaker Pro では、条件が真の場合に「1」、条件が偽の場合に「0」を返します。 GetField 関数は、別のフィールドの内容を返すこともできます。
その他の関数	フィールドとレイアウトオブジェクトに関する情報を返します。
取得関数	スクリプト内のエラーを監視したりファイルまたは実行されているステータスに関する情報を取得できます。取得関数の多くは定期的に変更される情報を結果として返します。 スクリプトで最新情報を取得する使い方が最も適切です。またはフィールドオプションの初期値の設定にも有用です。非保存でない計算タイプフィールドに Get (時刻) 関数が配置および保存される場合、新しいレコードが作成された場合にのみ時刻が更新されます。
デザイン関数	現在開いているカスタム App ファイルの構造に関する情報を返します。例えばすべてのレイアウト名やフィールド名を取得することができます。
モバイル関数	FileMaker Go での使用が前提です。

なお、種類のほかにも用途が似ていて結果が違うひとかたまりのグループがいくつかあります。1 つの関数の使用方法がわかると、他の関数についてもわかります。

よく使うものをいくつか記述します。さらに詳しい情報はヘルプを参照してください。

※ ここから、関数名の表記で「関数」という文言、計算式の例での参照フィールドの完全修飾名を一部省略しています。

10.4.3 UUID

FileMaker Pro のデフォルトのフィールドの自動作成では「主キー」フィールドに計算値自動入力で Get (UUID) 関数 が設定されています。新規レコードを作成すると「主キー」フィールドに次のような文字列が自動入力されます。

```
67574151-F1FC-4ECB-8787-3BE8AA9B7D3F
```

```
Get(UUID)
```

結果は16進数（128bit）のランダムな文字列です。評価されるたびに新しい文字列を返します。

また、別の関数も用意されています。次のような数字が結果として返ります。

```
625960635142039536788789136135223203998505948254052233911 2
```

```
Get(UUID番号)
```

結果は24進数（192ビット）のランダムな数字です。評価されるたびに新しい数字を返します。

UUID（Universally Unique IDentifier）は全世界でユニークになる（重複する確率が限りなく 0 である）識別子です。FileMaker Pro 独自の考え方ではなく標準化がなされている世界的なものです。この特性を主キーに活用できます。

主キーは照合フィールドとして関連の定義に使われることが多いのですが、文字列よりも数値の方が関連を利用する計算式や処理などが早くなる可能性があります。「主キー」フィールドに UUID を採用する場合、どちらの関数をフィールドオプションに設定するか検討してください。

やってみよう

どのようなデータが生成されるかデータビューアで確認しましょう。手順を記述します。

(1) ［ツール］–［データビューア］メニューを選択します。
(2) 「監視」タブをクリックします。
(3) 右下の［+］をクリックします。
(4) ［式の編集］ダイアログが表示されます。
(5) ［式］に `Get(UUID)` あるいは `Get(UUID番号)` を入力ます。
(6) 自動的に評価する が **オン** なら結果が表示されます。あるいは **オフ** の場合は［評価］ボタンをクリックします（評価の度に違う結果が表示されます）。

UUID の主キーの利点

例えば、オフラインとオンラインで作業するクライアントがそれぞれ複数あって、FileMaker Cloud や FileMaker Server 上のレコード（オンラインで作成された）とオフラインで作成されたレコードを一緒にするために同期するとします。このような場合、主キーをユニークに保つことは、一般的にはとても難しい作業です。しかし、UUID を使用するとその悩みが解決します（限りなくユニークだからです）。

10.4.4 数字関数

カスタム App では、よく数値を取り扱います。お金や個数、件数などです。例えば「見積作成管理 .fmp12」では、消費税額の端数は四捨五入しました。

```
Round( 数値 ; 桁数 )     →  四捨五入
```

```
Round( 123.45 ; 0 )
```

数字「123」が返ります。
桁数は、小数点以下を表します。小数点以下が不要な場合は桁数を「0」で式を指定します。
他にも端数処理に使用できる関数があります。

```
Truncate( 数値 ; 桁数 ) →  指定した小数点以下の桁で切り捨て
Ceiling( 数値 )         →  次の整数に切り上げた数値
Floor ( 数値 )          →  次に小さい整数に切り捨てられた数値
Int( 数値 )             →  数値の整数のみ
```

やってみよう

それぞれ「123.45」のような小数点以下の数値を使って任意の桁数を指定してデータビューアで結果を確認してみましょう。

他には、

```
Mod( 数値 ; 除数 )  →  数値を除数で割った余り
```

```
Mod( 7 ; 2 )
```

数字「1」を返します。
7÷2＝3で余り1なので、この余り部分を返します。日付や時間の計算に使用したり、2で割った余りが1なら数値は奇数であると判断するために使われることがあります。

```
Random  →  0以上1未満の乱数
```

```
.99035877187065557425
```

のような小数点以下の数字を返します。評価されるたびに新しい結果を返します。

例えば、0から9の間の乱数が必要な場合

```
Int ( Random * 10 )
```

0は不要で1から10の間の乱数が必要な場合は、次のような式で得られます。

```
Int ( Random * 10 ) + 1
```

やってみよう

Random をデータビューアで試してみましょう。

```
Abs( 数値 ) → 数値の絶対値
```

```
Abs ( -5 )
```
数字「5」を返します。
正数が結果として返ります。絶対値は「0」からの距離を捉えるのに役立ちます。よくある使用方法としてGet(システムプラットフォーム) 関数ではWindows プラットフォームの場合「-2」が返りますが、Abs 関数で絶対値化して取り扱うことがあります。

```
Sign( 数値 ) → 正か負か 0 か
```

```
Sign( -12345 )
```
数字「-1」を返します。
指定された数値が負の場合は「-1」、ゼロの場合は「0」、正の場合は「1」を返します。請求後の入金の残高有無などを確認するために役立ちます。

10.4.5 テキスト関数 1

豊富な関数が用意されています。

```
Length( テキスト ) → 指定したテキストの文字数（空白や改行も数えます）
```

```
Length( "あいうえお" )
```
数字「5」を返します ()。

例えば、「見積情報」テーブルの「件名」フィールドを 30 字までに制限するには「計算式で制限」で、
```
Length( 件名 ) ≦ 30
```
と、設定することもできます。この計算式は論理値（0 または 1）が結果として返ります。なお、結果の判断基準は、次のように考えています。
・30 文字までは「1」エラーではない
・31 字以上は「0」エラー

単語数を数える関数もあります。

```
WordCount( テキスト ) → 指定したテキストの単語数
```

```
WordCount ( "あいうえお　かきくけこ" )
```
数字「2」を返します。

```
WordCount ( "あい,うえお、かきく。けこ" )
```
数字「4」を返します。
(空白や改行と記号（, 、 。）などが単語の区切りになります)

やってみよう

「あいうえお　か、きく ¶ けこ」のような任意の文字列を指定してデータビューアで結果を確認してみましょう。

改行区切りの行数を数えるには、

```
ValueCount( テキスト ) → 改行区切りの値数（行数）
```

```
ValueCount( "カレー¶ハンバーグ¶オムライス¶ナポリタン" )
```
数字「4」が返ります。

関数を組み合わせて値一覧の行数を求めることも可能です。
```
ValueCount ( ValueListItems ( "10_見積作成管理" ; "見積状況" ) )
```
は、「10_見積作成管理.fmp12」の「見積状況」値一覧の行数の数字「3」が返ります。

空白を取り除く場合には次の関数が用意されています。

```
Trim( テキスト ) → 文字列の前後にあるスペースを削除

TrimAll( テキスト ; 全角詰め ; タイプ )
 → 文字列にあるスペースを削除する
    0：ローマ字以外の文字とローマ字の間のすべてのスペース
       （ローマ字の間には常にスペースが1つになるようにする）
    1：ローマ字以外の文字とローマ字の間に常に半角スペース
       （ローマ字の間には常にスペースが1つになるようにする）
    2：ローマ字以外の文字の間のスペース
       （ローマ字の間に複数スペースがある場合にスペースが1つになるようにする）
    3：すべてのスペース
```

いずれもテキストで結果が返ります。

やってみよう

「あ、いうえ　お¶か abc defg　きく　けこ」などの任意の文字列を指定しデータビューアで結果を確認してみましょう。

また、文字列から一部だけを取り出すには

```
Left( テキスト ; 文字数 )          → 左から指定した文字数を取り出す
```

```
Left( "あいうえお" ; 2 )
```
テキスト「あい」が返ります。　他の指定のための関数も用意されています。

```
Right( テキスト ; 文字数 )          → 右から指定した文字数を取り出す
Middle( テキスト ; 先頭文字位置 ; 文字数 )
 → 指定した位置から文字数を取り出す
LeftWords( テキスト ; 単語数 ) → 左から指定した単語数を取り出す
RightWords( テキスト ; 単語数 ) → 右から指定した単語数を取り出す
MiddleWords( テキスト ; 先頭単語位置 ; 単語数 )
 → 指定した位置から単語数を取り出す
```

いずれもテキストで結果が返ります。

任意の文字だけを取り出すには、

```
Filter( フィルタするテキスト ; フィルタテキスト ) → 指定したテキストを取り出す
```

例えば、数字文字列だけを取り出したいときは
`Filter( "(03)-1234-56789" ; "0123456789" )`
→ テキスト「03123456789」が返ります。

やってみよう
任意の文字列を指定してデータビューアで結果を確認してみましょう。

改行区切りの文字列は、様々に利用されます。改行区切りのテキストから任意の 1 行だけを取り出すには、

```
GetValue( 値一覧 ; 値番号 ) → 値番号の 1 行を取り出す（末尾に改行なし）
```

例えば、
`GetValue( "カレー¶ハンバーグ¶オムライス¶ナポリタン" ; 2 )`
→ テキスト「ハンバーグ」が返ります。

他にも、

```
LeftValues( テキスト ; 値数 )→ 上から指定した数の行を取り出す（末尾に改行あり）
```

例えば、
`LeftValues( "カレー¶ハンバーグ¶オムライス¶ナポリタン" ; 2 )`
→ テキスト「カレー¶ハンバーグ¶」が返ります。

```
RightValues( テキスト ; 値数 )
  → 下から指定した数の行を取り出す（末尾に改行あり）

MiddleValues( テキスト ; 値開始位置 ; 値数 )
  → 指定した開始位置からの数の行を取り出す（末尾に改行あり）
```

いずれもテキストで結果が返ります。

改行区切りから任意の行を取り出すには、

```
FilterValues( フィルタするテキスト ; フィルタ値 )
  → 指定したテキスト行を取り出す（末尾に改行あり）
```

例えば、「カレー¶ハンバーグ¶オムライス¶ナポリタン」の改行区切りテキストから「カレー」と「オムライス」を取り出したいときは、
`FilterValues( "カレー¶ハンバーグ¶オムライス¶ナポリタン" ; "カレー¶オムライス" )`
→ テキスト「カレー¶オムライス ¶ 」が返ります。

やってみよう

任意の文字列を指定してデータビューアで改行区切りの有無なども含めて結果を確認してみましょう。

10.4.6 日本語関数

日本語は、漢字、ひらがな、カタカナ、英数字、記号と、それらの全角半角を場面ごとに使い分けることがあります。例えば、縦書き印刷をする住所のために数字を漢数字にする場合などです。

```
Hiragana( テキスト )        →  指定したテキストの半角全角カタカナをひらがなにする
KanaHankaku( テキスト )     →  指定したテキストの全角カタカナを半角にする
KanaZenkaku( テキスト )     →  指定したテキストの半角カタカナを全角にする
KanjiNumeral( テキスト )    →  指定したテキストのアラビア数字を漢数字にする
Katakana( テキスト )        →  指定したテキストのひらがなを全角カタカナにする
RomanHankaku ( テキスト )
  →  指定したテキストの全角の英数字と記号を半角の英数字と記号にする
```

```
Hiragana( "アイウエオ" )
```
テキスト「あいうえお」が返ります。

```
KanjiNumeral( 1234 )
```
テキスト「一二三四」が返ります。

やってみよう

「さしスセソ　た¶ﾁﾂﾃﾄ 1234567890」のような任意の文字列を指定しデータビューアで結果を確認してみましょう。

フィールドオプションでも設定できますが、ふりがなも関数で取得できます。

```
Furigana( テキスト{ ; オプション } )
→  指定したテキストのふりがなを指定したオプションの文字種類のふりがなにする
   （思ったようなふりがなが結果として返らない場合もあります）
   1とオプション指定なし：ひらがな
   2：全角カタカナ
   3：全角ローマ字
   4：半角カタカナ
   5：半角ローマ字
```

```
Furigana( "山田太郎" ; 5 )
```
テキスト「yamadatarou」が返ります。

フィールドオプションを計算式で設定（入力値の自動化）する演習をします。

現在「商品情報」テーブルの「商品名」フィールドには商品名の前後に全角スペースが入力できてしまいます。本来、これは不要なものです。今後の入力時にデータの前後のスペースを自動的に削除するよう設定します。

「10 章開始時」フォルダ

演習 10-1　フィールドオプションに計算式を使う 1

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　［データベースの管理］ダイアログを表示します。

(2)　フィールドオプションを設定します。

a.　「商品情報」テーブルの「商品名」フィールドのフィールドオプションで「入力値の自動化」タブの［計算値］の右にある［指定 ...］をクリックして計算式を入力します。

```
Trim ( Self )
```

b.　［フィールドに既存の値が存在する場合は置き換えない］を　**オフ**　にします。

演習ここまで

引き続き、「顧客情報」テーブルの「郵便番号」フィールドと「顧客担当者」テーブルの「電話番号」「FAX 番号」フィールドに、半角数字と半角ハイフンだけが格納されるように設定します。

演習 10-2　フィールドオプションに計算式を使う 2

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　フィールドオプションを設定します。詳細な手順は省略します。

(2)　「顧客情報」テーブルの「郵便番号」フィールドの「入力値の自動化」オプションを設定します。

・［計算値］に次の式を設定

```
Filter ( RomanHankaku ( Self ) ; "-0123456789" )
```

・［フィールドに既存の値が存在する場合は置き換えない］を　**オフ**

(3)　「顧客担当者」テーブルの「電話番号」「FAX 番号」フィールドにも同様に設定します。

演習ここまで

「顧客情報」テーブルの「会社名」フィールドは、半角カタカナが入力されても全角で保存されるようにします。

演習 10-3　フィールドオプションに計算式を使う 3

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　フィールドオプションを設定します。詳細な手順は省略します。

(2)　「顧客情報」テーブルの「会社名」フィールドの「入力値の自動化」オプションを設定します。

・［計算値］に次の式を設定

```
KanaZenkaku ( Self )
```

・［フィールドに既存の値が存在する場合は置き換えない］を　**オフ**

演習ここまで

Self 関数を使うと、フィールド名を指定せずに計算式を作成できます。コピーと貼り付け（ペースト）を行っても同じように働きます。

「入力値の制限」を設定します。「見積情報」テーブルの「期限日」フィールドに、発行日より古い日付を入力できないようにします。

演習 10-4　フィールドオプションに計算式を使う 4

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　フィールドオプションを設定します。詳細な手順は省略します。

(2)　「見積情報」テーブルの「期限日」フィールドの「入力値の制限」オプションを設定します。

・［計算式で制限］の右にある［指定 ...］をクリックして次の式を設定

```
発行日　≦ Self
```

・［データの入力時にユーザによる上書きを許可する］を　**オフ**

カスタムメッセージも設定します。

・［制限値以外の入力時にカスタムメッセージを表示］を　**オン**
・「発行日より古い日付は設定できません。」を入力

(3)　［OK］をクリックして［データベースの管理］ダイアログを閉じます。

演習ここまで

レイアウトで入力を試します。

演習 10-5　フィールドオプションに計算式を使う 5

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「商品管理｜マスタ詳細」レイアウトへ切り替えます。
　a.　左側のポータルの任意の行をクリックして右側に商品情報を表示します。
　b.　商品名の　**前や後**　に全角や半角スペースを入力します。
　c.　フィールドの外でクリックすると前後のスペースがなくなります。しかし、商品名に含まれているスペースはなくなりません。

(2)　「顧客管理｜マスタ詳細」レイアウトへ切り替えます。
　a.　左側のポータルの任意の行をクリックして右側に顧客情報を表示します。
　b.　「会社名」フィールドに「ｱｲｳｴｵ」など任意の半角カタカナを入力します。
　c.　フィールドの外でクリックするとカタカナが全角に変わります。

　d.　「郵便番号」フィールドに数字以外の文字、「-（半角ハイフン）」を入力します。
　e.　フィールドの外でクリックすると、数字と「-（半角ハイフン）」にフィルタされることを確認します。

　f.　レイアウト下のポータル上の顧客担当者の「電話番号」「FAX 番号」フィールドでも同様の入力を試します。

(3)　「見積詳細」レイアウトへ切り替えます。1 件目のレコードに移動します。
　a.　「期限日」フィールドに、「発行日」よりも古い日付を入力します。
　b.　エラーメッセージが表示され「発行日」と同じか新しい日付の入力が必要です。

演習ここまで

演習で採り上げた項目はよく取り扱うものです。これらの基本的な情報は、書類の表示、郵便物の宛名などによく利用されるので、入力される文字の種類が統一されていると扱いやすいものです。

・会社名
・郵便番号
・住所
・電話番号
・FAX 番号
・E メールアドレス

ユーザに入力ルールを覚えてもらうよりも、カスタム App 側である程度補正します。全角半角、カタカナ、ひらがな、などについて、フィールドオプションを利用して自動で整える方がユーザも安心です。

テキストタイプフィールドとして取り扱う、郵便番号、電話番号、FAX 番号などは含まれるべき内容を考慮して、Filter 関数などで整形します。E メールアドレスは全体を半角にする RomanHankaku 関数が便利です。

10.5 レイアウトと計算式

レイアウト上の計算式について確認します。

10.5.1 ［次の場合にオブジェクトを隠す］

第 9 章で［次の場合にオブジェクトを隠す］を演習しました。この機能は、計算式の結果の論理値によって次のように表示をコントロールするものです。

・計算式の条件に一致する　「1（真 = True)」　→　非表示
・計算式の条件に一致しない「0（偽 = False)」　→　表示

「見積状況」フィールドに「結果確定済」が入力されるまでは「見積結果」フィールドは表示しない（入力できない）ようにしています。計算式は次の内容でした。

見積管理 **::** 見積状況 ≠ " 結果確定済 "　→　結果の論理値（0/1）が利用されている

2 つの文言の比較は、関数を使った計算式で記述することも可能です。

「10_6 から 8」フォルダ

演習 10-6　レイアウト上で計算式を使う 1（［次の場合にオブジェクトを隠す］）

- **使用ファイル：10_ 見積作成管理** .fmp12

(1)　「見積詳細」レイアウトでレイアウトモードに切り替えます。

(2)　［次の場合にオブジェクトを隠す］の設定を変更します。
（「見積結果」フィールドは、「注文あり　注文なし」のラジオボタンとして表示されています）

a.　「見積結果」フィールドをクリックします。
b.　右パネルのインスペクタの「データ」タブの［動作］の［次の場合にオブジェクトを隠す］を以下の計算式に変更します。

```
not Exact ( 見積管理::見積状況 ; " 結果確定済 " )
```

(3)　レイアウトを保存してブラウズモードへ切り替えて動作を確認してください。変更前と変更後に変わりはないはずです。

演習ここまで

```
Exact ( テキスト ; 比較テキスト )
  →　指定したテキストと比較テキストの比較結果「一致しているかどうか」を数字で返す
     結果は論理値（0/1）
     一致する　　「1（真 = True )」
     一致していない「0（偽 = False)」
```

「一致」は大文字、小文字も同じである必要があります。比較対象がオブジェクトタイプフィールドの場合、同じ設定で格納（埋め込みまたは参照で保存）されている必要があります。

not 演算子
Exact のような論理値を返す関数の結果を逆転させるために使用します。例えば、

```
not Exact ( 見積管理::見積状況 ; "結果確定済" )
      比較対象が一致している 「0（偽 = False)」
      比較対象一致していない 「1（真 = True )」
```

となります。また、次の式も同様の結果となります。

```
Exact ( 見積管理::見積状況 ; "結果確定済" ) = 0
```

比較演算子
演習以前の計算式では「≠」で比較していました。これは比較演算子の 1 つです。次の記号で 2 つのデータを比較する計算式は論理値「1（真 = True)」または「0（偽 = False)」を返します。

- =
- ≠　または　<>
- >
- <
- ≧　または　>=
- ≦　または　<=

10.5.2 条件付き書式

条件付き書式も論理値（0/1）を利用するものです。

- 計算式の条件に一致する　「1（真 = True)」　→　書式を適用する
- 計算式の条件に一致しない　「0（偽 = False)」　→　書式を適用しない

そして条件は、

- 値の範囲
- 計算式

これらを 1 つあるいは複数の設定が可能です。

一番優先の表示
条件付き書式はレイアウトオブジェクトに設定されているスタイルを上書きして表示されます。

例えば「見積詳細」レイアウトの右上にあるボタンバーでは、一番右のゴミ箱アイコンだけが赤です。

（[次の場合にオブジェクトを隠す] も設定されていると、さらに上位に表示／非表示が適用されます）

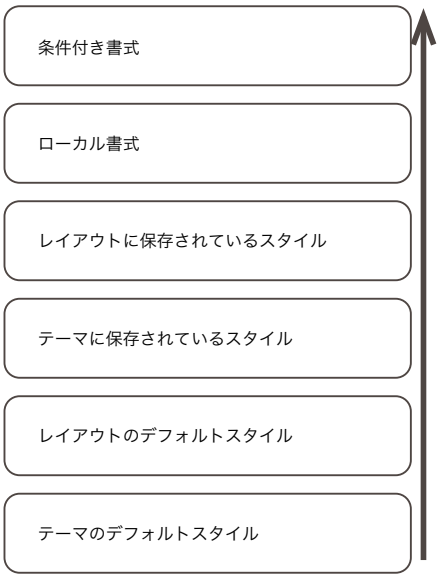


図 10-12 表示の優先順位では条件付き書式が一番上

ボタンバーでは、1 セグメントだけ別のスタイルを設定することはできません。なお、ゴミ箱アイコンのセグメントがアクティブに設定されているわけではありません。条件付き書式が設定されているのです（次図を参照）。

ゴミ箱アイコンのセグメントに設定されている条件付き書式は、「1」だけです。次の意味があります。

常に `1`（真 `=` `True`）→「いつも」という意味

論理値を利用して、常にアイコンが赤く表示されます。次の手順で設定を確認できます。

(1)「見積詳細」レイアウトでレイアウトモードに切り替えます。
(2) 画面右上のボタンバーのゴミ箱アイコンをダブルクリックします。
(3) [書式]–[条件付き ...] メニューを選択します。
(4) [選択されたオブジェクト条件付き書式] ダイアログが表示されます。

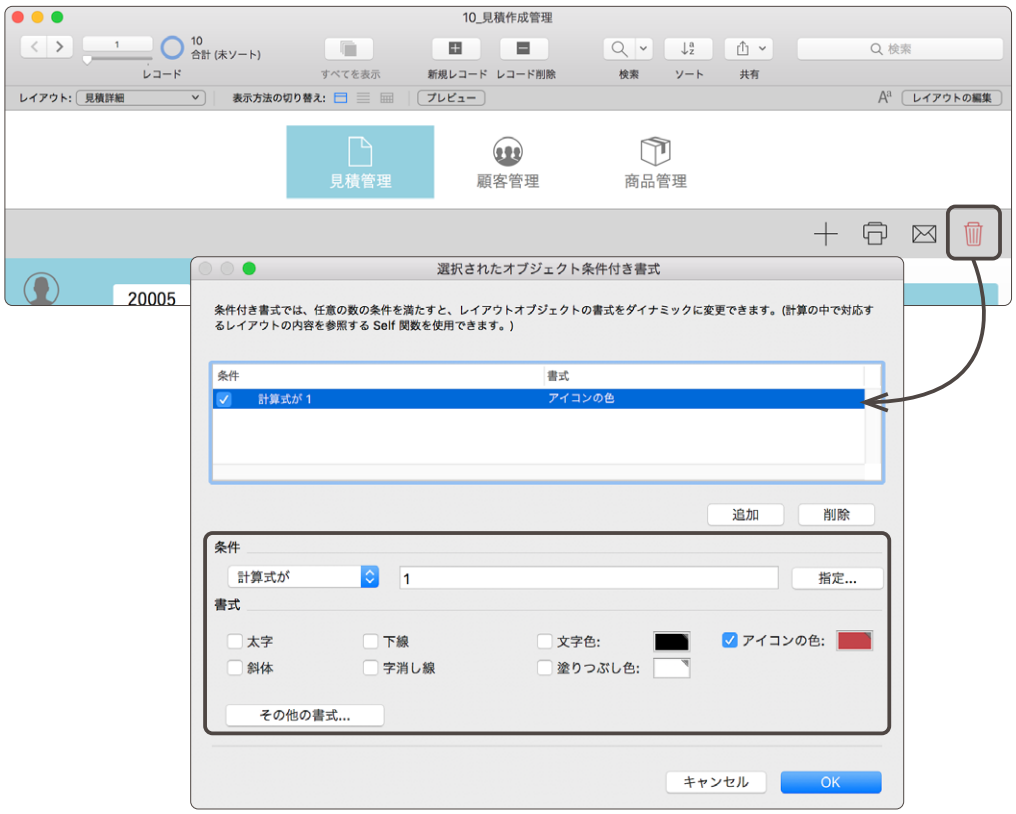


図 10-13 ゴミ箱アイコンだけ赤い／その条件付き書式

レイアウト上で「条件付き書式」を設定します。

商品は、「販売状態」フィールドが「1」の場合は「販売中」という前提です。

・「販売状態」フィールドが「1」　　そのまま
・「販売状態」フィールドが「1」ではない　　文字色をグレーにする

（演習 9-15 で商品番号 30002 から 30005 は販売停止としました）

演習 10-7　レイアウト上で計算式を使う 2（条件付き書式）

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「商品管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。

(2)　条件付き書式を設定します。

a.　左パネルの「オブジェクト」タブをクリックします。
（表示されていなければ、ステータスツールバーのボタンをクリックして表示してください）

b.　画面左にあるポータル（商品管理）の▶をクリックして「c_ 商品番号と商品名」フィールドをクリックします。

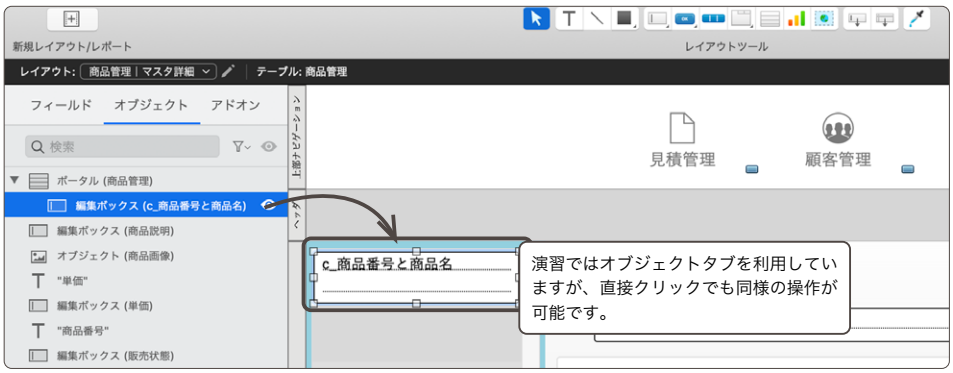


図 10-14　フィールドを選択

c.　右クリックで表示されるショートカットメニューから［条件付き書式 ..］を選択します。
（[書式]–[条件付き ...］メニューを選択でも同じです。）

d.　条件付き書式の設定を［選択されたオブジェクト条件付き書式］ダイアログが表示されます。

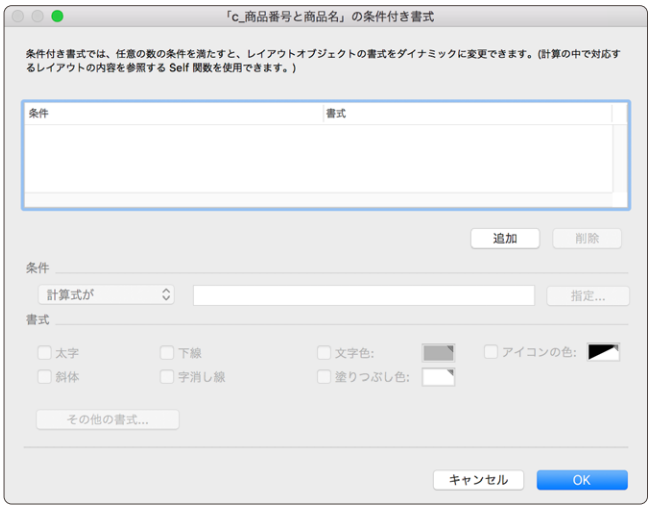


図 10-15　条件付き書式を設定するダイアログ

e.　［追加］ボタンをクリックします。

f. ［条件］のポップアップメニューで「計算式が」を選択します。
g. ［指定 ...］をクリックして［計算式の設定］ダイアログで次の計算式を記述します。
（Self ≥ "" and Self ≤ "" は、消去してください）

商品管理 **::** 販売状態 ≠ 1

h. ［OK］をクリックして、［計算式の設定］ダイアログを閉じます。

i. 「文字色」のチェックボックスをクリックして、右にある黒い色の四角形をクリックします。カラーリストが表示されるのでグレーの四角形を選択します。
j. ［OK］をクリックしてダイアログを閉じます。

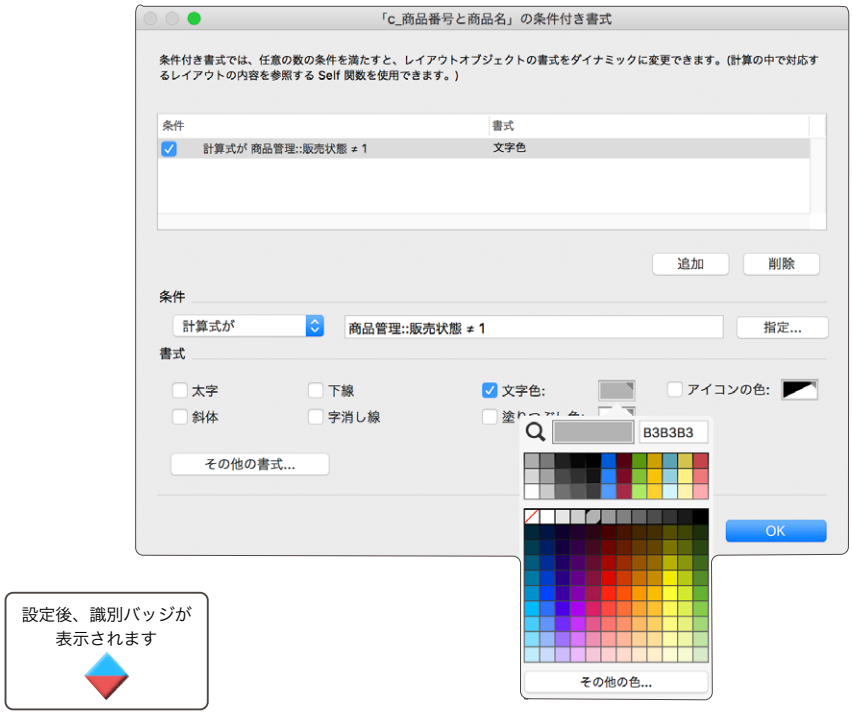


図 10-16 条件付き書式の設定後

(3) レイアウトを保存してブラウズモードへ切り替えます。

(4) ポータルの商品番号 30002 から 30005 の行で、文字が設定したグレーになっていることを確認してください。

演習ここまで

演習ファイルでは、この演習と同様の設定が「顧客管理｜マスタ詳細」レイアウトの会社名ポータルにも設定されています。次の計算式です。

顧客管理 **::** 取引状態 ≠ 1

条件付き書式は、複数の条件を設定できます。**別のレイアウトに、複数の条件を設定**します。

演習 10-8　レイアウト上で計算式を使う 3（条件付き書式）

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「見積リスト」レイアウトでレイアウトモードに切り替えます。

(2)　ボディに配置してあるフィールド全体をドラッグで選択します。
（あるいは、「オブジェクト」タブで、「フィールド」でフィルタしてすべてを選択）

(3)　右クリックでショートカットメニューから［条件付き書式 ..］を選択します。
次の 2 つの条件を設定します（Exact 関数で記述してもしなくても構いません）。

・「見積状況」フィールドが「お客様検討中」だったら　　文字を太字
・「見積状況」フィールドが「結果確定済」だったら　　文字色をグレー

1 つ目の条件を設定します。

a.　［追加］ボタンをクリックします。
b.　［条件］のポップアップメニューで「計算式が」を選択します。
c.　［指定 ...］をクリックして［計算式の設定］ダイアログで次の計算式を記述します。
（Self ≥ "" and Self ≤ "" は、消去してください）

```
Exact ( 見積管理::見積状況 ; "お客様検討中" )
```

d.　［OK］をクリックして、［計算式の設定］ダイアログを閉じます。
e.　「太字」のチェックボックスをクリックします。

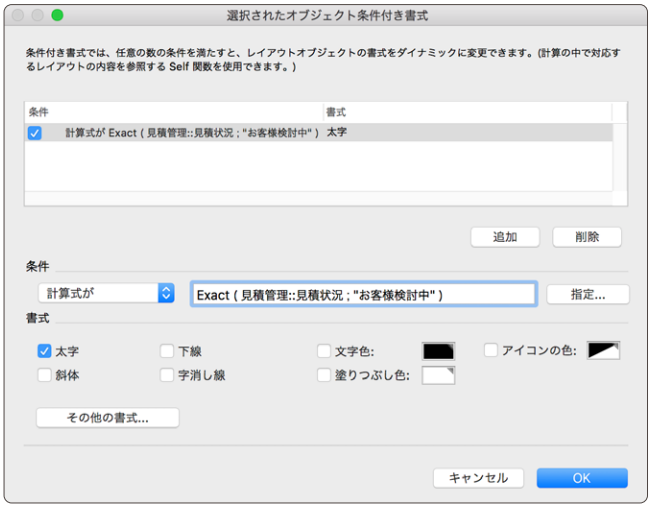


図 10-17　条件付き書式の設定後

2 つ目の条件を設定します。

f.　［追加］ボタンをクリックします。
g.　［条件］のポップアップメニューで「計算式が」を選択します。

h. ［指定 ...］をクリックして［計算式の設定］ダイアログで次の計算式を記述します。

```
Exact ( 見積管理::見積状況 ; "結果確定済" )
```

i. ［OK］をクリックして、［計算式の設定］ダイアログを閉じます。
j. 「文字色」のチェックボックスをクリックして、右にある黒い色の四角形をクリックします。カラーリストが表示されるのでグレーの四角形を選択します。

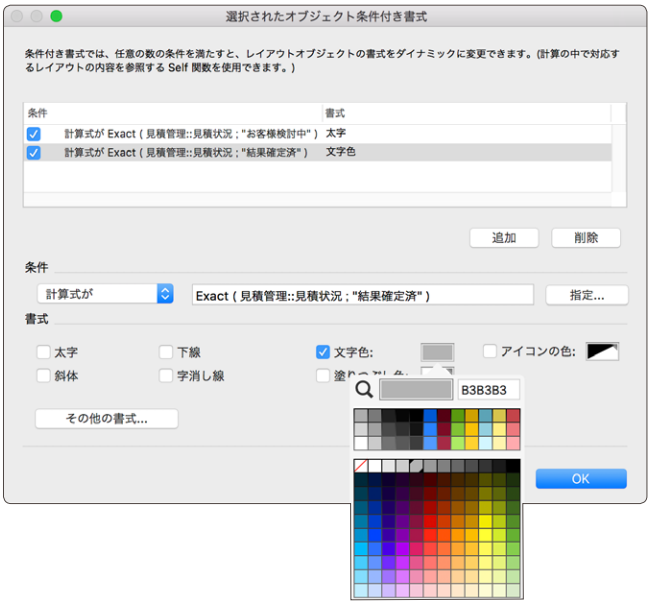


図 10-18 条件付き書式の設定後

k. ［OK］をクリックして条件付き書式を設定するダイアログを閉じます。

(4) レイアウトを保存してブラウズモードへ切り替えます。

(5) ブラウズモードに戻って確認してください。
（これまでにデータを変更している場合は、以下の限りではありません）

見積番号「10002」と「10005」は、太字で表示されます。
見積番号「10001」と「10004」は、グレーの文字で表示されます。

(6) 見積番号「10008」は、通常の表示です。データを変更します。
a. 8 件目のレコード「10008」のレコードに移動します。
b. 「見積詳細」レイアウトに切り替えます。
c. 「お客様検討中」のラジオボタンをクリックします。
d. 「見積リスト」レイアウトに切り替えます。
e. 見積番号「10008」は、太字で表示されます。

「見積リスト」レイアウトでは、「見積結果」フィールドに、［次の場合にオブジェクトを隠す］が設定されています。確かめます。

f. 「見積詳細」レイアウトに切り替えます。
g. 「結果確定済」のラジオボタンをクリックし、右に表示された「見積結果」フィールドの「注文あり」をクリックします。
h. 「見積リスト」レイアウトに切り替えます。
i. 見積番号「10008」は、グレーの文字で表示されます。



図 10-19 複数の条件付き書式の設定後

演習ここまで

複数の条件付き書式

複数の条件では、上から順番にすべてが評価されて合致した条件が適用されます。条件は行をクリックして上下に移動し、並べ替えをすることが可能です。

複数の条件を同じレイアウトオブジェクトに設定する場合、矛盾があると想定した結果になりません。必ず、いくつかのパターンのデータでテストして確認してください。

Case 関数との違い

Case 関数は、合致した条件で評価が終了して結果を返します。

先ほどの演習では「見積リスト」レイアウトの「ボディ」パート上にある 9 個のフィールドに対して条件付き書式を設定しました。これを取りやめて、1 つのレイアウトオブジェクトだけで表示を変更します。

条件付き書式をどこに設定するか

演習 10-8 のように複数のフィールドに条件付き書式を設定すると、それぞれが評価されます。多くの条件を設定すると評価時間がそれぞれの条件に（ほんの少しではありますが）かかります。特にリスト形式では複数のレコードが表示されるため、レコードごとに複数のフィールドの条件を評価することによって表示に時間がかかることがありますので注意してください。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 10_9 から 10」フォルダの中の「10_見積作成管理 .fmp12」ファイルをベースに進めます）

演習ファイルは、演習 10-8 までの条件付き書式を消去した内容です。別の条件付き書式でデータの表示をわかりやすくします。**ボタンオブジェクトを使った条件付き書式のを設定**します。

演習 10-9　レイアウト上で計算式を使う 4（条件付き書式）

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「見積リスト」レイアウトでレイアウトモードに切り替えます。

(2)　ボタンを 1 つ作成します。「ボディ」パートと同じ大きさです（幅 1024× 高さ 80pt）。

(3)　次図のように「ボディ」パートにぴったりに配置します。

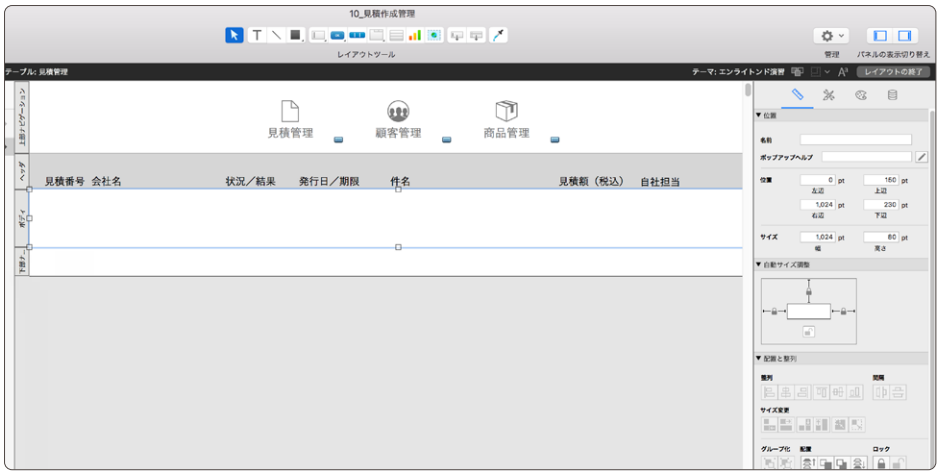


図 10-20 大きなボタンオブジェクトを作成して配置

右パネルの
「位置」タブ
・[位置]　「左辺は 0」、「上辺は 150」pt
・[自動サイズ調整]「左、上、右」

「外観」タブ
・「塗りつぶし」　「単色」で「白」
・「線」　「なし」
・「角丸の半径」　「0」

(4)　条件付き書式で、2 つの条件を設定してください。手順は省略します。
見積状況で色が変わるようにします（色は自由に指定して構いません）。

・「見積状況」フィールドが「お客様検討中」だったら　　塗りつぶし　ピンク
・「見積状況」フィールドが「結果確定済」だったら　　塗りつぶし　グレー

次の計算式になります。
見積管理 : : 見積状況 ＝ " お客様検討中 "
見積管理 : : 見積状況 ＝ " 結果確定済 "

(5)　レイアウトを保存します。

(6)　ブラウズモードで表示を確認します。フィールドは隠れていますが、ボタンオブジェクトの色が変わっています。

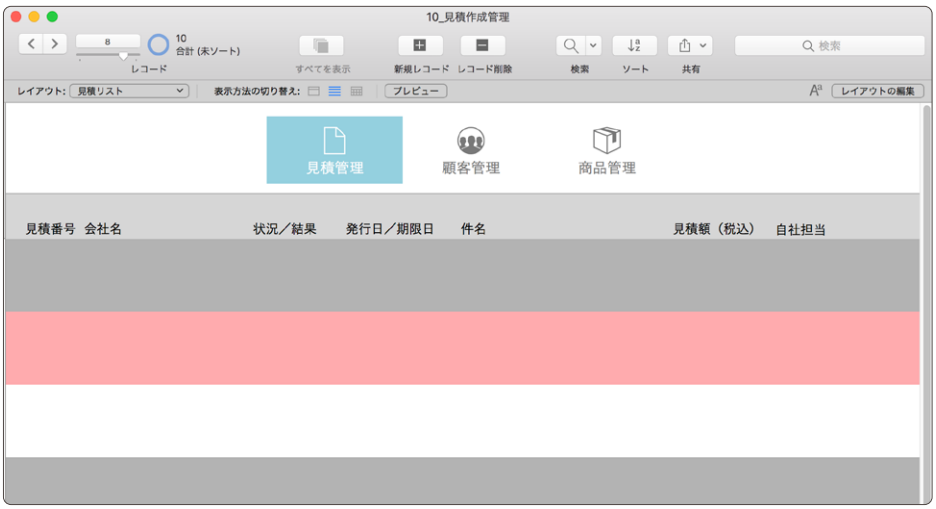


図 10-21　条件付き書式を確認

(7)　レイアウトモードに切り替えます。

(8)　ボタンを「オブジェクト」タブで最背面へ移動します。
（あるいは、インスペクタの「位置」タブの［配置と整列］で最背面ボタンを使います）

(9)　ボタンを透明に設定します。
(4) では「見積状況」フィールドが「作成」だった場合の塗りつぶし色は設定しませんでした。
指定された条件以外では色が変わらないように設定します。

a.　ボタンを選択します。
b.　塗りつぶしと線を設定します。
「外観」タブ
・「塗りつぶし」　　「なし」

(10)　レイアウトを保存してブラウズモードで表示を確認してください。

演習ここまで

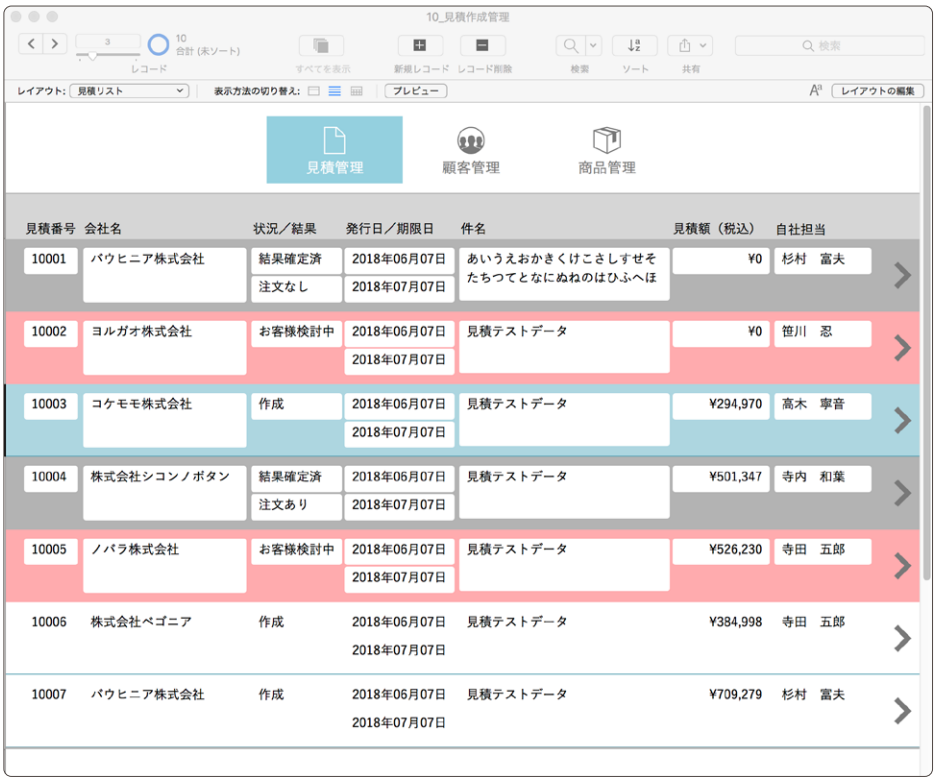


図 10-22 条件付き書式が設定された

ボタンオブジェクトを透明に塗ったので、現在のレコードは青い行になる、というボディに設定されたスタイルが見えるようになりました。

透明のオブジェクト

しかし、塗りつぶしも線も「なし」に設定するとレイアウト上でオブジェクトをクリックしづらくなります。そこにボタンオブジェクトがあるとわからなければ、「オブジェクト」タブでもわかりづらくなります。そこで、次のように工夫します。

・レイアウトモードではわかりやすい表示にする
・オブジェクト名を付与して「オブジェクト」タブで見つけやすくする

塗りと線が「なし」のオブジェクトをわかりやすくします。

FileMaker WebDirect での透明のオブジェクト

FileMaker WebDirect では、最前面のオブジェクトが透明な場合やオブジェクトグループの何もない場所をクリックした場合でも、他のレイアウトオブジェクトの背後にあるオブジェクトを選択することはできません。複数のオブジェクトを重ねて配置する代わりに、次の方法を検討してください。

・タブラベルに計算値を使用する
・ボタンの前にイメージを重ねて配置する代わりにボタンにアイコンを追加する
・複数のオブジェクトをグループ化してボタン設定を適用する

演習 10-10 レイアウト上で計算式を使う 5（条件付き書式）

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「見積リスト」レイアウトでレイアウトモードに切り替えます。

(2)　ボタンの塗りつぶし色を目立つ色（例えば緑）にします。

(3)　条件付き書式に 1 つ条件を追加して、塗りつぶしを「なし」にします（手順は省略します）。
次は、ブラウズモードだったらという計算式です。

```
Get ( ウインドウモード ) = 0
```

(4)　この条件を、一番上に並べてください。

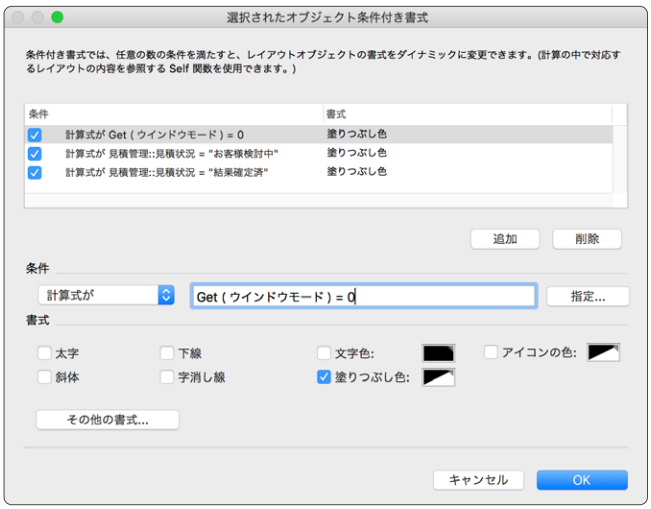


図 10-23 複数の条件付き書式の設定後

(5)　ついでにボタンオブジェクトをロックします。
（上に乗っているレイアウトオブジェクトを作業するときに動かしてしまわないように）

a.　ボタンオブジェクトをクリックします。

b.　インスペクタの「位置タブ」の[配列と整列]の「ロック」で錠マークをクリックします。

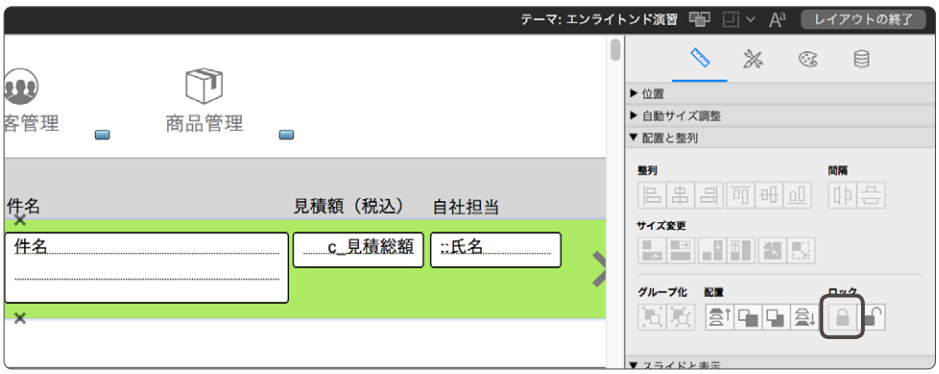


図 10-24 錠をクリックするとロックされる

(6)　レイアウトを保存して表示を確認してください。
ブラウズモードでは、条件付き書式が働きます。
レイアウトモードでは、塗りつぶしの緑色が表示されます。

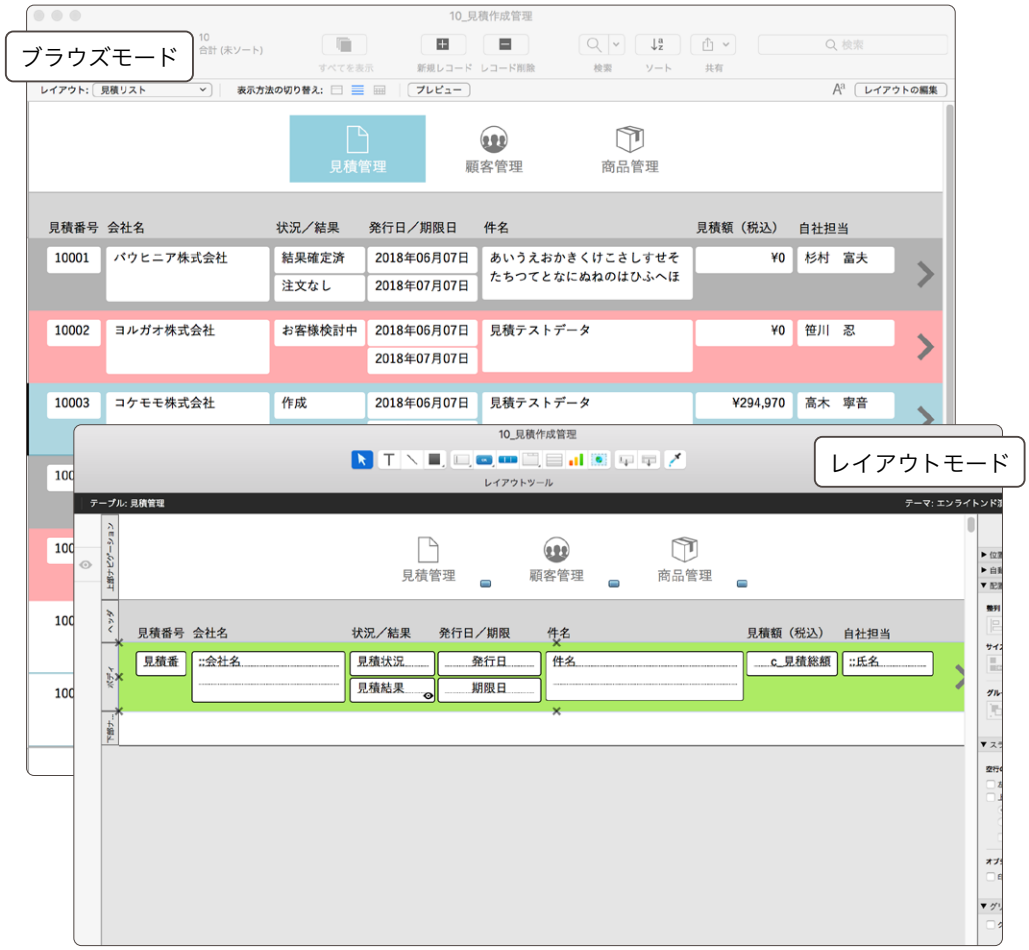


図 10-25 作業後のモードによる表示の違い

演習ここまで

この演習では、わかりやすい配色を使いましたが、濃い色がたくさんレイアウトに表示されて良いかどうかは実際にカスタム App を使うユーザと相談します。

ブラウズモードでのみボタンの塗りつぶしを透明にするという設定をしました。他にも検索モードやプレビューモードでも透明にする必要があります。

また、遠くから画面を確認する必要があれば、濃い色がわかりやすいかもしれません。明るい場所、暗い場所、色の多さ、少なさなど、ユーザの作業しやすさ、見やすさ（目の疲れやすさ）とあわせて検討します。

10.6　関数　その 2

また別のタイプの関数を確認します。

10.6.1　取得関数

この種類の関数は引数を指定せず、そのまま使います。主に「いま○○はどうなっているか」を知ることができる関数です。スクリプトでよく使います。

演習 10-10 で使用したのは以下です。

```
Get ( ウインドウモード )
  →  現在のウインドウモードを評価した値を返します。
     ブラウズモードは「0」
     検索モードは「1」
     プレビューモードは「2」
     印刷中「3」
     レイアウトモードは「4」（データビューアで確認するときに使います）
```

この関数は、まとめて解説しません。他の関数や演習、スクリプトなどで一緒に記述します。

10.6.2　テキスト関数 2

文字列の操作とは少し違う関数も用意されています。

```
GetAsDate( テキスト )    →  テキストに含まれる日付を返す
```

テキストタイプフィールドに "2018/1/1" と入力されている場合、計算式で日付として取り扱うには GetAsDate 関数を使います。あるいは、1（0001/1/1）から 1460970（4000/12/31）までの日付を表す数字を日付に変換することが可能です。

例えば、

```
GetAsDate("2018/1/1")
```

→ テキスト「2018/01/01」が返ります。

```
GetAsDate(789012)
```

→ テキスト「2161/03/29」が返ります。

これらの特性を利用して、経過日数を計算することも可能です

やってみよう

```
GetAsDate("2019/1/1") - GetAsDate("2018/1/1")
```

などの式を入力して、結果を確認してみましょう。

（この後、関数の説明ページに「やってみよう」は記述しませんが、気になる関数などを、どんどんデータビューアで試してみましょう）

GetAsNumber(テキスト) → テキストに含まれる数字を返す

例えば、
GetAsNumber("あ1い2う3")
→ 数字「123」が返ります。

Quote(テキスト)

指定されたテキストを「"（ダブルクォーテーションマーク）」で囲むことで、そのテキストがそのままテキスト形式で返ります。Evaluate関数などで評価されないように利用できます。

Quote (2+2)
→ テキスト「4」が返ります。

Quote ("2+2")
→ テキスト「2+2」が返ります。 → 「"」で囲んだためです。

Substitute (テキスト ; 検索テキスト ; 置換テキスト)
→ 指定したテキストを検索して置換

大文字と小文字を区別します。アルファベットを含む文字列を置換する場合には注意が必要です。

Substitute ("ファイトメーカー" ; "ト" ; "ル")
→ テキスト「ファイルメーカー」が返ります。

置換する文字列は一度に複数指定して記述できます。
置換の順番は、最初の置換をした結果に対して2つ目の置換、その結果に対して3つ目の置換...と評価されます。複数の置換を指定する場合は順番に気をつけます。

```
Substitute (
  "fileMaker pro" ;
   ["f" ; "F"] ; //この時点では「FileMaker pro」
   ["p" ; "P"] ; // この時点では「FileMaker Pro」
   ["Pro" ; "Pro 19"] // →「FileMaker Pro 19」
  )
```
→ テキスト「FileMaker Pro 19」が返ります。

10.6.3　論理式（0 か 1 か）

オブジェクトの［次の場合にオブジェクトを隠す］や条件付き書式では、設定した計算式の結果は、0 または 1 として返ります。その結果を元に FileMaker Pro が判断し動作します。他にも、If 関数や、ポータルフィルタも論理式の結果で動作が分岐しています。

論理式という概念です。真（True）か偽（false）のどちらかが結果として返る式です。この結果は数値で、

・真（True）は　　1
・偽（False）は　　0

です。真偽値などと呼びます。例えば、条件付き書式で記述した次の計算式では、

```
Exact ( 見積管理::見積状況 ; "お客様検討中" )
```

次のような判断がなされています。

・「見積状況」フィールドが「お客様検討中」→　真（True）は 1　　→　書式適用
・「見積状況」フィールドが上記以外　　→　偽（False）は 0　　→　書式非適用

論理式を書く場合、特にデータタイプに気をつける必要があります。

データタイプと論理式
文字列 (テキストタイプのデータ) を比較演算子で評価した論理式の結果は、少し曖昧な評価の結果が返ることに注意してください。比較演算子では文字列の大文字と小文字を区別しません。

以下の計算式は 1（真）を返します。

```
"A" = "a"
```

文字列の大文字と小文字を区別して比較するには、Exact 関数を 使います。次の式は 0（偽）を返します。

```
Exact ( テキスト ; 比較テキスト ) → Exact ( "A" ; "a" )
```

数値や日付、タイムスタンプタイプを比較演算子で評価する場合、フィールドのデータを利用するのであれば、その計算式に必要なフィールドタイプ（数値や日付、タイムスタンプ）が適切に設定されていないと正しい結果になりません。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 10_11 から 17」フォルダの中の「10_見積作成管理 .fmp12」ファイルをベースに進めます）

10.7　レイアウトと計算式 その 2

レイアウト上の計算式について、別の機能を確認します。

10.7.1　ポータルフィルタ

ポータルは、表示されるデータをフィルタ（絞り込み）が可能です。計算式で条件を指定します。

※「現在のテーブル」のポータルでは動作しません。

ポータルフィルタでは [ポータル設定] ダイアログに設定される計算式の結果が論理値「0 または 1」として評価されます。そして次のようにデータ表示に反映されます。

・偽（False）　0　→　フィルタ非表示　　計算式で指定した条件に合致しない
・真（True）　1　→　フィルタ表示　　計算式で指定した条件に合致する

ポータルフィルタの演習をおこないます。演習ファイルにテスト用のデータも追加しました。

「見積詳細」レイアウトに新しくタブコントロールをつくり、2 番目の「見積履歴」タブに、指定されている顧客の「過去の見積情報」レコードを表示するためのポータルを置きました。

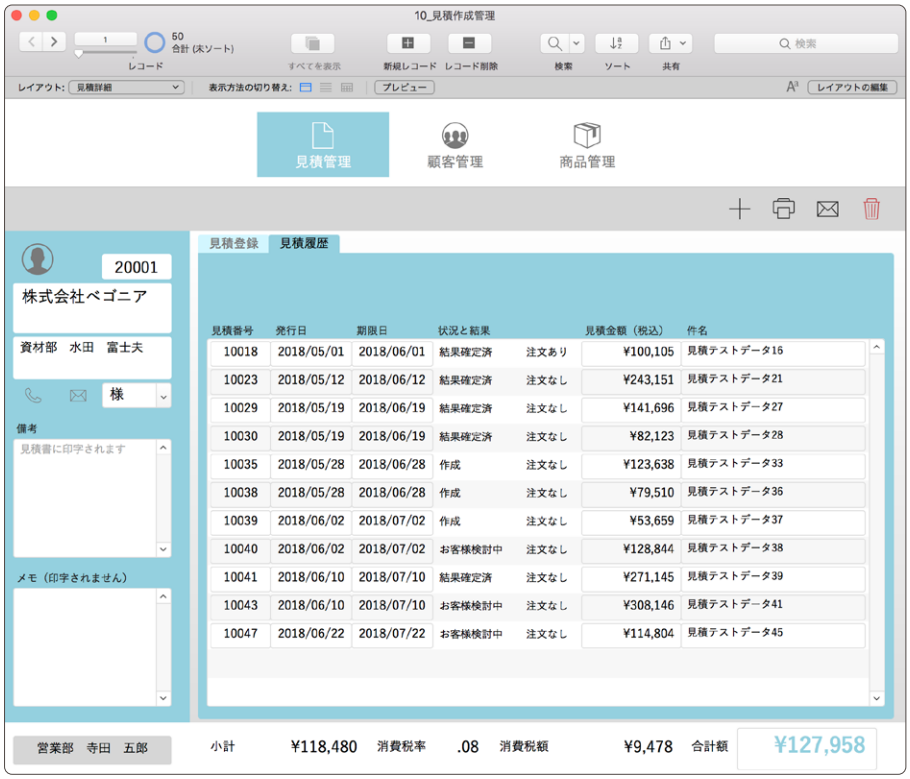


図 10-26 変更した「見積詳細」レイアウト

このポータルには、「見積管理」TO と 2 ペアの照合フィールドの関連が定義されている自己リレーションシップの TO が設定されています。次の関連レコードを表示するためです。

・現在の見積書と同じ顧客に提示した見積書
・ただし『現在の見積書以外』の情報に限定

次の定義です。

「同じ顧客外部キーを持つレコード」のための照合
「見積管理」TO の「顧客外部キー」＝「見積管理｜履歴｜自己」TO の「顧客外部キー」

「同じ主キーを持つレコード以外」のための照合
「見積管理」TO の「主キー」≠「見積管理｜履歴｜自己」TO の「主キー」

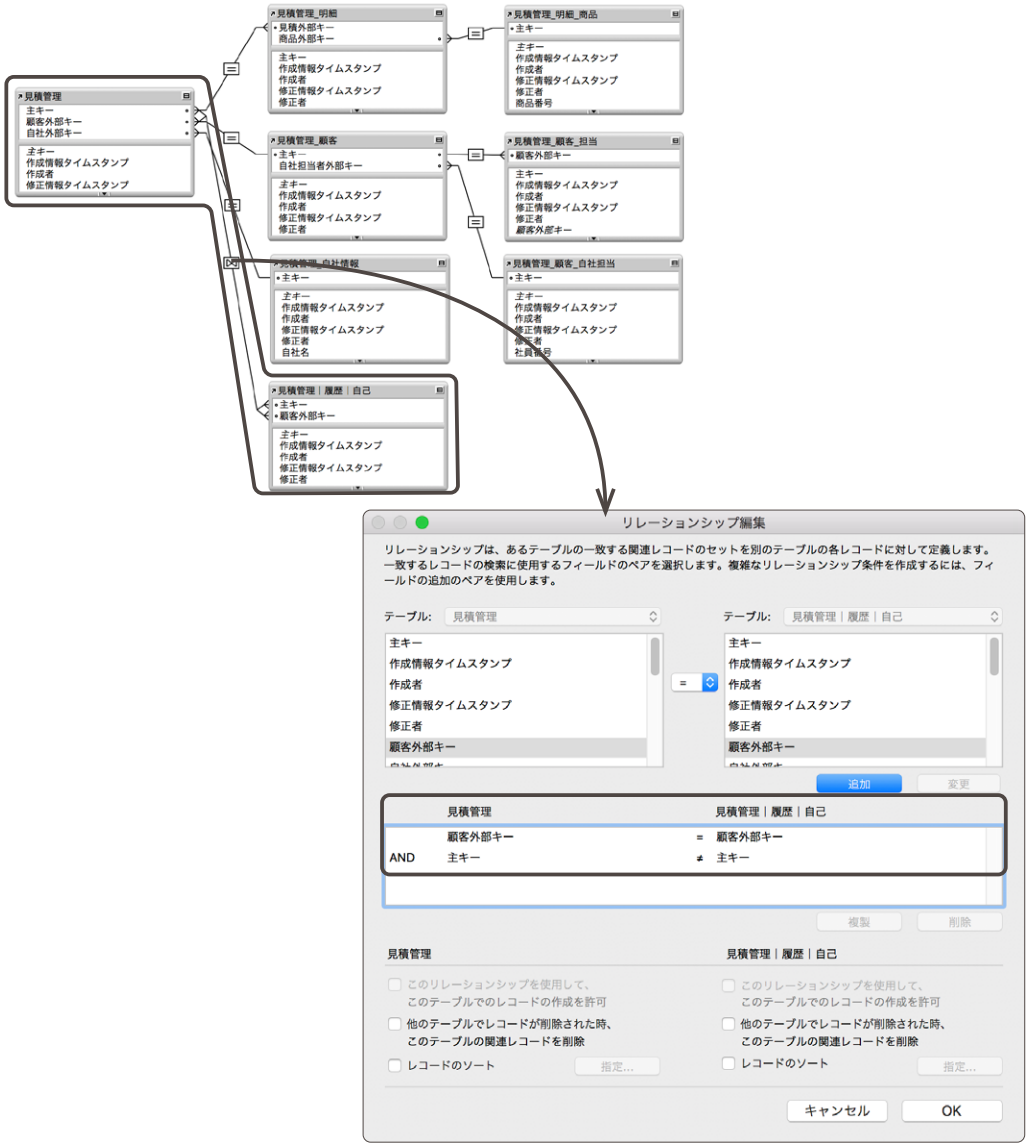


図 10-27 リレーションシップグラフの TOG と関連の定義

このポータルを「見積状況」フィールドのデータでフィルタします。

「演習 10_11 から 17」フォルダ

演習 10-11 ポータルフィルタで計算式を使う：準備

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) 「ユーティリティ」テーブルにポータルフィルタのためのフィールドが作成されていますので、確認します。

a. ［データベースの管理］ダイアログを開きます。

b. 「ユーティリティ」テーブルの「g_ 見積状況｜フィルタ用」フィールドを確認します。

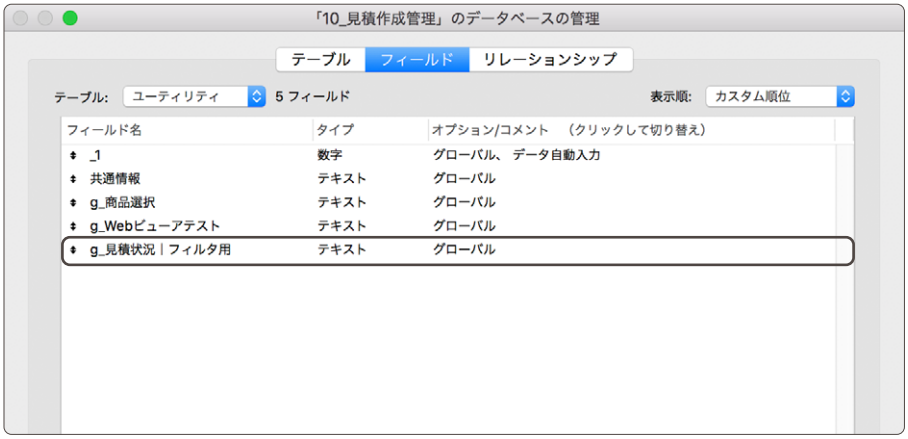


図 10-28 ポータルフィルタ用のグローバル格納フィールド

c. 確認したら［キャンセル］クリックしてダイアログを閉じます。

(2) グローバル格納フィールドに入力するフィルタ用データには「見積状況」値一覧を使います。

a. ［値一覧の管理］ダイアログで確認してください。

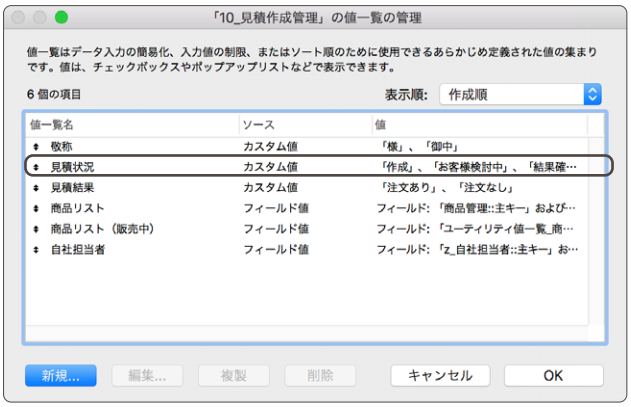


図 10-29「見積状況」値一覧

b. 確認したら［キャンセル］クリックしてダイアログを閉じます。

(3) 「見積詳細」レイアウトでレイアウトモードに切り替えて「見積履歴」タブをクリックします。

(4)　ポータルフィルタ用のフィールドをポータルに配置します（次図を参照）。

a.　左パネルの「フィールド」タブから配置します。

・現在のテーブルは　「z_ ユーティリティ」
・フィールドは　「g_ 見積状況｜フィルタ用」
・フィールド配置は　「縦」
・フィールドラベルは　「なし」
・コントロールスタイルは　「編集ボックス」

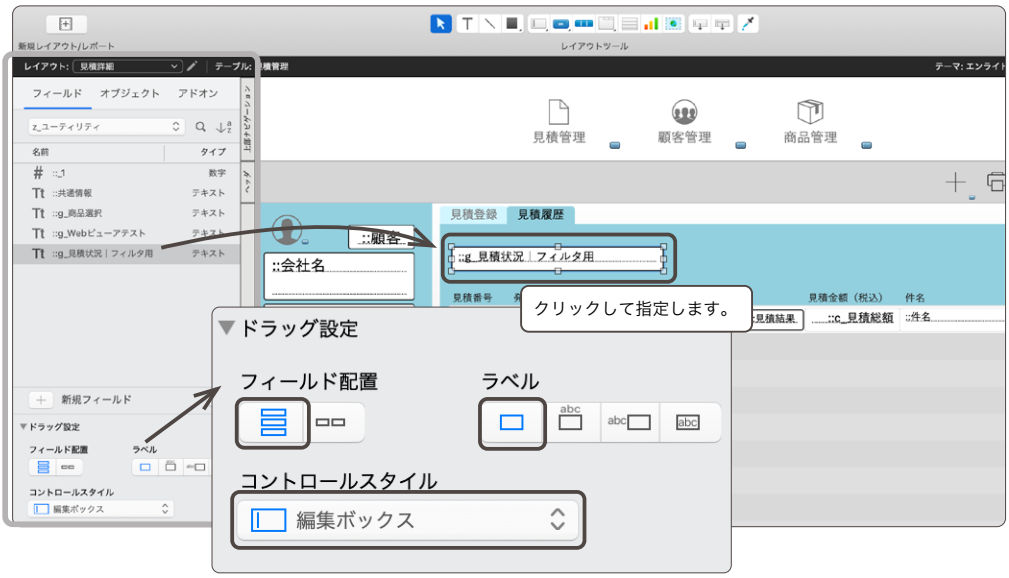


図 10-30 グローバル格納フィールドを配置

(5)　配置した「g_ 見積状況｜フィルタ用」フィールドをクリックしてインスペクタの「データ」タブの［コントロールスタイル］で値一覧を割り当てます。

・ポップアップメニュー
・値一覧は　「見積状況」
・［一覧の表示切り替え用矢印を表示］を　**オン**
・［値一覧でデータの書式を上書き］を　**オン**



図 10-31 値一覧を割り当て

(6)　ポータルをクリックしてインスペクタの「位置」タブの［位置］の［名前］に「見積履歴ポータル」と入力します。



図 10-32 オブジェクト名を付与

(7)　準備ができました。

演習ここまで

引き続きポータルフィルタを設定します。

演習 10-12 ポータルフィルタで計算式を使う 1

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「見積詳細」レイアウトでレイアウトモードに切り替えて「見積履歴」タブをクリックします。ポータルフィルタを設定します。

a.　「見積履歴ポータル」をダブルクリックします。

b.　［ポータル設定］ダイアログが表示されます。

c.　「ポータルレコードのフィルタ」の右の［指定 ...］をクリックします。



図 10-33 ポータルフィルタの設定箇所

d. 計算式を入力します。

```
If (
z_ユーティリティ::g_見積状況｜フィルタ用 = 見積管理｜履歴｜自己::見
積状況 ; 1 ;0 )
```

フィルタ用のフィールドを比較をする計算式です。

e. ［OK］をクリックしてダイアログを閉じます。

(2) レイアウトを保存してブラウズモードに戻ります。

演習ここまで

ポータルフィルタを作成しました。設定した計算式では、「g_ 見積状況｜フィルタ用」フィールドについて次のような条件でフィルタが実行されます。

・入力あり／「見積状況」フィールドと同じ → 真（True）1 → フィルタ適用（表示）
・入力あり／「見積状況」フィールドと違う → 偽（False） → フィルタ適用（非表示）

早速確認してみます。現在、ブラウズモードではなにも表示されていません。

演習 10-13 ポータルフィルタの確認

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) 「見積詳細」レイアウトで「g_ 見積状況｜フィルタ用」フィールドの値一覧で、「作成」を選択します。

(2) ポータルの表示は変わりません。

(3) 「g_ 見積状況｜フィルタ用」フィールドの値一覧で「お客様検討中」を選択します。

(4) やはりポータルの表示は変わりません。

(5) ウインドウの再表示を行います。
a. ［レコード］–［ウインドウ内容の再表示］メニューを選択します。
b. 表示が変わります。

(6) 次の作業を **何度か繰り返して** ポータルフィルタが動作し表示が変わること確認してください。
「g_ 見積状況｜フィルタ用」フィールドの値一覧で選択後、ウインドウ内容の再表示

演習ここまで

ポータルフィルタは動作していますが、毎回ウインドウ内容の再表示が必要です。操作がわかりづらい上に面倒です。これを自動で行うために「スクリプトトリガ」と「スクリプト」を使います。

ポータルフィルタを動作させるには「ウインドウ内容の再表示」ではなく「**ポータルの更新**」を実行すれば十分です。

［ポータルの更新］スクリプトステップ
関連レコードを表示しているポータルの内容を更新します。ポータルフィルタの計算式で使用されているグローバルフィールドが変更、保存、確定されたときなどに動作するようにスクリプトトリガと一緒に使うことがあります。
今回は、新しいデータが入力され確定したタイミングである「OnObjectSave」トリガでポータルを更新します。更新時に、設定した計算式の評価が起きて結果が返り、フィルタが動作します。

演習 10-14 ポータルフィルタでスクリプトトリガを使う

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「見積詳細」レイアウトでレイアウトモードに切り替えて「見積履歴」タブをクリックします。

(2)　フィールドにスクリプトトリガを割り当てて、実行するスクリプトを作成します。
a.　「g_ 見積状況｜フィルタ用」フィールドをクリックします。
b.　右クリックでショートカットメニューから［スクリプトトリガ設定 ...］を選択します。
c.　［スクリプトトリガ設定］ダイアログが表示されます。
d.　「OnObjectSave」を　**オン**　にします。
e.　［選択 ...］をクリックします。
f.　［スクリプト指定］ダイアログが表示されます。
g.　「＋」ボタンをクリックします。
h.　「見積履歴ポータル更新」と名前を入力します。
i.　次のスクリプトステップとオプション（オブジェクト名）を設定します（次図も参照）。

ポータルの更新　[見積履歴ポータル]

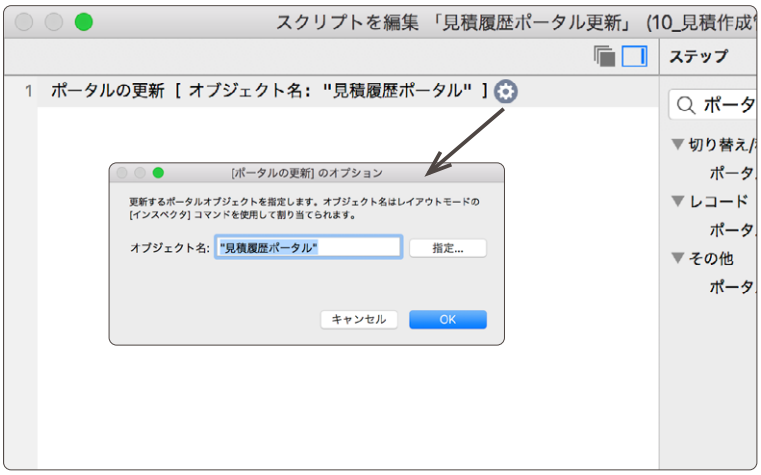


図 10-34　トリガに割り当てるスクリプトを作成

j.　［OK］を何度かクリックして各種設定ダイアログを閉じます。
k.　「g_ 見積状況」フィールドに、「OnObjectSave」トリガを割り当てました。

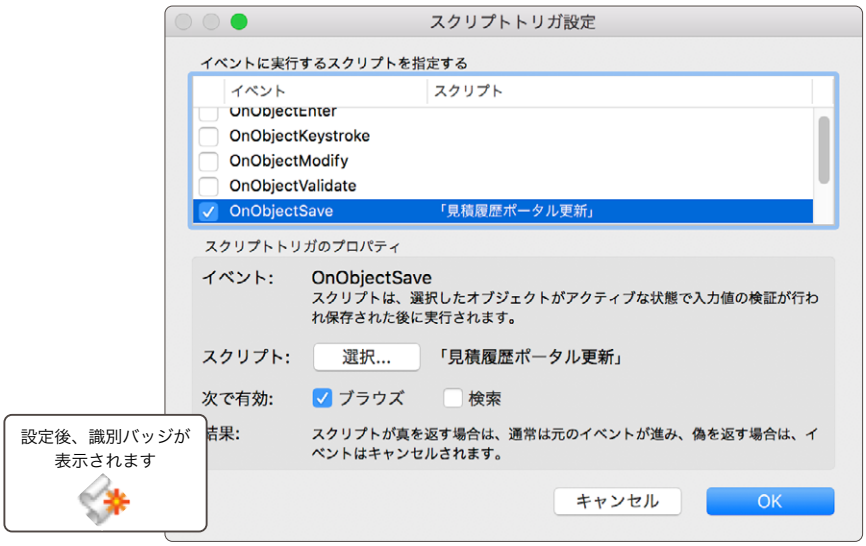


図 10-35［スクリプトトリガ設定］ダイアログ（割り当て後）

演習ここまで

これで自動でポータルフィルタが動作します。早速、ブラウズモードで確認します。

演習 10-15 ポータルフィルタの確認

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「見積詳細」レイアウトで「g_ 見積状況｜フィルタ用」フィールドの値一覧で、「作成」を選択します。

(2)　ポータルの表示が「作成」のみにフィルタされます。

(3)　「g_ 見積状況｜フィルタ用」フィールドの値一覧で「お客様検討中」を選択します。

(4)　ポータルの表示が「お客様検討中」のみにフィルタされます。

演習ここまで

ポータルのデータをフィルタせずに表示するにはどうしたらよいのでしょうか。状況ごとに確認したい場合と、顧客に発行されているすべての見積の状況を確認したい場合があります。

すべて、つまり「なにもフィルタしない」という条件は、次のように考えます。

・「g_ 見積状況｜フィルタ用」フィールドが空である（＝なにも指定されていない）

この条件をポータルフィルタの計算式に使う必要があります。まず、フィールドを空にするボタンを作成します。

演習 10-16 ポータルフィルタをリセットするボタン

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) 「見積詳細」レイアウトでレイアウトモードに切り替えて「見積履歴」タブをクリックします。

(2) ボタンオブジェクトを作成します。

a. ステータスツールバーの［ボタン］アイコンをクリックします。
b. 「g_ 見積状況｜フィルタ用」フィールドの右側でドラッグして四角形を描きます。
c. マウスボタンを離すと［ボタン設定］ダイアログが表示されます。
d. 右図のアイコンをクリックします。
e. 次のように設定します（右図も参照してください）。
・「アイコンだけ」
・［アイコン］は　「×」右図を参照
・［アイコンの大きさ］は　「18pt」



図 10-36［ボタン設定］ダイアログ

(3) 作成した「×」ボタンにスタイルを割り当てます。

a. ボタンをクリックします。
b. インスペクタの「スタイル」タブで「機能ボタン赤」を選択します。

(4) 作成した「×」ボタンにスクリプトを割り当てます。

a. ボタンをダブルクリックします。
b. ［ボタン設定］ダイアログが表示されます。
c. ［処理：］のポップアップメニューで「スクリプト実行」を選択します。
d. ［スクリプト指定］ダイアログが表示されます。
e. 「＋」ボタンをクリックします。
f. **「フィルタフィールドクリア」**と名前を入力します。
g. 次の 2 行のスクリプトステップとオプションを設定します。

```
消去　[ z_ユーティリティ ::g_見積状況｜フィルタ用　]
ポータルの更新　[ 見積履歴ポータル ]
```

［消去］スクリプトとステップのオプション
・内容全体を選択　は　**オン**
・指定フィールドへ移動　は　**オン**　z_ ユーティリティ ::g_ 見積状況｜フィルタ用　を指定
［ポータルの更新］スクリプトとステップのオプション
・オブジェクト名は　「見積履歴ポータル」

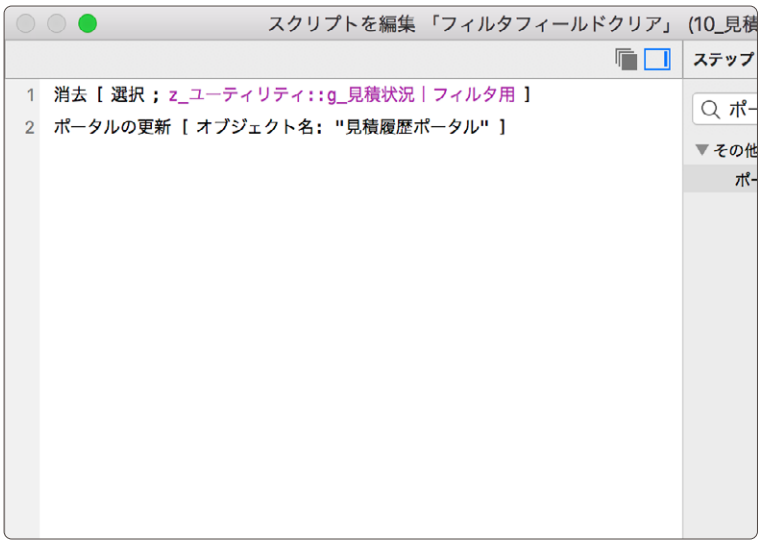


図 10-37 スクリプトを作成

h. [OK] を何度かクリックして各種設定ダイアログを閉じます。

(5) レイアウトを保存してブラウズモードに戻ります。

(6) 動作を確認します。

a. 1 レコード目に移動して「見積履歴」タブをクリックします。

b. 「g_ 見積状況｜フィルタ用」フィールドポップアップメニューで何か値を選択すると、ポータルがフィルタされて表示内容が変更されます。

c. 「×」ボタンをクリックして次のことを確認してください。

・「g_ 見積状況｜フィルタ用」フィールドがクリアされる

・ポータルになにも表示されない



図 10-38「×」ボタンができた

演習ここまで

さらにポータルフィルタの計算式にも空の場合の条件を追加します。

演習 10-17 ポータルフィルタで計算式を使う 2

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) 「見積詳細」レイアウトでレイアウトモードに切り替えて「見積履歴」タブをクリックします。
ポータルフィルタを設定します。
a. 「見積履歴ポータル」をダブルクリックします。
b. ［ポータル設定］ダイアログが表示されます。
c. 「ポータルレコードのフィルタ」の右の［指定 ...］をクリックします。
d. 計算式を入力します。
演習 10-12 の計算式を変更して、フィルタ用のフィールドが空欄だったら「1」つまり「すべてを表示する」という条件を加えます。

```
If (
IsEmpty ( z_ユーティリティ::g_見積状況｜フィルタ用 ) ; 1 ;
z_ユーティリティ::g_見積状況｜フィルタ用 = 見積管理｜履歴｜自己::見積状況 )
```

e. ［OK］をクリックしてダイアログを閉じます。

(2) レイアウトを保存してブラウズモードに戻ります。

演習ここまで

ポータルフィルタが出来上がりました。「g_ 見積状況｜フィルタ用」フィールドの値一覧を選択したり［×］ボタンでクリアしたりして、表示を確認します。

ポータルフィルタと関連
細かい条件分岐によるデータの表示を、計算式を利用して簡単に実現できることがポータルフィルタの良い点です。

なお、注意点があります。フィルタは設定されたポータルオブジェクトのみで機能しています。ポータルフィルタでは関連レコードの表示後、レイアウト上でフィルタを動作させています。

例えば「見積管理｜履歴｜自己」TO の関連レコードのデータを List 関数で取得するとリレーションシップに基づいたフィルタされていない結果が返ります。

この演習で設定した計算式では、「g_ 見積状況｜フィルタ用」フィールドについて次のような条件でフィルタが実行されます。

・空欄（入力なし） → 真（True）1 → フィルタ適用（全表示）
・入力あり／「見積状況」フィールドと同じ → 真（True）1 → フィルタ適用（一部表示）
・入力あり／「見積状況」フィールドと違う → 偽（False） → フィルタ適用（一部非表示）

いろいろな計算式の書き方

演習 10-12 では次の計算式でした。

```
If (
z_ユーティリティ::g_見積状況｜フィルタ用 = 見積管理｜履歴｜自己::見積状況 ;
1 ;0 )
```

演習 10-17 で「空欄だったら」という条件を加えました。

```
If (
IsEmpty ( z_ユーティリティ::g_見積状況｜フィルタ用 ) ; 1 ;
z_ユーティリティ::g_見積状況｜フィルタ用 = 見積管理｜履歴｜自己::見積状況 )
```

空欄でない場合の計算式は次の内容です。演習 10-12 と違い、If 関数が省略されています。

```
z_ユーティリティ::g_見積状況｜フィルタ用 = 見積管理｜履歴｜自己::見積状況
```

この式でも「=」の左右のデータの比較評価の結果として演習 10-12 の計算式と同じ「0 または 1」が返ります。つまり、同じ結果になるので省略して評価式のみの記述としています。

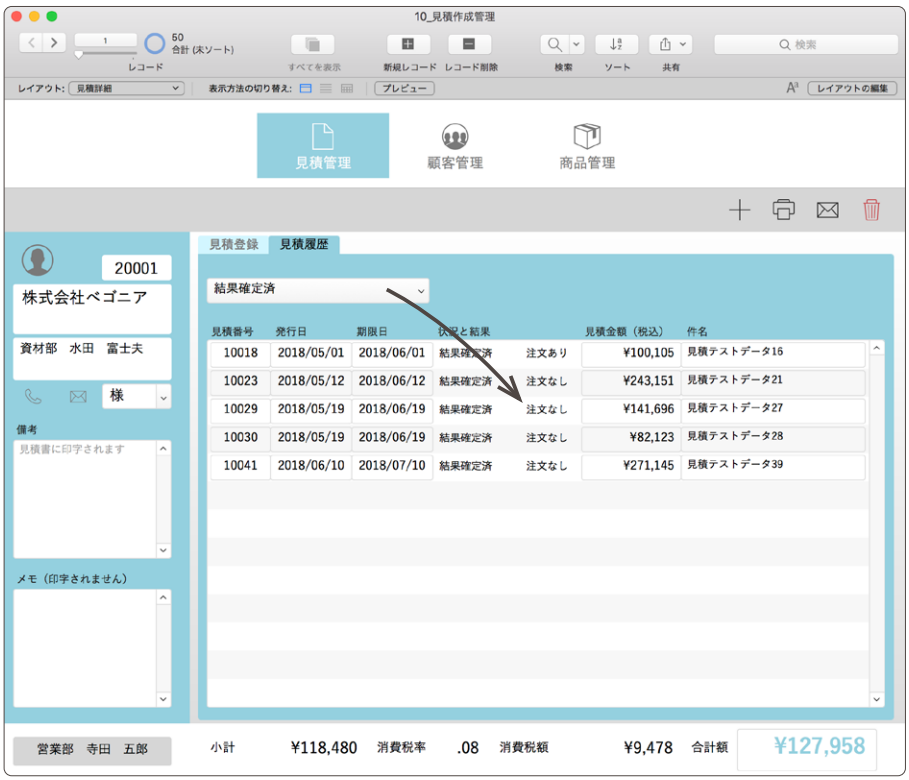


図 10-39 ポータルフィルタ完成後

演習ファイルを閉じて、次のファイルを開いてください。

10.7.2　ポータルフィルタと集計タイプフィールド

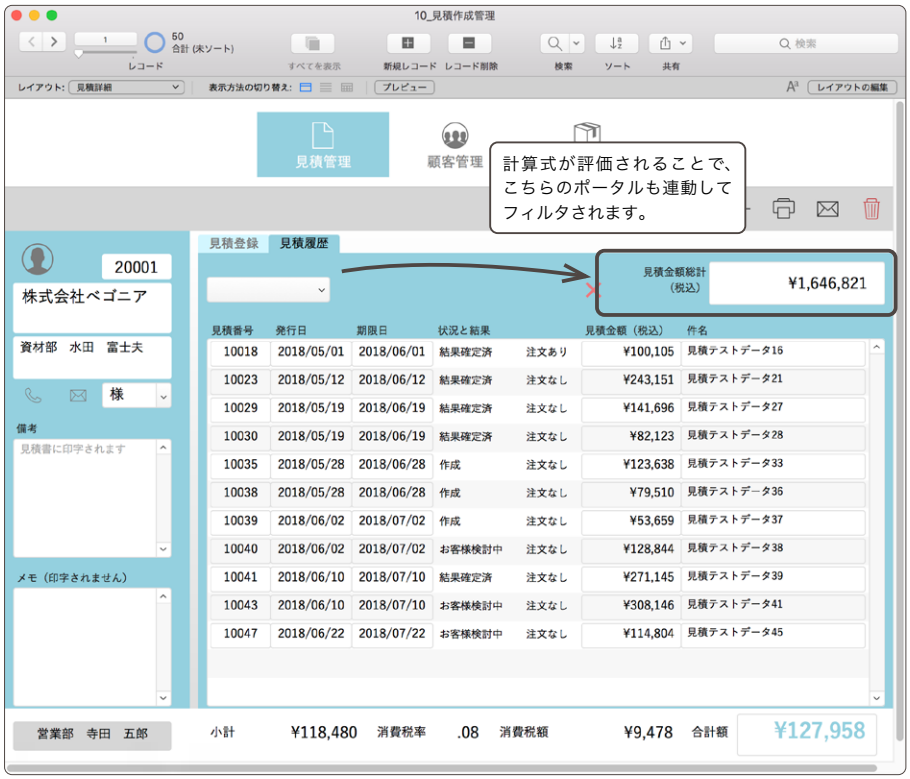
ポータルフィルタで表示されているデータと連動した集計結果を表示するにはどうしたらよいでしょうか。フィルタは設定されたポータルオブジェクトでのみ動作します。集計タイプフィールドをレイアウト上に配置するだけでは集計結果は連動しません。

（ここから、本書に付属している演習ファイルの「演習 10_18」フォルダの中の「10_ 見積作成管理 .fmp12」ファイルをベースに進めます）

演習ファイルに、さらに機能を追加しました。「見積詳細」レイアウトで 1 レコード目に移動して「見積履歴」タブをクリックします。タブの右上に集計タイプの「s_ 見積額」フィールドがあります。

「見積詳細」レイアウト

「g_ 見積状況」フィールドにポップアップメニューを選択してデータを入力したり「×」ボタンをクリックして表示される金額の変化を確認してください。ポータルフィルタと連動します。



見積番号	発行日	期限日	状況と結果		見積金額（税込）	件名
10018	2018/05/01	2018/06/01	結果確定済	注文あり	¥100,105	見積テストデータ16
10023	2018/05/12	2018/06/12	結果確定済	注文なし	¥243,151	見積テストデータ21
10029	2018/05/19	2018/06/19	結果確定済	注文なし	¥141,696	見積テストデータ27
10030	2018/05/19	2018/06/19	結果確定済	注文なし	¥82,123	見積テストデータ28
10035	2018/05/28	2018/06/28	作成	注文なし	¥123,638	見積テストデータ33
10038	2018/05/28	2018/06/28	作成	注文なし	¥79,510	見積テストデータ36
10039	2018/06/02	2018/07/02	作成	注文なし	¥53,659	見積テストデータ37
10040	2018/06/02	2018/07/02	お客様検討中	注文なし	¥128,844	見積テストデータ38
10041	2018/06/10	2018/07/10	結果確定済	注文なし	¥271,145	見積テストデータ39
10043	2018/06/10	2018/07/10	お客様検討中	注文なし	¥308,146	見積テストデータ41
10047	2018/06/22	2018/07/22	お客様検討中	注文なし	¥114,804	見積テストデータ45

図 10-40　ポータルフィルタと集計額が連動する

まず［データベースの管理］ダイアログを開いて、新しく作成されたフィールドの内容を確認します。ポータルに表示されている見積データの見積総額を合算する集計タイプフィールドです。

・「s_ 見積額」フィールド　→　「c_ 見積総額」フィールドの合計を集計

集計フィールドのオプションは、次図の内容です。確認後、［キャンセル］をクリックしてダイアログを閉じてください。

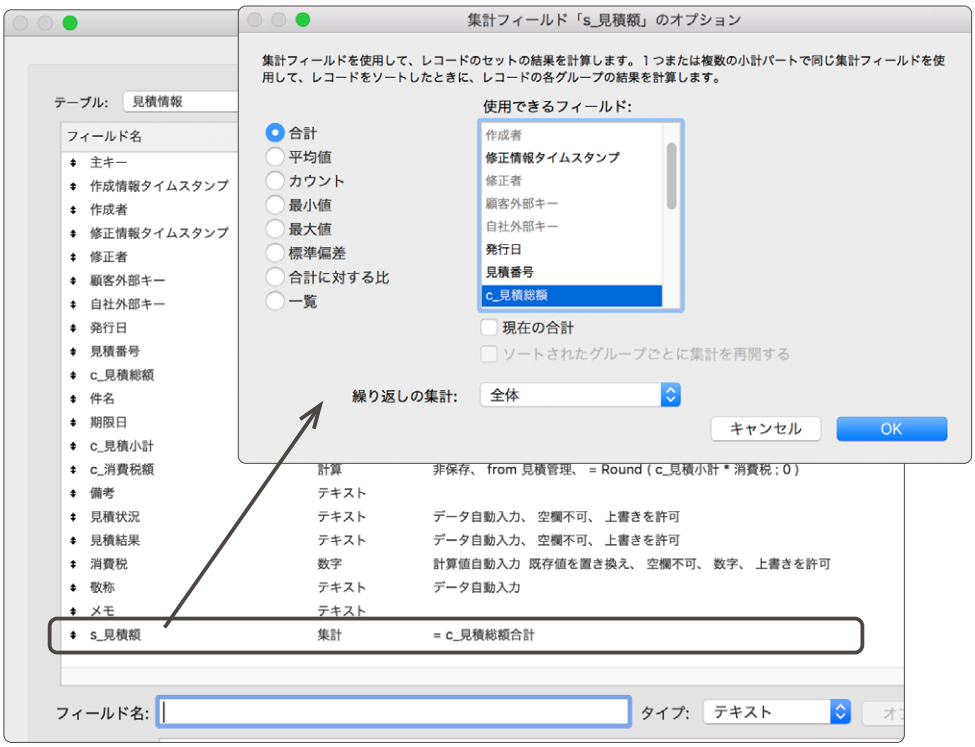


図 10-41「s_ 見積額」フィールド

「見積詳細」レイアウトでレイアウトモードに切り替えてください。

「見積履歴」タブの「s_ 見積額」フィールドは「見積履歴」ポータルとは別のポータル上にあります（わかりやすいようにフィールドよりも大きめにポータルを作成しました）。

このポータルをダブルクリックして［ポータル設定］ダイアログを確認します。

・「見積履歴」ポータルと同じ「見積管理｜履歴｜自己」TO のレコードを表示している
・フィルタに設定されている計算式が「見積履歴」ポータルと同じ

2 つのポータルの計算式が同じなので「g_ 見積状況」フィールドのデータが変わると同時にフィルタされていることがわかります。

・「見積履歴」ポータル内のデータ
・別のポータル上の「s_ 見積額」フィールドの集計結果

確認後、［キャンセル］をクリックしてダイアログを閉じてください。

やってみよう

「見積管理｜履歴｜自己」TO の「s_ 見積額」フィールドを「見積履歴」タブに新たに配置すると、どのようなデータが表示されるか確認してみましょう。「g_ 見積状況」フィールドの値を変更しても連動せず、関連レコードすべての集計結果が常に表示されるはずです。

10.7.3 ポータルフィルタ（2 つのフィールドの条件）

演習ファイルには「顧客管理｜マスタ詳細」レイアウトも作成されていますので、レイアウトを切り替えてください。「見積履歴」タブをクリックしてください。ポータルがあります。このポータルは、「見積詳細」レイアウトの「見積履歴」タブと同じような構造です。

ポータルの関連

「顧客管理」TO と関連が定義されている「顧客管理 _ 見積」TO（見積情報テーブル）が設定されています。次の関連レコードを表示するためです。

・現在の顧客に提示した見積書
　→ 「顧客管理」TO の「主キー」＝「顧客管理 _ 見積」TO の「顧客外部キー」

フィルタするデータを入力するグローバルタイプフィールド

ポータルフィルタに利用しているのは、「ユーティリティ」テーブルの 2 つのフィールドです。

・「g_ 見積状況｜フィルタ用」フィールド
・「g_ 見積結果｜フィルタ用」フィールド

2 つのフィールドのポップアップメニューを選択したり「×」ボタンをクリックして表示を確認してください。ポップアップメニューの選択が片方でも両方でもフィルタは動作します。

フィルタの条件

次の条件が計算式で設定されています（If 関数を「and」演算子で繋げて記述しています）。

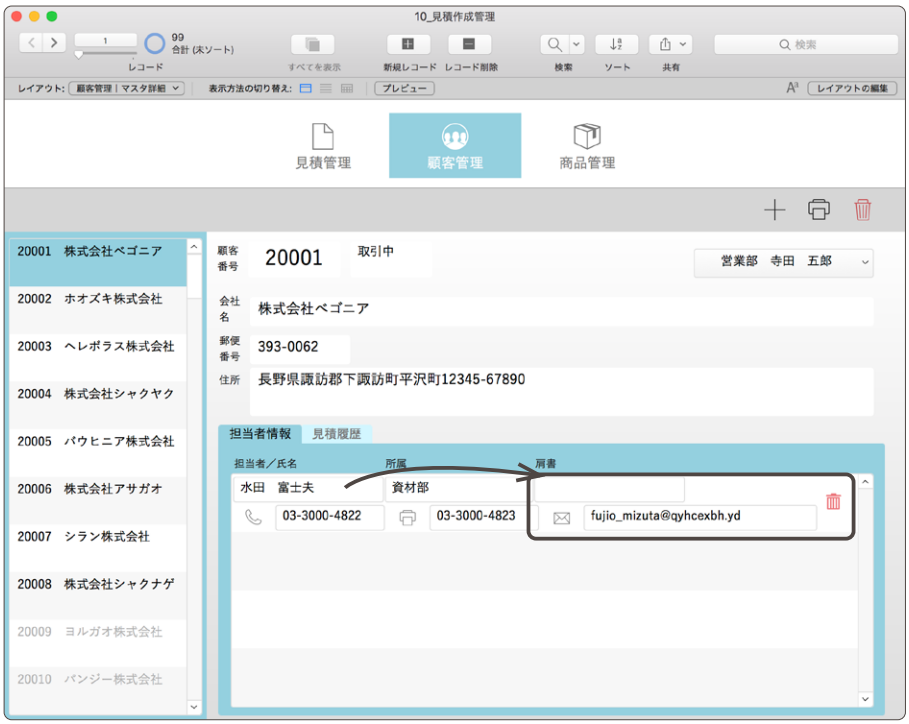
「g_ 見積状況｜フィルタ用」フィールド
・空（入力なし）　→ 真（True）1　→ フィルタ適用（全表示）
・入力あり／「見積状況」フィールドと同じ　→ 真（True）1　→ フィルタ適用（表示）
・入力あり／「見積状況」フィールドと違う　→ 偽（False）0　→ フィルタ適用（非表示）

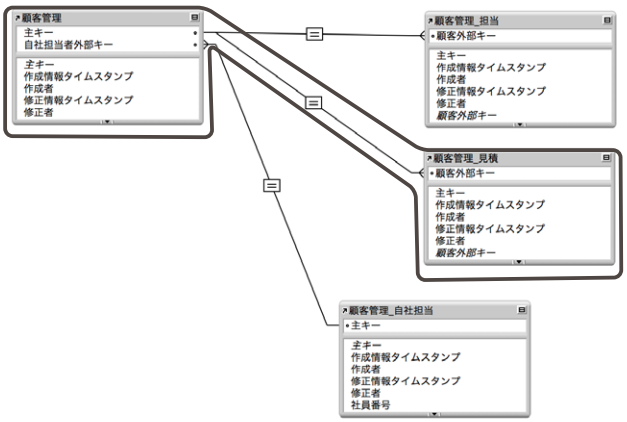
「g_ 見積結果｜フィルタ用」フィールド
・空（入力なし）　→ 真（True）1　→ フィルタ適用（全表示）
・入力あり／「見積結果」フィールドと同じ　→ 真（True）1　→ フィルタ適用（表示）
・入力あり／「見積結果」フィールドと違う　→ 偽（False）0　→ フィルタ適用（非表示）

```
If (
    IsEmpty ( z_ユーティリティ::g_見積状況｜フィルタ用 ) ; 1 ;
    z_ユーティリティ::g_見積状況｜フィルタ用 = ユーティリティ管理_顧客｜選択_見積::見積状況 )
and
If (
    IsEmpty ( z_ユーティリティ::g_見積結果｜フィルタ用 ) ; 1 ;
    z_ユーティリティ::g_見積結果｜フィルタ用 = ユーティリティ管理_顧客｜選択_見積::見積結果 )
```

and 演算子

計算式では「&」は、左右を連結するための演算子ですが、「and」は検索条件の and と同じ意味で使われます。





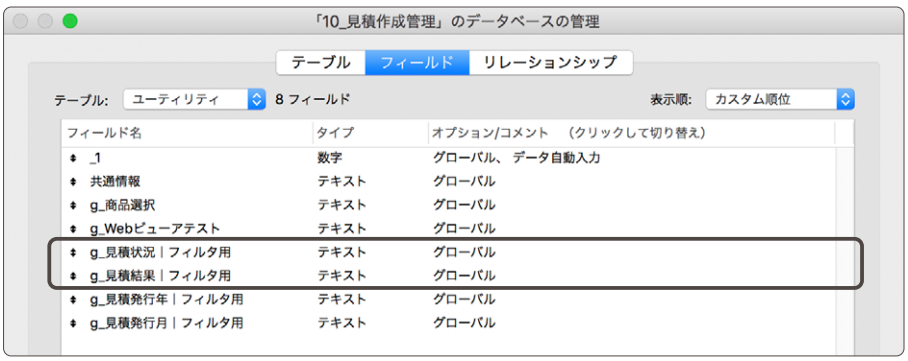


図 10-42「顧客管理｜マスタ詳細」レイアウトなど

10.8 関数 その 3

別のタイプの関数を確認します。

10.8.1 日付と時刻に関する関数

日付、時刻、日付と時刻のタイプスタンプは検索や、スクリプトの条件分岐によく使われます。

・「今日」や「2020/11/25」のようなピンポイントの日付
・「任意の年」「任意の年月」の日付

などです。年、月、日は、それぞれ別のフィールドを作成するのではなく、日付や日付タイプフィールドから取り出すこともできます。

```
Year( 日付 )    →  指定した日付の 4 桁の年
Month( 日付 )   →  指定した日付の月（1 から 12 の数字）
Day( 日付 )     →  指定した日付の日（1 から 31 の数字）
```

日付の情報についても関数でわかります。

```
WeekOfYear( 日付 )      →  指定した日付の年の 1/1 からの経過週数
YearName( 日付 ; 形式 ) →  指定した日付の和暦年（指定した形式）
                           （「令和」では　0 は「令和」、1 は「(令)」、2 は「R」と表示
                             なお「1」年は「元」年表示）

MonthName( 日付 )       →  指定した日付の月（8 月などの名前）

DayOfYear( 日付 )       →  指定した日付の年の 1/1 からの経過日数
DayOfWeek( 日付 )       →  指定した日付の曜日（1 から 7 の数字）
                           （1 は「日曜日」から 7 は「土曜日」）

DayName( 日付 )         →  指定した日付の曜日（「日曜日」などの名前）
```

任意の数字を指定することで日付を作成します。この関数はよく使用します。覚えておきましょう。

```
Date( 月 ; 日 ; 年 )    →  指定した月、日、年のカレンダー上の日付
```

「カレンダー上の日付」とはどのような意味でしょうか。これは、次のような指定をしても結果はエラーになりません。

```
Date( 2 ; 31 ; 2020 ) →  2020/03/02  が返ります
Date( 15 ; 1 ; 2018 ) →  2019/03/01  が返ります
```

存在しない日付は、カレンダーにある日付として結果を返します。また、Date と他の日付関数、取得関数の組み合わせも便利です。

```
Date(Month(Get( 日付 )) ; 1 ; Year(Get( 日付 )))    →  今月1日
Date(Month(Get( 日付 )) ; 1 ; Year(Get( 日付 )))-1 →  先月末日
```

「今月1日」や「先月末日」が簡単にわかります。先月末日の式では「-1」をしています。このように日付は足し算、引き算が可能です。

和暦や曜日をレイアウト上で表示されることが目的であれば、インスペクタの「データ」タブの［データの書式設定］で見た目を設定することもできます。

日付、時刻と数字

FileMaker Pro は内部的には日付や時刻を数値として扱っています。日付は 1 日単位、時刻、タイムスタンプは秒単位で計算ができます。覚えておきましょう。

日付や時刻、タイムスタンプを計算式で使用したとき、結果タイプの指定によっては次のような数値として表示されますが、正しいタイプを指定すれば正しく表示されるので大丈夫です。

・日付は、0001 年 1 月 1 日を「1」として経過した日数分カウントした数値
・時刻は、午前 0：00：00 を「1」として経過した秒数をカウントした数値
・タイムスタンプは、0001 年 1 月 1 日の午前 0：00：00 を「1」としてカウントした数値

時刻も日付のような関数が用意されています。

```
Hour( 時刻 )           →  指定した時刻の時（0から23）
Minute( 時刻 )         →  指定した時刻の分（0から59）
Seconds( 時刻 )        →  指定した時刻の秒（0から59）
Time ( 時 ; 分 ; 秒 ) →  指定した時、分、秒の値から計算した時刻
```

```
Time(Hour(Get( 時刻 )) ; 1 ; 1 )          →  現在時の1分1秒
Time(Hour(Get( 時刻 )) ; 1 ; 1)＋3600     →  現在時1時間後の1分1秒
```

日付関数と取得関数を組み合わせて使用できます。1時間は3600秒で取り扱います。

タイムスタンプには以下の関数があります。取得関数もあります。

```
Timestamp ( 日付 ; 時刻 )  →  指定した日付と時刻のタイムスタンプ
Get ( タイムスタンプ )          →  現在のタイムスタンプ
Get ( ホストのタイムスタンプ ) →  現在のホストのタイムスタンプ
```

タイムスタンプから日付や時刻だけを取り出すことができます。以下のテキスト関数を使用します。

```
GetAsDate ( テキスト )  →  指定したテキストの日付
GetAsTime ( テキスト )  →  指定したテキストの時刻
GetAsTimestamp ( テキスト )    →  指定したテキストのタイムスタンプ
```

次のように使用できます。

```
GetAsDate ( Get( タイムスタンプ ) )  →  今日の日付
```

何の日付や時刻なのか
取得関数が返す日付や時刻、タイムスタンプは、クライアントまたはサーバー（Get (ホストのタイムスタンプ) のみ）のパソコンの日付や時刻の設定に基づいています。

さらに時刻としては以下の取得関数があります。1 秒以下が計れます。

```
Get ( 現在の時刻UTC ミリ秒 )
  →  現在の協定世界時刻 (UTC) にミリ秒単位で適合させた近似値
```

ミリ秒単位の結果、例えば「63601728323332」を返します。スクリプトで処理時間を計るときなどに便利です。

日付の関数を使ったポータルフィルタの演習をします。演習ファイルにフィールドと値一覧を作成しておきました。これらを使います。

フィルタするデータを入力するグローバルタイプフィールド
「ユーティリティ」テーブルの 2 つのフィールドです。
・「g_ 見積発行年｜フィルタ用」フィールド
・「g_ 見積発行月｜フィルタ用」フィールド

値一覧
・「年」値一覧（カスタム値　2017¶2018¶2019¶2020¶2021）
・「月」値一覧（カスタム値　1¶2¶3¶4¶5¶6¶7¶8¶9¶10¶11¶12）

「演習 10_18」フォルダ

演習 10-18 ポータルフィルタで計算式を使う：日付の関数

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「見積詳細」レイアウトでレイアウトモードに切り替えて「見積履歴」タブをクリックします。

(2)　「g_ 見積状況｜フィルタ用」フィールドの幅を狭め、「×」ボタンの位置を移動しておきました。

(3)　「g_ 見積状況｜フィルタ用」フィールドの右に、ポータルフィルタ用のフィールドをポータルに配置します（次図を参照）。
レイアウトエリアの外に一旦置いて、横幅を「120」pt にしてからタブコントロール上に移動すると楽です。

左パネルの「フィールド」タブから配置します。
・現在のテーブルは　「z_ ユーティリティ」
・フィールドは　「g_ 見積発行年｜フィルタ用」と「g_ 見積発行月｜フィルタ用」
・フィールド配置は　「横」
・フィールドラベルは　「なし」
・コントロールスタイルは　「編集ボックス」

(4)　配置したフィールドをクリックしてインスペクタの「データ」タブの［コントロールスタイル］で値一覧を割り当てます。

「g_ 見発行年｜フィルタ用」
・ポップアップメニュー
・値一覧は　「年」
・［一覧の表示切り替え用矢印を表示］を　**オン**

「g_ 見発行月｜フィルタ用」
・ポップアップメニュー
・値一覧は　「月」
・［一覧の表示切り替え用矢印を表示］を　**オン**

(5)　配置したフィールドにスクリプトトリガを割り当てて、トリガで実行するスクリプトを指定します。
a.　「g_ 見積発行年｜フィルタ用」フィールドをクリックします。
b.　右クリックでショートカットメニューから［スクリプトトリガ設定 ...］を選択します。
c.　［スクリプトトリガ設定］ダイアログが表示されます。
d.　「OnObjectSave」を　**オン**　にします。
e.　［スクリプト指定］ダイアログが表示されます。
f.　「見積履歴ポータル更新」を選択します。
g.　［OK］を何度かクリックしてダイアログを閉じます。

h.　同様の手順で、「g_ 見積発行月｜フィルタ用」フィールドにも「OnObjectSave」トリガで「見積履歴ポータル更新」スクリプトを割り当てます。



図 10-43 グローバル格納フィールドを配置、値一覧を割り当て

(6)　配置したフィールドにインスペクタでプレースホルダテキストを付加します。
（「データ」タブの［フィールド］の［プレースホルダテキスト（フィールドが空の場合）］で設定します）

・「g_ 見積状況｜フィルタ用」フィールド　→「状況でフィルタ」
・「g_ 見積発行年｜フィルタ用」フィールド →「年でフィルタ」
・「g_ 見積発行月｜フィルタ用」フィールド →「月でフィルタ」

(7)　2 つのポータルにフィルタを設定します。

この計算式は長いので入力に時間がかかるかもしれません。
演習時に時間が足りない場合は、「s_ 見積額」フィールドが配置されているポータルに計算式が入力されていますので、コピーと貼り付け（ペースト）をして利用してください。

a. 「見積履歴ポータル」をダブルクリックして、
b. ［ポータル設定］ダイアログが表示されます。
c. 「ポータルレコードのフィルタ」の右の［指定 ...］をクリックします。
d. 計算式を入力します（次図も参照してください）。

```
If (
  IsEmpty ( z_ユーティリティ::g_見積状況｜フィルタ用 ) ; 1 ;
   z_ユーティリティ::g_見積状況｜フィルタ用  =  見積管理｜履歴｜自己::
見積状況 )

and

If (
  IsEmpty ( z_ユーティリティ::g_見積発行年｜フィルタ用 ) ; 1 ;
  z_ユーティリティ::g_見積発行年｜フィルタ用 = Year ( 見積管理｜履
歴｜自己::発行日 ) )

and

If (
  IsEmpty ( z_ユーティリティ::g_見積発行月｜フィルタ用 ) ; 1 ;
  z_ユーティリティ::g_見積発行月｜フィルタ用  = Month ( 見積管理｜
履歴｜自己::発行日 ) )
```

e. 集計フィールドがあるポータルにも同じ計算式を設定します。
計算式は　/*　*/　でコメントアウトされているので、上記の計算式だけが有効になるように作業してください（次図も参照してください）。

(8)　レイアウトを保存してブラウズモードに戻ります。

(9)　1 レコード目に移動して「見積履歴」タブをクリックします。

(10)　ポップアップメニューを選択すると、ポータルがフィルタされて 2 つのポータルの表示内容が変化することを確認してください。

演習ここまで

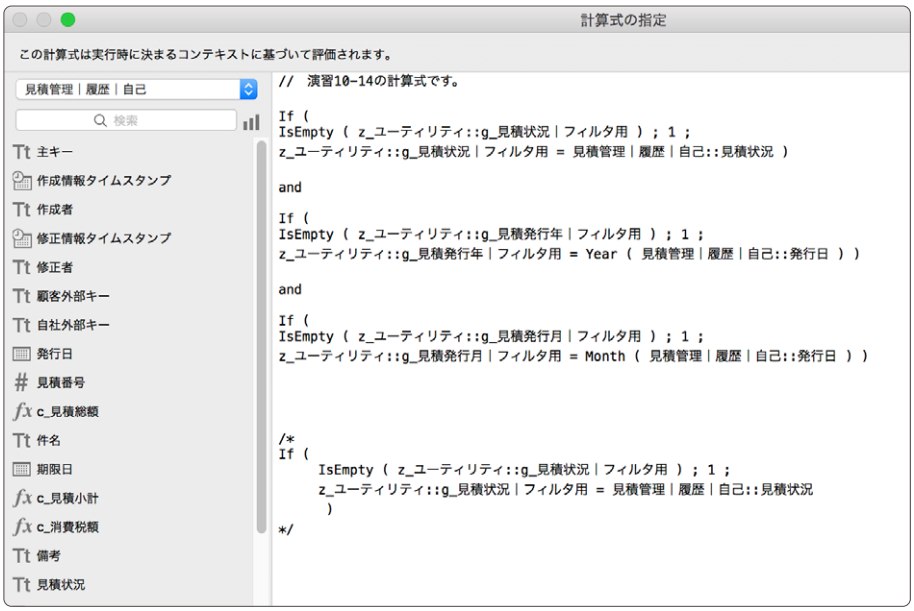


図 10-44 ポータルフィルタの計算式（前の計算式を無効にした例）

ポータルフィルタに設定した計算式では、次のような関数が使われています。これらは、「発行日」フィールドから「年」だけ、または「月」だけを取り出しています。

```
Year ( 見積管理｜履歴｜自己::発行日 )
Month ( 見積管理｜履歴｜自己::発行日 )
```

「発行年」「発行月」などのフィールドが作成されていなくても「年」だけ、または「月」だけなどを利用することができます。

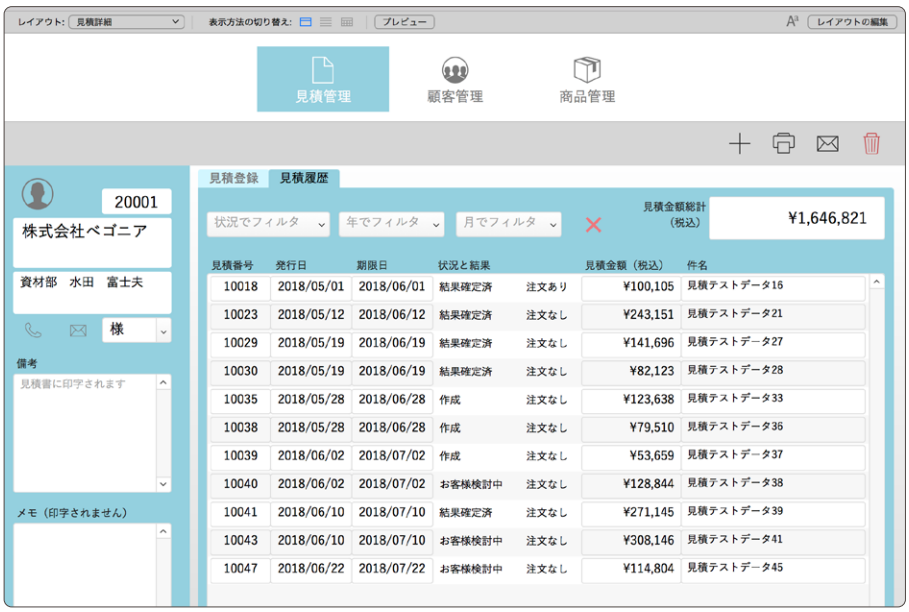


図 10-45 ポータルフィルタの作業後

10.8.2 論理関数 1

これまで、Self、IsEmpty、Let、If などの論理関数を使用してきました。その時に論理式や論理値が登場しました。論理関数は、指定した条件を評価し、その論理値（真か偽か）に従って結果を出す関数です。これは条件の分岐のために使用可能です。

If 関数

```
If ( 条件 ; 結果 1 { ; 結果 2} )
  → 条件が1（真 = True ）なら結果1、0（偽 = False）なら結果2
```

If 関数は、2 つの結果を指定できます。2 つ以上の結果が必要なら Case 関数を使用します。

Case 関数

```
Case ( 条件1 ; 結果1 {;条件2 ; 結果2 ; ... ; デフォルト値} )
  → それぞれの条件を順に評価し1（真 = True ）の条件が見つかったらその条件の結果
    （どの条件も真でなければデフォルト値）
```

例えば、「消費税額」フィールドの端数処理の方法を、「端数処理」フィールドに 1, 2, 3 のいずれかの数値で指定する場合は、次のようにします（「端数処理」フィールドに 1、2、3 以外の数値が入力されているか、空欄だと端数処理されません）。

```
Case (
 端数処理 = 1 ; Round( 消費税額 ; 0 ) ;        //四捨五入
 端数処理 = 2 ; Truncate( 消費税額 ; 0 ) ;     //切り捨て
 端数処理 = 3 ; Ceiling( 消費税額 ) ;          //切り上げ
 消費税額 )
```

Case 関数では記述された順番に条件を評価し、条件が真になったら結果を返します。計算式の結果がでるまで時間がかかるようなら、結果がでるまで時間がかからない条件が先に評価されるように先に記述します。

また、数値や日付の範囲が条件だったら、条件を大きい順あるいは小さい順に並べることで単純に記述できます。例えば「金額」フィールド（数字）に入力される数値で、割引率を結果として出す計算式は、次のように記述します。

```
Case (
 金額 < 1000 ; "" ; // 割引なし
 金額 < 2000 ; ".01" ; // 1%割引
 金額 < 3000 ; ".02" ; // 2%割引
 ".05")//5%割引
```

一番上の条件が偽であれば、次の条件は「2000 未満」かどうかだけで、「1000 以上」という条件指定は必要ありません。

Case 関数に似た構文では Choose 関数があります。

Choose 関数

```
Choose ( 条件 ; 結果0 { ; 結果1 ; 結果2...} )
  →  条件の整数値に従った1つの結果
```

Choose関数ではCase関数のように複数の条件を指定しません。この関数を使って「評価」フィールドに文字列を入力するためには、以下のように計算式を作成します。「評価レベル」フィールドに0/1/2/3 のいずれかの数値で指定する形です。

```
Choose (
 評価レベル  ; //条件
 "評価なし" ;                    //結果0または空欄
 "不可" ;                        //結果1
 "可" ;                          //結果2
 "良" ;                          //結果3
  )
```

「評価レベル」フィールドが0、空欄の場合は「評価なし」になります。また1/2/3以外のデータでは結果は返りません。

論理値を返す関数

次のような関数が用意されています。

```
IsEmpty ( フィールド )
  →  数字が返ります。
     指定したフィールドが空欄か他のエラーが発生したら「1（真 = True )」
     それ以外の場合は「0（偽 = False)」

IsValid ( フィールド )
  →  数字が返ります。
     指定したフィールドが無効なら「0（偽 = False)」
     それ以外の場合は「1（真 = True )」

GetAsBoolean ( データ )
  →  数字が返ります。
     指定したデータがゼロ以外の数値なら
     「1（真 = True )」それ以外は「0（偽 = False)」
     指定したデータがオブジェクトタイプフィールドでデータが含まれている場合は
     「1（真 = True )」それ以外は「0（偽 = False)」
```

not 演算子

これらに not 演算子をつけると結果を逆転させることができます。例えば、「件名」フィールドが空欄でないかどうかを確認するには、次の計算式となります。

```
not IsEmpty ( 件名 )
  →  空欄でなかったら「1」 空欄だったら「0」 が返ります。
```

IsEmpty 関数や IsValid 関数は、フィールドのデータを評価するだけではなく、関連レコードの有無や関連、関連ファイルがあるかどうかを判断することに使用できます。

例えば「顧客管理」TO と関連している「顧客管理 _ 見積」TO があり、関連レコードがあるかどうかを確認するには次のように判断できます。

```
IsEmpty ( 顧客管理_見積::顧客外部キー )
    →  関連レコードがなければ「1」、あれば「0」が返ります。

IsValid ( 顧客管理_見積::顧客外部キー )
    →  関連レコードがなければ「0」、あれば「1」が返ります。
```

あるいは「顧客管理」TO の取引状態にデータが入力されているかどうかを確認するには

```
GetAsBoolean ( 顧客管理::取引状態 )
    →  「1」が入力されていれば「1」、空欄あるいは「0」が入力されていれば「0」
```

真偽値（フラグ）
レコードの状態を示すものとして数字タイプフィールドを 0/1 あるいは空欄 /1 のどちらかが入力されるようにフィールドオプションで設定したりレイアウト上でコントロールすることで「1」の場合に関連が成立する照合フィールドとしたり、検索条件として扱うことが可能です。

そのような項目を真偽値、フラグなどと呼びます。レイアウト上に配置したフィールドをボタンオブジェクトにしてクリックされたときに GetASBoolean 関数の結果が「1」のときに「0 または空欄」、「0」のときに「1」を入力するスクリプトを割り当てることで、データのコントロールをおこなうことが可能です。空欄が「1」の判断であれば、以下の計算式と同じ結果です。

```
If (IsEmpty ( 顧客管理::取引状態 ) ;  1 ; 0 )
```

しかし、GetAsBoolean 関数なら「なにか入力されているか／ 0 または空欄か」を 1 つの関数の結果で取得できるので便利です。なお、レイアウト上で文言を表示するなら、インスペクタの「データ」タブの「データの書式設定」の「論理値」で文言表示を設定可能です。

Self 関数

これまでにも使用した Self 関数は、独特のものです。フィールドオプション、条件付き書式、ポップアップヘルプ、プレースホルダテキストで使用します。Self 関数を使用した計算式は、特定のフィールド名を持たないためコピーと貼り付け（ペースト）で使用できます。

Evaluate 関数

指定した式を計算式として評価します。

```
Evaluate ( 式 { ; [ フィールド1 ; フィールド2 ; フィールド3 ; ...] } )
  →「式 ; 」のあとの「フィールド」が変更されると計算式が再評価されて結果が更新されます。
```

評価の違いを簡単に比較すると、

```
Evaluate ( 2+2 )は「4」
Evaluate ( Quote ( "2+2" ))は「2+2」
```

あるいは、例えば「足し算」フィールドがあるとします。テキストタイプフィールドです。「2 + 2」の文字列が入力されています。

```
Evaluate ( 足し算 ) は「4」
Evaluate ( Quote (足し算)) は「2+2」
Evaluate ( "足し算" ) は「2+2」
Evaluate ( GetField ( "足し算" ) ) は「4」
```

「1*かける数」が入力された「かけ算」フィールド（テキストタイプ）と「2」が入力された「かける数」フィールドがあるとします。

```
Evaluate ( かけ算 ; [かける数] ) は「2」
  → 「かける数」フィールドに 3 を入力すると「3」を返します。

Evaluate ( Get(タイムスタンプ) ; [名前] )
  → 「名前」フィールドが変更されると、タイムスタンプを返します。
```

サーバーサイドで実行するスクリプトの注意

Evaluate 関数をサーバーサイドで実行するスクリプトで利用する場合には、評価する関数名は英語でなければいけません。スクリプトの作成時や後述するカスタム関数の作成時に注意してください。

Let 関数

変数を取り扱う関数です。次のページも参照してください。

```
Let ( {[] 変数1 = 式1 { ; 変数2 = 式2...]} ; 計算 )
```

この式は、以下のような形で記述します。変数はいくつでも指定できます。

```
Let (
     変数を指定する ;
     指定した変数を使用した計算式
 )
```

10.8.3 その他の関数

次のような関数が用意されています。

フィールド名を指定する関数

スクリプトなどで完全修飾名のフィールドを指定する場合に使用します。TO（完全修飾名）付きのテキストが結果として返るので、計算式やスクリプトのコンテキストに合致するように指定できます。

```
GetFieldName ( フィールド名 ) → 指定したフィールドの完全修飾名
```

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 10_19 から 22」フォルダの中の「10_見積作成管理 .fmp12」ファイルをベースに進めます）

10.9 変数

変数は、ある宣言をおこなって、宣言された変数名を使用した計算式が記述されたとき、その式の評価時に宣言された内容で変数名に代入するためのものです。

この計算式では、3 つの関数が入れ子になります。

```
Filter ( RomanHankaku (
Substitute ( Self ; ["－" ; "-"];["-" ; "-"] ) ; "-0123456789"))
```

しかし、Let 関数を使うと、シンプルです。

```
Let ( [
  x = Substitute ( Self ; [ "－" ; "-" ];[ "-" ; "-" ] ) ;
  y = RomanHankaku ( x ) ] ;
  Filter ( y ; "-0123456789" )
)
```

`Substitute`関数で文字列を置換し、さらに`RomanHankaku`関数で英数記号を半角に変換した文字列を`Filter`関数を使って指定した文字列だけにしています。

また、このような Date 関数の式では、

```
Date(Month( Get( 日付 ) ) ; Day( Get( 日付 ) ) ; Year( Get( 日付 ) ))
```

Let 関数を使えば、同じ関数（Get(日付)）を 3 回実行することなく、1 回で済ませることができるので、パフォーマンス的にも効率的な計算式となります。

```
Let (
  x = Get( 日付 ) ;
  Date ( Month( x ) ; Day( x ) ; Year( x ) )
)
```

3 種類の変数

Let 関数では、3 種類の変数を指定できます。

- **変数**　→　任意の文字列を変数名とし、Let 関数の計算式の中でのみ使用するもの
- **$ 変数**　→　**ローカル変数**　宣言があったスクリプトが終了するまで有効なもの
- **$$ 変数**　→　**グローバル変数**　宣言があったファイル内で有効なもの

スクリプトと変数を使った演習を行います。オブジェクトフィールドのバイナリファイルを、デスクトップへ書き出すスクリプトを作成します（変数は、すべて「ローカル変数」です）。

演習 10-19 スクリプトの作成（ファイルパスに変数を使う）：書き出し

「演習 10_19 から 22」フォルダ

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　「商品管理｜マスタ詳細」レイアウトに切り替えます。

(2)　オブジェクトフィールドのフィールド内容のエクスポートを手作業で試します。
a.　1 レコード目に移動して、画像（「商品画像」フィールド）を右クリックします。
b.　「フィールド内容のエクスポート ...」を選択します。
c.　ダイアログが表示されますので、デスクトップに保存します。



図 10-46 保存ダイアログ

(3)　今おこなった作業を**「画像をデスクトップに書出」スクリプト**として作成します。

(4)　スクリプトと、それを割り当てるボタンバーを作成しておきました。



図 10-47 ボタンバーとフィールド

(5)　書き出しのためのスクリプトを作成します。
a.　レイアウトモードに切り替えます。
b.　新しく配置したボタンバーの左側のセグメントをダブルクリックします。

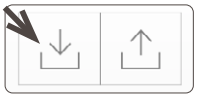


図 10-48 左側のセグメント

c. ［ボタンバー設定］ダイアログが表示されます。
d. 「画像をデスクトップに書出」スクリプトが割り当てられていますので、スクリプト名をクリックします。
e. ［スクリプトを編集］ダイアログが表示されます。
スクリプトは、ほとんど作成されていません。目次のようなものだけを書いておきました。

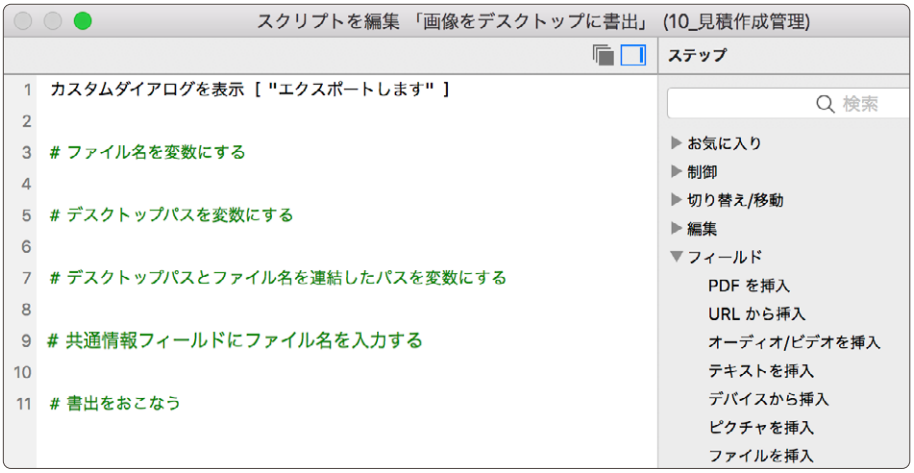


図 10-49「画像をデスクトップに書出」スクリプト

オートコンプリートを使った手順を記述します。スクリプトパネルを使った操作でも構いません。

(6) 4 行目をクリックします。
a. 「へんすう」や「変数」と入力して、［変数を設定］スクリプトステップを選択します。

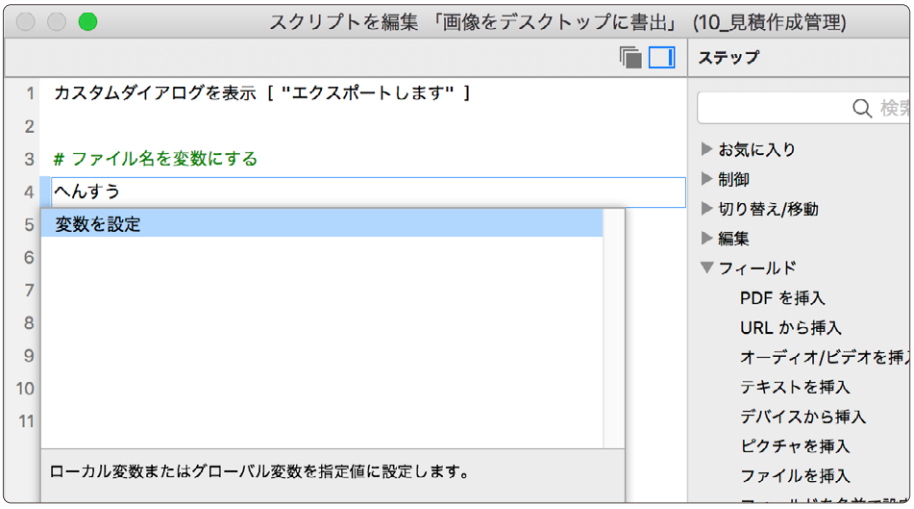


図 10-50 スクリプトステップを選択

b. 歯車アイコンをクリックします。
c. 「[変数を設定] のオプション」ダイアログが表示されます。
・[名前：] は 「$ バイナリファイル名」($ は、半角で入力してください)
・[値：] は [指定 ...] をクリックして、計算式を入力

```
GetContainerAttribute ( 商品管理::商品画像 ; "filename")
```

d. [OK] をクリックします。

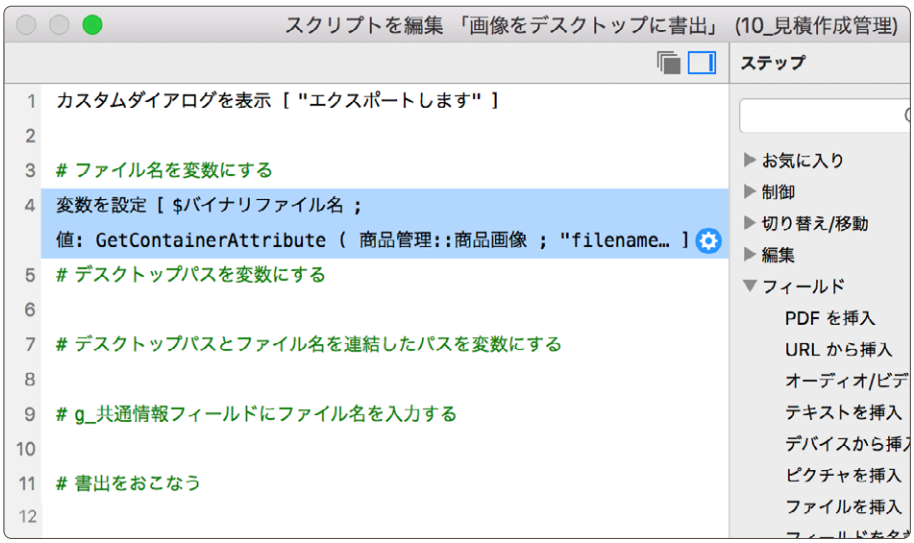


図 10-51 4 行目を設定後

(7) 6 行目をクリックします。
a. 同様に [変数を設定] スクリプトステップを入力します。
b. 歯車アイコンをクリックします。
c. 「[変数を設定] のオプション」ダイアログが表示されます。
・[名前：] は 「$ デスクトップパス」($ は半角、後ろにスペース無し)
・[値：] は [指定 ...] をクリックして、計算式を入力

```
Case (
   Get ( システムプラットフォーム ) = 1 ; "filemac:" ;
   Get ( システムプラットフォーム ) = -2 ; "filewin:" ;
   "" )
&
Get ( デスクトップパス )
```

d. [OK] をクリックします。

(8) 8 行目をクリックします。
a. [変数を設定] スクリプトステップを入力します。
b. 歯車アイコンをクリックします。
c. 「[変数を設定] のオプション」ダイアログが表示されます。
・[名前：] は、「$ パスとファイル名」($ は半角、後ろにスペース無し)

・[値：] は、[指定 ...] をクリックして、計算式を入力

$ デスクトップパス & $ バイナリファイル名

d. [OK] をクリックします。

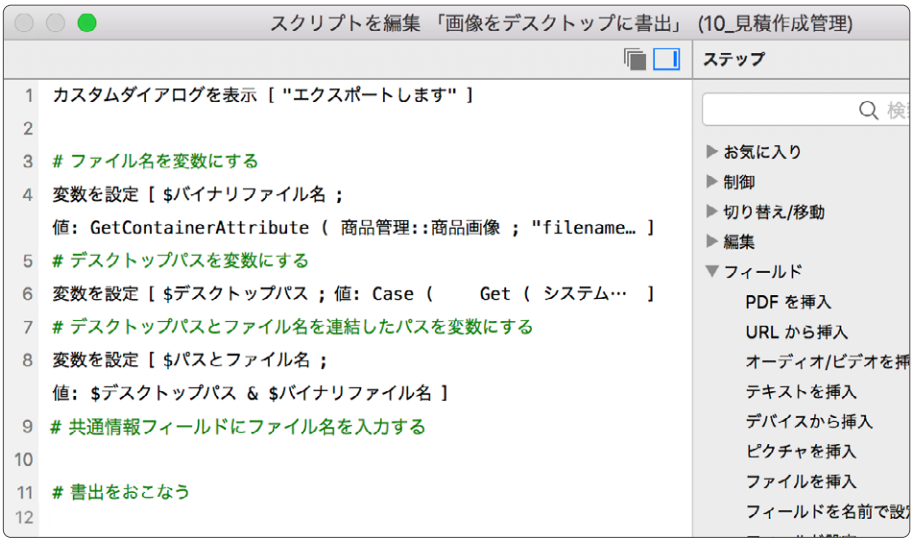


図 10-52 6、8 行目を設定後

(9) 10 行目をクリックします。

a. [フィールドを設定] スクリプトステップを入力します。

b. 歯車アイコンをクリックします。

c. オプション」ダイアログが表示されます。

・[ターゲットフィールドの指定] は 「z_ ユーティリティ :: 共通情報」フィールド

・[計算結果：] は

$ バイナリファイル名

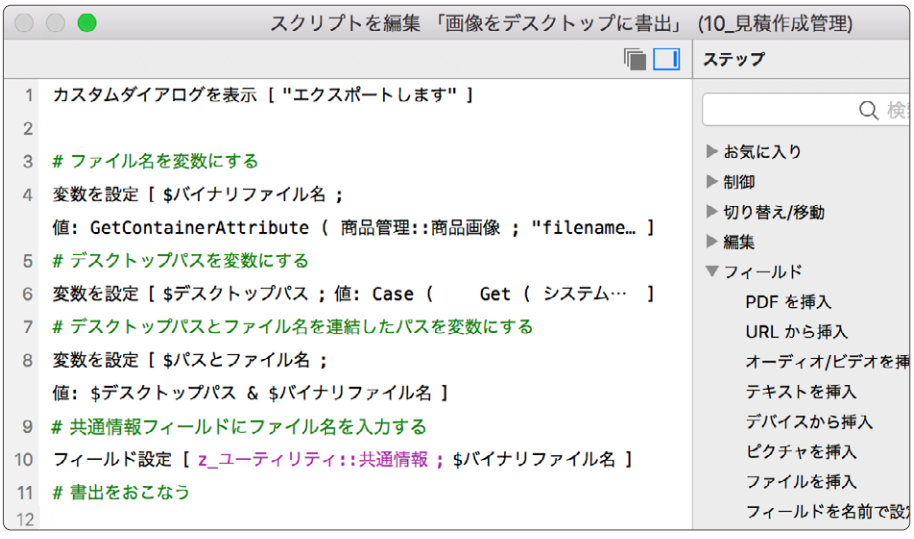


図 10-53 10 行目を設定後

（10） 12 行目をクリックします。
a. ［フィールド内容のエクスポート］スクリプトステップを入力します。
b. 歯車アイコンをクリックします。
c. 「［フィールド内容のエクスポート］のオプション」ダイアログが表示されます。
・［ターゲットフィールドの指定］は　「商品管理 :: 商品画像」フィールド
・［出力ファイルの指定］は　$ パスとファイル名
・［ディレクトリの作成］は　**オフ**

（11） ［OK］をクリックしてダイアログを閉じてスクリプトを保存します。
できあがったスクリプトの内容を記述します。

```
カスタムダイアログを表示 [ タイトル: "エクスポートします"; デフォルトボタン:「OK」, 確定:「いいえ」 ]

#ファイル名を変数にする
変数を設定 [ $バイナリファイル名 ;
値:GetContainerAttribute ( 商品管理::商品画像 ; "filename" )]

#デスクトップパスを変数にする
変数を設定 [ $デスクトップパス ;
値:Case ( Get ( システムプラットフォーム ) = 1 ; "filemac:" ; Get ( システムプラットフォーム ) = -2 ; "filewin:" ; "" ) & Get ( デスクトップパス ) ]

#デスクトップパスとファイル名を連結したパスを変数にする
変数を設定 [ $パスとファイル名 ;
値:$デスクトップパス & $バイナリファイル名 ]

#共通情報フィールドにファイル名を入力する
フィールド設定 [ z_ユーティリティ::共通情報 ;  $バイナリファイル名 ]

#書出をおこなう
フィールド内容のエクスポート [ 商品管理::商品画像 ;「$パスとファイル名」 ]
```

（12） レイアウトを保存してブラウズモードに戻ります。

（13） 左のポータルの商品番号「30001」を選択して、右側にデータを表示します。

（14） 新しく配置したボタンバーの左側のセグメントをクリックします。ダイアログは［OK］をクリックします。

（15） デスクトップに画像が書き出され「共通情報」フィールドに「v17m-note01.png」と入力されます。

演習ここまで

オブジェクトフィールドへデスクトップからバイナリファイルを取り込むスクリプトを作成します。デスクトップにあるファイル名をあらかじめ指定して取り込む方法で記述します。

演習 10-20 スクリプトの作成（ファイルパスに変数を使う）：取り込み

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) オブジェクトフィールドへのファイルの挿入を手作業で試します。

a. 9 レコード目に移動して画像（「商品画像」フィールド）を右クリックします。

b. 「ピクチャを挿入 ...」を選択します。

c. ダイアログが表示されますので、デスクトップから演習 10-15 で保存した「v17m-note01.png」を選択してインポートします。



図 10-54 選択ダイアログ

(2) 今実施した作業を**「画像をデスクトップから取込」スクリプト**として作成します。

(3) 「商品管理｜マスタ詳細」レイアウトのボタンバーの右側のセグメントに割り当てられている「画像をデスクトップから取込」スクリプトを編集します。一部の手順は省略します。

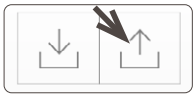


図 10-55 左側のセグメント

(4) 同様に、目次のようなものだけを書いておきました。

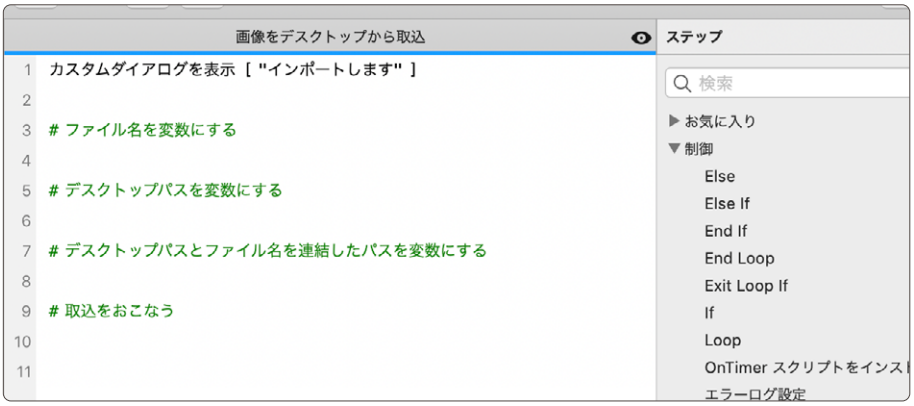


図 10-56「画像をデスクトップから取込」スクリプト

(5)　4 行目は、[変数を設定] スクリプトステップです。
オプションは、

・[名前：] は　「$ バイナリファイル名」
・[値：] は　　[指定 ...] をクリックして、計算式を入力

```
z_ユーティリティ::共通情報
```

(6)　6 行目は、[変数を設定] スクリプトステップです。
(「画像をデスクトップに書出」スクリプトと同じなのでコピーと貼り付け（ペースト）しても構いません)
オプションは、

・[名前：] は　「$ デスクトップパス」
・[値：] は　　[指定 ...] をクリックして、計算式を入力

```
Case (
   Get ( システムプラットフォーム ) = 1 ; "filemac:" ;
   Get ( システムプラットフォーム ) = -2 ; "filewin:" ;
   "" )
&
Get ( デスクトップパス )
```

(7)　8 行目は、[変数を設定] スクリプトステップです。
(「画像をデスクトップに書出」スクリプトと同じなのでコピーと貼り付けしても構いません)
オプションは、

・[名前：] は　「$ パスとファイル名」
・[値：] は　　[指定 ...] をクリックして、計算式を入力

```
$デスクトップパス & $バイナリファイル名
```

(8)　10 行目をクリックします。
a.　[ファイルを挿入] スクリプトステップを入力します。
b.　歯車アイコンをクリックします。
c.　オプションダイアログが表示されます。
d.　ダイアログオプションの [指定 ...] をクリックします。
・[カスタムダイアログのタイトル] は　そのまま
・[格納オプション：] は　「挿入」
・[表示：] は　　「ファイルの内容（表示可能な場合）」
・[圧縮：] は　　「圧縮（可能な場合）」
e.　[OK] をクリックします。

f.　ターゲット [指定 ...] をクリックします。
・[フィールド] ラジオボタン　**オン**
・「商品画像」フィールドをクリック
g.　[OK] をクリックします。

h. ソースファイルの指定［指定 ...］をクリックします。
・$ パスとファイル名
i. ［OK］をクリックします。

(9) ［OK］をクリックしてダイアログを閉じてスクリプトを保存します。
できあがったスクリプトの内容を次に記述します。

```
カスタムダイアログを表示 [ タイトル: "インポートします"; デフォルト
ボタン:「OK」, 確定:「いいえ」 ]

#ファイル名を変数にする
変数を設定 [ $バイナリファイル名 ; 値:z_ユーティリティ::共通情報 ]

#デスクトップパスを変数にする
変数を設定 [ $デスクトップパス ;
値:Case (Get ( システムプラットフォーム ) = 1 ; "filemac:"
; Get ( システムプラットフォーム ) = -2 ; "filewin:" ;
"" ) & Get ( デスクトップパス ) ]

#デスクトップパスとファイル名を連結したパスを変数にする
変数を設定 [ $パスとファイル名 ; 値:$デスクトップパス & $バイナリ
ファイル名 ]

#画像を取り込む
ファイルを挿入 [ 挿入 ; 内容を表示 ; 可能な場合圧縮 ; 商品管理::
商品画像 ; 「$パスとファイル名」 ]
```

(10) レイアウトを保存してブラウズモードに戻ります。

(11) 左のポータルの商品番号「30010」を選択して、右側にデータを表示します。
(画像が入力されていません)

(12) ボタンバーの右側のボタンをクリックします。ダイアログは［OK］をクリックします。

(13) 「v17m-note01.png」の画像が挿入されます。

演習ここまで

これらの演習では、ユーザがデスクトップを選択することなく、書き出しや取り込みがおこなわれました。Get 関数で、ユーザのパソコン OS によってデスクトップパスが異なる場合でも対応可能です。

「画像をデスクトップへ書出」と「画像をデスクトップから取込」スクリプトでは、6 行目が同じです。デスクトップパスの部分です。ユーザのパソコンの OS が、カスタム App を使っている間に変わってしまうことはほとんどありません。ユーザがカスタム App を起動する度にデスクトップパスを作成して保持しておいても良いかもしれません。

グローバル変数でデスクトップパスを保持する演習を行います。まず、どのようなグローバル変数が保持されるのかを簡単に確かめられるようにレイアウトにマージ変数を配置します。

演習 10-21 スクリプトの作成（ファイルパスに変数を使う）：グローバル変数：準備

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) 「商品管理｜マスタ詳細」レイアウトで、レイアウトモードに切り替えます。

(2) 変数をレイアウトに配置します。
   a. [挿入]–[マージ変数] メニューを選択します。
   b. レイアウトに <<$$>> の記号が配置されます。
   c. 「ヘッダ」パートに配置します。
   d. <<$$ デスクトップパス >> という内容にします。（記号と文字列の間に半角スペースなどは入力しないでください）

(3) わかりやすいように文字の大きさを大きめにします。
   a. <<$$ デスクトップパス >> をクリックします。
   b. インスペクタの「スタイル」タブで「タイトルテキスト 1」を選択します。
   c. サイズを調整します。（横幅は「800」pt ぐらいに長くします）
   d. レイアウトを保存してブラウズモードに戻ります。



図 10-57 マージ変数を置いた

演習ここまで

ボタンをクリックしてグローバル変数でデスクトップパスを保持できるようにします。

演習 10-22 スクリプトの作成（ファイルパスに変数を使う）：グローバル変数

- **使用ファイル：10_ 見積作成管理** .fmp12

(1) ボタンオブジェクトを作成します。
a. ステータスツールバーの［ボタン］アイコンをクリックします。
b. 「上部ナビゲーション」パートの左側でドラッグして四角形を描きます。
c. マウスボタンを離すと［ボタン設定］ダイアログが表示されます。
d. ボタンの名前は「パスの作成」と入力
e. ［処理：］のポップアップメニューで「スクリプト実行」を選択します。
f. ［スクリプト指定］ダイアログが表示されます。
g. 「＋」ボタンをクリックします。
h. **「グローバル変数でパスを保存」**と名前を入力します。

(2) スクリプトステップとオプションを設定します。
a. ［変数を設定］スクリプトステップを入力します。
b. 歯車アイコンをクリックします。
c. 「[変数を設定］のオプション」ダイアログが表示されます。
・［名前：］は、**「$$ デスクトップパス」**
・［値：］は、［指定 ...］をクリックして、計算式を入力
（「画像をデスクトップに書出」スクリプトの「$ デスクトップパス」と同じなので計算式をコピーと貼り付けしても構いません）

```
Case (
   Get ( システムプラットフォーム ) = 1 ; "filemac:" ;
   Get ( システムプラットフォーム ) = -2 ; "filewin:" ;
   "" )
&
Get ( デスクトップパス )
```

d. ［OK］をクリックします。

e. 次の行に［ウインドウの再表示］スクリプトステップを入力します。
（オプションなどはなにも設定しません）

(3) ［OK］をクリックしてダイアログを閉じてスクリプトを保存します。

```
変数を設定 [ $$デスクトップパス;
値:Case ( Get ( システムプラットフォーム ) = 1 ; "filemac:" ;
Get ( システムプラットフォーム ) = -2 ; "filewin:" ; "" )
& Get ( デスクトップパス ) ]

ウインドウ内容の再表示
```

(4) レイアウトを保存してブラウズモードに戻ります。

(5)　新しく配置した「パスの作成」ボタンをクリックします。

(6)　グローバル変数「$$ デスクトップパス」の内容が表示されます。

(7)　演習ファイルを閉じて、また開いて「商品管理｜マスタ詳細」レイアウトで確認するとグローバル変数が消えていることを確かめてください。

演習ここまで

グローバル変数の管理

グローバル変数は、ファイルを閉じると消えます。または空欄のグローバル変数を宣言しても消えます。グローバル変数をポータルフィルタに使用している場合、ユーザが A レイアウトでポータルフィルタを使用して B レイアウトに切り替えてなにか作業をして A レイアウトに戻ってきたときに、元のままのフィルタリングのままか、フィルタリングされていない方が良いかなどを考える必要があります。

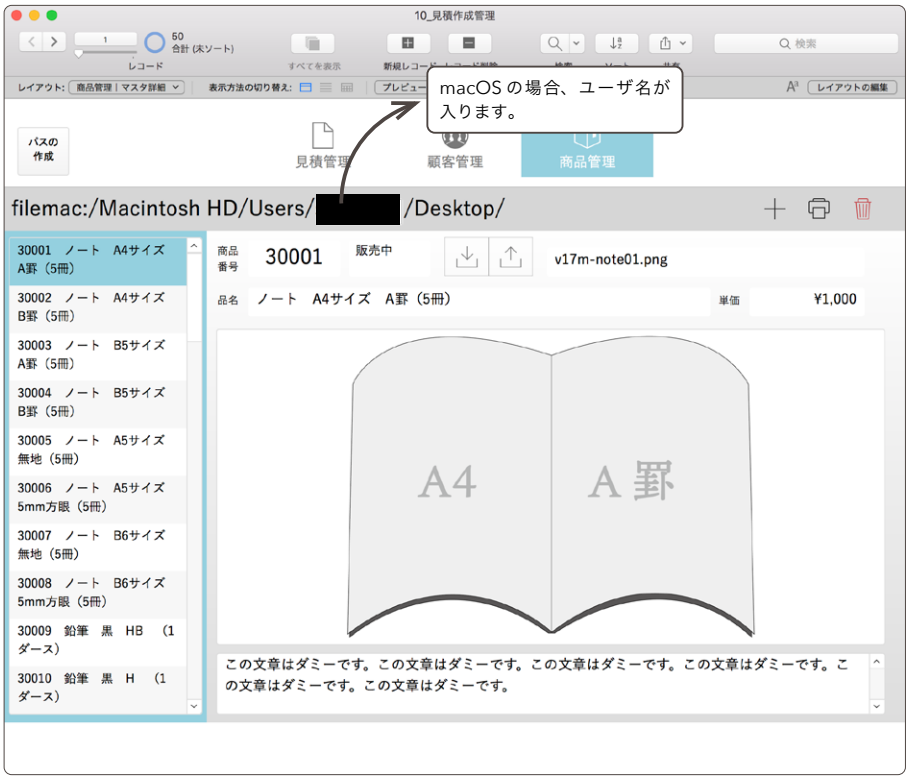


図 10-58 グローバル変数をマージ変数でレイアウトに表示

演習 10-18 のボタンで宣言するグローバル変数を使ったスクリプトを確認します。演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 10_19 から 22」フォルダの中の「10_見積作成管理 10_22 完了時 .fmp12」ファイルをベースに進めます）

ファイルを開くと、「商品管理｜マスタ詳細」レイアウトが次図のようになっています。



図 10-59「10_ 見積作成管理 10_18 完了時 .fmp12」ファイルの、「商品管理｜マスタ詳細」レイアウト

「パスの作成」ボタンの右に黒いボタンバーが配置されています。

これは、「画像をデスクトップへ書出｜グローバル変数」と「画像をデスクトップから取込｜グローバル変数」スクリプトの動作を確認するためです。演習 10-15 と 16 のスクリプトのパスにグローバル変数を使った内容のスクリプトです。

(1)「パスの作成」ボタンをクリック（パスがレイアウト上に表示される）
(2) 左のポータルの商品番号「30017」を選択して、右側にデータを表示します。
(3) 黒のボタンバーの左側のセグメントをクリックします。ダイアログは［OK］をクリックします。
(4) デスクトップに「30017」フォルダが作成されて、v17m-note08.png が書き出されます。
(5)「商品管理｜マスタ詳細」レイアウトの商品番号「30017」の画像を消去します（いま書出した画像です）。
(6) 黒のボタンバーの右側のボタンをクリックします。ダイアログは［OK］をクリックします。
(7)「v17m-note08.png」の画像が挿入されます。

［スクリプトワークスペース］からスクリプトの内容を確認してください。ローカル変数「$ デスクトップパス」は使っていません。代わりに「パスの作成」ボタンで作成されているはずのグローバル変数「$$ デスクトップパス」を利用しています。

なお、「画像をデスクトップへ書出｜グローバル変数」では、

・[フィールド内容のエクスポート] スクリプトステップの［ディレクトリの作成］ → **オン**

のオプションを使っています。書き出すパスには、次の計算式でフォルダを加えています。このオプションを利用するとデスクトップに同名のフォルダがなければ作成されます。

```
$$デスクトップパス & 商品管理::商品番号 & "/" & $バイナリファイル名
```

さらに、もう 1 つスクリプトを作成しました。「パスの作成」ボタンの右に灰色のボタンが配置されています。

「画像をデスクトップから取込｜グローバル変数｜変数に挿入」では、

・[フィールド内容のエクスポート] スクリプトステップの［ターゲット］ → 変数

のオプションを使っています。挿入した画像をグローバル変数「$$ 画像」に格納し、さらに「ユーティリティ」テーブルの「g_ 画像｜変数に挿入テスト」フィールドにフィールド設定しています。

(1)「パスの作成」ボタンをクリック（パスがレイアウト上に表示される）
(2) 左のポータルの商品番号「30018」を選択して、右側にデータを表示します。
(3) 黒のボタンバーの左側のセグメントをクリックします。ダイアログは［OK］をクリックします。
(4) デスクトップに「30018」フォルダが作成されて、v17m-note09.png が書き出されます。
(5) 灰色のボタンバーをクリックします。ダイアログは［OK］をクリックします。
(6)「v17m-note09.png」の画像が右に表示され、さらに右に「$$ 画像」に保存されている内容（画像のテキスト）が表示されます。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 10_23 から 24」フォルダの中の「10_見積作成管理 .fmp12」ファイルをベースに進めます）

10.10 関数 その 4

10.10.1 論理関数 2

特別な関数も用意されています。

ExecuteSQL 関数
この関数は SQL クエリーを実行するものです。

SQL（Structured Query Language）
データベースに対して処理を命令するための言語です。

クエリー（query）
データベースへの問い合わせのことです。つまり SQL クエリーは、SQL という言語で記述されたデータベースへの問い合わせ命令のことです。

ExecuteSQL 関数は、以下のような構文の関数です。

```
ExecuteSQL ( SQLクエリー ; フィールド区切り ; 行区切り { ; 引数...} )
    フィールド区切りは結果のフィールドの区切り
    行区切りは、結果が複数レコードの時の区切り文字
    引数はSQLクエリー内に「?」を記述してその代入のデータを設定するダイナミック引数
```

「SQL クエリー」の部分に ExecuteSQL 関数を使用するポイントがあります。ここには、FileMaker Pro の関数とは別の命令語、構文、文法が定められています。この定めに従って記述される命令文のことを「ステートメント」と呼びます。SQL クエリーにはいくつかの種類がありますが ExecuteSQL 関数で使用できるものは限定されています。「SELECT」のみが実行可能です。

SELECT
データベースのテーブルのレコードを抽出するものです。以下のような構文である必要があります。

```
SELECT
[ 抽出するフィールドの指定 ]
[ FROM 抽出するフィールドのテーブルのTO名 ]
[ WHERE 条件式 ]
（構文の中の、FROMやWHEREは、FROM句、WHERE句と呼ばれます）
```

ExecuteSQL 関数とリレーションシップグラフ
FileMaker Pro では直接テーブルに対してクエリーを実行するのではなく TO を使用してテーブルに命令を出します。しかし、この関数で実行する SQL クエリーは指定した TO のコンテキストには縛られません。リレーションシップグラフで関連が設定されていても、関係ありません。テーブル同士の関係は、FROM 句の中で JOIN を使って結合（関連）を指定します。

また、いくつかの注意点があります。

予約語
SQL には制限や予約語が FileMaker Pro とは別に存在します。予約済みの言葉は使えません。

日付の書式
日付は「YYYY-MM-DD」で取り扱います。ExecuteSQL 関数の日付の返り値も同様の書式です。

エラー
文法が間違っているなどの理由でエラーになった場合「?」を返します。

「?」のダイナミック引数
SQL クエリーの中に「?」を指定して、ダイナミック引数を使用して代入することができます。

例えば「見積管理」TO の「発行日」フィールドが 2020 年 8 月 4 日のレコードを抽出するには、次のように記述します（SQL クエリーは文字列で記述されるため、「"」を使用するには「\"（macOS）」「¥"(Windows)」と記述します。以下は macOS の場合です）。

```
ExecuteSQL (
  "SELECT *
   FROM \"見積管理\"
   WHERE \"発行日\" = '2020-08-04' "
; "" ;"" )
```

（TO 名やフィールド名が全角文字、記号から始まる、「.」が含まれるなどの場合、「"」で囲む必要があります）
（「2020-08-04」のような文字列を記述するには「'」で囲む必要があります）

「*」
ワイルドカードです。「SELECT * 」と記述すると FROM で指定したすべての列（フィールド）を取得するという意味になります。任意のフィールドを指定するには

```
SELECT id, \"発行日\",\"期限日\"
```

のように「,」で区切って指定します。

ダイナミック引数の利用
WHERE 句で、発行日が「今日」のものを抽出したいのであれば、Get (日付) で取得した「YYYY/MM/DD」を「YYYY-MM-DD」に変更しなければならないのでしょうか。

```
ExecuteSQL (
  "SELECT *
   FROM \"見積管理\"
   WHERE \"発行日\" = '" &
   Year(Get(日付))&"-"&Month(Get(日付))&"-"&Day(Get(日付))&
"'" ; "" ; "" ; )
```

（「"」の前後には半角スペースなどを入力しないでください）
ダイナミック引数を使用したほうが簡単です。

```
ExecuteSQL (
 "SELECT *
  FROM \"見積管理\"
  WHERE \"発行日\" = ?"
; "" ;"" ; Get(日付))
```

「""」の前後には半角スペースなどを入力しないでください。

「?」は、いくつでも使用できます。引数の指定は？が記述された順番に指定します。ダイナミック引数には、フィールドを指定できます。注意点としては、このフィールドは他の関数と同じです。ダイナミック引数にフィールドを使用する場合は、ExecuteSQL 関数を実行するコンテキストに注意をする必要があります。

返ってくるデータ

ExecuteSQL を実行して返ってくるデータは、以下のようなものです。区切りの指定をしないと、フィールドは「,」、レコードは改行で区切られます。

```
22761E61-342D-4183-969D-322222E67A31,2018-06-18 17:21:09,
Admin,2018-06-28 12:17:53,Admin,2018-06-07,10001,127958,見
積テストデータ1,2018-07-07,118480,9478,,結果確定済,注文あり,50E70A0E-
9970-48BC-9333-98BD9BB7BD24,8BC80074-AB63-4CC5-B6DF-
BDEB424C3C1B,.08,,様,127958
```

レコードの順番は ORDER BY 句でソート順を指定しなければ作成順になります。

ExecuteSQL についての更に詳しい説明は、FileMaker Pro のヘルプ、製品のマニュアルや、「Claris FileMaker SQL リファレンスガイド」（https://help.claris.com/ja/sql-reference.pdf）を参照してください。

10.10.2 オブジェクト関数

これまでにも少し解説しました。

```
GetContainerAttribute( ソースフィールド ; 属性名 )
  →  指定したフィールドの、指定した属性名のメタデータ
```

多くの属性名が用意されています。注意点としてメタデータはバイナリファイルによっては付加されていないことがあります。

オブジェクトがイメージの場合に大きさの情報は以下の関数でわかります。

```
GetHeight ( ソースフィールド ) → イメージの高さ（ピクセル）
GetWidth ( ソースフィールド ) → イメージの幅（ピクセル）
```

イメージの縦横比はそのままで指定したポイントサイズ以下のサムネールを取得するには

```
GetThumbnail ( ソースフィールド ; 幅に合わせる ; 高さに合わせる )
（ソースフィールドはオブジェクトタイプフィールドか、サムネールを取得したいイメージへのファイルパスを指定）
```

サムネール

FileMaker Pro はイメージのサムネールを自動的に生成してレイアウトの表示に使用しています。［オブジェクトの］管理ダイアログで、そのサムネールの持ち方を設定することが可能です。

以下は論理値を返します。

```
VerifyContainer ( ソースフィールド )
  →  オブジェクトデータが FileMaker 以外を使用して変更または削除された場合
    「0（偽 = False)」、それ以外の場合は「1（真 = True )」
```

イメージと Base64 についての関数は

```
Base64Encode ( ソースフィールド )
  →  指定したフィールドに格納されているイメージをBase64フォーマットのテキストに変換
Base64Decode ( テキスト {; 拡張子を含むファイル名} )
  →  Base64 フォーマットでエンコードされたテキストからオブジェクトに変換
```

イメージを文字列にすること

Base64 はデータを 64 種類の英数字と記号で表現しています。画像を文字列として転送に使用することが可能です（Base64 の文字列に変換して転送しデコードして画像に戻す）。

10.10.3 統計関数

統計処理を行う関数です。Sum 関数はすでに演習で使用しました。統計関数は、1 テーブルの任意のフィールドを複数指定することもできますが、主に関連レコードのフィールドのデータを集計するためによく使用されます。

```
Sum ( フィールド {; フィールド...} )
→ 指定したフィールド内の空白を除いた値すべての合計
```

「見積情報」テーブルの「c_ 見積小計」フィールドでは関連の「見積明細」テーブルの「c_ 金額」フィールドのデータを Sum 関数で集計していました。他には、

```
Average ( フィールド {; フィールド...} )
  → 指定したフィールドの空白を除く有効な値すべての平均
Count ( フィールド {; フィールド...} )
  → 指定したフィールドの空白を除く有効な値の数
Max ( フィールド {; フィールド...} )
  → 指定したフィールドの最大値
Min ( フィールド {; フィールド...} )
  → 指定したフィールドの最小値
```

関連レコードのフィールドについて統計関数で集計する場合、非保存の計算結果になります。また関連テーブルのレコード数が増えるに従って時間がかかるようになります。

例えば、レイアウトに統計関数で関連レコードを指定している非保存の計算タイプフィールドや集計タイプフィールドをたくさん配置すると表示に時間がかかることがあります。これを短縮するには、非保存の計算タイプフィールドや、集計タイプフィールドの配置を減らしたり、ポップオーバー、ポップアップヘルプなど、ユーザが任意のタイミングで行うアクションをきっかけに結果が表示されるようにレイアウトで工夫します。

また、計算タイプフィールドとして集計するよりも、スクリプトで集計してからデータを入力するようにすると入力されたフィールドに索引が作成されるため、表示や検索に時間がかからなくなります。

Count するなら
関連レコード数を数えるために Count 関数を使用する場合は、主キーのような空欄不可のフィールドを指定します。

List 関数
とても便利な関数です。関連レコードの指定したフィールドのデータの改行区切りテキストをスクリプトで使用するために利用できます。

```
List ( フィールド {; フィールド...} ) → フィールド内の空白を除く値の改行リスト
```

統計関数の演習をおこないます。ポップアップヘルプで使います。

「演習 10_23 から 24」フォルダ

演習 10-23 レイアウトで統計関数を使う 1

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) 「見積詳細」レイアウトでレイアウトモードに切り替えます。

(2) 「会社名」フィールドを選択します。

(3) ポップアップヘルプに計算式を設定します。

a. インスペクタの「位置」タブの位置のポップアップヘルプの右にある鉛筆アイコンをクリックします。

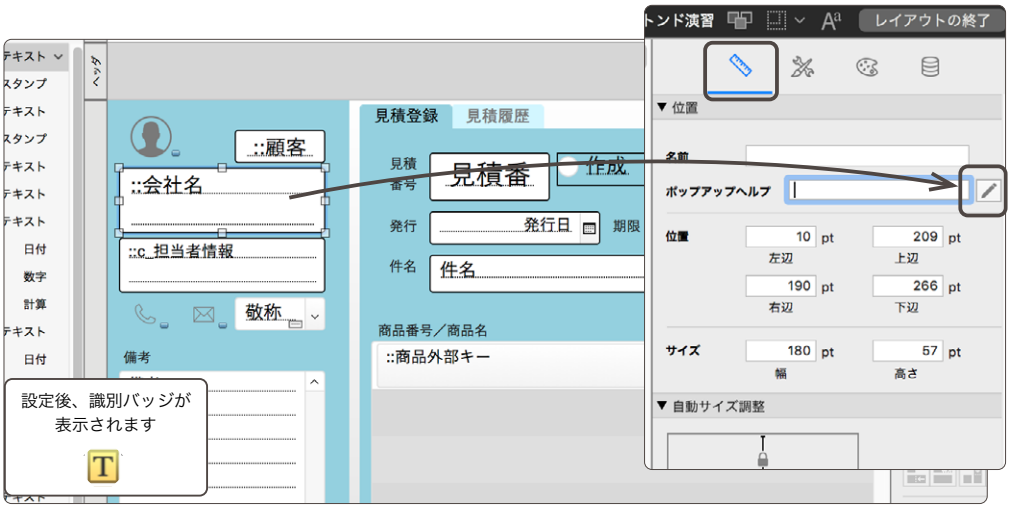


図 10-60 ポップアップヘルプの設定

b. ［計算式の指定］ダイアログが表示されます。

c. 以下の計算式を記述します。

```
Let ( [
   x = Count ( 見積管理｜履歴｜自己::主キー ) ;
   y = Sum ( 見積管理｜履歴｜自己::c_見積総額 )
] ;
   "件数：" & x & "件¶¶" &
   "合計：" & y & "円¶¶" &
   "平均：" & Round ( y/x ; 0 ) & "円"
)
```

d. ［OK］をクリックしてダイアログを閉じます。

(4) レイアウトを保存してブラウズモードに戻ります。

演習ここまで

ポップアップヘルプ
レイアウトオブジェクトに指定して使用します。ポップアップヘルプを表示するには、ユーザがカーソルをそのレイアウトオブジェクト上に置いて少しだけ待ちます。演習では計算式を使用しましたが、任意のテキストだけを指定することもできます。

計算式が指定されている場合、ユーザがカーソルをレイアウトオブジェクト上に置いてポップアップヘルプを表示しようとしているときに計算式が評価されます。

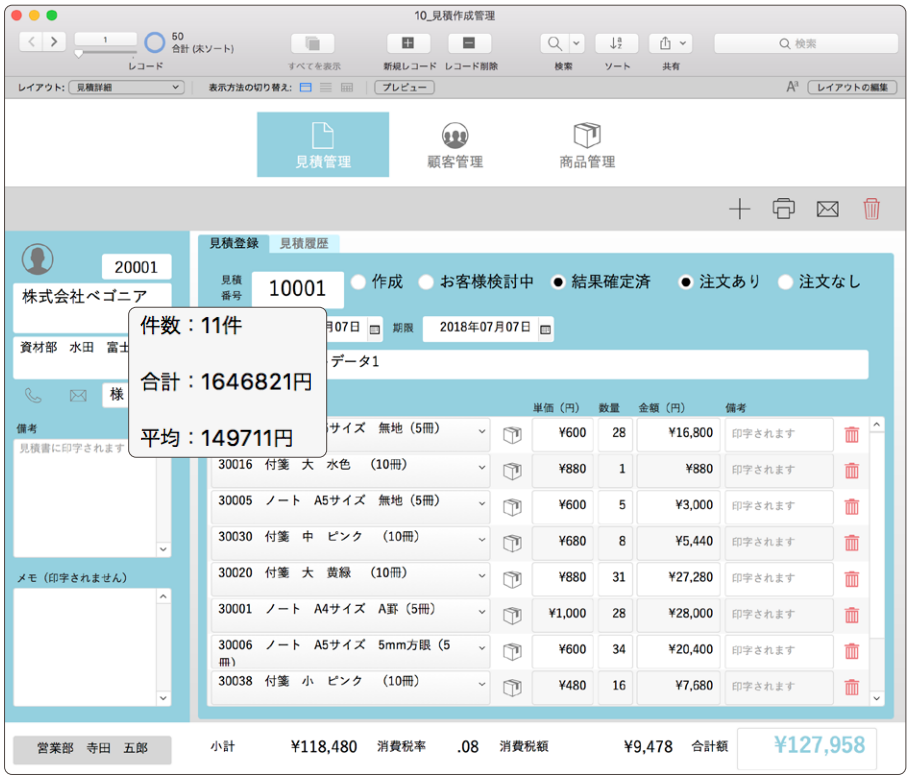


図 10-61 ポップアップヘルプが表示

この方法は、「見積情報」テーブルに関連レコードのデータを統計するためのフィールドを作成する必要がありません。同じ内容のポップアップヘルプを何度も設定する必要がある場合は、変数を利用したりフィールドを作成した方がメンテナンスしやすくなります。

続いて、**タブコントロールで統計関数を使います。**

演習 10-24 レイアウトで統計関数を使う 2

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) 「見積詳細」レイアウトでレイアウトモードに切り替えます。

(2) タブコントロールのタブ名に計算式を設定します。

a. タブコントロールをダブルクリックして［タブコントロール設定］ダイアログを表示します。
b. ［タブコントロール設定］ダイアログの表示
c. タブパネルの名称に計算式を使った設定をします。
d. 「見積履歴」をクリックして［指定 ...］をクリックします。
e. ［計算式の指定］ダイアログが表示されます。
f. 以下の計算式を記述します。

```
Let ( [
   x = Count ( 見積管理｜履歴｜自己::主キー ) ;
   y = Sum ( 見積管理｜履歴｜自己::c_見積総額 )
] ;
   "件数：" & x & "件/" &
   "合計：" & y & "円/" &
   "平均：" & Round ( y/x ; 0 ) & "円" )
```

g. ［OK］をクリックして［計算式の指定］ダイアログを閉じます。
h. ［名前変更］をクリックします。
i. ［タブ揃え］を「両端揃え」に変更します。

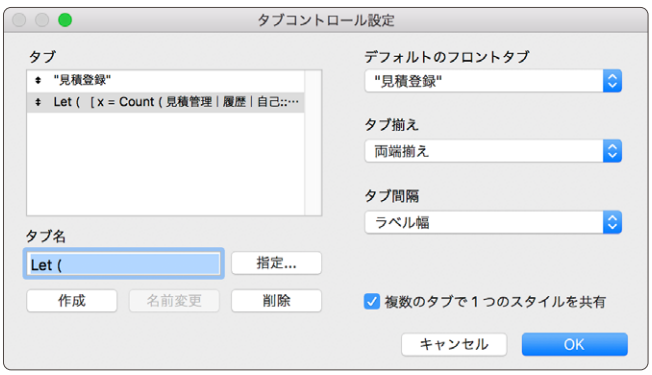


図 10-62［タブコントロール設定］ダイアログ

j. ［OK］をクリックして［タブコントロール設定］ダイアログを閉じます。

(3) レイアウトを保存してブラウズモードに戻ります。表示を確認してください。

演習ここまで

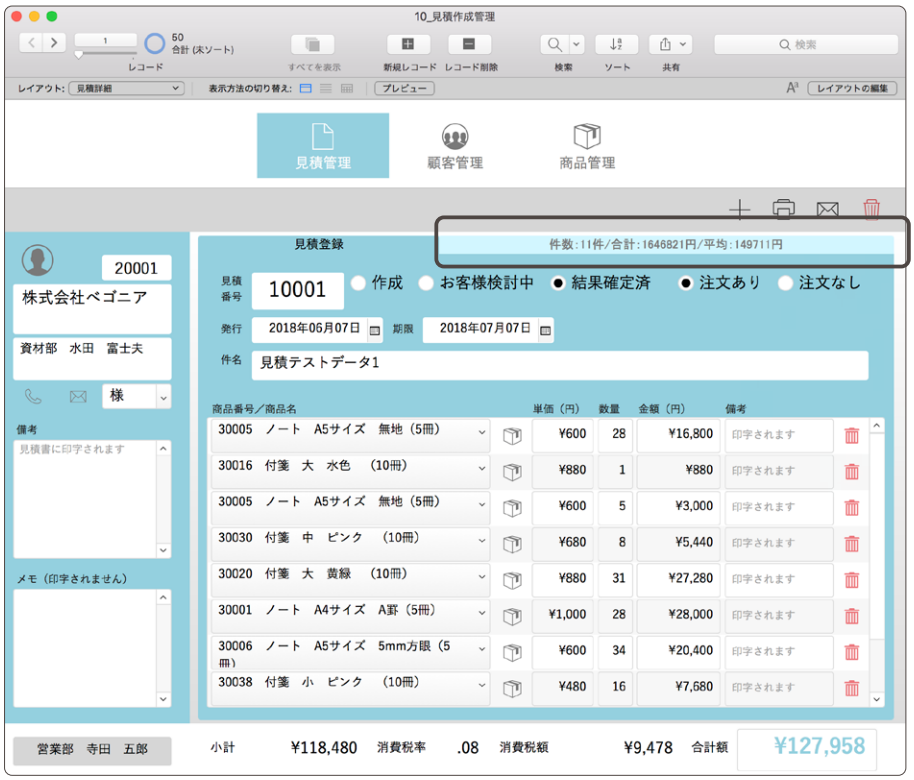


図 10-63 タブパネルの名前が計算式で表示される

タブコントロールやボタンの名前に計算式を使う

ボタンやボタンバー、タブコントロール、ポップオーバーのタイトルなど、計算式で名前を指定できる場所があります。

その場合、レイアウトモードでは、「計算」と表示されます。しかし、[表示]–[オブジェクト]–[サンプルデータ］のチェックを**オン**にするとブラウズモードの表示か確認できます。

10.10.4 JSON 関数

JSON データを解析および変更するための関数がいくつか用意されています。

JSON とは

JSON (JavaScript Object Notation) は、

・名前（要素）：値　のペア

で構成されたデータの記述方式です。データサイズが軽量でデータオブジェクトを転送する様々な Web サービスで使用されています。詳しくは「JSON の紹介」（https://www.json.org/json-ja.html）を参照してください。

JSON データは、REST API を使用して Web サービスなどから取得します。

REST API

REST(REpresentational State Transfer) API（Application Programming Interface）

API

簡単に言えば、ある OS やアプリケーションや Web サービスの機能について外部から呼びだして利用できるようにするインターフェースのことです。インターフェースには、ルールが決まっています。

REST

Web サービスの設計モデルです。簡単に説明すると、ある特定の URL に HTTP メソッドで問い合わせを行うと HTTP メソッドで答えが返ってくるようなもの、と考えてください。URL には処理（POST や GET など）と、必要に応じて変数（パラメータ）を指定します。

例えば、国土地理員の「標高 API」サービスで富士山の高さを JSON データで取得すると、

```
{"elevation":3751.9,"hsrc":"5m\uff08\u30ec\u30fc\u30b6\uff09"}
```

このような答えが返ってきます。これが「標高API結果」フィールドに保存されていることとします。

```
JSONFormatElements(json)
  →  JSON データ内の要素を読みやすい形に書式設定します。
```

```
JSONFormatElements( 標高API結果 )
```

次の結果が返ります。「名前（要素）」と「値」のペアがわかりやすくなりました。

```
{
  "elevation" : 3751.9,
  "hsrc" : "5m（レーザ）"
}
```

（ここから、本書に付属している演習ファイルの「演習 10_25 から 28」フォルダの中の「10_再帰関数例 .fmp12」ファイルをベースに進めます）

10.10.5 論理関数 3　再帰の関数

繰り返して処理を実行するための関数が用意されています。「**再帰**」のためのものです。再帰は、関数の中で自分自身を呼び出して処理を繰り返すことです。

While 関数

条件が真（1）の間、［ロジック］を繰り返してから、結果を返します。

```
While ( [ 初期変数 ] ; 条件 ; [ ロジック ] ; 結果 )
```

→ 次のように動作します。

1 初期変数引数を評価します。
2 条件引数を評価します。
3 条件が：
　真（0 以外の数値結果）の場合、ロジックを評価してから 2 番目のステップを繰り返す
　偽（0）の場合、繰り返しを停止して結果を返す

条件

条件が正しく評価されることに注意します。なんらかにより停止条件が偽（0）にならない場合、規定数まで処理を繰り返して「?」を返して再帰を強制終了します。

繰り返し回数は 50000 回

While 関数の繰り返しは 50000 回までに制限されています。または SetRecursion 関数で最大繰り返し数を設定することもできます。

SetRecursion 関数

式の再帰とループの最大繰り返し数を設定します。

```
SetRecursion ( 式 ; 最大繰り返し ) )
```

式が最大繰り返しを超えると「?」を返します

簡単にこれらの関数を使ってみましょう。

「演習 10_25 から 28」フォルダ

演習 10-25 再帰の計算式の作成

- **使用ファイル：10_ 再帰関数例 .fmp12**

(1)　計算式を作成します。

a.　［ツール］–［データビューア］メニューを選択して「データビューア」を表示します。
b.　右下の［+］ボタンをクリックします。
c.　［式の編集］ダイアログが表示されます。
d.　次ページの計算式を作成します。
e.　すぐに結果の 45 が表示されます。

もしも作成がうまくいかなければ演習ファイルの「copy 演習 10-25」をクリックすると、計算式がコピーされます。[式の編集] ダイアログにペーストしてください。

1 から順にたし算をおこなう計算式

```
SetRecursion (

While (
      //初期変数
      [ count = 0 ;
        x = 0  ] ;
      //条件
      count < 9 ;
      //
      [ count = count+1 ;
        x = x + count ] ;
      // 結果
      x
       )
; 10 )
```

この計算式は `count` が `9` より小さい数字の場合、`x` に `count` を足し続けるものです。

(2) 繰り返しの回数を変更して試します。

a. //条件 行の下にある `count < 9` を `count < 10` に書き換えます。
b. 結果は 55 に変わります。
c. 一番下の行 `; 10 )` を `; 9 )` に書き換えます。
d. 結果は？に変わります。これは SetRecursion 関数の式（While 関数の再帰）が 9 回を超えているためです。

演習ここまで

While 関数と SetRecursion 関数を簡単に使った例を演習しました。再帰は、[loop][Exit loop if] ステップの処理と似ています。しかし [Exit loop If] スクリプトステップでは、条件が真（1）になるとループを抜けて終了します。While 関数では条件が偽（0）になると再帰を終了します。

スクリプト内で [loop][Exit loop if] ステップなどで処理している部分を、再帰の計算式のみのシンプルな内容に置き換えることができるかもしれません。例えば、アカウント管理のための初期登録パスワード（ランダムな 12 文字の文字列）の自動生成の機能などです。

他の使い方も試します。**JSON データからデータを取り出すために、再帰の計算式を使ってみましょう。**

演習ファイルの「再帰関数例」レイアウトには、JSON データが入力されたフィールドが配置されています。ここからデータを取り出します。

演習 10-26 再帰の計算式の例

- **使用ファイル：10_ 再帰関数例 .fmp12**

(1) データビューアで計算式を作成します。
a. 右下の［+］ボタンをクリックします。
b. ［式の編集］ダイアログが表示されます。

演習 10-25 同様に計算式がコピーされるボタンを作成しました。「copy 演習 10-26」をクリックすると、計算式がコピーされます。［式の編集］ダイアログにペーストしてください

(2) すぐに次の結果が表示されます。

ドーナツ
チョコレートケーキ
バゲット

JSON データからデータを取り出す計算式

```
While (
// 初期変数
       [
       x = 再帰関数例::jsonTextEx01 ;
       y = "" ;
       count = 0 ] ;
// 条件
      count < 3 ;
// ロジック
       [
       y = y  & JSONGetElement ( x ; "製菓店.製品[" & count & "]名前
"
      ) & ¶ ;
        count = count +1 ] ;
//  結果
y )
```

(3) count が 3 より小さければ x に 「名前」要素のデータを取り出すものです。

演習ここまで

JSON データでは、配列は 0 から始まります。この計算式では count を 1 から始めると次の結果となります。

チョコレートケーキ
バゲット

やってみよう

SetRecursion 関数で最大繰り返し数を追加してみましょう。

やってみよう

参考として、Let 関数と組み合わせた計算式（「価格」要素のデータを合計するもの）も作成しました。「copy 参考」ボタンをクリックしてデータビューアで内容や結果を確認してみましょう。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 10_25 から 28」フォルダの中の「10_見積作成管理 .fmp12」ファイルをベースに進めます）

10.11 カスタム関数

FileMaker Pro では標準で用意されている関数に加え、開発者が独自の関数を作成することができます。次のクライアントで利用できます。

※ 高度なツールを　**オン**　にしているとカスタム関数を作成可能です（方法は 9 ページを参照）。

・FileMaker Pro
・FileMaker Go
・FileMaker WebDirect

10.11.1 カスタム関数に必要なもの

カスタム関数はファイルごとに定義が必要です。作成、設定などに、次のものが必要です。

・［完全アクセス］アクセス権セットのアカウント

カスタム関数は開発者が定義しておけば良く、ユーザはファイルにカスタム関数が定義されているだけで利用可能です。

カスタム関数には以下の特徴があります。

・長くて複雑な計算式を、シンプルな関数名に置き換える
・何度も利用する計算式を、シンプルな関数名に置き換える

Let 関数に似ています。そして、メンテナンスの効率も良くなります。計算の論理に変更があったとき、1 回の変更で対応ができるからです。そして、カスタム関数で可能なことは、原則として FileMaker Pro の標準の関数、スクリプトステップで可能です。まずそれらで動作を確認してからカスタム関数を作成する方が確実です。

10.11.2 カスタム関数の作成

カスタム関数は［カスタム関数を編集］ダイアログで作成します。

表示方法：

(1)　［ファイル］–［管理］–［カスタム関数］メニューを選択
(2)　［カスタム関数の管理］ダイアログが表示
(3)　［新規 ...］または［編集 ...］をクリック

簡単なカスタム関数を作成します。「開発者かどうか」を論理値で返す関数です。

「演習 10_25 から 28」フォルダ

演習 10-27 カスタム関数の作成 1

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1)　カスタム関数を作成します。
　a.　［ファイル］–［管理］–［カスタム関数］メニューを選択します。
　b.　［カスタム関数の管理］ダイアログが表示されます。

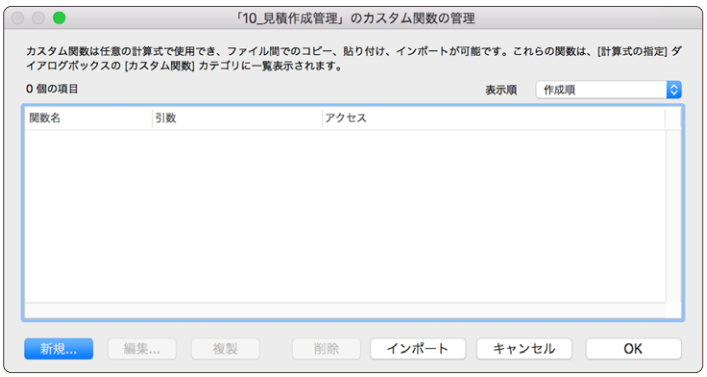


図 10-64［カスタム関数の管理］ダイアログ

　c.　［新規 ...］をクリックします。
　d.　［カスタム関数を編集］ダイアログが表示されます。
　　・［関数名］は　　「開発者かどうか」
　　・［関数の引数］は　空欄（この関数には引数がありません。）
　　・［新規関数］は　　以下の計算式を入力します（次ページの図も参照してください）。

```
Get ( 現在のアクセス権セット名 ) = "[Full Access]"
```

　e.　［OK］を何度かクリックしてダイアログを閉じます。

(2)　データビューアで動作を確認します。
　a.　右下の「+」をクリックします。
　b.　［式の編集］ダイアログが表示されます。

　c.　［式：］に、右の関数パネルの一番下にある「開発者かどうか」関数をダブルクリックして入力します。

d.　すぐに［結果：］に「1」が表示されるはずです。

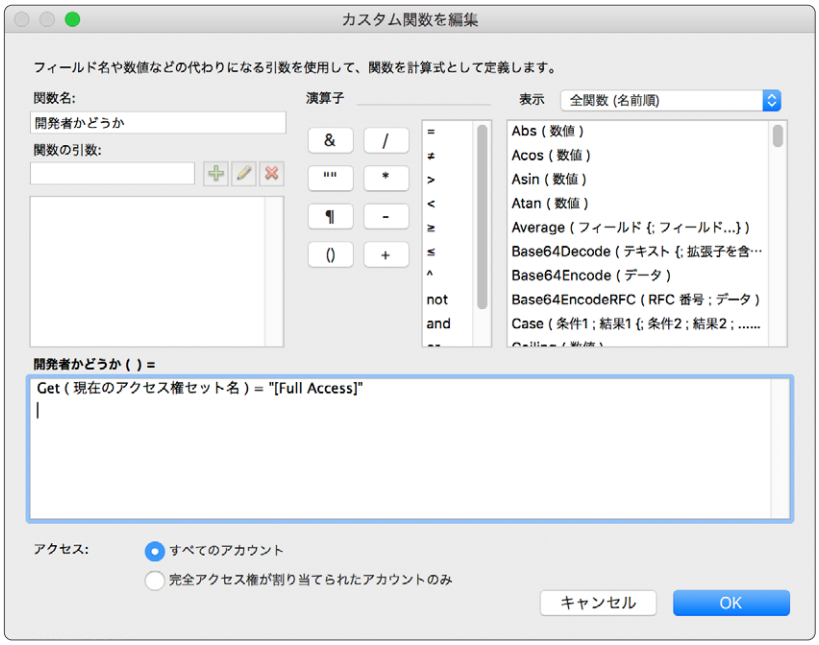


図 10-65［カスタム関数を編集］ダイアログに関数名と計算式を指定する

演習ここまで

カスタム関数ができました。このカスタム関数は、

・現在のユーザが［完全アクセス］アクセス権セットである＝「開発者」である＝「1」

という意味です。

必ずテストする
カスタム関数を作成したとき、すべてのアクセス権セット、すべてのクライアントのハードウェア OS で確認します。

ファイルごとに作成する
カスタム関数は、ファイルごとに定義します。他のファイルのものを利用することはできません。しかし、他のファイルからインポートができます。

誕生日を指定して年齢を計算する「CF 年齢（誕生日)」

```
/*
CF年齢（誕生日）

作成者　山田太郎
作成日　YYYY/MM/DD
*/


Let ( [
  w = GetAsDate( 誕生日 ) ;
  x = Get (日付) ;
  y = Date ( Month ( w ) ; Day ( w ) ; Year ( x )) ;
  z = not ( y ≤ x )
] ;


Case (

  IsEmpty ( w ) ; "" ; Year(x)-Year(w)-z

))

/*
誕生日が空欄の場合は年齢も空欄
そうでなければ
今日の「年」から「誕生年」を引いた結果に、今日が誕生日より大きいかどうかの論理値を利用して
0または1を引く
*/
```

対象日付を指定して消費税率を計算する「CF 消費税（対象日付)」

```
/*
CF消費税（対象日付）

作成者　山田太郎
作成日　YYYY/MM/DD
*/

Case ( GetAsDate( 対象日付 ) < Date ( 10 ; 1 ; 2019 ) ; .08 ; .1
)
```

更に**再帰のカスタム関数を作成します**。引数ありで、Case 関数や If 関数などで指定します。

再帰の回数
なんらかにより再帰が終わらない場合、50000 回まで処理を繰り返し、それ以降は再帰を終了して「?」を返します。作成後、必ずデータビューアで結果を確認することをお勧めします。

演習 10-28 再帰のカスタム関数

- **使用ファイル：10_ 見積作成管理 .fmp12**

(1) Case 関数を使ったカスタム関数を作成します。手順を一部省略します。
[カスタム関数を編集] ダイアログで設定します。
・[関数名] は 「CF再帰ドット」
・[関数の引数] は 「テキスト」「ドット数」の順番で登録します。
・[新規関数] は 以下の計算式

```
Case (
    ドット数 > 0 ;
    CF再帰ドット ( テキスト & "・" ; ドット数−1 ) ; テキスト )
```

(2) If 関数を使ったカスタム関数を作成します（手順は省略）。
・[関数名] は 「CF再帰左から文字」
・[関数の引数] は 「文字列」「数字」 の順番で登録します。
・[新規関数] は 以下の計算式

```
If(
    数字 > 0 ;
    CF再帰左から文字 ( 文字列 ; 数字−1 ) &
    Left ( 文字列 ; 数字 )& ¶ )
```

(3) [OK] を何度かクリックしてダイアログを閉じます。2 つの再帰関数を登録しました。

(4) データビューアで結果を確認します。

(5) [ツール]–[データビューア] メニューを選択します。

(6) CF再帰ドット関数をテストします。次のように確認してください。

```
CF再帰ドット ( "あいうえお" ; 10 )
  →  あいうえお・・・・・・・・・・  と結果が表示されます。
```

(7) CF再帰左から文字関数をテストします。次のように確認してください。

```
CF再帰左から文字 ( "あいうえお" ; 4 )
  →  あ
     あい
     あいう
     あいうえ
      と結果が表示されます。
```



図 10-66 データビューアで結果を確認

演習ここまで

再帰の計算式でカスタム関数をつくる

「10_ 見積作成管理 10_28 完了時 .fmp12」ファイルに、演習 10-28 の「CF 再帰左から文字」と同じ結果となる While 関数と SetRecursion 関数を使った「CF 再帰左から文字 2」カスタム関数をつくっておきました。データビューアで引数を演習 10-28 と同様に書き換えて結果を確認してみましょう。

「文字列」から、1 から数字まで分の文字を Left 関数で取り出す（ただし 100 文字以内）

```
SetRecursion (

    While (
    //初期変数
    [
     x = 文字列 ;
     y = 数字 ;
     z = "" ;
     count = 1 ] ;
    //条件
    count ≤ y ;
    //ロジック
    [
     z = z & Left ( x ; count ) & ¶ ;
     count = count + 1 ] ;
    //結果
    z )

; 100 )
```

10.11.3 カスタム関数の利用

［計算式の指定］ダイアログの右側の関数パネルに表示され、FileMaker Pro に標準で用意されている関数と同じ方法で計算式を指定します。

定数的に利用する

カスタム関数を定数として利用することができます。例えば、ファイルのリリースバージョンなどです。

アクセス権セットを制限する

［完全アクセス］アクセス権セットが付与されているアカウントのみがアクセスできるように設定することが可能です。例えば、従業員の年収額など、誰もがが利用できてはいけないものは、この設定が必要か考えます。

コメントを書く

自分で作成したカスタム関数でも作った理由などを忘れてしまうかもしれません。あるいは、別の開発者が他のファイルで利用するかもしれません。そのためにコメントを書いておきます。

```
Get ( 現在のアクセス権セット名 ) = "[Full Access]"
/*
開発者かどうか
引数無し
完全アクセス権かどうかを確認する
作成者　山田太郎
作成日　YYYY/MM/DD
*/
```

このような書き方でも良いですし、

```
/*
CF再帰ドット（テキスト ; ドット数）
「テキスト」の後に、指定した数のドット（「ドット数」）を付加する
作成者　山田太郎
作成日　YYYY/MM/DD
*/
Case (
    ドット数 > 0 ;
    CF再帰ドット ( テキスト & "・" ; ドット数 -1 ) ; テキスト
)
/*
CF再帰ドット（"あいうえお" ; 5）="あいうえお・・・・・"
のようになる。
*/
```

このように使用例を記述してもわかりやすくなります。

```
/*
CF再帰左から文字（文字列 ; ドット数）
「文字列」から、1から数字まで分の文字をLeft関数で取り出す
作成者　山田太郎
作成日　YYYY/MM/DD
CF再帰左から文字（"あいうえお" ; 3）= "あ¶あい¶あいう"
のようになる。
*/
If (
     数字 > 0 ;
     CF再帰左から文字( 文字列 ; 数字-1 ) & Left ( 文字列 ; 数字 ) & ¶ )
```

このようにまとめて書いても良いです。更新したら、更新日と更新者をコメントに追加します。

10.11.4 まとめてファイルに置いておく

“あのカスタム関数を作ったのはどのファイルだったっけ……”と探すよりも、カスタム関数用のファイルを 1 つ作成してまとめて保管しておくと便利です。必要になったら保管用のファイルからインポートするだけです。また、他の開発者がこれまでに作成したカスタム関数が公開されている Web サイトがあります。代表的なものをご紹介します。

・Brian Dunning's FileMaker Custom Functions
https://www.briandunning.com/filemaker-custom-functions/recentlist.php

演習ファイルを閉じてください。

第 11 章 スクリプトと自動化

機能の作成には、ほとんどの場合スクリプトが欠かせません。一連の作業を自動化してユーザをサポートします。

11.1 スクリプトも論理

スクリプトは、カスタム App にひとつの流れを持った作業を自動化した「**機能**」を作成するための重要なツールです。おおまかに言えば FileMaker Pro に『なにをさせるか』を記述するものです。計算式と同じくスクリプトも合意した要件や設計した仕様に従って記述します。

機能の設計
要件（必要な条件）に従って、具体的に何が実現されるか（仕様）を決定する、ということです。例えば、

要件（必要な条件）
見積書をメールなどでも送れるようにしたい。

↓

仕様（具体的な実現）
ユーザが「見積詳細」レイアウトで「PDF 保存」ボタンをクリックすると現在のレコードが「見積書」レイアウトの内容で「見積番号 - タイムスタンプ .pdf」名でデスクトップに PDF 形式でエクスポートされる。

曖昧な部分があるとうまく作成できないことがあります。要件やヒアリングが足りないのかもしれません。不明な部分はレイアウトで動作をユーザに見せながら確認することも有効です。

11.1.1 スクリプトの基本

機能のための「**スクリプト**」とはどのようなものでしょうか。特別なプログラミング言語を習得していなくてもすぐに作成できるものです。次のような特徴があります。

・用意されているコマンドを選択して組み合わせるので簡単に命令を作成できる
・コマンドは日本語名なので内容が直感的に理解できる
・コマンドに必要な条件をコマンド選択時にオプション設定できるのでわかりやすい

FileMaker Pro には、「**標準の機能**」があります。「新規レコード作成」「エクスポート」「検索実行」などがそれです。しかし多くの場合、それだけではなく、カスタム App「**独自の機能**」を作成する必要があります。

独自の機能は、標準の機能を組み合わせて順番に実行することで実現可能なことがほとんどです（例えば、ボタンをクリックすると、ある決まった検索条件で実行し、あるレイアウトへ切り替える）。その組み合わせと順序を、ユーザが毎回手作業で実行するのではなく『一部あるいはすべてを自動的に実行できるようにすること』が「**自動化**」です。そのために作成されるのが「**スクリプト**」です。

※ ここから「機能」と記述されることは、独自の機能のことです。

作業の順番を自動化することは、つまり、

・ユーザが意識していること／していないことを含めスクリプトステップに置き換える

ということです（ユーザが普段何気なく行っていることが重要だったりします）。そして、機能を作成する側としては実行される、

・コンテキスト（TO）

と、次のことを考慮する必要があります。

・どのタイミングで実行するか

スクリプト実行のタイミングは、任意のタイミング（ボタンクリック）、『ユーザが入力を開始したとき』などの細かいイベント（スクリプトトリガ）に連動させる、決まった時間に実行、が可能です。

スクリプトステップ
スクリプトの作成用にあらかじめ用意されている標準の機能を実行するコマンドが「**スクリプトステップ**」です。

メニューから実行できる項目を元に作成されていて、スクリプトステップ名はメニューの表示名とほぼ同じです。例えば［レイアウト切り替え］や［印刷］など、日本語でわかりやすく単純な最小単位のものです。そして、オプションが用意されています。あらかじめ選択肢が決まっているものや、選択、指定できるものがあります。

さらに、スクリプトステップには、自動化において重要なコマンド（メニューにはありません）も用意されています。「**制御コマンド**」です。

・条件分岐
・ループ（反復）

『ある条件』に合致した／しない場合の処理や、反復した処理は作業の自動化にとても有用です。

※「スクリプト」と「スクリプトステップ」は、正確には意味や実行できる場所、実行の手順が違いますが、自動処理が簡単にできるという意味で区別せず「スクリプト」と呼ばれる場合があります。

組み合わせは無限
ある 1 つの機能をつくるのに正解は特にありませんが、よく使われるスクリプトステップの順番というのはあり得ます。様々な組み合わせを試してください。スクリプトステップの組み合わせが可能にすることは無限といっても良い程たくさんあります。

演習ファイルを開いてください。

（ここから、本書に付属している演習ファイルの「11 章開始時」フォルダの中の「11_ 見積作成管理 .fmp12」ファイルをベースに進めます）

これまでの演習でスクリプトをいくつか利用しました。例えば、

「見積詳細」レイアウトの 2 つ目のタブパネル上のポータルの更新（スクリプトトリガで実行）
→ 「見積履歴ポータル更新」スクリプト

「見積詳細」レイアウトの 2 つ目のタブパネル上のポータルフィルタの計算式で指定しているフィールドのクリア（ボタンで実行）
→ 「フィルタフィールドクリア」スクリプト

などです。いずれも数行のシンプルな内容でした。スクリプトは、簡単なものから複雑なものまで様々なものが作成できます。

11.1.2 スクリプトの実行

スクリプトを「**実行**」するためには、なにかに「**割り当て**」します。対象はいくつかあります。

・ボタン
・トリガ
・スクリプトから
・メニュー（スクリプトメニュー、カスタムメニュー）

実行
スクリプトを動作させることを「実行」と呼びます。

11.1.3 スクリプトの流れ

スクリプトを「**実行**」すると、命令、処理、結果の流れが起こります。

(1) 命令する
(2) FileMaker Pro が処理する
(3) 結果を得る

ということです。つまり次図です。レイアウトを切り替えるという命令の例です。

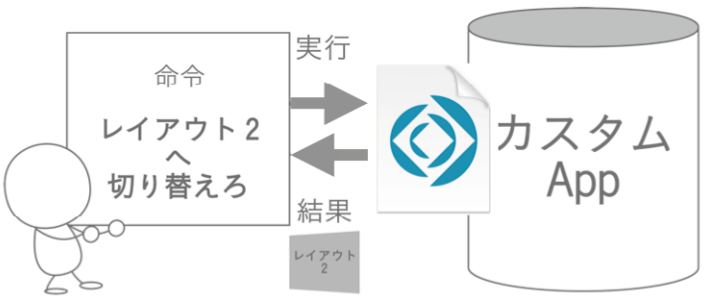


図 11-1 レイアウト変更の命令を実行する

命令する
「命令」が、スクリプトの重要な部分です。新規レコードの作成、レコードの検索、レコードのソート、などの実行を命令します。そして『命令を出すべきかどうか』を判断するには「**条件分岐**」を行います。これは大きく分けて次になります。

・命令に必要なデータによる分岐
・スクリプトとユーザの対話による分岐

FileMaker Pro が処理する
スクリプトの処理は、FileMaker Pro あるいは FileMaker Cloud ／ FileMaker Server が実行します。
実行する場所（サーバー側かクライアント側か）を開発者が指定できます。
※ FileMaker Cloud ／ FileMeker Server で実行するスクリプトは本書では取り扱いません。

結果
命令が実行され処理が完了すると、結果が返ります。検索結果が表示されたり、レイアウトが切り替えられたり、印刷されたり、などか、あるいは『エラーが発生して想定した結果が得られない』ことがあります。

11.1.4 スクリプトとエラー

スクリプトでは実行時の「**エラー**」への対応も重要なポイントです。まず、作成時に

・スクリプト内容のエラー（計算式で言うところの構文エラー）

を解決します。次に、『実行中のエラーをどのように処理するか』を決定しなければなりません。そのことを「**エラー処理**」と呼びます。

エラーの種類
実行中のエラーの原因は大きく分けて 2 種類です。

・命令を処理するために必要なデータが揃っていない
・データは揃っているが想定していない結果になる

これらは要件と関係します。

・どのようなデータが必要か
・どのような結果がでるべきなのか

を把握し、

・必要なデータが揃っているか／いないか
・どこで［キャンセル］をユーザが選択できるか／できないか
・FileMaker としてはエラーだが作業としてはエラーではない（検索結果が 0 件など）

などのパターンを考えて対応します。

11.1.5 スクリプトの終了

スクリプトの終了には、2 種類があります。これらはその名称通りの働きをします、

［現在のスクリプトを終了］スクリプトステップ
［全スクリプト終了］スクリプトステップ

トリガのキャンセルや、サブスクリプト実行、などでは上記のどちらを使用するべきか検討してください。

なるべく条件分岐でこれらのステップを使わずにスクリプトが終了することが可能かどうかも考えます。なお、［全スクリプト終了］スクリプトステップを使用する場合は、充分にテストしてください（どこで実行されるかによって思わぬことが起きるかもしれません）。

11.2 ［スクリプトワークスペース］

スクリプトの作成は、［スクリプトワークスペース］で行います。

ボタンやボタンバーにスクリプトを指定する［スクリプト指定］ダイアログから表示したり、メニューを選択して表示したりできます。そして、このウインドウは独立しています。

モードに関係なく任意のタイミングで表示したり閉じたりできます。つまりカスタム App の表示などを確認しながらスクリプトの作成や編集などが可能です。また、［スクリプトワークスペース］の作業中は、表示メニューがスクリプト専用のメニューに切り替わります。

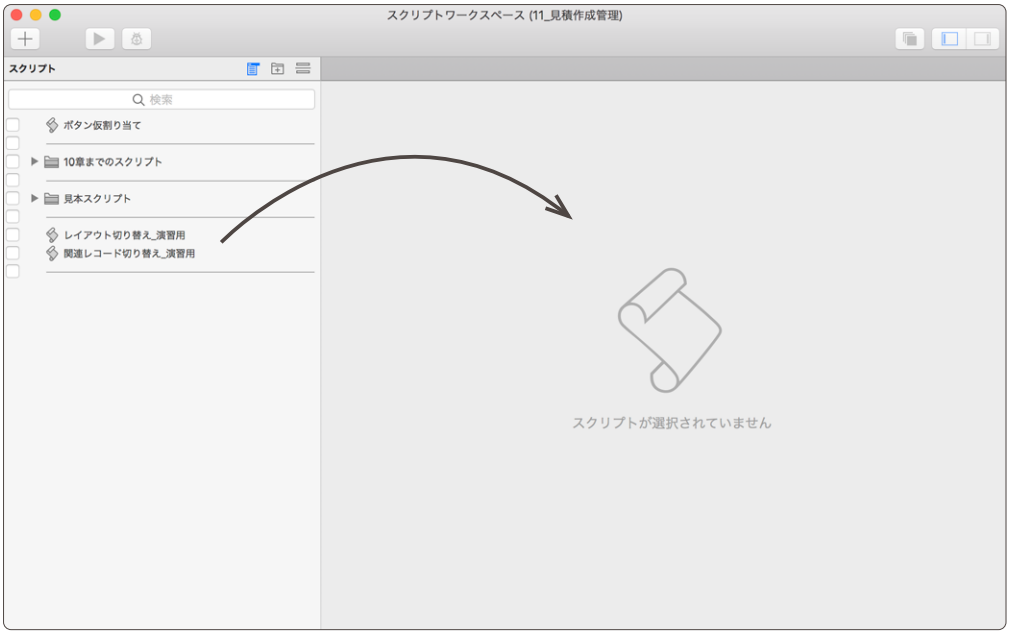


図 11-2［スクリプトワークスペース］

左側のスクリプトパネルで、いずれかのスクリプトが選択されると右側に、スクリプト編集エリアとスクリプトステップパネルが出現します。

演習ファイルの［スクリプトワークスペース］を開いてください。各部の名称などを解説します。

11.2.1 スクリプトパネル

一番左のパネル部分です。作成されているスクリプト名の一覧が表示されます。次の操作が可能です。



図 11-3 スクリプトが選択されている［スクリプトワークスペース］

スクリプトの複製、削除など

目的のスクリプトを選択して次のいずれかで実行します。

・［編集］メニュー（Windows）［スクリプト］メニュー（macOS）
・ショートカットメニュー
・キーボードショートカット

スクリプトの移動

選択されているスクリプトをドラッグすると任意の場所に移動できます。

まとめて選択

スクリプトやフォルダは、まとめて選択することもできます。複数選択するには［Ctrl］キー（Windows）または［command］キー（macOS）を押しながら各スクリプト名をクリックします。

スクリプト作成

左上の［+］ボタンをクリックします。「新規スクリプト」というスクリプトが作成されスクリプト編集パネルが開きます。

スクリプトの実行
選択しているスクリプトを、ここから実行できます。［スクリプトワークスペース］の左上にある［▶］ボタンをクリックします。ユーザの使用ではなく特別な実行や動作の確認での使用をお勧めします。

スクリプトのデバッグ
選択しているスクリプトを「デバッグ」するときにクリックします。
※「高度なツール」を**オン**にしている場合（9 ページを参照）のみ利用できます。

スクリプトの検索
検索ボックスに文字列を入力すると、表示されているスクリプト名から検索します。検索ボックス右の［X］ボタンをクリックすると検索文字がクリアされ、検索も解除されすべてのスクリプトが表示されます。

スクリプトパネルの検索ボックス右上には 3 つのボタンが並んでいます。

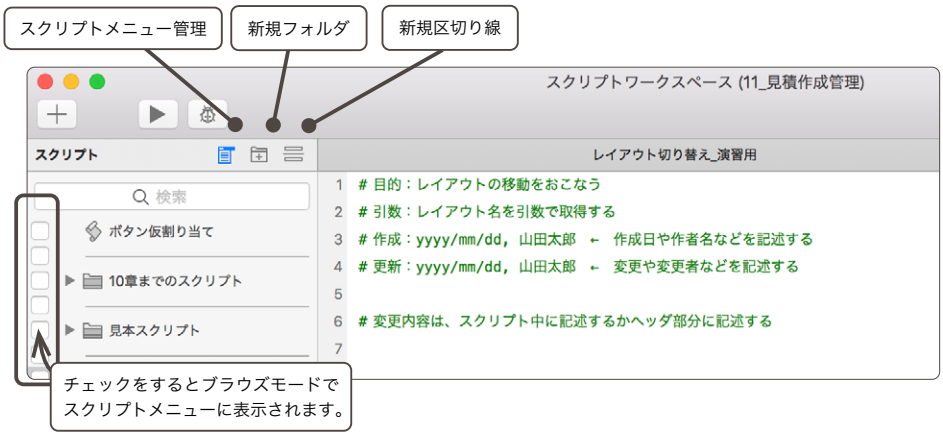


図 11-4 3 つのボタン

［スクリプトメニュー管理］ボタン
クリックすると、各スクリプトの先頭にチェックボックスが表示されます。スクリプトメニューに表示しない場合はチェックを　**オフ**　にします。［スクリプトワークスペース設定］で新規にスクリプトを作成した状態のオン／オフを設定できます（通常は　**オフ**　です）。

［新規フォルダ］ボタン
たくさんあるスクリプトをフォルダにまとめて管理できます。スクリプトメニューに表示すると階層化されてフォルダの下のスクリプトがサブメニューとして表示されます（スクリプトメニューに表示チェックがされている場合）。フォルダの中にさらにフォルダを入れ子にすることもできます。

［新規区切り線］ボタン
選択したスクリプトの下に区切り線を作成し表示します。スクリプト間の区切りを明確にできます。スクリプトメニューに表示することもできます。

11.2.2 スクリプト編集パネル

スクリプトパネルでスクリプトの記述や編集をします。複数のスクリプトをタブで開くことができます。



図 11-5 スクリプト編集パネル

スクリプトの表示

ちょっとした違いがあります。

- **スクリプト名をクリック**
  単に選択されているスクリプトの内容を表示します。なにも変更しないで別のスクリプトを選択するとそのスクリプトの内容に切り替わります。タブタイトルの右側にビュー記号が表示されます。

- **スクリプト名をダブルクリック**
  ［X］ボタンでスクリプトを閉じるまで開いています。タブタイトルにビュー記号はありません。別のスクリプトを選択すると別のタブでスクリプトが表示されます。

スクリプト編集パネルがたくさん開いてウインドウ内に表示しきれなくなると、右端に［>>］が表示されます。［>>］から隠れているタブを選択できます。

スクリプトを編集

表示されているスクリプトを編集すると右端に「*(アスタリスク)」が表示されます。スクリプトを保存すると「*」が消えます。

変更を元に戻す／やり直す
スクリプト編集パネルでは、スクリプトを保存または実行した後でも変更を元に戻す／やり直すことができます。しかし、スクリプト名タブを閉じる、［スクリプトワークスペース］を閉じるなどの操作をおこなうと変更を元に戻すことができなくなります。なお、スクリプトパネルのスクリプト名の変更やスクリプトの削除は元に戻すことはできません。

スクリプトの＜○○がみつかりません＞
TO、レイアウト、フィールド、スクリプトの削除は、慎重におこないます。スクリプトで使用されている場合は、確認メッセージが表示されます。
なお、他のファイルからスクリプトをコピーしてきたときや、TO、レイアウト、フィールド、スクリプトなどが削除されてスクリプト上で『見つからない』状態になると　＜○○がみつかりません＞とスクリプトに表示されます。後述の「スクリプトの問題をチェック」で発見できます。

ショートカットメニュー
スクリプト編集パネルのタブタイトルを右クリックするとメニューが表示されます。

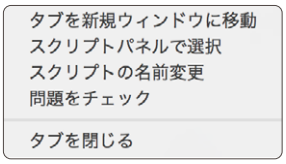


図 11-6 右クリックで表示されるメニュー

タブを新規ウインドウに移動
そのスクリプトを編集するための専用の新しいウインドウが開きます。

スクリプトパネルで選択
編集中のスクリプトをスクリプトパネルで選択し、隠れていればスクロールして表示します。

スクリプトの名前変更
スクリプト名を編集します。タブタイトルをダブルクリックしても変更できます。

問題をチェック
スクリプトの内容が、文法的に問題がないかチェックできます。あくまでも文法的なチェックです。実行後、正しい結果になるということではありません。問題のある行は、色が付いて強調表示されます。

タブを閉じる
スクリプト編集タブを閉じます。タブの上にマウスポインタを移動すると右に［X］ボタンが表示されるのでこれをクリックしても閉じることができます。

スクリプトの保存
主な方法のみ記述します。スクリプトはこまめに保存することを推奨します。

方法 1：メニュー
・［ファイル］メニューの［スクリプトの保存］（Windows）
・［スクリプト］メニューの［スクリプトの保存］（macOS）

方法 2：ショートカット
［Ctrl］＋［S］（Windows）あるいは［command］＋［S］（macOS）を使います。

11.2.3 スクリプトステップパネル

スクリプトに使用するスクリプトステップの一覧が表示されます。

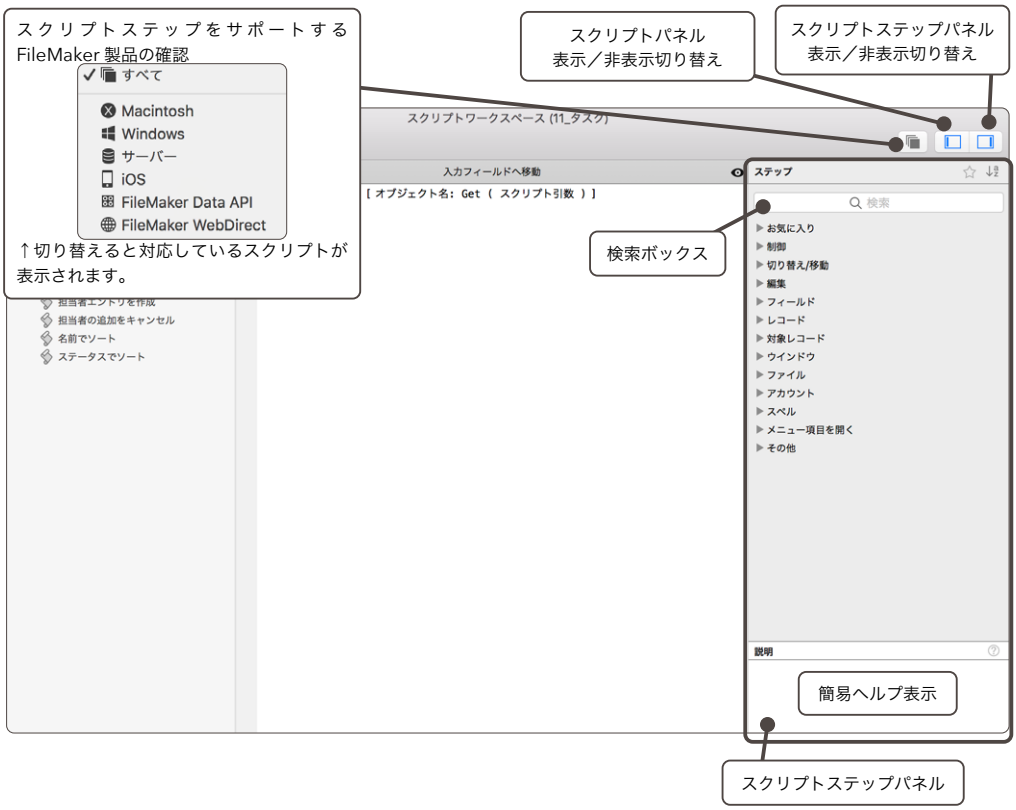


図 11-7［スクリプトワークスペース］

簡易ヘルプ
スクリプトステップを選択すると下に簡単なヘルプが表示されます。

スクリプトの検索
検索ボックスに文字列を入力するとスクリプトステップ名から検索します。検索ボックス右の［X］ボタンをクリックすると検索文字をクリアし検索が解除されすべてのスクリプトステップが表示されます。

検索ボックスの右上のボタンでスクリプトステップの表示を切り替えることができます。表示順をアルファベット順に並び替えたり、お気に入り機能を使うこともできます。

パネルの表示
スクリプトステップパネルが表示されていない場合、ウインドウ右上のスクリプトステップパネル表示切り替えボタンをクリックします。ほかにも可能なことがあります。
・スクリプトをコピーと貼り付け（ペースト）
・他ファイルからスクリプトをインポート

［スクリプトワークスペース］には、スクリプト作業を効率良く作業できるよう、あらゆる機能が詰め込まれています。

オートコンプリート

スクリプト編集パネルには、オートコンプリート機能が用意されています。入力を開始すると、スクリプトステップの名前（単語）の前方一致検索が行われて候補のスクリプトステップが表示されます。英語では頭字語でも候補が表示されます。例えば Go To Related Record だったら GTRR です。日本語は、例えば［関連レコードへ移動］スクリプトステップの場合、かんれん、kanren、関連、などで表示されます。

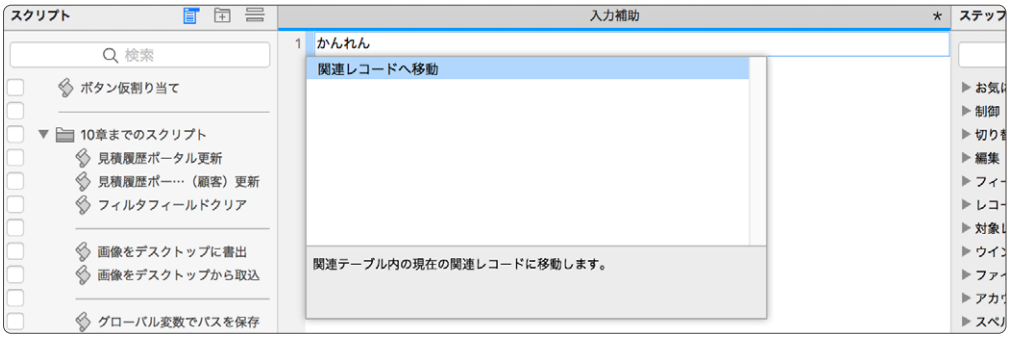


図 11-8 オートコンプリート

11.2.4 スクリプト作成のショートカット

覚えていると便利なものを表にしました。他のショートカットについてはヘルプを参照してください。

目的	macOS	Windows
現在の行の下に空白行を挿入	return または command + return	Enter または Ctrl + Enter
現在の行の上に空白行を挿入	Shift + return または command + Shift + return	Shift + Enter または Ctrl + Shift + Enter
オートコンプリートの候補から選択	↑ ↓	↑ ↓
オートコンプリートの候補を決定	return	Enter
オートコンプリートの候補から選択し最初の引数の選択候補を選択	Tab	Tab
オートコンプリートキャンセル	Esc	Esc
選択したスクリプトまたはスクリプトステップ行を削除する	Delete または Backspace	Delete または Backspace
選択したスクリプトステップを無効にする	command + /	Ctrl + /
空の行にコメントを追加する	#	#
すべてのタブを閉じる	command + option + W	Ctrl + Alt + W
現在のスクリプトのスクリプトオプションを選択する	return またはスペースバー	Enter またはスペースバー
スクリプトステップのオプションを表示する	+	+
［スクリプトパネル］内のすべてのグループを展開する／たたむ	command キーを押しながら 展開／たたむアイコンをクリック	Ctrl キーを押しながら 展開／たたむアイコンをクリック

※ 高度なツールを **オン** にしていないと使えないものがあります（ツールオンの方法は 9 ページを参照してください）。

11.2.5 スクリプトの設計

スクリプトは、単純なものであれば頭の中で設計しつつスクリプトステップを入力して並べただけでできてしまいます。エラーが起きたり、想定しないような結果になることはほとんどありません。しかし、スクリプトは、その流れや詳細を設計したほうが効率良く作成できます。

なにをするのかを基本に考えます。「機能要件」や「機能設計」と呼ばれます。

・扱うデータ
・コンテキスト（TO とレイアウト）
・処理内容
・ユーザの操作
・帳票などの出力の形式

などが要素として必要です。前述の、

（1）命令する
（2）FileMaker Pro が処理する
（3）結果を得る

これらついて 5W1H になぞらえて考えると次のようになります。

（1）いつ　　実行のタイミング
（2）どこで　　どのレイアウト（TO）か
（3）誰が　　どのユーザか
（4）なにを　　どのレコードが対象か
（5）なぜ　　目的
（6）どのように　　実行する内容（命令と命令のオプション）

この中でも「どのように」が、スクリプトの処理です。目指すのは、構文エラーがなく想定された結果を出すスクリプトです。

フローチャートを書くと整理できる
業務フローチャートと同じように処理フローチャートを考えると頭が整理できます。手書きでも図を描くためのアプリケーションでもなんでも、どのような順番でなにをするのか、どのように処理の分岐や反復をするのか、条件を設定するかなどを考えます（次ページ図を参照）。

処理にはパターンがある
スクリプトの設計や作成はある程度の経験値が蓄積されていたほうがスムーズに進みます。スクリプトステップの使い方、全体の流れの設計を何度も行うことで適切なスクリプトステップの組み合わせがパッと思い浮かぶようになるからです。そして、よく使う処理の流れにはスクリプトステップの組み合わせが同じようなパターンになることに気づきます。その処理パターンをうまく使用して『汎用性の高い』スクリプトを作成できるようになります。

はじめは個々のスクリプトについてのみ考えていて良いのです。そのうちパターンがあることに気付くと、すべてのスクリプトを俯瞰して眺める視点を持つことができ効率良く使用することを考えることができます。

11.2.6 スクリプトのテスト

設計を行ってから作成を開始したとしても、試行錯誤（つくる←→テスト）になります。テストはとても重要です。充分に時間をとるようにします。

複数の結果をすべて試す

複数の結果が想定されているなら、すべての結果をテストします。

どのような結果を想定したらよいか、は、要件定義時や、設計時にわかるはずです。レアケース（ほとんどあり得ない結果やエラー）も含めてすべてを網羅する必要があるかどうかについては慎重に考えてください（網羅するとスクリプトは複雑な内容になることが多いためです）。

テストデータ

スクリプトの実行テストでは、あるべき結果がわかっていないと構文エラーだけを解決することが目的になりがちですが、それだけでは不十分です。次のことをお勧めします。

・『実行したら、この結果になる』というテストデータを準備してテストする

前述のレアケースについてはデータを準備すること自体ができないかもしれません、だとするとそのレアケースはスクリプトの対象範囲外でも良い、という判断ができるかもしれません。

他の人にテストや確認をお願いする

自分以外の開発者が、テストやスクリプトの内容をレビュー（査読）する時間がとれるとベストです。自分では気付かない視点からの助言や指摘をもらえるからです。なお、レビュー時にスクリプトの説明資料として処理フローを合わせて提示すると、

・なにが作りたかったか
・結果としてなにができたか

が、わかりやすくなります。テストやスクリプト内容の理解に役立ちます。

スクリプトの作成に習熟すると、自分と自分以外の開発者がメンテナンスができるような記述（コメント）も考えられるようになりますが、それまでには、ある程度の時間がかかります。

設計、テスト、できあがったスクリプト全体を眺めるという作業はとても重要です。余裕をもって取り組めるように可能な限り充分な作業期間をとることをお勧めします。

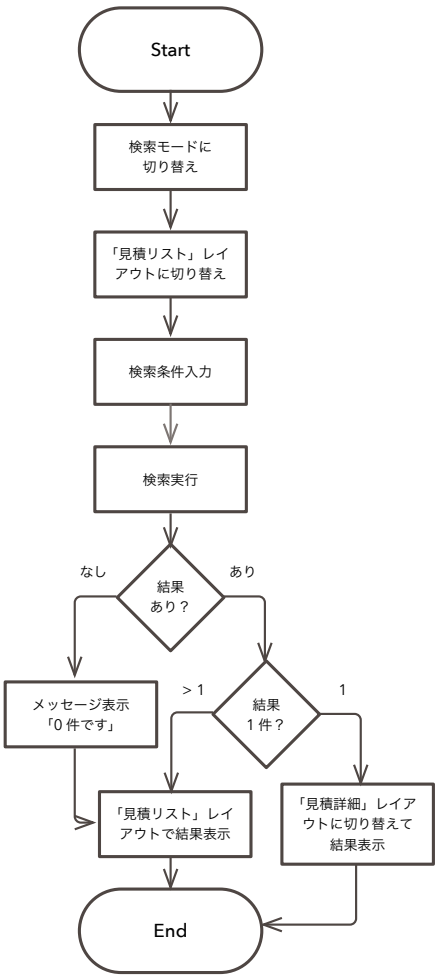


図 11-9 処理フローの例（検索）

11.2.7 スクリプトデバッガ

スクリプトのテストに便利なツールあります。「**スクリプトデバッガ**」です。

デバッグ
プログラムのバグ（bug、虫、つまりはエラーのことです）を探して取り除くことを言います。バグ取りなどと言うこともあります。デバッガはバグ取りをサポートするツールのことです。またはそのような作業をおこなう人のことをデバッガーと呼んだりします。

前述したように［スクリプトワークスペース］での作業時に構文的にエラーにならなくても、実際に処理を実行しているとエラーになることがあります。例えば、

・想定していないレイアウトに切り替えられてしまう、あるいは切り替えされていない結果、処理が（TO が異なる、などの理由で）エラーになる

と、いうようなことです。スクリプトデバッガはこのようなエラーを確認するのに最適なツールです。スクリプトステップを 1 行ずつ実行できるのでエラーになった原因箇所を特定することが容易になります。

※高度なツールを　**オン**　にしていないと使えません（ツールオンの方法は 9 ページを参照してください）。

スクリプトデバッガは以下の方法で表示します。

方法 1：
［スクリプトワークスペース］で任意のスクリプトを指定して、［スクリプト］–［スクリプトのデバッグ］メニューを選択します。指定しているスクリプトが実行されて、デバッグできます。

方法 2：
［ツール］–［スクリプトデバッグ］メニューを選択します。なにもスクリプトを指定してなくても表示されます。なにかスクリプトを実行するとデバッグできます。
（再度選択するとスクリプトデバッガが表示されなくなります）

方法 3：
［スクリプトワークスペース］で任意のスクリプトを指定して、スクリプトパネルの上にある虫マークのボタンをクリックします。指定しているスクリプトが実行されて、デバッグできます。

スクリプトデバッガに配置されているボタンなどは次ページ図のとおりです。難しいことはなにもありません。何度も使ううちにボタンの役割などが自然と覚えられるはずです。

ボタン／カスタムメニューの「単一ステップ」のデバッグ
両方ともスクリプトデバッガでデバッグできます。

（次図と同じ状態を確認するには、演習ファイルの［スクリプトワークスペース］のスクリプトパネルの「10 章までのスクリプト」フォルダの「画像をデスクトップに書出」スクリプトをクリックし、スクリプトパネルの上にある虫ボタンをクリックで、［スクリプトデバッガ］を表示してください）

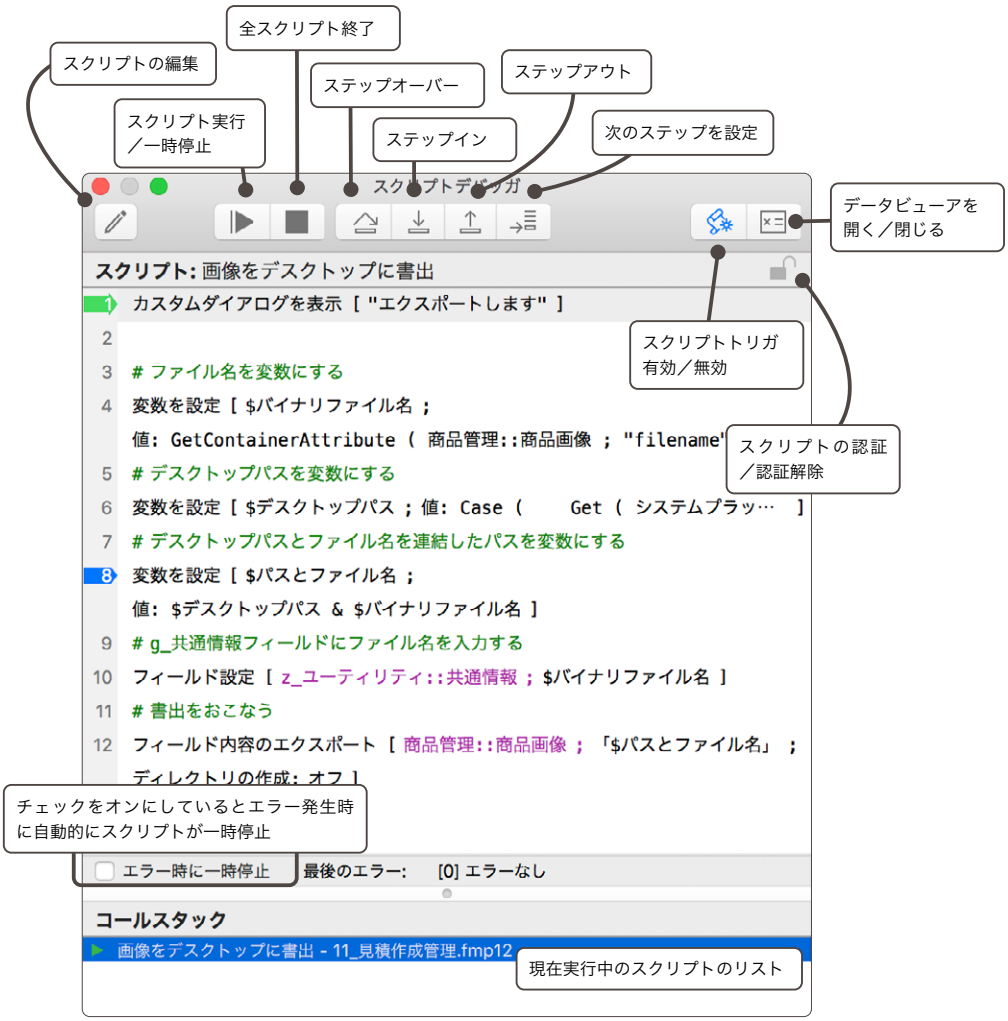


図 11-10 スクリプトデバッガ

ステップオーバー
スクリプトを 1 ステップごとに実行します。実行対象のスクリプトステップが［スクリプト実行］の場合、サブスクリプトを実行して呼び出しもとのスクリプトの次のステップへ進みます。

ステップイン
スクリプトを 1 ステップごとに実行します。実行対象のスクリプトステップが［スクリプト実行］の場合、サブスクリプトの 1 ステップ目へ進みます。

ステップアウト
実行中の現在のスクリプトのすべてのスクリプトステップを実行します。現在のスクリプトがサブスクリプトの場合、呼び出し元のスクリプトの［スクリプトの実行］スクリプトステップの次のステップへ進みます。

次のステップを設定
現在選択されているスクリプトステップに実行ポインタ（緑の矢印マーク）を設定します。

スクリプトトリガを有効／無効にする
デバッグ時に、ファイル内のすべてのスクリプトトリガの有効／無効を切り替えます。無効にするには［完全アクセス］アクセス権セットを持ったアカウントのログインが必要です。無効後、スクリプトデバッガを閉じると有効に戻ります。

スクリプトの認証／認証解除
デバッガを起動した現在のアカウントがスクリプトを編集することができない権限である場合、スクリプトデバッガまたはデータビューアのロックを解除するために使用します。ロックの解除には［完全アクセス］アクセス権セットを持ったアカウントのログインが必要です。編集権限は、スクリプトデバッガおよびデータビューア両方を閉じるまで持続します。

ブレークポイント
デバッグ中に、任意のステップで一時停止するために使用します。ブレークポイントはスクリプトに保存されますが、スクリプトデバッガ以外のスクリプト実行ではブレークポイントは無視されます。ブレークポイントは、スクリプトの行番号をクリックすると挿入されます。

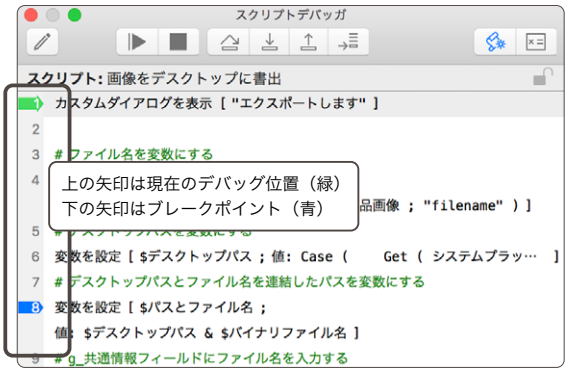


図 11-11 ブレークポイント

11.2.8 データビューア

もう 1 つ、便利なツールがあります。「**データビューア**」です。計算式の結果確認でも使用しました。「監視」タブでした。データビューアには 2 つのタブがあります。スクリプトデバッガでスクリプトを実行しているときは「現在」タブで、そのスクリプトに記述されている

・フィールド名と値
・変数名と値

を確認できます。

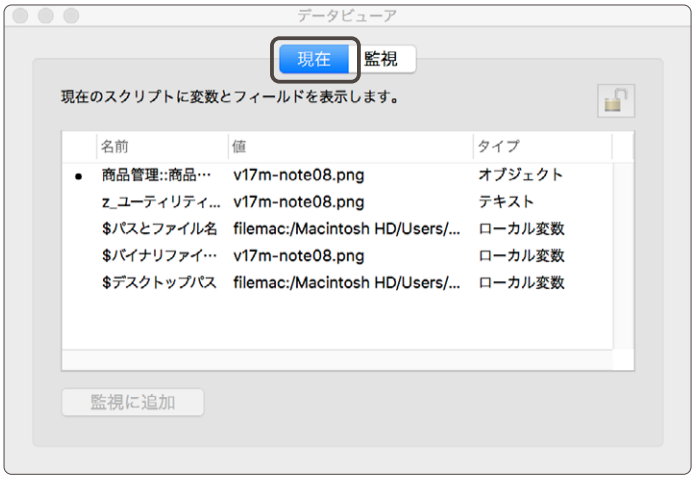


図 11-12 データビューアの「現在」タブ

11.3 スクリプトトリガ

スクリプトトリガについて解説します（ボタンやボタンバーへスクリプトを割り当てる作業は、本書をお読みになっている方は何度も行っていると思いますので、改めて説明はしません）。

「**トリガ**」とは、つまり特定の「イベント（きっかけ）」です。このトリガが発生したときにスクリプトが実行されるようにできます。イベントは、例えば、フィールドへのデータの入力を確定することなどです。そのイベントが実行される直前または直後にスクリプトが実行されます。

トリガは、25 個用意されていて大きく分けて次の 3 種類です。

・レイアウト
・レイアウトオブジェクト
・ファイルオプション

例えば、「OnObject...」ではじまるトリガの「Object」は、レイアウトオブジェクトのことです。あるいは「OnLayout...」ではじまるものは、レイアウトに関わるものです。これらは、レイアウト上で設定します。あるレイアウトで設定しても他のレイアウトでは適用されないので気をつけてください。

ユーザとスクリプトトリガ
トリガは、ユーザが任意にボタンをクリックすることと違います。ユーザが意識しないところでスクリプトが実行されるため、とても便利で有用です。しかし、逆にユーザが戸惑うこともあります。どのようなことが起きるのか、必ずユーザの立場に立ってテストしてください。

想定したタイミングでスクリプトが実行されるかどうか、つまり適切なスクリプトトリガが設定されているか、を確認するにはデバッガを使用するとわかりやすく確認できます。

スクリプトトリガが実行されないもの
次のような操作では実行されません。

・インポート、全置換、再ルックアップなど、レイアウト上ではなく FileMaker Pro がテーブルのレコードに対して実行する操作
・［カスタムダイアログを表示］、［フィールド設定］、［フィールドを名前で設定］、［Web ビューアの設定］でフィールドのデータを変更するスクリプトステップが実行されたとき

本項目の表について
以降のスクリプトトリガの説明中の表の「可能なクライアント」については、カスタム Web 公開は明記していません。また、次の略語を使用します。

・FMP → FileMaker Pro
・FMS → FileMaker Server ／ FileMaker Cloud
・FMG → FileMaker Go
・FWD → FileMaker WebDirect

本節の解説よりもさらに詳しい内容はヘルプを参照してください。

11.3.1 スクリプトトリガの順番

スクリプトトリガの評価には順番があります。複数のスクリプトトリガが有効になる状況では、FileMaker Pro が 1 つずつ実行するためです。同じイベントで複数のスクリプトトリガが実行されると、スクリプトトリガは次の順で実行されます。

(1) OnFirstWindowOpen
(2) OnWindowOpen
(3) OnLayoutEnter
(4) OnModeEnter
(5) OnRecordLoad
(6) OnLayoutSizeChange
(7) OnViewChange

次のスクリプトトリガでは、トリガイベントによって順序が制御されます。

(1) OnObjectEnter
(2) OnObjectKeystroke
(3) OnLayoutKeystroke
(4) OnGestureTap
(5) OnObjectAVPlayerChange
(6) OnFileAVPlayerChange
(7) OnExternalcommandReceived
(8) OnPanelSwitch
(9) OnObjectModify
(10) OnObjectValidate
(11) OnObjectSave
(12) OnObjectExit
(13) OnRecordCommit または OnRecordRevert
(14) OnModeExit
(15) OnLayoutExit
(16) OnWindowClose
(17) OnLastWindowClose

スクリプトトリガとイベント

スクリプトトリガの実行はイベントの「前」と「後」の 2 種類があります。

クライアントがなにか、どの順番で実行されるものか、イベントの前か後か、を考えて設定します。1 回で 1 つずつ設定していきますが、結果として複数のスクリプトトリガが有効になることがあります。充分にテストをします。

トリガのキャンセル

イベントの前に実行されたスクリプトが [現在のスクリプト終了] スクリプトステップで結果が 0 (偽 = False) で終了するとイベントそのものがキャンセルされます。[現在のスクリプト終了] スクリプトステップで結果を 1 (真 = True) で終了するか、結果がないか、このスクリプトステップが含まれていないスクリプトであった場合、イベントは実行されます。必ず条件分岐を行います。

11.3.2 レイアウトのスクリプトトリガ

12 種類あります。次表を参照してください。

名称	イベント		可能なモード		クライアントのサポート			
	前	後	ブラウズ	検索	FMP	FMS	FMG	FWD
OnRecordLoad		○	○	○	○	○	○	○
OnRecordCommit	○		○	○	○	○	○	○
OnRecordRevert	○		○	○	○	○	○	○
OnLayoutKeystroke	○		○	○	○		△	
OnLayoutEnter		○	○	○	○	○	○	○
OnLayoutExit	○		○	○	○	○	○	○
OnLayoutSizeChange		○	○	○	○		○	○
OnModeEnter		○	○	○	○		○	○
OnModeExit	○		○	○	○		○	○
OnViewChange		○	○	○	○	○	○	○
OnGestureTap	○		○	○	○		○	
OnExternalcommandReceived		○	○	○			○	

スクリプトトリガを設定する場所

レイアウトモードで［レイアウト設定］ダイアログの「スクリプトトリガ」タブを使います。

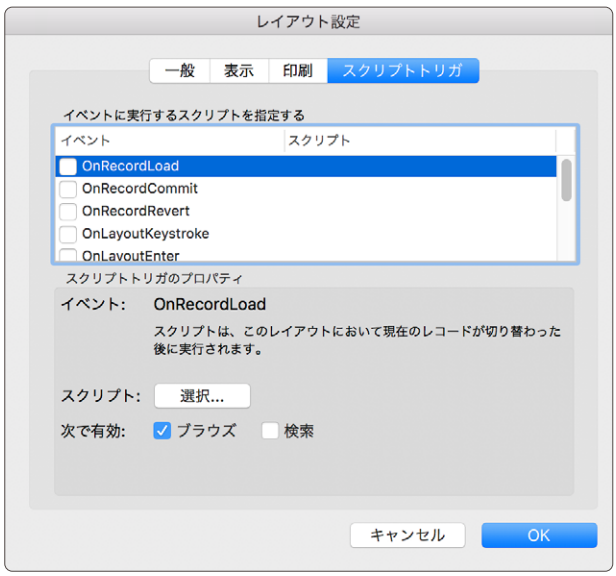


図 11-13 レイアウトのスクリプトトリガ設定

```
OnRecordLoad
```

レイアウトにレコードをロード（読み込み）した直後にスクリプトが実行されます。例えば「一覧画面」から「詳細画面」へレイアウトを移動する場合に、選択したレコードの種類（顧客情報だったら、それが個人か企業か）で表示するレイアウトを違うものにする、などで使用できます。

OnRecordCommit

変更があったレコードの確定の直前にスクリプトが実行されます。

OnRecordRevert

一連のレコードが復帰（入力前の状態に戻される）の直前に、スクリプトが実行されます。

OnLayoutKeystroke

レイアウトでなにか1つ以上の文字が入力を感知したときにスクリプトが実行されます。プレビューモードでも動作します。以下の関数でキーを判断します。

```
Code ( テキスト ) 関数
Char ( 数値 ) 関数
Get ( トリガキー修飾キー ) 関数
Get ( トリガキー入力 ) 関数
```

(1) キーボードでキーが押される
(2) クライアントのOSが判断する（例えばOS のショートカットに該当するとそれが実行されます。するとスクリプトトリガは実行されません）
(3) FileMaker Proが判断する（ショートカットに該当するとそれが実行されます。するとスクリプトトリガは実行されません）
(4) OnObjectKeystrokeトリガが設定されているアクティブなレイアウトオブジェクトからスクリプトトリガが有効になる
(5) 上記のイベント後、OnLayoutKeystrokeトリガが有効になる

上記は、スクリプトが実行されるまでの流れになります。なお、実行されたスクリプト内のエラーでスクリプトが終了できなかった場合、トリガしたキーは処理されません
FileMaker Goの場合、ユーザが入力したときにフィールドが有効である場合に限りアクティブになります。

OnLayoutEnter

レイアウトのロード（表示）直後にスクリプトが実行されます。（このトリガで実行されるスクリプトの最初の行を［新規ウインドウ］スクリプトステップとすると再帰になるので何らかの方法でスクリプトを強制終了しなければ無限に反復されることになります）
例えば「一覧画面」へレイアウトを移動したらソートや検索を実行する、などに使用できます。
なお、プレビューモードでも動作します。

OnLayoutExit

レイアウトを切り替える直前にスクリプトが実行されます。プレビューモードでも動作します。

OnLayoutSizeChange

レイアウトまたはウインドウのサイズが変更された直後にスクリプトが実行されます。

FileMaker Goでは、iOS/iPadOSデバイスを回転、ステータスツールバーの表示／非表示、ウインドウが最初に開いたときにアクティブになります。

FileMaker ProとFileMaker WebDirectでは、次の場合にアクティブになります。
・ユーザによるサイズ変更
・スクリプトステップ
・メニューコマンド
・ショートカット
・スクリプトでのステータスツールバーまたは書式設定バーの表示 / 非表示によるレイアウトやウインドウのサイズ変更、
・ウインドウが最初に開いたとき
（マウスを使用してウインドウのサイズを変更した場合、ウインドウのサイズ変更が終了したときにのみアクティブになります）

OnModeEnter

ブラウズモード、検索モード、プレビューモードへ切り替えた直後にスクリプトが実行されます。例えば、検索モードに切り替えた直後に、検索条件を初期設定する、などに使用できます。

OnModeExit

ブラウズモード、検索モード、プレビューモードへ切り替える直前にスクリプトが実行されます。例えば、検索実行の直前に、検索条件を初期設定する、あるいは、プレビューモードからブラウズモードへ戻る直前にウインドウを調整する、などに使用できます。

OnViewChange

フォーム形式、リスト形式、表形式に表示を切り替えた直後にスクリプトが実行されます。プレビューモードでも動作します。

OnGestureTap

レイアウト上でタップジェスチャが受信されたときにスクリプトを実行します（Windows とiOS/iPadOSのみ）。次のジェスチャで、このトリガが有効になります。
・1、2、または 3本の指でのタップ（iOS/iPadOS）
・1本の指でのダブルタップ（iOS/iPadOS）
・2本の指でのタップ（Windows）
1本の指でダブルタップする場合は、1回目と2回目のタップのときに2回評価されます。

`OnExternalcommandReceived`

ユーザが、ロック画面または外部デバイスで次のいずれかのボタンをクリックしたときにスクリプトが実行されます。停止、再生、一時停止、再生／一時停止を切り替え、次のメディアを再生、前のメディアを再生、前方または後方への検索を開始または終了。次の関数で、何のイベントかを判断します。

`Get ( トリガ外部イベント ) 関数`

11.3.3 レイアウトオブジェクトのスクリプトトリガ

8 種類あります。次表を参照してください。

名称	イベント		可能なモード		クライアントのサポート			
	前	後	ブラウズ	検索	FMP	FMS	FMG	FWD
OnObjectEnter		○	○	○	○	○	○	△
OnObjectKeystroke	○		○	○	○		○	
OnObjectModify		○	○	○	○	○	○	△
OnObjectValidate	○		○	○	○	○	○	○
OnObjectSave		○	○	○	○	○	○	○
OnObjectExit	○		○	○	○	○	○	△
OnPanelSwitch	○		○	○	○	○	○	○
OnObjectAVPlayerChange		○	○	○			○	

スクリプトトリガを設定する場所

レイアウトモードでフィールド、タブコントロール／パネルコントロール、ポップオーバー、Web ビューア、ポータル、ボタンバーなどをクリックして［書式］–［スクリプトトリガ設定 ...］メニューを選ぶか、右クリックでショートカットメニューから［スクリプトトリガ設定 ...］を選択します。

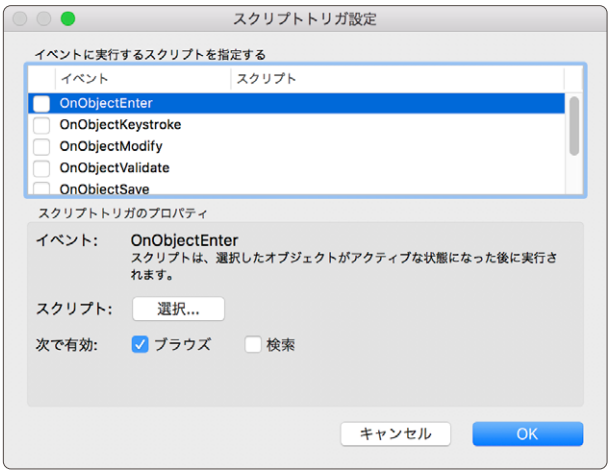


図 11-14 レイアウトオブジェクトのスクリプトトリガ設定

OnObjectEnter

レイアウトオブジェクトがアクティブになった直後にスクリプトが実行されます。
ユーザがフィールドのデータを編集する前に、編集するフィールドと編集内容を取得できます。例えば、レコードの変更ログを記録する機能に使用することができます。ポータルでは、ポータル行がアクティブになる度に評価します。
アクティブになっていないボタンやタブコントロールは、それらをクリックしただけでは評価されません。

OnObjectKeyStroke

レイアウトオブジェクトがアクティブになり、なにか 1 つ以上の文字の入力を感知したときにスクリプトが実行されます（OnLayoutKeystroke を参照）。
フィールドオプションの［入力値の制限］は、アクティブなフィールドから抜けないと評価が起きません。そのため、このトリガが役立つ場合があります。

OnObjectModity

レイアウトオブジェクトの値が変更された直後にスクリプトが実行されます。

OnObjectValidate

アクティブなレイアウトオブジェクトが検証および保存される直前にスクリプトが実行されます。
このスクリプトトリガは、選択されたフィールドのデータが変更された場合にのみ有効になりなす。

OnObjectSave

アクティブなレイアウトオブジェクトが検証と保存の直後にスクリプトが実行されます。

OnObjectExit

アクティブなレイアウトオブジェクトが検証を終了する直前にスクリプトが実行されます。
［現在のスクリプト終了］スクリプトステップで、結果が 0（偽＝False）で終了するとレイアウトオブジェクトから出られなくなるので注意してください。

OnPanelSwitch

タブコントロール、パネルコントロールでパネルが切り替えられる直前にスクリプトが実行されます。アクティブでないパネルをクリックしたりスクリプトで移動したときです。

```
OnObjectAVPlayerChange
```

レイアウトオブジェクトのメディアの状態を変更したときにスクリプトが実行されます。

11.3.4 ファイルのスクリプトトリガ

5 種類あります。次表を参照してください。

名称	イベント		クライアントのサポート			
	前	後	FMP	FMS	FMG	FWD
OnFirstWindowOpen		○	○	○	○	○
OnLastWindowClose	○		○	○	○	○
OnWindowOpen		○	○	○	○	○
OnWindowClose	○		○	○	○	○
OnFileAVPlayerChange		○			○	

スクリプトトリガを設定する場所

[ファイル]–[ファイルオプション ...] メニューを選び、[ファイルオプション] ダイアログの「スクリプトトリガ」タブで設定します。

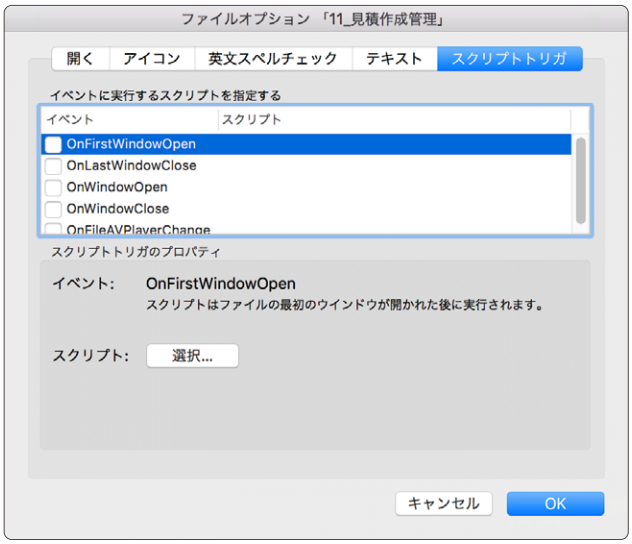


図 11-15 レイアウトオブジェクトのスクリプトトリガ設定

```
OnFirstWindowOpen
```

ファイルの最初のウインドウが開いた直後にスクリプトが実行されます。ほとんどの場合は起動時です。関連ファイルが裏側で開いている（ウインドウが非表示になっている）と、最初にウインドウが表示されたときに有効になります。

`OnLastWindowClose`

ファイルの最後のウインドウが閉じる直前にスクリプトが実行されます。ほとんどの場合はファイルを閉じるときです。関連ファイルの場合はウインドウが非表示で開いている状態になります。

`OnWindowOpen`

ウインドウが開いた直後にスクリプトが実行されます。

`OnWindowClose`

ウインドウが閉じる直前にスクリプトが実行されます。

`OnFileAVPlayerChange`

メディアファイルがフィールドまたは URL から再生されていて、ユーザまたはスクリプトステップの設定が再生状態を変更したときにスクリプトが実行されます。

11.3.5 OnTimer

OnTimer スクリプトはこれまでのものとは違います。次のスクリプトステップを利用します。

［OnTimer スクリプトをインストール］スクリプトステップ

（1）実行するスクリプトを指定
（2）指定したスクリプトが実行されるまでの時間間隔を指定（秒単位）

の順番に動作します。(1)、(2) はタイマーがキャンセルされるまで繰り返されます。例えば、次のような用途に使えます。

・アンケートを実施するためのレイアウトへ切り替えたら 1 時間ごとにカスタムメッセージを表示して 3 時間経ったら終了する

OnTimer は、次のような特徴があります。

・［OnTimer スクリプトをインストール］スクリプトステップが実行されたときにアクティブなウインドウに OnTimer スクリプトが設定される
・1 ウインドウに 1 つの OnTimer スクリプトだけを設定できる
・既に OnTimer スクリプトが設定されているウインドウに別の OnTimer スクリプトを設定すると、前の OnTimer スクリプトは取り消される
・既に OnTimer スクリプトが設定されているウインドウを閉じると取り消される
・複数のウインドウで、それぞれ OnTimer スクリプトを実行させることができる
・既に OnTimer スクリプトが設定されているウインドウをアクティブにしてユーザやスクリプトが新規ウインドウを作成すると新しいウインドウに OnTimer スクリプトが引き継がれる

OnTimer スクリプトのキャンセル
OnTimer スクリプトをキャンセルするには次の方法があります。

・ウインドウを閉じる
・［OnTimer スクリプトをインストール］スクリプトステップで実行するスクリプトを指定しない
・［OnTimer スクリプトをインストール］スクリプトステップで時間間隔を 0 秒または空欄を指定

時間のカウント
タイマーで設定した時間内にユーザが作業をしている場合、FileMaker Pro は、その作業を待ってからスクリプトを実行します。そしてスクリプトが開始したら次のタイマーが開始します。つまり、正確な間隔ではない場合があります。

11.4 スクリプトの作成

「見積作成管理 .fmp12」に、スクリプトを作成していきます。

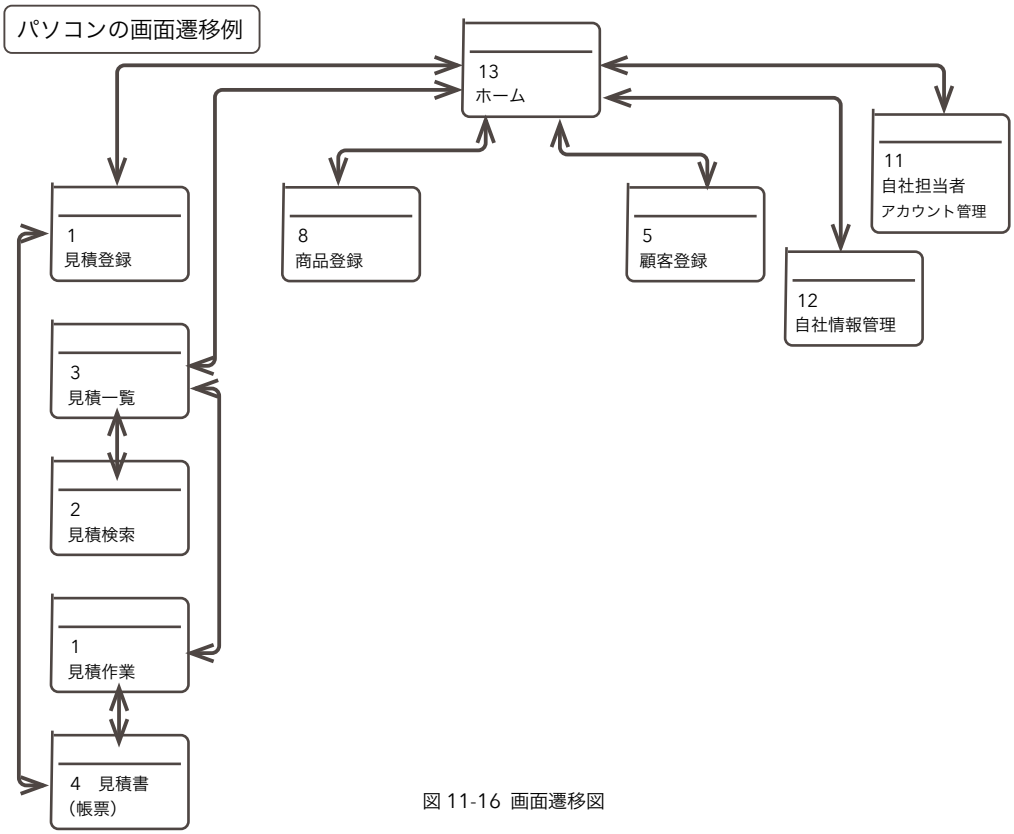


図 11-16 画面遷移図

	画面名	レイアウト種類	パソコン	モバイル
1	見積登録・作業画面 → 見積詳細	フォーム形式	○	×
2	見積検索画面	フォーム形式	○	×
3	見積一覧画面 → 見積リスト	リスト形式	○	×
4	見積書（帳票）(印刷／ PDF)	フォーム形式	○	×
5	顧客登録・作業画面 → 顧客管理｜マスタ詳細	フォーム形式 (マスタ / 詳細)	○	×
~~6~~	~~顧客検索画面~~	~~フォーム形式~~	⊖	~~×~~
~~7~~	~~顧客一覧画面 → 顧客リスト~~	~~リスト形式~~	⊖	~~×~~
8	商品登録・作業画面 → 商品管理｜マスタ詳細	フォーム形式 (マスタ / 詳細)	○	×
~~9~~	~~商品検索画面~~	~~フォーム形式~~	⊖	~~×~~
~~10~~	~~商品一覧画面~~	~~リスト形式~~	⊖	~~×~~
11	自社担当者・アカウント情報登録・作業画面	フォーム形式 (マスタ / 詳細)	○	×
12	自社情報管理画面	フォーム形式	○	×
13	ホーム画面 → ホーム	フォーム形式	○	×
(モバイル画面は省略)				

画面遷移は第 8 章で考えた流れから少し変わりました。マスタ／詳細レイアウトを取り入れたためです。画面数が少なくなっています。

なお、本書の演習では、すべての機能を作成していません。ピックアップしたものを作成します。

11.4.1 スクリプト作成の注意点

いくつか注意点があります。改めて確認します。

コンテキスト、TO を常に考える

「いまどのレイアウトにいるのか」は、そのレイアウトに指定されている TO によって、

・現在のウインドウ
・現在のウインドウに表示されている、レイアウト
・現在のウインドウのレイアウトの、現在の対象レコード
・現在のウインドウのレイアウトの対象レコードの、現在のレコード
・現在のウインドウのレイアウトの対象レコードの現在のレコードの、アクティブなフィールド

すべてが関係します。TO（コンテキスト）を適切に考慮しスクリプト内で指定をおこなわなければなりません。そしてそれらを把握するために、いくつもの Get 関数が用意されています。

```
Get ( ウインドウ名 )
Get ( レイアウト名 )、Get ( レイアウト番号 )、Get ( レイアウトテーブル名 )
Get ( 対象レコード数 )
Get ( レコード番号 )、Get ( アクティブレコード番号 )、Get ( アクティブポータル行番号 )
Get ( アクティブレイアウトオブジェクト名 )、Get ( アクティブフィールド名 )
GetFieldName( フィールド )
```

などです。他に用意されている関数についてはヘルプを参照してください。

ユーザのクリップボードは当てにしない

あたり前のことですが、ユーザはパソコンで FileMaker Pro、iPad や iPhone で FileMaker Go を使用します。

フィールドにデータを入力することについて、次のスクリプトステップの使用は注意が必要です。

```
［コピー］スクリプトステップ
［切り取り］スクリプトステップ
［貼り付け］スクリプトステップ
```

これらは、ユーザの使用している OS のクリップボードを利用します。例えば、ユーザがスクリプトの途中で他のアプリケーションでコピーや切り取りを行った後に［貼り付け］スクリプトステップが実行されたら……。想定とは違うデータが入力されるかもしれません。

クリップボード

OS が一時的に保存しているデータです。FileMaker Pro だけのものではありません。

コピー、切り取り、貼り付けは、次のスクリプトステップを使用して実行することもできます。クリップボードと関係がありません。安全に作業ができるものです。

［変数を設定］スクリプトステップ　→　データの一時保存先として使えます
［フィールド設定］スクリプトステップ　→　フィールドにデータを入力できます
［フィールドを名前で設定］スクリプトステップ　→　フィールド名でデータを入力できます

グローバル格納フィールド
グローバル格納フィールドは、TO やコンテキストは関係ありません。変数と同様にデータの一時保存先としてスクリプトでも便利に使えます。

11.4.2 「切り替え」や「移動」

例えば、ホーム画面でボタンをクリックしてレイアウトを「切り替える」ことはユーザにとっては、『作業を開始する』ための何気ない作業です。しかし、この操作は FileMaker Pro にとって、

・コンテキスト（TO）を切り替える

という意味を持ちます。そして、入力するためにフィールドをクリックすることは、

・レイアウトオブジェクトへ「移動」する

ということになります。スクリプトでは、切り替えや移動はとても重要なものです。次のスクリプトステップは、切り替えや移動に関するスクリプトステップです。

［レイアウト切り替え］スクリプトステップ
［レコード／検索条件／ページへ移動］スクリプトステップ
［関連レコードへ移動］スクリプトステップ
［ポータル内の行へ移動］スクリプトステップ
［オブジェクトへ移動］スクリプトステップ
［フィールドを選択］スクリプトステップ

また、ウインドウが複数開いている場合も選択が必要です。

・どのウインドウ（コンテキスト／ TO）を選択して作業をするか

そのために次のスクリプトステップが用意されています。

［ウインドウを選択］スクリプトステップ

これらの切り替え先／移動先／選択先の指定は、スクリプトステップのオプションでおこないます。

・レイアウト、フィールド、レイアウトオブジェクト、ウインドウの名前を任意、計算式で指定
・最初、最後、前、次、何番目のように指定したり、計算式で指定

などの方法が用意されています。

スクリプトで、切り替えや移動が重要なのは、スクリプトステップの中には『どの TO か』を指定するオプションがないものがあるからです。例えば、

［新規レコード／検索条件］スクリプトステップ
［レコード／検索条件削除］スクリプトステップ

これらは、その実行の前に適切なレイアウトに切り替えてられていないと目的の結果が得られません。

レイアウト、ウインドウ、レイアウトオブジェクトと、コンテキスト（TO）が「現在何であるか」は常に意識することが必要です。

11.4.3 スクリプト引数

スクリプトにも引数があります。引数を使う例として、レイアウトの切り替えに関するスクリプトを考えます。

「見積作成管理 .fmp12」では、ファイル起動直後の画面（今後は「ホーム」レイアウトと呼びます）から各メニューへ遷移し、作業後「ホーム」レイアウトに戻る画面遷移を考えていました。そして、できれば、見積、商品、顧客の 3 メニューも横断して行き来できると良いとも考えています。

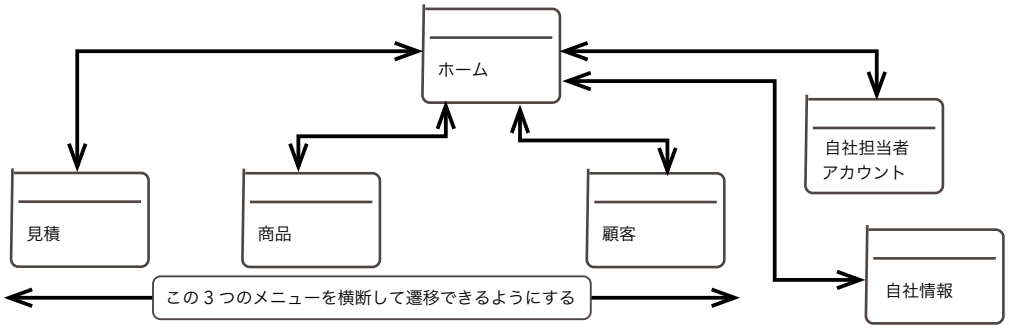


図 11-17 見積、商品、顧客を横断するイメージ

レイアウトの切り替えの流れは、ユーザ側から考えると、

（1）ボタンをクリックする
（2）スクリプトが実行されてレイアウトが切り替わる

ですから、切り替える前にレイアウト名がわからないといけませんので、

（1）ボタンをクリックする　→　**切り替えるレイアウト名を指定**
（2）スクリプトが実行されてレイアウトが切り替わる

ボタンに「切り替え先のレイアウト名」を持たせれば良さそうです。ここに「スクリプト引数」を使います。

引数は、次の場所で実行するスクリプト名と一緒に指定します。

・ボタン
・ボタンバー
・スクリプトトリガ
・スクリプト内

次の内容で渡します。

・任意の文字列
・フィールドのデータや計算式や変数

スクリプト引数を利用すると、1 つのスクリプトに複数の役割を持たせることが可能です。改めて、レイアウト切り替えのためのスクリプトについて考えます。

なにを引数にするか
例えば、「顧客管理｜マスタ詳細」レイアウトに切り替えるためには、

(1) いつ　　ユーザの任意
(2) どこで　　ユーザが現在表示しているレイアウト
(3) 誰が　　ユーザ
(4) なにを　　対象レコード関係なし
(5) なぜ　　レイアウトを切り替える
(6) どのように　「顧客管理｜マスタ詳細」レイアウトに切り替え

と考えられますが、引数を使うのであれば具体的にしなくても良い部分があります。

(1) いつ　　ユーザの任意
(2) どこで　　ユーザが現在表示しているレイアウト
(3) 誰が　　ユーザ
(4) なにを　　対象レコード関係なし
(5) なぜ　　レイアウトを切り替える
(6) どのように　レイアウト名を指定して~~「顧客管理｜マスタ詳細」レイアウトに~~切り替え

(6) です。どのレイアウトからも［顧客管理］ボタンの移動先は「顧客管理｜マスタ詳細」レイアウトです。ですから引数は「切り替えるレイアウト名」となります。

どのような流れが必要でしょうか。少し細かく見ます。

(1) ボタンをクリックする
　(1-1) 実行するスクリプトと引数（レイアウト名）が指定される
(2) スクリプトが実行される
　(2-1) 引数に従って、指定されたレイアウトに切り替える
(3) レイアウトが切り替わる

となります。引数を変えれば「見積詳細」でも「商品管理｜マスタ詳細」でも同じ動作で良いので 1 つのスクリプトで複数の画面遷移が可能です。

渡された引数の内容を得る

スクリプト内で、渡されたスクリプト引数の内容を得るためには、次の関数を使います。

```
Get ( スクリプト引数 ) 関数
```

引数を使ったレイアウト切り替えのためのスクリプトを作成します。

本章からの演習ファイルでは、スクリプト名に「_ 見本」と付加されているものが「演習結果例」のスクリプトです。「_ 演習用」と付加されているものに演習でスクリプトを作成していきます。

演習ファイルに「ホーム」レイアウトを作成しておきました。ボタンだけのシンプルなものです。これまでレイアウトの切り替えはレイアウトバーから手作業で行っていましたが、ボタンで可能にします。

「11 章開始時」フォルダ

演習 11-1　スクリプト引数でレイアウトを切り替える 1

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　「ホーム」レイアウトに切り替えます。

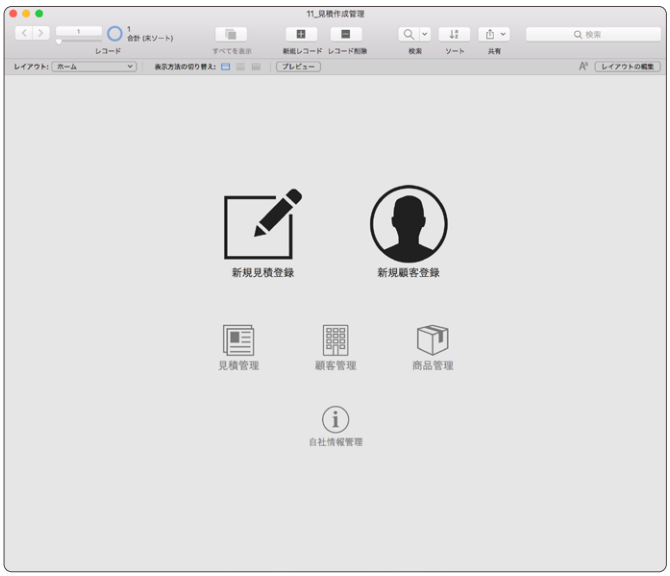


図 11-18「ホーム」レイアウト

(2)　レイアウトモードに切り替えます。

(3)　画面の中央にあるボタンバーの「見積管理」ボタンのスクリプトを指定します。

a.　ボタンバーの一番左の「見積管理」をダブルクリックして［ボタンバー設定］ダイアログを表示します。
b.　現在は「ボタン仮割り当て」スクリプトが割り当てられています。
c.　スクリプト名の右のボタンをクリックします。
d.　［スクリプト指定］ダイアログが表示されます。

e. 「レイアウト切り替え _ 演習用」スクリプトを選択します。
f. 歯車アイコンをクリックして［編集 ...］を選択します。
g. ［スクリプトを編集］ダイアログが表示されます。スクリプトには、# コメントによるヘッダ情報だけで、なにも内容はありません。

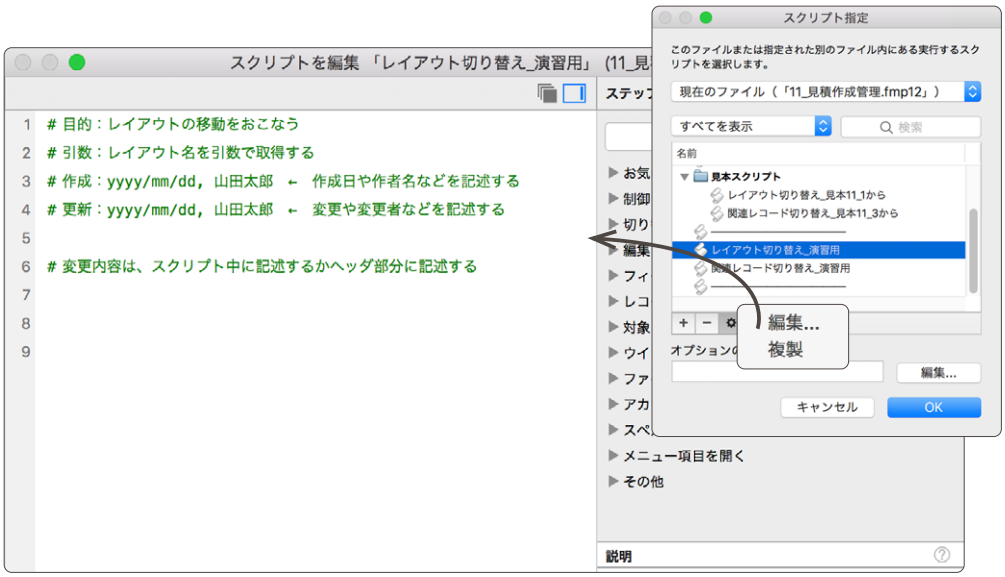


図 11-19 表示されるダイアログ

オートコンプリートとキーボードを使った手順を記述します。スクリプトパネルを使った操作でも構いません。

(4) スクリプトを編集します。
a. 8 行目をクリックして「レイアウト」と入力すると、候補のスクリプトステップが表示されます。
b. 下矢印を押して次のスクリプトステップを選び、[Enter] や［return］キーを押します。

```
レイアウト切り替え
```

c. そのままスペースバーを押すか、「元のレイアウト」をクリックすると表示されるポップアップメニューから「計算によるレイアウト名 ...」を選択します。
d. ［計算式の指定］ダイアログが表示されます。
e. 関数を入力します。

```
Get( スクリプト引数 )
```

f. オプションの「アニメーション」は何も変更しません。
g. スクリプトの完成形は次の内容です。

```
レイアウト切り替え [ Get ( スクリプト引数 ) ]
```

h. ［OK］をクリックして［スクリプト指定］ダイアログへ戻ります。

(5)　スクリプト引数を指定します。

a.　［オプションのスクリプト引数：］に切り替え先のレイアウト名である「**見積リスト**」と入力します。

b.　［OK］をクリックして［ボタンバー設定］ダイアログへ戻ります。

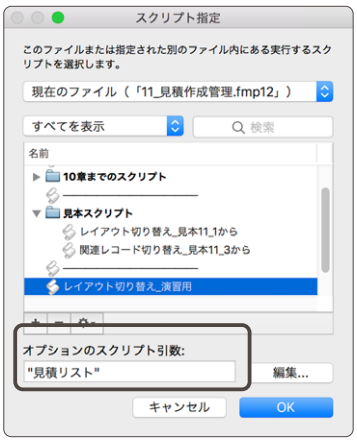


図 11-20 表示されるダイアログ

(6)　「見積管理」ボタンにスクリプトと引数を指定しました。

(7)　引き続き、他のセグメントにも作業を行います。

a.　［ボタンバー設定］ダイアログの［>］ボタンをクリックすると、作業対象のセグメントが切り替ります。

b.　次のように設定します（手順は省略します）。

「顧客管理」セグメント
・割り当てるスクリプト
「レイアウト切り替え _ 演習用」スクリプト
・設定する引数
「顧客管理｜マスタ詳細」

「商品管理」セグメント
・割り当てるスクリプト
「レイアウト切り替え _ 演習用」スクリプト
・設定する引数は、
「商品管理｜マスタ詳細



図 11-21 セグメントの切り替え

(8)　レイアウトを保存してブラウズモードに戻ります。

(9)　動作を確認します。
正しくレイアウトに切り替えできるかボタンバーの 3 つのボタンをクリックしてください
（遷移後のレイアウトで家アイコンのボタンをクリックすると「ホーム」レイアウトへ戻れます）。

演習ここまで

文字列のスクリプト引数
［編集 ...］をクリックせずにスクリプト引数に文字列を直接入力して指定すると「"」が自動で設定されます。

画面の遷移ができるようになりました。スクリプト引数と、Get (スクリプト引数) 関数を使用して複数のレイアウト上で動作します。なお、「見積リスト」、「顧客管理｜マスタ詳細」、「商品管理｜マスタ詳細」レイアウトの「上部ナビゲーション」パートのボタンバーには、すべてのセグメントに「レイアウト切り替え _ 見本 11_1 から」スクリプトを割り当てました。引数が違うだけで同様に動作します。また「見積リスト」レイアウトの「>」ボタンも同じです。

引数に使っている名称の変更には注意
レイアウト名を変更しても引数は連動しません。切り替え先のレイアウトが見つからないため、スクリプトエラーになります。レイアウト名やレイアウトオブジェクト名などを変更する場合、スクリプト内やスクリプト引数で使われていないか確認してください。

11.4.4 ［ウインドウの調整］スクリプトステップ

ウインドウの大きさを調整するものです。このスクリプトステップを使ってスクリプトを改善します。次の作業を行ってください。

(1)「ホーム」レイアウトで［見積管理］ボタンをクリックします。

「見積リスト」レイアウトでのウインドウは「ホーム」レイアウトと同じ大きさのままです。

(2) 任意の行の右端にある［>］ボタンをクリックし「見積詳細」レイアウトに切り替えます。
(3)「見積詳細」レイアウトで左上にある「見積リスト」ボタンをクリックして「見積リスト」レイアウトに切り替えます。すると、ウインドウは縦に長くなります。

(2)(3) を繰り返すと、レイアウトによりウインドウの大きさが変わることがわかります。これは「レイアウト切り替え _ 見本 11_1 から」スクリプトに［ウインドウの調整］スクリプトステップが記述されているからです。

「レイアウト切り替え _ 演習用」スクリプトでも**［ウインドウの調整］スクリプトステップ**を使います。

演習 11-2 スクリプト引数でレイアウトを切り替える 2

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1) ［スクリプトワークスペース］を開いてください

(2) **「レイアウト切り替え _ 演習用」スクリプト**を編集します。
   a. スクリプトパネルで、「レイアウト切り替え _ 演習用」スクリプトをクリックします。
   b. ［レイアウト切り替え］の下に改行を何回か入力します。
   c. 10 行目ぐらいに次のスクリプトステップを入力します

```
ウインドウの調整
```

   d. 「元に戻す」をクリックして表示されるポップアップメニューから「収まるようにサイズ変更」を選択します。

(3) スクリプトを保存して閉じてください。スクリプトの完成形は次の内容です。

```
レイアウト切り替え [ Get ( スクリプト引数 ) ]

ウインドウの調整 [ 収まるようにサイズ変更 ]
```

(4) 動作を確認します。
   a. 「ホーム」レイアウトで「見積管理」ボタンを押すと「見積リスト」レイアウトでウインドウサイズが変わります。
   b. 「ホーム」ボタンを押して戻ります。
   c. 「顧客管理」、「商品管理」ボタンで切り替えるとウインドウサイズは変わりません。

演習ここまで

オプションで「収まるようにサイズ変更」を選択すると表示形式やレイアウトエリアのサイズに合わせてウインドウサイズが調整されます。

なお、ウインドウを調整するかどうかはユーザによって好みが分かれる場合がありますので実際に使ってもらい、希望を確認します。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_3 から 7」フォルダの中の「11_ 見積作成管理 .fmp12」ファイルをベースに進めます）

11.4.5 ［関連レコードへ移動］スクリプトステップ

現在／対象レコードの、関連レコードへ移動するものです。このスクリプトステップは、

・どの TO（参照元）から

スクリプト内の現在のレイアウトです。そして、次のことも重要です。

・どの TO（参照先）の関連レコード（とレイアウト）へ移動するか

オプションの設定について確認します。

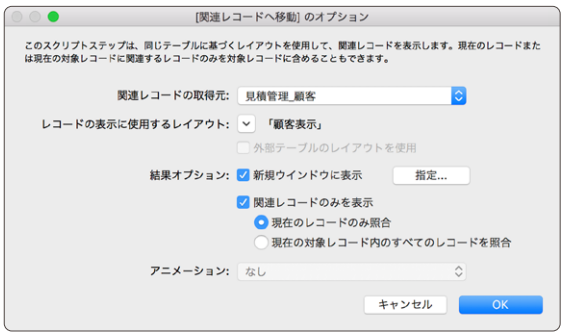


図 11-22 スクリプトのオプション設定ダイアログ

(1) スクリプト内の「現在のレイアウトの TO」から「どの TO 関連レコードへ移動したいか」を選択 (この TO 間には関連が設定されていなければなりません)
(2) 移動後に関連レコードを表示するレイアウト（(1) で指定した TO と同じ実体テーブルを持つ TO が指定されているレイアウトから選択）

次に結果についてを設定します。

(3) (2) で指定したレイアウトを、現在のウインドウに表示するか新しいウインドウに表示するかを選択
(4) 現在のレイアウトの現在のレコードの関連レコードを表示するか、現在の対象レコードすべての関連レコードを表示するかを選択

このスクリプトステップを使って、別の画面遷移を作成します。例えば「見積詳細」レイアウトから関連レコードを「顧客表示」レイアウトで表示するには、

(1) いつ　　ユーザの任意
(2) どこで　　「見積詳細」レイアウト
(3) 誰が　　ユーザ
(4) なにを　　現在表示している見積先の「顧客情報」レコード
(5) なぜ　　データの確認
(6) どのように　関連レコードが対象レコードの「顧客表示」レイアウトへ切り替える

スクリプトの流れは、次のようになります。

(1) ボタンをクリックする
(2) スクリプトが実行されて指定した TO の関連レコードと、指定した TO の実体テーブルの TO が設定されているレイアウトへ切り替える

ポータルの任意の行や、現在のレコードから「関連レコード」を対象レコードにしつつレイアウトの切り替えを可能にします。

演習ファイルの「見積詳細」「顧客管理」レイアウトの 4 箇所（次図）です。ボタンのうちいくつかには、演習用のスクリプトを割り当て済みです。なお、顧客情報を 1 つずつ表示する「顧客表示」レイアウトを切り替え先として作成しました。

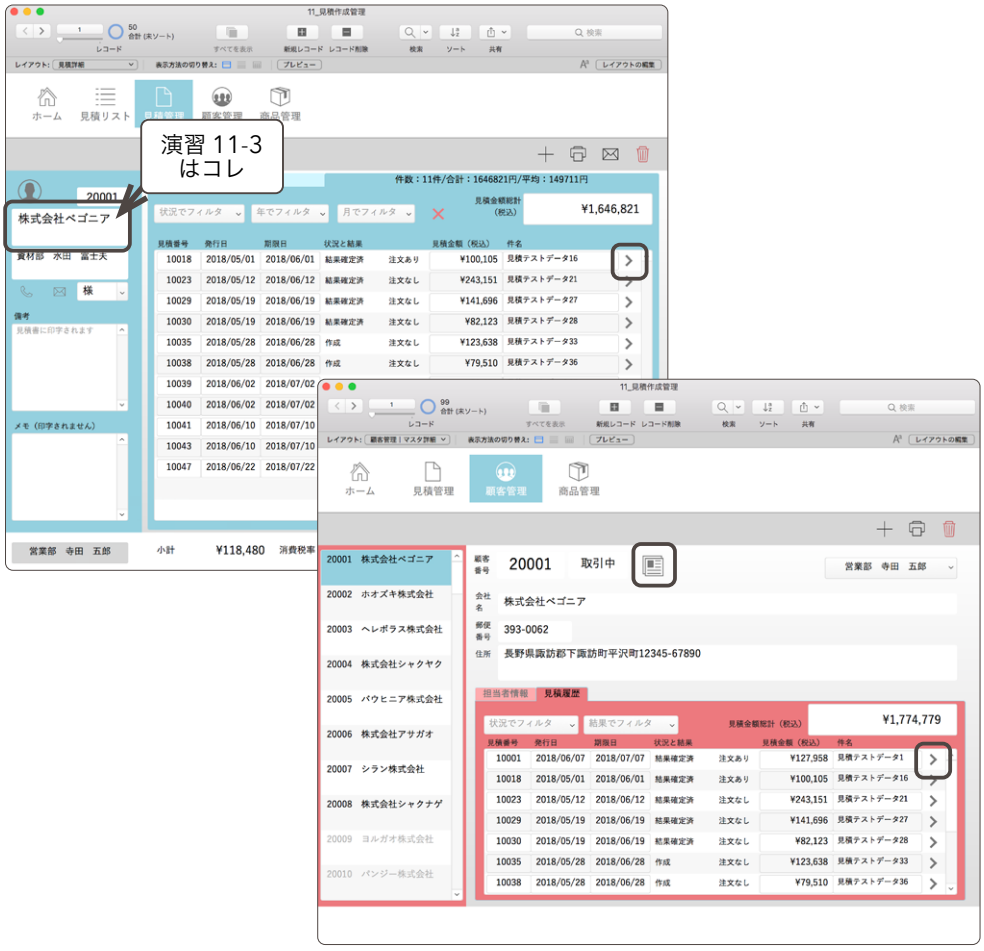


図 11-23 関連レコードを対象にしつつレイアウトを切り替えたい場所

「見積詳細」レイアウトの現在のレコードの「見積先顧客情報（関連レコード）」へ移動するスクリプトの演習をおこないます。

「演習 11_3 から 7」フォルダ

演習 11-3　関連レコードへ移動する 1：見積から顧客（レイアウト切り替え）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　どの TO の関連レコードへ移動するのか確認します。

(2)　［データベースの管理］ダイアログを開きます。
「見積詳細」レイアウトの TO は、「見積管理」TO です。
「会社名」のボタンで移動したい関連レコードは、
見積先の「顧客情報」テーブルのレコード＝「見積管理 _ 顧客」TO です。

なお、切り替えたいレイアウトは「顧客表示」レイアウトです。

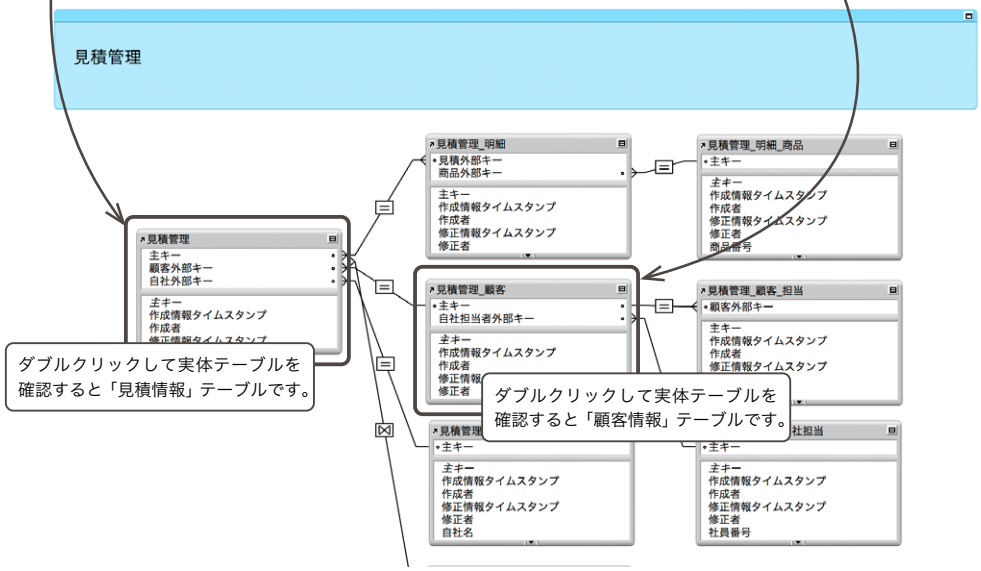


図 11-24　どの関連レコードへ移動するか

(3)　［キャンセル］をクリックして［データベースの管理］ダイアログを閉じます。

(4)　「見積詳細」レイアウトで、レイアウトモードに切り替えます。

「関連レコード切り替え _ 演習用」スクリプトを作成しておきました。「見積詳細」レイアウトの現在レコードの「見積先顧客情報（関連レコード）」へ移動するスクリプトとして使います。

(5)　フィールドをボタンにします。
a.　前ページ図の囲った部分、「見積管理 _ 顧客 :: 会社名」フィールドを右クリックします。
b.　「ボタン設定」を選択します。
c.　［ボタン設定］ダイアログが表示されます。

d. ［処理：］の下をクリックして［スクリプト実行］を選択します。
e. ［スクリプト指定］ダイアログが表示されます。
f. **「関連レコード切り替え _ 演習用」スクリプト**を選択します。
g. 歯車アイコンをクリックして［編集 ...］をクリックします。
h. スクリプトが表示されます。

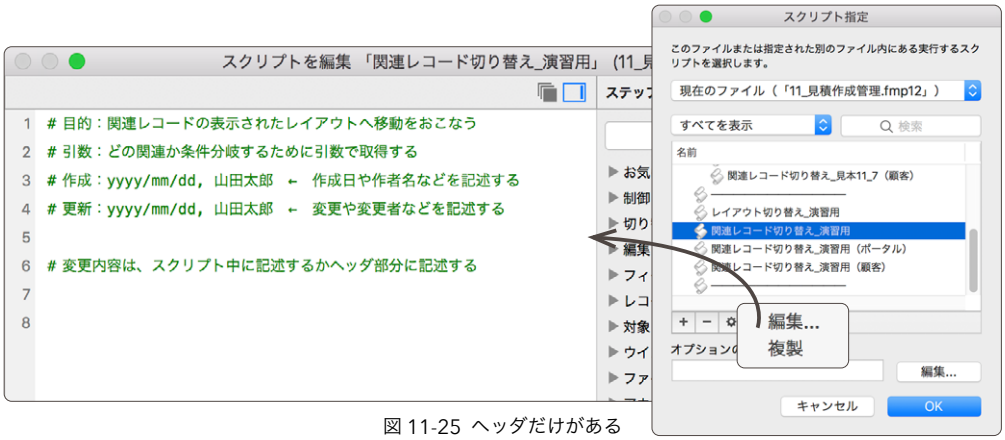


図 11-25 ヘッダだけがある

(6) スクリプトを編集します。
a. スクリプトにはヘッダ情報だけで、なにも内容はありません。
b. 8 行目をクリックして「かんれん」と入力すると、［関連レコードへ移動］スクリプトステップだけが表示されます。
c. ［Enter］や［return］キーを押して選択します。
d. そのままスペースバーを押します。

e. ［関連レコードへ移動］スクリプトステップのオプションを設定します。
・［関連レコードの取得元］は　「見積管理 _ 顧客」
・［レコードの表示に使用するレイアウト］は　「顧客管理｜マスタ詳細」
・［アニメーション］は　そのまま
・［結果オプション：］は「新規ウインドウに表示」を **オフ**　「関連レコードのみを表示」を **オン** で「現在のレコードのみ照合」をクリック

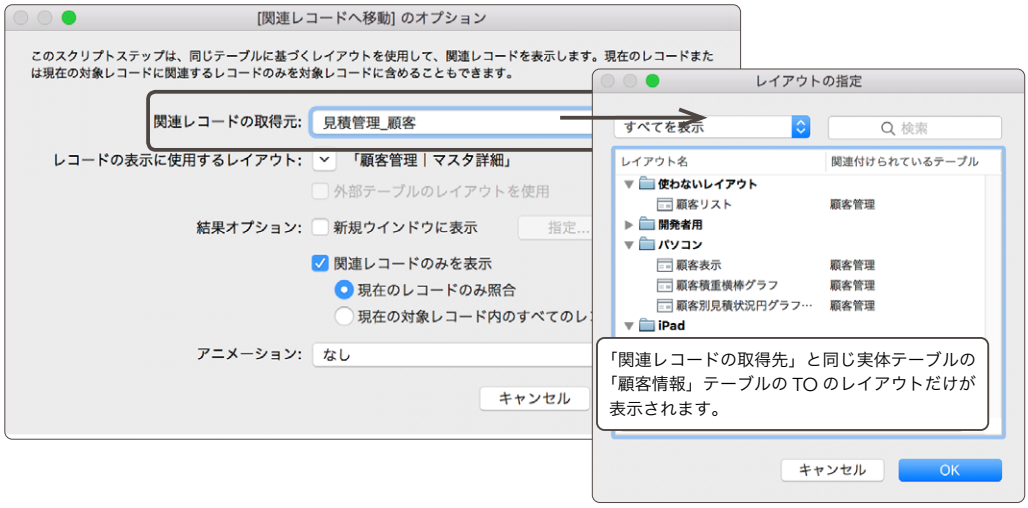


図 11-26［関連レコードへ移動］スクリプトステップオプションの設定内容

f. ［OK］をクリックしてオプションを保存します。
g. スクリプトの完成形は次の内容です。スクリプトを保存します。

```
関連レコードへ移動 [ テーブル：「見積管理_顧客」; 使用するレイアウト：
「顧客管理｜マスタ詳細」 ( 顧客管理 ) ]
```

(7) ブラウズモードに切り替えて動作を確認します。
a. 1 レコード目に移動します。見積先の会社名は「株式会社ベゴニア」です。
b. 設定した会社名ボタン（「見積管理 _ 顧客 :: 会社名」フィールド）をクリックします。
c. 「顧客管理｜マスタ詳細」レイアウトへ切り替えます。対象レコードは、1 件です。
d. 会社名は「株式会社ベゴニア」です。

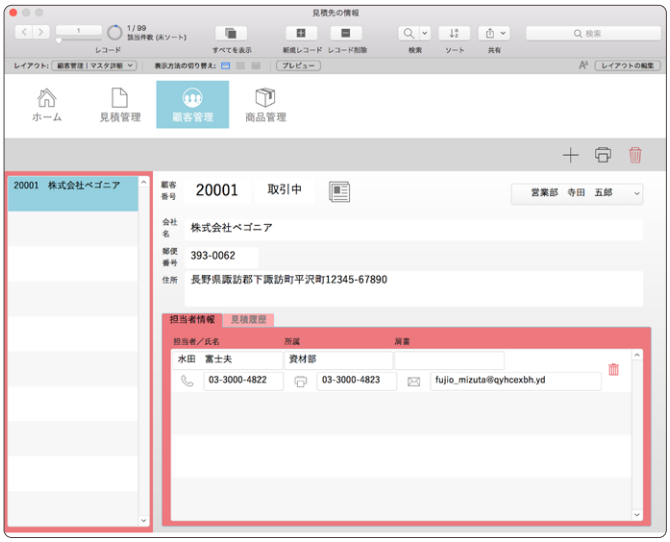


図 11-27「顧客管理｜マスタ詳細」レイアウト

(8) デバッガとデータビューアを表示して確認します。
ステップインで 1 行ずつ確認します。

演習ここまで

関連レコードへ移動できました。顧客の詳細情報が確認できます。

ユーザが迷わないように
注意点があります。関連レコードを表示しながらレイアウトを切り替えるとユーザは、

・いま自分が画面遷移のどこにいるのか

が、よくわからなくなってしまうことがあります。または、

・関連（参照先）の情報を、ちょっと確認したら、また元のレイアウトに戻りたい

と、考えているかもしれません。ユーザがどのように操作したいか、を考慮します。

関連レコードへ移動後も作業が続くのであれば関連レコードへ移動したままとします。しかし、『ちょっと確認して戻る』のであれば、カードウインドウを利用するとユーザにわかりやすい動作になります。

カードウインドウ
アクティブなウインドウ（親ウインドウ）の現在のレイアウトの一番手前に表示されるタイプのウインドウです。なお、他のウインドウを選択して作業することができます。親ウインドウで作業するにはカードウインドウを閉じる必要があります。

・FileMaker Pro では、ステータスツールバー、書式設定バー、タイトルバー、ウインドウの枠は表示されません。

・FileMaker Go でカードのメニューバーやツールバーを表示／非表示するには、カードを表示する前にバーを表示／非表示する必要があります。

関連レコードをカードウインドウで表示するようにスクリプトの動作を変更します。「顧客表示」レイアウトをつかいます。

演習 11-4　関連レコードへ移動する 2：見積から顧客（カードウインドウ表示）

• **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］を表示します。

(2)　**「関連レコード切り替え _ 演習用」スクリプト**を編集します。
a.　スクリプトパネルで、「関連レコード切り替え _ 演習用」スクリプトをクリックします。
b.　［関連レコードへ移動］スクリプトステップのオプションを変更します。

・［結果オプション：］は「新規ウインドウに表示」を　**オン**
「関連レコードのみを表示」を　**オン**

［新規ウインドウ］のオプション　ダイアログが表示されます。
・［ウインドウスタイル：］は　「カード」
・［ウインドウ名：］は　「見積先の顧客情報」
・［レイアウト：］は　すでに選択済み
・［サイズ：］は　そのまま
・［位置：］は　「上端から」を「0」「左端から」は、なにも設定しない
・［ウインドウオプション］は　「閉じる」「親ウインドウ淡色表示」の 2 つを　**オン**
［OK］をクリックしてダイアログを閉じます。

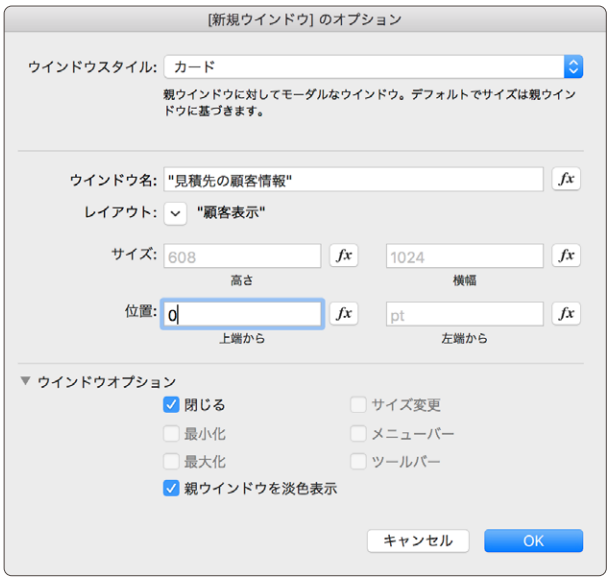


図 11-28［新規ウインドウ］のオプション　ダイアログ

c. ［OK］をクリックして［関連レコードへ移動］スクリプトステップのオプションダイアログを閉じます。

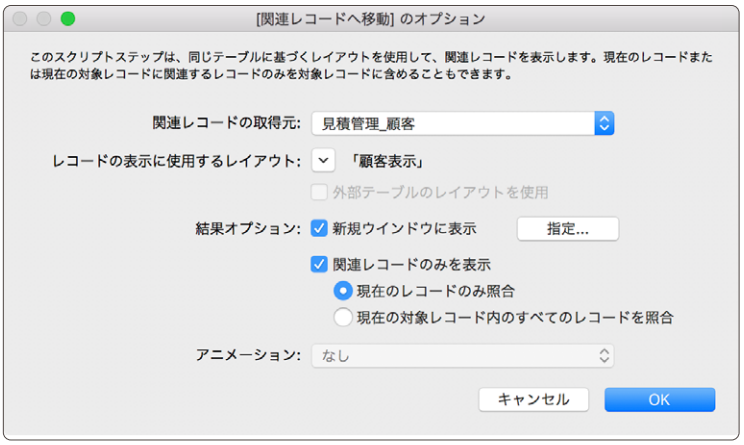


図 11-29［関連レコードへ移動］スクリプトステップオプションの設定内容

d. スクリプトを保存します。スクリプトの完成形は次の内容です。

```
関連レコードへ移動 [ テーブル：「見積管理_顧客」; 使用するレイアウト：
「顧客表示」 ( 顧客管理 ) ] [ 関連レコードのみを表示; 新規ウインドウ ]
```

(3) 「見積詳細」レイアウトで動作を確認します。

a. 1 レコード目に移動します。見積先の会社名は「株式会社ベゴニア」です。

b. 会社名ボタンをクリックします。

c. 「顧客表示」レイアウトがカードウインドウで表示されます。会社名は「株式会社ベゴニア」です。

d. ウインドウを縦横に広げて元のウインドウの上にカードウインドウが表示されていることを確認してください。
e. 左上にある白い丸に × のボタンをクリックすると、カードウインドウが閉じます。

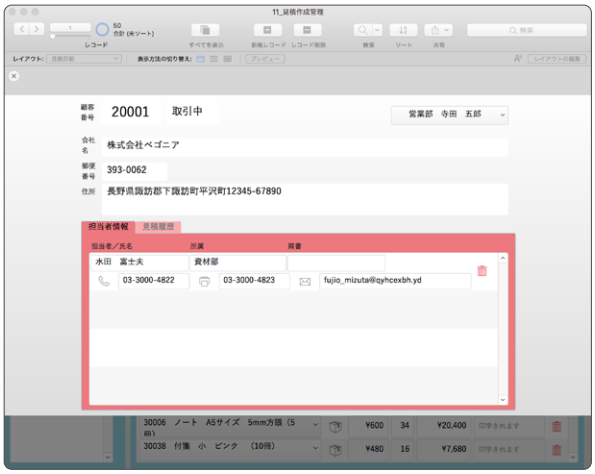


図 11-30「顧客表示」レイアウトがカードウインドウで表示、関連レコードが表示

演習ここまで

この方法なら、ユーザが『ちょっと確認して戻る』で画面遷移の迷子になることはありません。あるいは、関連先のレコードを、現在のウインドウとは別のウインドウで表示することも可能です。**関連レコードを別のウインドウで表示するようにスクリプトの動作を変更**します。

演習 11-5　関連レコードへ移動する 3：見積から顧客（ドキュメントウインドウ表示）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］を表示します。

(2)　**「関連レコード切り替え _ 演習用」スクリプト**を編集します。
a. ［スクリプトワークスペース］を表示します。
b. スクリプトパネルで、「関連レコード切り替え _ 演習用」スクリプトをクリックします。
c. ［関連レコードへ移動］スクリプトステップオプションを変更します。

新規ウインドウに表示の［指定 ...］をクリックします。

［新規ウインドウ］のオプション　ダイアログが表示されます。
・［ウインドウスタイル：］は　「ドキュメント」
・［位置：］は　「上端から」「左端から」は、なにも設定しない
・［ウインドウオプション］は　「閉じる」を　**オン**
［OK］をクリックしてダイアログを閉じます。

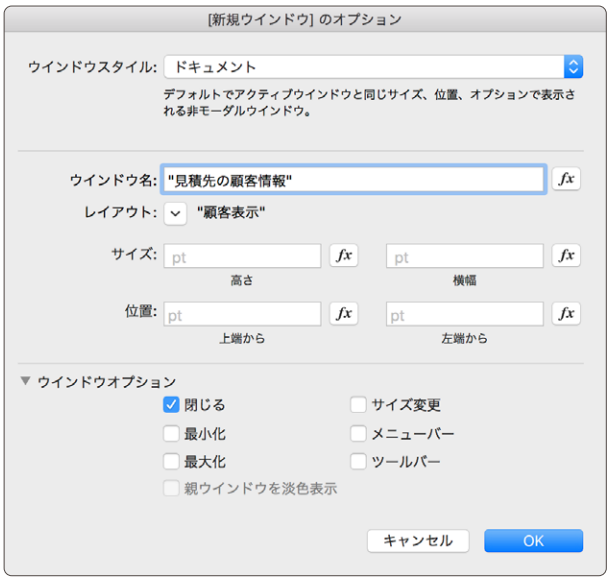


図 11-31［新規ウインドウ］のダイアログ

d. ［OK］をクリックして［関連レコードへ移動］スクリプトステップのオプションダイアログを閉じます。

e. スクリプトを保存します。スクリプトの完成形は次の内容です。

```
関連レコードへ移動 [ テーブル：「見積管理_顧客」； 使用するレイアウト：
「顧客表示」 ( 顧客管理 ) ] [ 関連レコードのみを表示； 新規ウインドウ ]
```

(3) 「見積詳細」レイアウトで動作を確認します。

a. 1 レコード目に移動します。見積先の会社名は「株式会社ベゴニア」です。

b. 会社名ボタンをクリックします。

c. 「顧客表示」レイアウトが別のウインドウで表示されます。会社名は「株式会社ベゴニア」です。

d. ウインドウを閉じます。

演習ここまで

顧客情報が別のウインドウで表示できるようになりました。ウインドウスタイルは、状況に応じて選択をします。

ウインドウの数

別のウインドウを表示する場合、注意が必要です。例えば、演習 11-5 の「ドキュメント」ウインドウ形式では、複数のウインドウがあるとユーザは自由に選択できます。状況によってはユーザが、

・いまどのウインドウを操作しているか

がわからなくなるかもしれません。複数のウインドウが開かれたままでも大丈夫かどうか、などをあらかじめ考えておく必要があります。「フローティングドキュメント」「ダイアログ」のタイプを選択する場合でも、それぞれの特徴を考慮します。

「見積詳細」レイアウトの現在のレコードの「見積先顧客情報が同じ別のレコード」（関連レコード）へ移動するスクリプトの演習をおこないます。2 つ目のタブパネルにある見積履歴ポータルの上です。

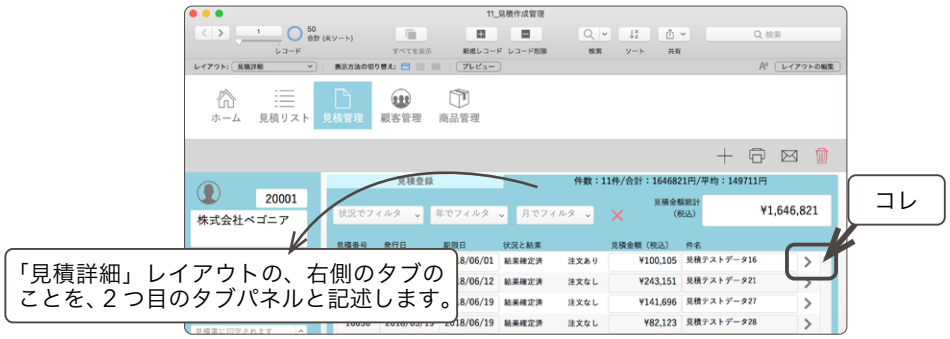


図 11-32 演習 11-6 で作業するボタンとなるフィールド

演習 11-6　関連レコードへ移動する 4：見積から見積（自己リレーション／レイアウト切り替え）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　どの TO の関連レコードへ移動するのか確認します。

(2)　［データベースの管理］ダイアログを開きます。
「見積詳細」レイアウトの TO は、「見積管理」TO です。
関連レコードへ移動したいポータルの「>」ボタンで移動したい先は、
見積先の「顧客情報」テーブルのレコード＝「見積管理｜履歴｜自己」TO です。

なお、切り替えたいレイアウトは、同じ「見積詳細」レイアウトです。

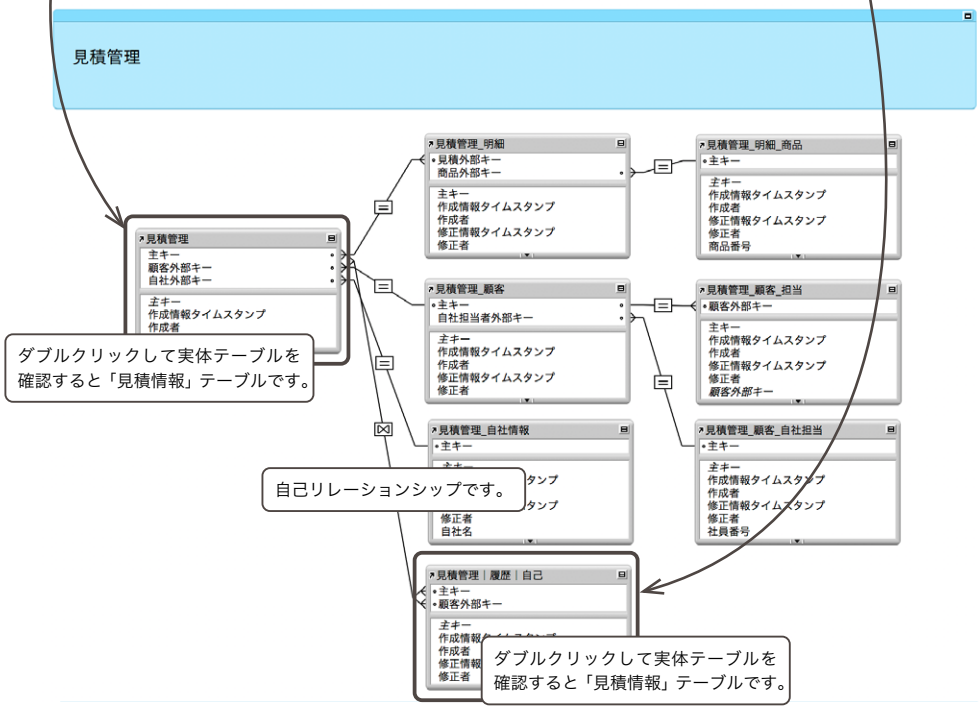


図 11-33 どの関連レコードへ移動するか

(3)　［キャンセル］をクリックして［データベースの管理］ダイアログを閉じます。

「見積詳細」レイアウトの現在のレコードの「見積先顧客情報が同じ別のレコード」（関連レコード）へ移動するスクリプトとして、「関連レコード切り替え _ 演習用（ポータル）」スクリプトを作成して左図の「>」ボタンに割り当てておきました。これを使います。

(4)　［スクリプトワークスペース］を表示し**「関連レコード切り替え _ 演習用（ポータル）」スクリプト**を編集します（詳細な手順は省略します）。

a.　8 行目に次のスクリプトステップを入力します。

```
関連レコードへ移動
```

オプションを設定します。

・［関連レコードの取得元］は　「見積管理｜履歴｜自己」
・［レコードの表示に使用するレイアウト］は　< 現在のレイアウト >　そのまま
・「関連レコードのみを表示」を　**オン**
・「現在のレコードのみ照合」をクリック　そのまま
・［アニメーション］は　そのまま
・［結果オプション：］は「新規ウインドウ表示」を　**オフ**　「関連レコードのみを表示」を　**オン**　で「現在のレコードのみ照合」をクリック

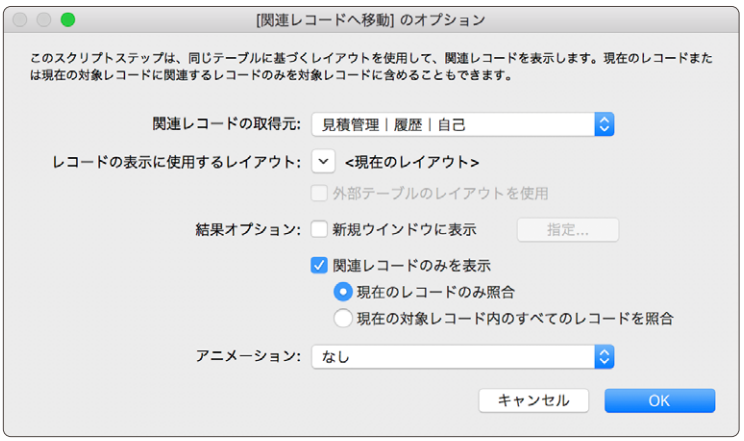


図 11-34［関連レコードへ移動］スクリプトステップオプションの設定内容

b.　［OK］をクリックしてオプションを保存します。
c.　スクリプトを保存します。

スクリプトの完成形は次の内容です。

```
関連レコードへ移動　[　テーブル：「見積管理｜履歴｜自己」；使用するレイアウト：<現在のレイアウト>　]
```

(5)　「見積詳細」レイアウトに切り替えて動作を確認します。
(時間があればレイアウトモードで「>」ボタンに割り当てられているスクリプトが「関連レコード切り替え _ 演習用（ポータル）」であることを確認してください)

a.　ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。
b.　3 レコード目の「見積番号 10003」に移動します。
c.　2 つ目のタブパネルをクリックします。
d.　ポータルには 5 行が表示されています。
e.　ポータル行の 2 行目「見積番号 10021」の「>」ボタンをクリックします。
f.　(レコードが切り替ります)
g.　タブパネルの「見積登録」タブをクリックすると「見積番号 10021」が表示されているはずです。
h.　対象レコードは 5 件に絞り込まれました（ポータルの表示と同じ件数です)。

(6)　関連レコードへ移動できました。

移動できましたが、タブパネルが切り替らないのですぐに見積番号を確認できません。関連レコードへタブパネルが変わるように変更します。

タブパネルに名前（オブジェクト名）を付けて移動できるようにします。

(7)　「見積詳細」レイアウトで、レイアウトモードに切り替えます。

(8)　タブにオブジェクト名を付けます。
a.　左パネルの「オブジェクト」タブを漏斗のマークをクリックして「パネルコントロール」を選択してフィルタします。
b.　「見積登録」を右クリックしてショートカットメニューで「オブジェクト名の指定」を選択します。
c.　「見積情報タブ」と入力し設定します。
d.　同じく「計算」は、「見積履歴タブ」と設定します。
e.　レイアウトを保存します。



図 11-35 オブジェクト名を設定

(9) ［スクリプトワークスペース］を表示し「関連レコード切り替え _ 演習用（ポータル）」スクリプトを編集します。

a. ［関連レコードへ移動］スクリプトステップの下に次のスクリプトステップを入力します。

```
オブジェクトへ移動
```

オプションを設定します。
・［オブジェクト名：］は 「見積情報タブ」

b. ［OK］をクリックしてオプションを保存します。
c. スクリプトを保存します。完成形は次の内容です。

```
関連レコードへ移動 [ テーブル：「見積管理｜履歴｜自己」; 使用するレイアウト：＜現在のレイアウト＞ ]

オブジェクトへ移動 [ オブジェクト名："見積情報タブ" ]
```

(10) 「見積詳細」レイアウトに切り替えて動作を確認します。

a. ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。
b. 3 レコード目の「見積番号 10003」に移動します。
c. 2 つ目のタブパネルをクリックします。
d. ポータルには 5 行が表示されています。
e. ポータル行の 2 行目「見積番号 10021」の「>」ボタンをクリックします。
f. レコードが切り替わり、タブも切り替って「見積番号 10021」が表示されているはずです。
g. 対象レコードは 5 件に絞り込まれました（ポータルの表示と同じ件数です）。

(11) デバッガとデータビューアを表示して確認します。
ステップインで 1 行ずつ確認します。

(12) どのレコードに移動したのかわかりやすくなりました。

(13) 最後にステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

演習ここまで

演習 11-3 から 6 で［関連レコードへの移動］スクリプトステップの移動方法をいくつか演習しました。機能を作成するとき、どのように移動したらユーザが作業しやすいか、を考えて適切な移動方法を選択します。

「顧客管理｜マスタ詳細」レイアウトの「現在の顧客データ」の「見積情報（関連レコード）」へ移動するスクリプトの演習を行います。次図の場所です。

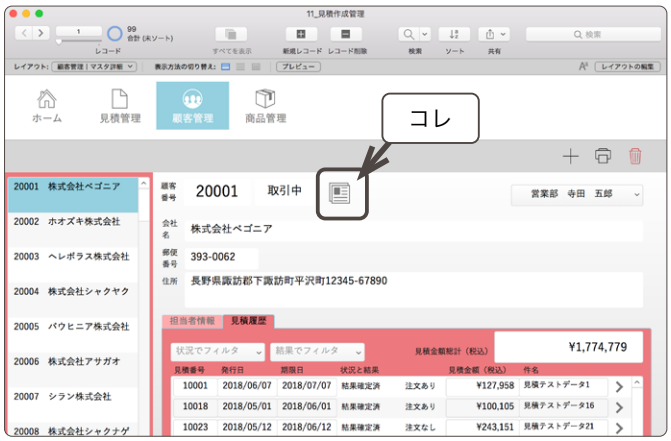


図 11-36 演習 11-7 で作業するボタンとなるフィールド

演習 11-7　関連レコードへ移動する 5：顧客から見積（レイアウト切り替え）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　どの TO の関連レコードへ移動するのか確認します。

(2)　［データベースの管理］ダイアログを開きます。
「顧客管理｜マスタ詳細」レイアウトの TO は、「顧客管理」TO です。
関連レコードへ移動したい書類アイコンボタンで移動したい先は、
顧客の「見積情報」テーブルのレコード＝「顧客管理 _ 見積」TO です。

なお、切り替えたいレイアウトは、「見積詳細」レイアウトです。

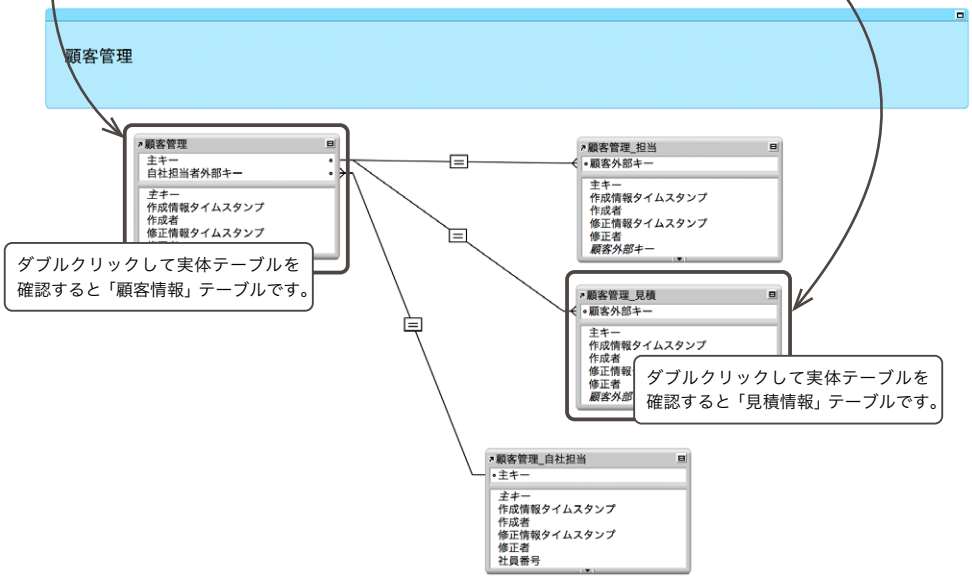


図 11-37 どの関連レコードへ移動するか

(3)　［キャンセル］をクリックして［データベースの管理］ダイアログを閉じます。

「顧客管理｜マスタ詳細」レイアウトの「現在の顧客データ」の「見積情報（関連レコード）」へ移動するスクリプトとして、「関連レコード切り替え _ 演習用（顧客）」スクリプトを作成して左図のボタンに割り当てておきました。これを使います。

(4)　［スクリプトワークスペース］を表示し**「関連レコード切り替え _ 演習用（顧客）」スクリプト**を編集します（詳細な手順は省略します）。

a.　8 行目に次のスクリプトステップを入力します。

関連レコードへ移動

オプションを設定します。

・［関連レコードの取得元］は　「顧客管理 _ 見積」
・［レコードの表示に使用するレイアウト］は　「見積詳細」
・「現在のレコードのみ照合」をクリック　そのまま
・［アニメーション］は　そのまま
・［結果オプション：］は「新規ウインドウ表示」を **オフ**　「関連レコードのみを表示」を **オン**　で「現在のレコードのみ照合」をクリック

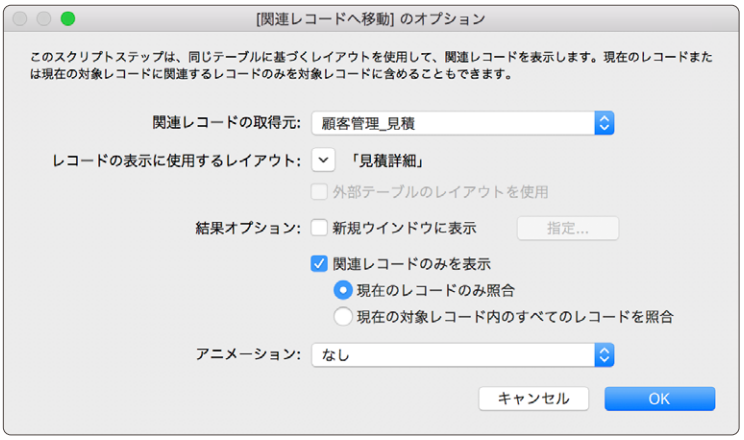


図 11-38［関連レコードへ移動］スクリプトステップオプションの設定内容

b.　［OK］をクリックしてオプションを保存します。
c.　スクリプトを保存します。

スクリプトの完成形は次の内容です。

関連レコードへ移動　[　テーブル：「顧客管理_見積」；　使用するレイアウト：「見積詳細」（見積管理）　]

(5)　「顧客管理｜マスタ詳細」レイアウトに切り替えます。

(6)　レイアウトモードで書類アイコンのボタンに割り当てられているスクリプトが「関連レコード切り替え _ 演習用（顧客）」であることを確認します。

(7)　ブラウズモードで動作を確認します。

a.　もし、全レコードが表示されていなければ、ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。
b.　2 レコード目の「顧客番号 20002」に移動します。
c.　「見積履歴」タブをクリックします。関連のレコードは 6 件です。
d.　書類アイコンのボタンをクリックします。
e.　「見積詳細」で「顧客番号 20002」の「見積番号 10003」レコードが表示されているはずです。
f.　対象レコードは 6 件です。表示されているのは 1 件目のレコードです。
g.　ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

「顧客管理｜マスタ詳細」レイアウトには、もう 1 つボタンがありました。「上部ナビゲーション」パートの「顧客管理」ボタンをクリックしてレイアウトを切り替えて確認します。

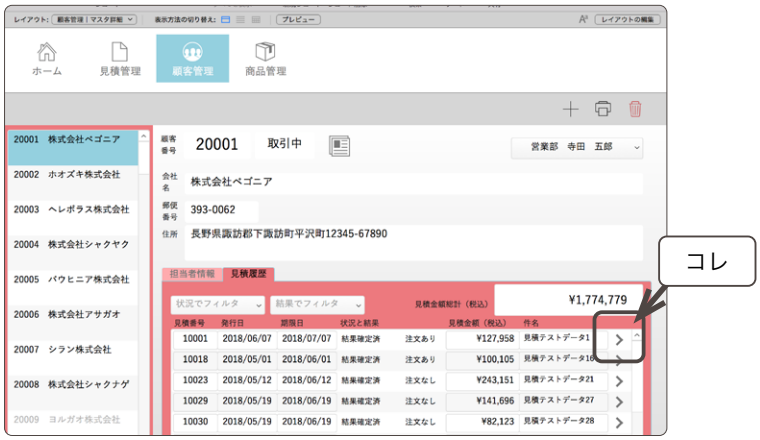


図 11-39 見積情報を表示するボタン（ポータル）

(8)　このボタンにも同様に「関連レコード切り替え _ 演習用（顧客）」スクリプトが割り当てられていることをレイアウトモードで確認してください。

(9)　ブラウズモードで動作を確認します。

a.　2 レコード目の「顧客番号 20002」に移動します。会社名は「ホオズキ株式会社」です。
b.　「見積履歴」タブをクリックすると、ポータルには 6 行が表示されています。
c.　ポータル行の 3 行目「見積番号 10021」の「>」ボタンをクリックします。
d.　「顧客番号 20002」の見積データ「見積番号 10021」レコードが表示されているはずです。会社名は「ホオズキ株式会社」です。
e.　対象レコードは 6 件です。表示されているのは 3 件目のレコードです。
f.　ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

(10)　デバッガとデータビューアを表示して確認します。
ステップインで 1 行ずつ確認します。

演習ここまで

1 つのレイアウトの 2 つのボタンに、「関連レコード切り替え _ 演習用（顧客）」スクリプトが割り当てられています。そして正しく動作していました。どちらのボタンも同じ目的で作成されているため、動作します。

・「顧客管理｜マスタ詳細」レイアウトの「現在の顧客データ」の「見積情報（関連レコード）」へ移動する

演習 11-7 で動作を確認した、

・「顧客番号 20002」のレコード

の、対象レコード数は同じ 6 件です。但し、表示されるレコードが異なります。

・書類アイコンボタンは「見積情報」レコードの作成順の 1 件目「見積番号 10003」
・ポータルの「>」ボタンはボタンをクリックした「見積番号 10021」

ボタンがどこに配置されるかにより表示されるレコードが異なることを覚えていてください。

演習で採り上げなかったオプションの「現在の対象レコードすべてのレコードを照合」は、例えば、次のような場合に便利です。

・2020 年 6 月に見積書を作成し注文があった顧客をすべて表示したい

何らかの形で対象レコードを作成し、その対象レコードの、すべての関連レコード（対象レコード）を表示すること、つまり、検索と組み合わせて使用することが多くなります。

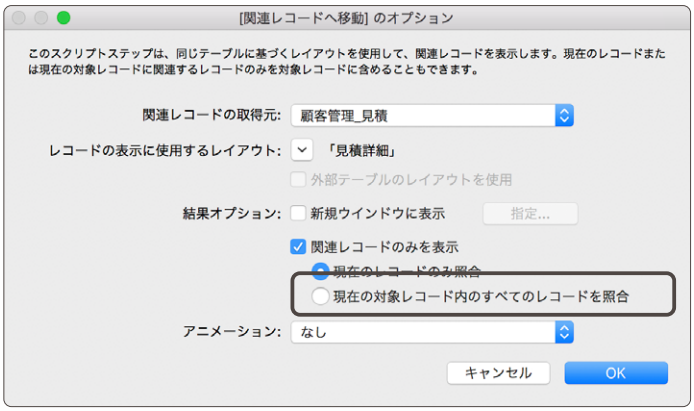


図 11-40 見積情報を表示するボタン（ポータル）

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_8」フォルダの中の「11_ 見積作成管理 .fmp12」ファイルをベースに進めます）

関連レコードへ移動することについてエラーになるテストをします。

(1)「顧客管理｜マスタ詳細」レイアウトで顧客番号「20006」の顧客を選択します。
(2) 書類アイコンのボタンをクリックします。
(3) なにも起きません。
(4)「見積履歴」タブをクリックします。
(5) ポータルにはレコードが表示されていません。この顧客には関連の見積がありません。

関連先にレコードが無い場合
関連レコードへ移動は、関連レコードが無い場合、ボタンを押しても、なにも起きません。

ポータルのボタン
ポータル上に配置されているボタンは関連レコードが無ければ表示されません。ボタンを押せません。

このままでも移動しないだけなので良いのかもしれません。しかし、スクリプトによっては関連レコードに移動後、ソートしたり、フィールドにデータを入力したり、レコードを削除したりするかもしれません。エラーが表示されたり、想定とは違うレコードを削除してしまうかもしれません。それを防ぐためには、

```
IsEmpty ( フィールド ) 関数
```

を使用します。**「エラーになる前に」エラー処理**を行います。

「演習 11_8」フォルダ

演習 11-8　関連レコードへ移動する 6：エラー処理

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　[スクリプトワークスペース] を表示し**「関連レコード切り替え _ 演習用（顧客）」スクリプト**を編集します。

(2)　準備します。
8 行目をクリックし［Enter］や［return］キーを押して 3 行ほど空行を入力します。

(3)　入力を開始します。
a.　10 行目位で「if」と入力すると候補のスクリプトステップが表示されます。
b.　矢印と［Enter］や［return］キーを押して［If］選択します。
c.　［If］と［End If］がセットで表示されます。
d.　［If］行に計算式を指定します。

```
IsEmpty ( 顧客管理_見積::主キー )
```

e.　スクリプトは次の状態です。

```
関連レコードへ移動 [ (省略) ]

If [IsEmpty ( 顧客管理_見積::主キー )]
End If
```

f. ［If］行の下に［Enter］や［return］キーを押して空行をいれます。
g. 「else」と入力して次の状態にします。

```
関連レコードへ移動 [ （省略） ]

If [IsEmpty ( 顧客管理_見積::主キー )]
Else
End If
```

h. ［関連レコードへ移動］スクリプトステップを［Else］スクリプトステップの後ろにドラッグして移動します。

```
If [ IsEmpty ( 顧客管理_見積::主キー ) ]
Else
  関連レコードへ移動 [ （省略） ]
End If
```

関連レコードが無いことがわかりやすいようにします。

i. ［If］行をクリックして［Enter］や［return］キーを押して次のスクリプトステップを入力します。

```
[カスタムダイアログを表示]
```

オプションを設定します。

・［タイトル：］は　「見積なし」
・［メッセージ：］は　「見積はありません」
・「デフォルトボタン」は　OK　［データを確定］のチェック　**オフ**
・「ボタン 2」は " キャンセル " を消去して［データを確定］のチェック　**オフ**
・「ボタン 3」は　そのまま

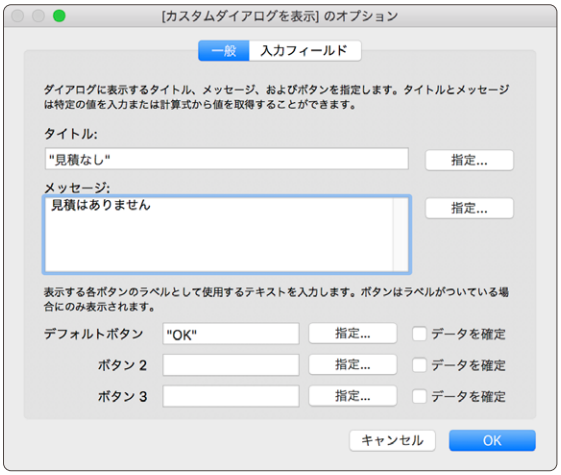


図 11-41［カスタムダイアログを表示］スクリプトステップオプションの設定内容

j. ［OK］をクリックしてオプションを保存します。
k. スクリプトを保存します。

スクリプトの完成形は次の内容です。

```
If [ IsEmpty ( 顧客管理_見積::主キー ) ]
  カスタムダイアログを表示 [ "見積なし" ; "見積はありません" ]
Else
  関連レコードへ移動 [ テーブル:「顧客管理_見積」; 使用するレイアウト:
  「見積詳細」( 見積管理 ) ]
End If
```

(4) 「顧客管理｜マスタ詳細」レイアウトに切り替えて動作を確認します。
a. 6 レコード目の「顧客番号 20006」に移動します。会社名は「株式会社アサガオ」です。
b. 「見積履歴」タブをクリックします。ポータルにはなにも表示されていません。
c. 書類アイコンのボタンをクリックします。
d. カスタムダイアログが表示されます。
e. ［OK］をクリックしてダイアログを閉じます。
f. 2 レコード目の「顧客番号 20002」に移動します。会社名は「ホオズキ株式会社」です。
g. 「見積履歴」タブをクリックします。関連のレコードは 6 件です。
h. 書類アイコンのボタンをクリックします。
i. 「見積詳細」で「顧客番号 20002」の「見積番号 10003」レコードが表示されているはずです。会社名は「ホオズキ株式会社」です。
j. 対象レコードは 6 件に絞り込まれました（ポータルの表示と同じ件数です）。
k. ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

(5) デバッガを表示して確認します。
（[ツール]–[デバッガ］メニューを選択して表示します）
a. ステップインで 1 行ずつ確認します。
b. 正しく分岐しているか確認してください。

(6) エラーが回避され正しく動作するようになりました。エラー処理できました。

演習ここまで

演習 11-8 では関連レコードが無いことがわかりやすいようにカスタムダイアログを表示しました。ユーザの立場では『関連の見積情報が無いならなにも起きなくても良い』かもしれません。毎回[OK]をクリックするのが面倒に感じることがあるためです。関連レコードへ移動を利用した機能では、関連レコードの有無によるその後の流れも考えて作業します。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_9 から 11」フォルダの中の「11_見積作成管理 .fmp12」ファイルをベースに進めます）

11.5　条件分岐、変数、引数

スクリプトでは、演習 11-8 で利用した［If］スクリプトステップなどを使って「**条件分岐**」をおこなうことが多々あります。スクリプトを実行する内容を文字通り「条件で分岐」することです。分岐する条件は、

・命令に必要なデータによる分岐（引数、変数、フィールドなどのデータ）
・スクリプトとユーザの対話による分岐（カスタムダイアログでユーザが選ぶ）

です。いずれも、論理値として評価されるように［If］あるいは［Else If］スクリプトステップに計算式を記述します。前の演習でも IsEmpty 関数を使って論理値で条件を分岐しました。

```
If [ IsEmpty ( 顧客管理_見積::主キー ) ]
  カスタムダイアログを表示 [（省略）]
Else
  関連レコードへ移動 [（省略）]
End If
```

［If］スクリプトステップに設定された計算式の結果が

・「1（真 = True ）＝関連レコードがない　　＝カスタムダイアログを表示
・「1 以外」（偽 =False)」＝関連レコードがある　＝関連レコードへ移動する

というものでした。

（この節では「データによる分岐」を演習で取り扱います）

11.5.1　［If］、［Else If］、［Else］、［End If］スクリプトステップ

関数の If と構造はほぼ同じです。計算式の結果は論理値として返ります。

・「1（真 = True)」であるかどうか

で実行する内容が決まります。

```
If [ ここに条件（計算式）を設定する ]
  条件（計算式）の結果が「1（真 = True)」だったら実行する内容
End if
```

これが最もシンプルな使用方法です。

```
If [ ここに条件（計算式）を設定する ]
  条件（計算式）の結果が「1（真 = True)」だったら実行する内容
Else
  条件（計算式）の結果が「1以外」（偽 = False）だったら実行する内容
End if
```

これが一番よく使用される方法です。

さらに Else If は、

```
If [ ここに条件（計算式）を設定する ]
  条件（計算式）の結果が「1（真 = True)」だったら実行する内容
Else If [ 2番目の論理計算式 ]
  2番目の条件（計算式）の結果が「1（真 = True)」だったら実行する内容
Else If [ 3番目の論理計算式 ]
  3番目の条件（計算式）の結果が「1（真 = True)」だったら実行する内容
Else If [ 4番目の論理計算式 ]
  4番目の条件（計算式）の結果が「1（真 = True)」だったら実行する内容
Else
  これまでの条件（計算式）すべてが「1以外」(偽=False）だったら実行する内容
End if
```

このような形で、[Else If] スクリプトステップはいくつでも挟めます。複数を設定すると、最初に 1（真 = True ）に一致した条件（計算式）の後に記述されているスクリプトステップを実行します。Case 関数に似ています。

さて、本章では、次のスクリプトについて作業してきました。

・「関連レコード切り替え _ 演習用」　→「見積詳細」の会社名ボタン
・「関連レコード切り替え _ 演習用（ポータル)」→「見積詳細」のポータルの「>」ボタン
・「関連レコード切り替え _ 演習用（顧客)」　→「顧客管理｜マスタ詳細」の書類アイコンボタンとポータルの「>」ボタン

別々のスクリプトのままでも間違いではありませんが、**条件分岐を利用して 1 つのスクリプトにします。**演習ファイルには、「関連レコード切り替え _ 演習用 (分岐)」スクリプトを作成しておきました。

スクリプト引数を使って条件分岐を行います。引数は演習 11-1 のような、レイアウト名などの命令に直結するものだけでなく「条件分岐」のためにも使用します。3 つのスクリプトを 1 つとして考えると、

(1) いつ　ユーザの任意
(2) どこで　「見積詳細」または「顧客管理｜マスタ詳細」レイアウト
(3) 誰が　ユーザ
(4) なにを　「見積詳細」または「顧客管理｜マスタ詳細」レイアウトの現在のレコード
(5) なぜ　データの確認
(6) どのように　関連レコードへ移動（必要な場合レイアウトも切り替え）

この場合、引数にするべきなのは、次の箇所です。

(1) いつ　ユーザの任意
(2) どこで　**スクリプトを指定して　←ここ**
(3) 誰が　ユーザ
(4) なにを　「見積詳細」または「顧客管理｜マスタ詳細」レイアウトの現在のレコード
(5) なぜ　データの確認
(6) どのように　関連レコードへ移動（必要な場合レイアウトも切り替え）

「演習 11_9 から 11」フォルダ

演習 11-9　スクリプト引数でレイアウトを切り替える 1（スクリプトの作業）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］を表示し**「関連レコード切り替え _ 演習用（分岐）」スクリプト**を編集します。内容は次図です。条件分岐だけを行えるように記述しました。



図 11-42 スクリプトの内容

(2)　入力を開始します。

a.　8 行目位をクリックして［If］[End If] スクリプトステップを入力します。

b.　［If］行に計算式を指定します。

```
Get ( スクリプト引数 ) = "見積から顧客"
```

c.　［If］行の下に［Enter］や［return］キーを押して空行をいれます。

d.　「else If」と入力して次の状態にします。

```
If[Get ( スクリプト引数 ) = "見積から顧客"]
Else If[ ]
End If
```

e. ［Else If］行に計算式を指定します。

```
Get ( スクリプト引数 ) = "見積から見積"
```

f. ［else If］行の下に 2 番目の［else If］行を入力し、計算式を指定します。

```
Get ( スクリプト引数 ) = "顧客から見積"
```

g. 次の状態になります。

```
If[Get ( スクリプト引数 ) = "見積から顧客"]
Else If[Get ( スクリプト引数 ) = "見積から見積"]
Else If[Get ( スクリプト引数 ) = "顧客から見積"]
End If
```

(3) これまでの演習で作成したスクリプトステップを移動して次のようにしてください。

```
If[Get ( スクリプト引数 ) = "見積から顧客"]

  #見積から顧客を表示
  #演習 11-4 の内容
  関連レコードへ移動 [ （省略）]

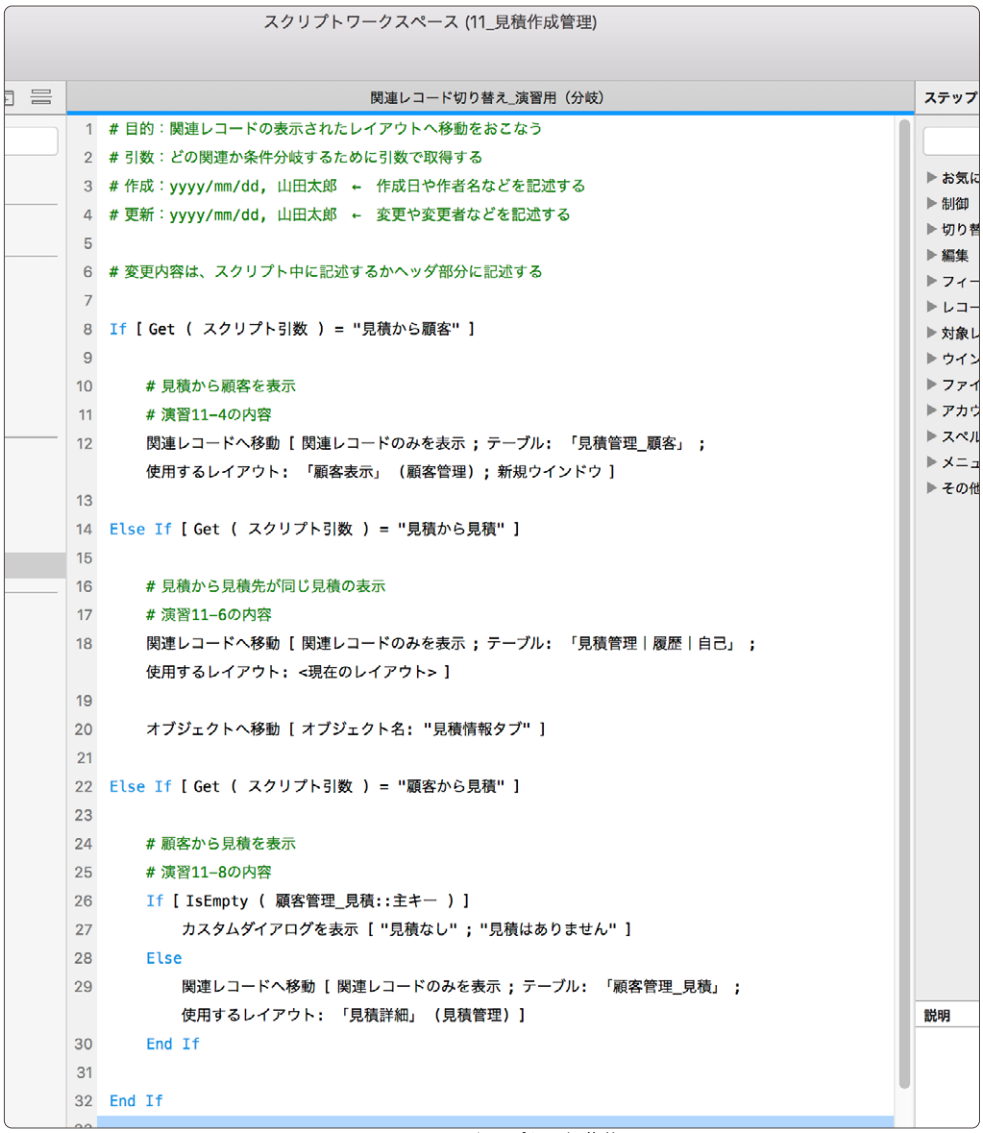
Else If[Get ( スクリプト引数 ) = "見積から見積"]

  #見積から見積先が同じ見積の表示
  #演習 11-6 の内容
  関連レコードへ移動 [ （省略）]
  オブジェクトへ移動 [ （省略） ]

Else If[Get ( スクリプト引数 ) = "顧客から見積"]

  #顧客から見積を表示
  #演習 11-8 の内容
  If [ IsEmpty ( 顧客管理_見積::主キー ) ]
    カスタムダイアログを表示 [（省略） ]
  Else
    関連レコードへ移動 [ （省略）]
  End If

End If
```




```
# 目的：関連レコードの表示されたレイアウトへ移動をおこなう
# 引数：どの関連か条件分岐するために引数で取得する
# 作成：yyyy/mm/dd, 山田太郎 ← 作成日や作者名などを記述する
# 更新：yyyy/mm/dd, 山田太郎 ← 変更や変更者などを記述する

# 変更内容は、スクリプト中に記述するかヘッダ部分に記述する

If [ Get ( スクリプト引数 ) = "見積から顧客" ]

    # 見積から顧客を表示
    # 演習11-4の内容
    関連レコードへ移動 [ 関連レコードのみを表示 ; テーブル: 「見積管理_顧客」 ;
    使用するレイアウト: 「顧客表示」（顧客管理）; 新規ウインドウ ]

Else If [ Get ( スクリプト引数 ) = "見積から見積" ]

    # 見積から見積先が同じ見積の表示
    # 演習11-6の内容
    関連レコードへ移動 [ 関連レコードのみを表示 ; テーブル: 「見積管理 | 履歴 | 自己」 ;
    使用するレイアウト: <現在のレイアウト> ]

    オブジェクトへ移動 [ オブジェクト名: "見積情報タブ" ]

Else If [ Get ( スクリプト引数 ) = "顧客から見積" ]

    # 顧客から見積を表示
    # 演習11-8の内容
    If [ IsEmpty ( 顧客管理_見積::主キー ) ]
        カスタムダイアログを表示 [ "見積なし" ; "見積はありません" ]
    Else
        関連レコードへ移動 [ 関連レコードのみを表示 ; テーブル: 「顧客管理_見積」 ;
        使用するレイアウト: 「見積詳細」（見積管理）]
    End If

End If
```

図 11-43 スクリプトの編集後

(4)　スクリプトを保存します。

演習ここまで

演習 11-9 のように［If］スクリプトステップは入れ子にすることも可能です。

何段階も入れ子にする場合でも、すべての条件分岐をテストします。たくさんの入れ子があるとスクリプトは複雑になって後からの変更、修正が難しくなることがあります。入れ子を少なくする工夫ができないかを考えることも重要です。

11.5.2 引数を使った条件分岐

スクリプトをボタンに割り当てて 4 つのボタンに次の引数を設定します。

・見積から顧客　「見積詳細」レイアウトの会社名ボタン　→　「見積から顧客」
・見積から見積　「見積詳細」レイアウトのポータルの「>」ボタン　→　「見積から見積」
・顧客から見積　「顧客管理｜マスタ詳細」レイアウトの 2 つのボタン　→「顧客から見積」

演習 11-10 スクリプト引数でレイアウトを切り替える 2

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　「見積詳細」レイアウトでレイアウトモードに切り替えます。

(2)　スクリプトを割り当てて、引数を変更します。

a. 画面左上の会社名のボタンをダブルクリックして［ボタン設定］ダイアログを表示します。
b. スクリプト名の右のボタンをクリックすると「関連レコード切り替え _ 演習用（分岐）」スクリプトが選択されています。
c. ［オプションのスクリプト引数:］の［編集 ...］をクリックすると［計算式の指定］ダイアログが表示されます。
d. " 見積から顧客 "　と入力します。
e. ［OK］を何度かクリックします。

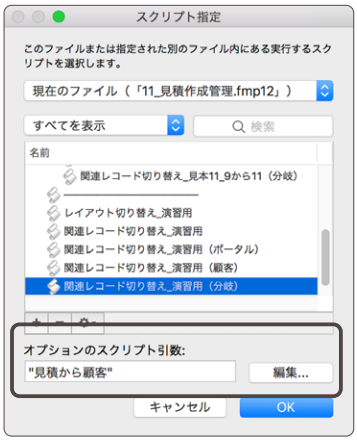


図 11-44 スクリプトと引数を指定

別の方法で引数を設定します。

f. 2 つ目のタブパネル上の「>」ボタンをクリックして［ボタン設定］ダイアログを表示します。
g. スクリプト名の右のボタンをクリックして「関連レコード切り替え _ 演習用（分岐）」スクリプトを選択します。
h. ［OK］をクリックします。
i. スクリプト名の下の［オプションのスクリプト引数 :］の下の入力欄をクリックします
j. ［計算式の指定］ダイアログが表示されます。
k. " 見積から見積 "　と入力します。
l. ［OK］をクリックします。

（このようにどちらの方法でも引数を設定できます）

m. レイアウトを保存します。



図 11-45 ボタン設定ダイアログで引数を指定

(3) 「顧客管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。

(4) スクリプトを割り当てて、引数を変更します（詳細は省略します）。
a. 画面上の書類アイコンのボタンに、
・スクリプトは 「関連レコード切り替え _ 演習用（分岐）」スクリプト
・引数は " 顧客から見積 "
b. 「見積履歴」タブ上のポータルの「>」アイコンのボタンに、
・スクリプトは 「関連レコード切り替え _ 演習用（分岐）」スクリプト
・引数は " 顧客から見積 "

c. レイアウトを保存します。

(5) ブラウズモードで動作を確認します。「見積詳細」レイアウトに切り替えてこれまでの演習でも行っていたテスト作業を一通り実施します。

会社名ボタンの確認
a. ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。
b. 1 レコード目に移動します。見積先の会社名は「株式会社ベゴニア」です。
c. 会社名ボタンをクリックします。
d. 「顧客表示」レイアウトが、ドキュメントウインドウで表示されます。会社名は「株式会社ベゴニア」です。
e. 閉じるボタンをクリックしてウインドウを閉じます。

タブ上のポータルの「>」ボタンの確認
f. 3 レコード目の「見積番号 10003」に移動します。
g. 2 つ目のタブパネルをクリックします。
h. ポータルには 5 行が表示されています。
i. ポータル行の 2 行目「見積番号 10021」の「>」ボタンをクリックします。
j. タブも切り替って「見積番号 10021」レコードが表示されます。
k. 対象レコードは 5 件です（ポータルの表示と同じ件数です）
l. ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

(6) 「顧客管理｜マスタ詳細」レイアウトに切り替えます。

書類アイコンボタンの確認
a. 「顧客番号 20006」に移動します。会社名は「株式会社アサガオ」です。
b. 「見積履歴」タブをクリックします。ポータルにはなにも表示されていません。
c. 書類アイコンのボタンをクリックします。
d. カスタムダイアログが表示されます。
e. [OK] をクリックしてダイアログを閉じます。

f. 「顧客番号 20002」に移動します。会社名は「ホオズキ株式会社」です。
g. 「見積履歴」タブをクリックすると、ポータルには 6 行が表示されています。
h. 書類アイコンのボタンをクリックします。
i. 「顧客番号 20002」の見積データ「見積番号 10003」レコードが表示されているはずです。会社名は「ホオズキ株式会社」です。
j. ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

(7)　「顧客管理｜マスタ詳細」レイアウトに切り替えます。

タブ上のポータルの「>」ボタンの確認

a. 「顧客番号 20002」に移動します。会社名は「ホオズキ株式会社」です。
b. 「見積履歴」タブをクリックすると、ポータルには 6 行が表示されています。
c. ポータル行の 3 行目「見積番号 10021」の「>」ボタンをクリックします。
d. 「顧客番号 20002」の見積データ「見積番号 10021」レコードが表示されているはずです。会社名は「ホオズキ株式会社」です。
e. 対象レコードは 6 件です（「顧客管理｜マスタ詳細」レイアウトのポータルの表示と同じ件数です）。
f. ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

(8)　デバッガとデータビューアを表示して確認します。
ステップインで 1 行ずつ確認します。変数の内容などを確認してください。

演習ここまで

動作確認では、これまでの演習を一通り確認しました。スクリプトで多数の分岐を使う場合は、このようにすべての分岐結果を確認する必要があります。

11.5.3 スクリプトと変数

「関連レコード切り替え _ 演習用（分岐）」スクリプトには改善点があります。第 10 章の Let 関数の解説で、3 種類の変数がありました。

・**変数**　→　任意の文字列を変数名とし、Let 関数の計算式の中でのみ使用するもの
・**$ 変数**　→　**ローカル変数**　宣言があったスクリプトが終了するまで有効なもの
・**$$ 変数**　→　**グローバル変数**　宣言があったファイル内で有効なもの

また、以下の式を例示しました。

```
Let (
        x = Get( 日付 ) ;
        Date ( Month( x ) ; Day( x ) ; Year( x ) )
 )
```

同じ関数を 3 回実行することなく、1 回で済ませることができるので、効率的な計算式となります。これはスクリプトでも同じです。

スクリプト引数を変数としてスクリプト内で利用する演習を行います。

演習 11-11 スクリプトで変数を使う

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］を表示してスクリプトパネルで、**「関連レコード切り替え _ 演習用（分岐）」スクリプト**をクリックします。

(2)　最初の［If］の前に空白行を入れて次のスクリプトステップを入力します。

［変数を設定］

オプションを設定します。
・［名前：］は　「$ 画面遷移」（$ は半角）
・［値：］は　［指定 ...］をクリックして、計算式を入力

```
Get( スクリプト引数 )
```



図 11-46［変数を設定］スクリプトステップオプションの設定内容

(3)　スクリプトを引数を使った内容に書き換えます（詳細な手順は省略します）。
Get (スクリプト引数) を変数と置き換えます。

a.　最初の［if］スクリプトステップは、

```
$画面遷移 = "見積から顧客"
```

b.　2 番目の［if］スクリプトステップは、

```
$画面遷移 = "見積から見積"
```

c.　3 番目の［if］スクリプトステップは、

```
$画面遷移 = "顧客から見積"
```

d.　次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
変数を設定 [ $画面遷移; 値:Get( スクリプト引数 ) ]

If[ $画面遷移 = "見積から顧客"]

  #見積から顧客を表示
  #演習 11-4 の内容
  関連レコードへ移動 [（省略）]

Else If[ $画面遷移 = "見積から見積" ]

  #見積から見積先が同じ見積の表示
  #演習 11-6 の内容
  関連レコードへ移動 [（省略）]
  オブジェクトへ移動 [（省略）]

Else If[ $画面遷移 = "顧客から見積" ]

  #顧客から見積を表示
  #演習 11-8 の内容
  If [ IsEmpty ( 顧客管理_見積::主キー ) ]
    カスタムダイアログを表示 [（省略） ]
  Else
    関連レコードへ移動 [（省略）]
  End If

End If
```

(4)　スクリプトを保存し、演習 11-10 と同じく動作を確認します（489 ページを参照）。

演習ここまで

ボタンの引数は変更する必要はありません。スクリプト内で変数を使用するように書き換えただけなので、引数には影響を与えません。

Let 関数のように関数の評価回数が少なくなりました。演習 11-10 のスクリプトでは、Get (スクリプト引数) 関数は［If］スクリプトステップの 3 箇所で評価していましたが、演習 11-11 のスクリプトでは［変数を設定］スクリプトステップの 1 箇所になりました。

変数の管理に注意
変数名を間違えて入力するとスクリプトが想定どおりに動作しません。変数名の管理やチェックをきちんと行います。なお、**変数名は英文字では大文字小文字の区別はありません。**

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_12」フォルダの中の「11_ 見積作成管理 .fmp12」ファイルをベースに進めます）

11.5.4 サブスクリプト

スクリプト内で、さらにスクリプトを呼び出して実行できます。呼び出される側のスクリプトを「**サブスクリプト**」と呼びます。

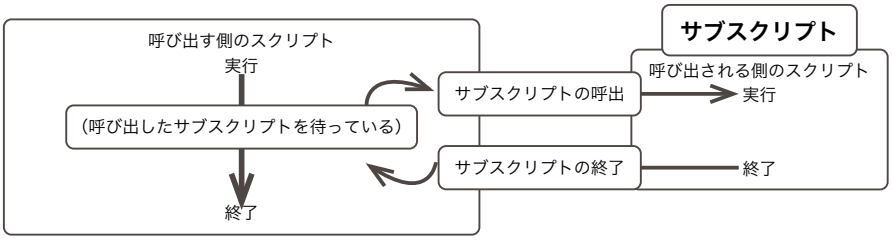


図 11-47 スクリプトからサブスクリプトを呼び出す

スクリプト（呼び出した側）は、サブスクリプトが終わるのを待ちます。サブスクリプトは、よく使うスクリプトを「**モジュール**」として扱うときに便利です。

モジュール
ひとまとまりの機能のことを指します。サブスクリプトを 1 つあるいは複数組み合わせて 1 つの機能をつくると考えてください。難しく考えることはありません。印刷のための用紙サイズ、印刷ダイアログ、引数を使用して任意のレイアウトオブジェクトへ移動するなど、あちこちで呼び出されるようなものをサブスクリプトにすると便利です。開発をはじめた当初から『モジュール化しよう！』と頑張らなくてもだんだん見えてきます。

サブスクリプトは、次のスクリプトステップで呼び出します。

［スクリプト実行］スクリプトステップ

サブスクリプトへ引数を渡す
サブスクリプトにもオプションとして引数を設定できます。Get(スクリプト引数) 関数で、現在のスクリプトからサブスクリプトへ引数をそのまま引き渡せます。また、現在のスクリプトの変数やフィールドのデータなど、別の引数を設定することも、もちろん可能です。

サブスクリプトと実行結果
サブスクリプトを実行後に結果を受け取ることができます。

Get(スクリプトの結果) 関数

を使います。そして、実行したサブスクリプトの結果で条件分岐することもできます。

この関数が取得するのは、最後に実行されたサブスクリプトの結果です。複数のサブスクリプトの結果を比較して条件分岐する場合などには、それぞれの結果を変数やグローバル格納フィールドを使って保持します。

条件分岐とサブスクリプトを使って演習 11-11 と同内容のスクリプトを作成する演習を行います。

演習ファイルには、「関連レコード切り替え _ 演習用（分岐／サブスクリプト）」スクリプトを作成し、これまでの 4 箇所のボタンに割り当てておきました（引数は変えていません）。

「演習 11_12」フォルダ

演習 11-12 サブスクリプトを実行する

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1) ［スクリプトワークスペース］を表示し**「関連レコード切り替え _ 演習用（分岐／サブスクリプト）」**を編集します。内容は次図です。途中まで書いておきました。

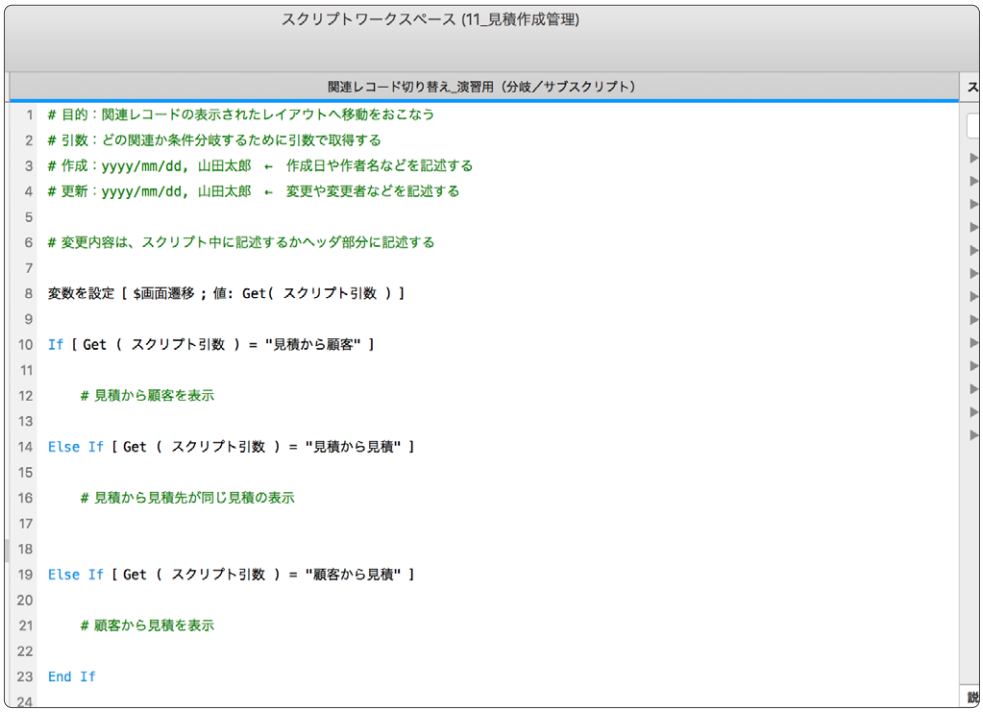


図 11-48 スクリプトの内容

(2) コメント行「見積から顧客を表示」の次行にスクリプトステップを入力します。

スクリプト実行

オプションを設定します。
・［指定：］は　「一覧から」 そのまま
・＜不明＞をクリックして「関連レコード切り替え _ 演習用」を選択します。
・［引数は：］は　なし

(3) コメント行「見積から見積先が同じ見積の表示」の次行にも［スクリプト実行］スクリプトステップを入力します。
オプションを設定します。
・［指定：］は　「一覧から」 そのまま
・＜不明＞をクリックして「関連レコード切り替え _ 演習用（ポータル）」を選択します。
・［引数は：］は　なし

(4)　その下に［オブジェクトへ移動］スクリプトステップを入力します。
　　オプションを設定します。
　　・［オブジェクト名：］は　「見積情報タブ」

(5)　コメント行「顧客から見積を表示」の次行にも［スクリプト実行］スクリプトステップを入力します。
　　オプションを設定します。
　　・［指定：］は　　「一覧から」 そのまま
　　・＜不明＞をクリックして「関連レコード切り替え _ 演習用（顧客）」を選択します。
　　・［引数は：］は　　なし

(6)　次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
変数を設定 [ $画面遷移; 値:Get(スクリプト引数) ]

If[ $画面遷移 = "見積から顧客"]

  #見積から顧客を表示
  スクリプト実行[ 指定：一覧から ;「関連レコード切り替え_演習用」 ]

Else If[ $画面遷移 = "見積から見積" ]

  #見積から見積先が同じ見積の表示
  スクリプト実行[ 指定：一覧から ;「関連レコード切り替え_演習用（ポータル）」 ]

  オブジェクトへ移動 [ オブジェクト名 : "見積情報タブ" ]

Else If[ $画面遷移 = "顧客から見積" ]

  #顧客から見積を表示
  スクリプト実行[ 指定：一覧から ;「関連レコード切り替え_演習用（顧客）」 ]

End If
```

(7)　スクリプトを保存し、演習 11-10 と同じく動作を確認します（489 ページを参照）。

演習ここまで

ボタンの引数は変更する必要はありません。この変更も引数には影響を与えません。スクリプトがすっきりしました。サブスクリプトは機能を分けて管理をできますし、複数のスクリプトから実行するような部品的な使い方も可能です。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_13 から 14」フォルダの中の「11_見積作成管理 .fmp12」ファイルをベースに進めます）

11.5.5 複数の引数

スクリプト引数はとても便利です。そして、カスタム App の機能では 2 つ以上のスクリプト引数を使いたいことが多々あります。しかし、スクリプト引数は、スクリプトごとに 1 つしか指定できませんし、Get (スクリプト引数) 関数だけでは、スクリプトの中でうまく取り扱えません。

他の関数を使って工夫をすれば 1 つの引数を複数に分けて使用できます。

例えば、

```
GetValue ( 値一覧 ; 値番号 ) 関数
```

で行ごとに分ける方法です。つまり、次のような引数（改行区切りのテキスト）があるとします。

```
"猫" & ¶ &
"オス" & ¶ &
"4.5"
```

これを［変数を設定］スクリプトステップで次のようにデータを取得して利用できます。

```
$変数A → GetValue ( Get ( スクリプト引数 ) ; 1 )  → 結果は 「猫」
$変数B → GetValue ( Get ( スクリプト引数 ) ; 2 )  → 結果は「オス」
$変数C → GetValue ( Get ( スクリプト引数 ) ; 3 )  → 結果は「4.5」
```

改行区切りの何行目に、どのデータがあるかを理解していなければなりません。または、違う関数を使う方法もあります。JSON 関数です。引数には、次の関数を使います。

```
JSONSetElement ( json ; キーまたは索引またはパス ; 値 ; タイプ ) 関数
```

つまり、JSONSetElement 関数の結果としての、次の引数（JSON フォーマット）があるとします。

```
{"動物":"猫","性別":"オス","体重":"4.5"}
```

これを［変数を設定］スクリプトステップで次の関数を使ってデータを取得して利用できます。

```
JSONGetElement ( json ; キーまたは索引またはパス ) 関数
```

```
$変数A
  → JSONGetElement ( Get ( スクリプト引数 ) ; "動物" )  → 結果は「猫」
$変数B
  → JSONGetElement ( Get ( スクリプト引数 ) ; "性別" )  → 結果は「オス」
$変数C
  → JSONGetElement ( Get ( スクリプト引数 ) ; "体重" )  → 結果は「4.5」
```

JSON 関数を使う方が、「名前（要素）」と「値」として引数を取り扱えるために便利です。

これらを演習で使ってみます。

本章の演習で作成したスクリプトは 2 種類です。次の 2 つのスクリプトに集約されています。

・レイアウト切り替え　→ 「レイアウト切り替え _ 演習用」
・関連レコード移動　→ 「関連レコード切り替え演習用（分岐／サブスクリプト）」

これを、複数の引数をつかって 1 つのスクリプトにします。まず、これまでの引数を整理します。

機能の種類	レイアウト切り替え	関連レコード移動
スクリプト名	「レイアウト切り替え _ 演習用」	「関連レコード切り替え _ 演習用 （分岐／サブスクリプト）」
引数	レイアウト名を指定 「ホーム」 「見積リスト」 「見積詳細」 「顧客管理｜マスタ詳細」 「商品管理｜マスタ詳細」	動作を指定 「見積から顧客」 「見積から見積」 「顧客から見積」
スクリプト内の変数	なし	$ 画面遷移
動作	引数のレイアウトへ切り替え	引数を変数にして条件分岐し、スクリプトを実行 「関連レコード切り替え _ 演習用」 「関連レコード切り替え _ 演習用（ポータル）」 「関連レコード切り替え _ 演習用（顧客）」

まず、実行するスクリプトを指定し、そしてスクリプト内で使われるべき引数を指定すれば良さそうです。すると、

・1 つ目は「遷移種類」 → 「レイアウト切り替え」または「関連レコード移動」
・2 つ目は「遷移内容」 → 1 行目に指定した種類によって適切な引数を設定（上表参照）

ということになります。つまり引数の組み合わせは、次のどちらかです。

```
" レイアウト切り替え "
  と
" ホーム " または " 見積リスト " または " 見積詳細 " または " 顧客管理｜マスタ詳細 " または " 商品管理｜
マスタ詳細 "
```

```
" 関連レコード移動 "
  と
" 見積から顧客 " または " 見積から見積 " または " 顧客から見積 "
```

改行区切りの 2 行の引数を設定し、GetValue 関数で取り出して変数として利用します。演習ファイルには、「画面遷移 _ 演習用（GetValue）」スクリプトを作成し各所のボタンに割り当てておきました。

なお引数は、設定箇所が多いため、演習用の一部のもの以外は設定済みです。

・「ホーム」レイアウト　→ 3 つ → 演習で設定
・「見積リスト」レイアウト　→ 5 つ → 設定済み
・「見積詳細」レイアウト　→ 7 つ → 演習で設定
・「商品管理｜マスタ詳細」レイアウト → 4 つ → 設定済み
・「顧客管理｜マスタ詳細」レイアウト → 6 つ → 設定済み

「演習 11_13 から 14」フォルダ

演習 11-13 複数の引数を取り扱う：GetValue 関数　1

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］を表示し**「画面遷移 _ 演習用（GetValue)」スクリプト**を編集します。内容は次図です。

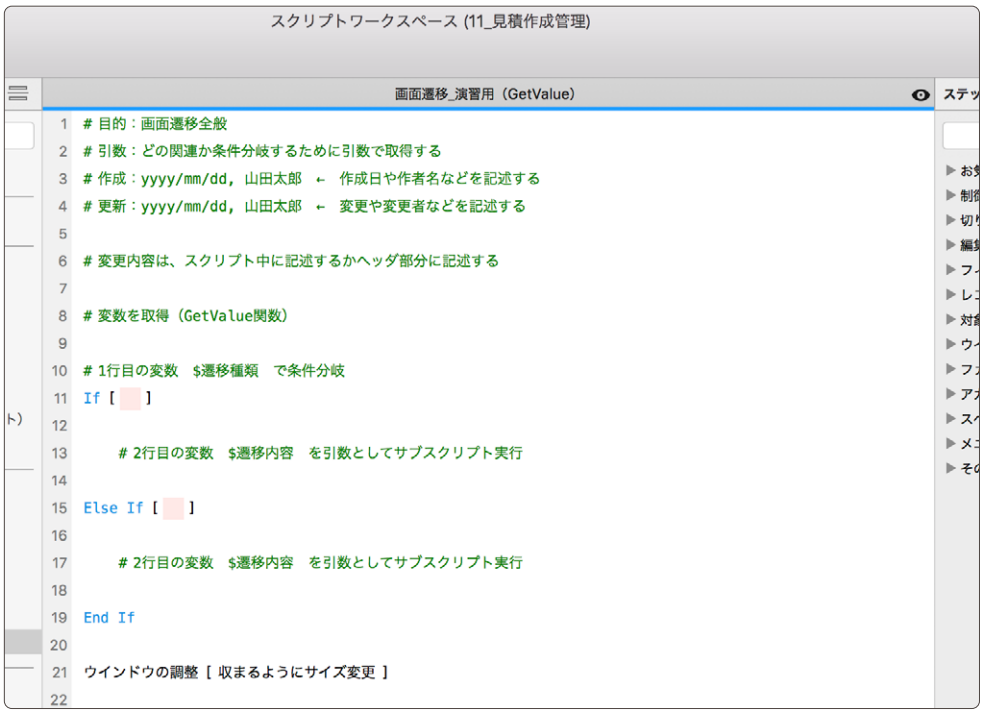


図 11-49 スクリプトの内容

(2)　コメント行「変数を取得（GetValue 関数）」の次行にスクリプトステップを入力します。

変数を設定

オプションを設定します。
・［名前：］は　「$ 遷移種類」
・［値：］は　　［指定 ...］をクリックして、計算式を入力

```
GetValue ( Get( スクリプト引数 ) ; 1 )
```

(3)　次の行も［変数を設定］スクリプトステップを入力します。
オプションを設定します。
・［名前：］は　「$ 遷移内容」
・［値：］は　　［指定 ...］をクリックして、計算式を入力

```
GetValue ( Get( スクリプト引数 ) ; 2 )
```

(4)　実行するサブスクリプトのための条件分岐を記述します。

a. 最初の［if］スクリプトステップは、

$遷移種類 = "レイアウト切り替え"

b. その下のコメント行「2 行目の変数 $遷移内容 を引数としてサブスクリプト実行」の次行にスクリプトステップを入力します。

［スクリプト実行］スクリプトステップ

オプションを設定します。
・［指定：］は 「一覧から」 そのまま
・＜不明＞をクリックし「レイアウト切り替え _ 演習用」を選択します。
・［引数は：］は $遷移内容

c. 2 番目の［if］スクリプトステップは、

$遷移種類 = "関連レコード移動"

d. その下のコメント行「2 行目の変数 $遷移内容 を引数としてサブスクリプト実行」の次行に［スクリプト実行］スクリプトステップを入力しオプションを設定します。
・［指定：］は 「一覧から」 そのまま
・＜不明＞をクリックし「関連レコード切り替え _ 演習用（分岐／サブスクリプト）を選択します。
・［引数は：］は $遷移内容

(5) 次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
#変数を取得（GetValue関数）
変数を設定 [ $遷移種類; 値:GetValue ( Get(スクリプト引数) ; 1 ) ]
変数を設定 [ $遷移内容; 値:GetValue ( Get(スクリプト引数) ; 2 ) ]

#1行目の変数 $遷移種類 で条件分岐
If [ $遷移種類 = "レイアウト切り替え" ]

    # 2行目の変数 $遷移内容 を引数としてサブスクリプト実行
    スクリプト実行 [ 「レイアウト切り替え_演習用」; 引数: $遷移内容 ]

Else If [ $遷移種類 = "関連レコード移動" ]

    # 2行目の変数 $遷移内容 を引数としてサブスクリプト実行
    スクリプト実行 [ 「関連レコード切り替え_演習用（分岐／サブスクリプト）」;
引数: $遷移内容 ]

End If

ウインドウの調整 [ 収まるようにサイズ変更 ]
```

演習ここまで

スクリプト引数を設定します。

演習 11-14 複数の引数を取り扱う：GetValue 関数　2

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　ボタンに引数を設定します（いずれも詳細な手順は省略します）。

(2)　「ホーム」レイアウトでレイアウトモードに切り替えます。
真ん中のボタンバー（3 セグメント）に「画面遷移 _ 演習用（GetValue）」スクリプトを割り当ててあります。**引数を指定します**（引数は下図を参照）。

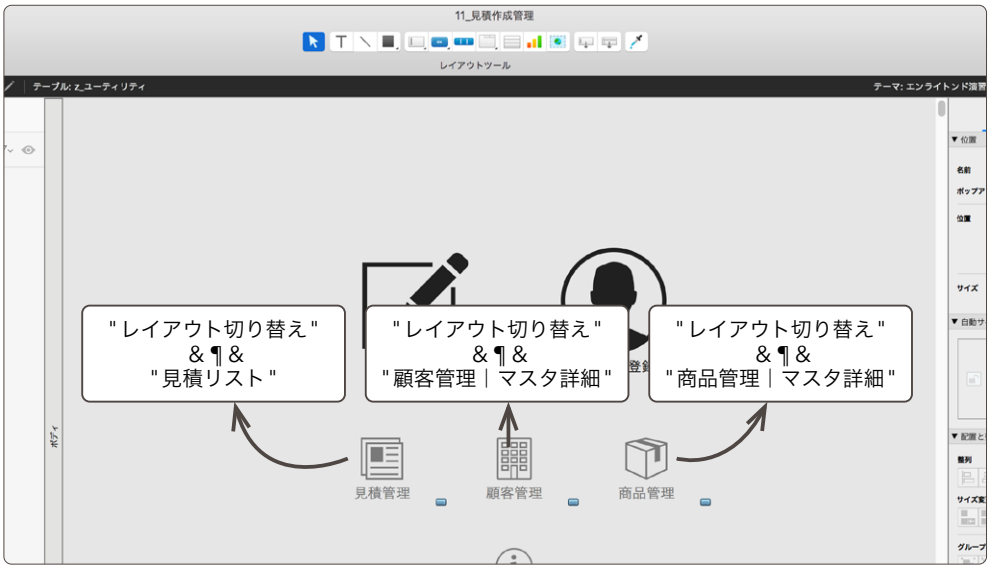


図 11-50「ホーム」レイアウトのボタンへのスクリプト割り当てと引数

(3)　レイアウトを保存します。

(4)　「見積リスト」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（3/4 セグメント）と、ボディの「>」ボタンに、スクリプトと引数が割り当て済みであることを確認してください。

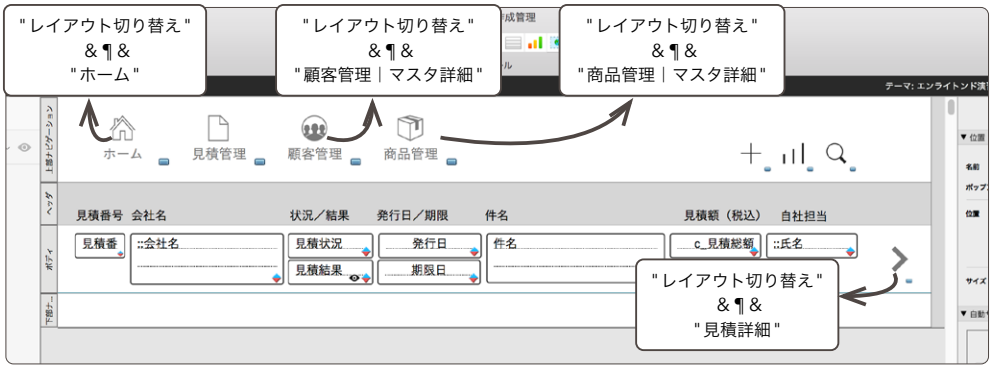


図 11-51「見積リスト」レイアウトのボタンへのスクリプト割り当てと引数

(5)　「見積詳細」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（4/5 セグメント）に「画面遷移 _ 演習用（GetValue)」スクリプトを割り当ててあります。**引数を指定します**（引数は下図を参照)。

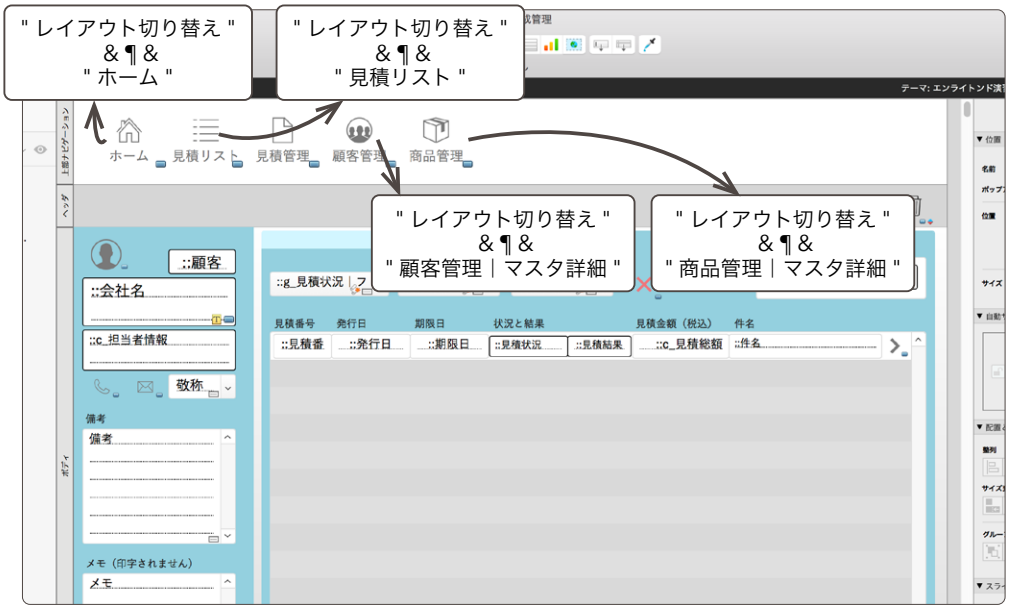


図 11-52「見積リスト」レイアウトのボタンへのスクリプト割り当てと引数 1

会社名ボタン、2 つ目のタブパネルのポータル上の「>」ボタンにも**引数を指定します**（引数は下図を参照)。



図 11-53「見積リスト」レイアウトのボタンへのスクリプト割り当てと引数 2

(6)　レイアウトを保存します。

(7)　「商品管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（3/4 セグメント）に、スクリプトと引数が割り当て済みであることを確認してください。



図 11-54「商品管理｜マスタ詳細」レイアウトのボタンへのスクリプト割り当てと引数

(8)　「顧客管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（3/4 セグメント）と、ボディの「>」ボタンに、スクリプトと引数が割り当て済みであることを確認してください。

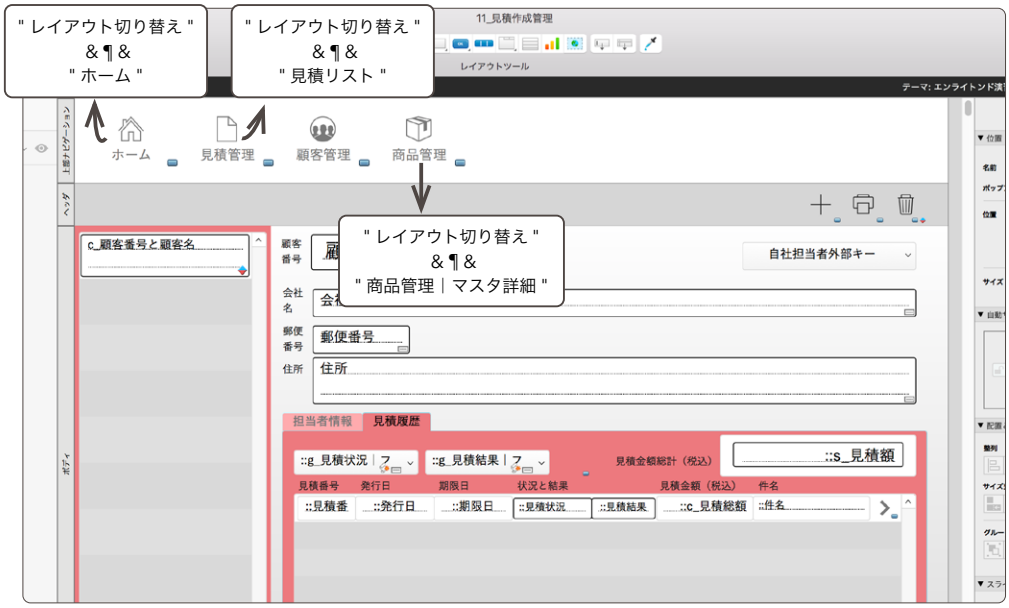


図 11-55「顧客品管理｜マスタ詳細」レイアウトのボタンへのスクリプト割り当てと引数 1

書類アイコンボタン、「見積履歴」タブのポータル上の「>」ボタンにスクリプトと引数が割り当て済みであることを確認してください。

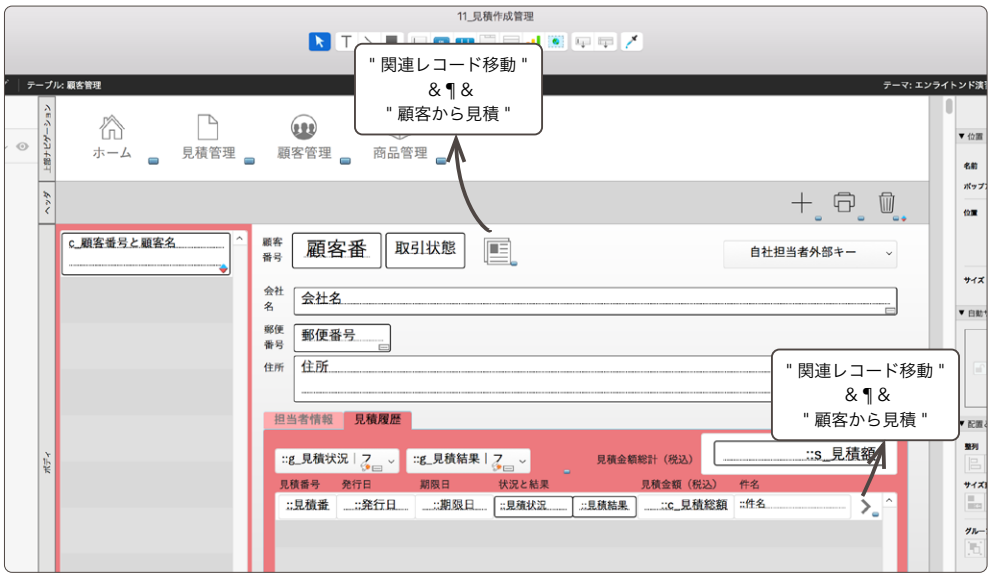


図 11-56「顧客品管理｜マスタ詳細」レイアウトのボタンへのスクリプト割り当てと引数 2

(9)　「ホーム」レイアウトに切り替えます。

(10)　動作を確認します。

「ホーム」レイアウトのボタンの確認

a.　「見積管理」ボタンをクリックします。
b.　「見積リスト」レイアウトが表示されます。
c.　「ホーム」ボタンをクリックします（「ホーム」レイアウトが表示されます）。
d.　「顧客管理」ボタンをクリックします。
e.　「顧客管理｜マスタ詳細」レイアウトが表示されます。
f.　「ホーム」ボタンをクリックします（「ホーム」レイアウトが表示されます）。
g.　「商品管理」ボタンをクリックします。
h.　「商品管理｜マスタ詳細」レイアウトが表示されます。
i.　「ホーム」ボタンをクリックします（「ホーム」レイアウトが表示されます）。

……このあとの手順はとても長いので、ここでは省略します。スクリプトと引数を設定したレイアウトですべてのパターンを試します。

(11)　デバッガとデータビューアを表示して確認します。
ステップインで 1 行ずつ確認します。変数の内容などを確認してください。

演習ここまで

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_15 から 16」フォルダの中の「11_見積作成管理 .fmp12」ファイルをベースに進めます）

演習 11-14 を JSON 関数を使った引数の指定と、変数への設定で行います。

演習ファイルには、「画面遷移 _ 演習用（JSON）」スクリプトを作成し各所のボタンに割り当てておきました。なお引数は、設定箇所が多いため演習用の一部のもの以外は設定済みです。

JSONSetElement 関数で、どのように引数を設定したら良いのでしょうか。

" レイアウト切り替え " → $ 遷移種類
" ホーム " または " 見積リスト "（……省略） → $ 遷移内容

と、なるのですから「レイアウト切り替え _ 演習用」スクリプトで「見積リスト」レイアウトへ切り替えたいなら、次の計算式をスクリプト引数として設定します。

```
JSONSetElement ( "{}" ;
  ["遷移種類" ; "レイアウト切り替え" ; JSONString];
  ["遷移内容" ; "見積リスト" ; JSONString] )
```

JSONString は、データの型です。
JSONString → 文字列
JSONNumber → 数字
JSONBoolean → 論理値
などがあります。詳しくはヘルプを確認してください。

以下の結果になります。JSON フォーマットデータ、つまり「名前（要素）と値（データ）」のペアです。

```
{"遷移内容":"見積リスト","遷移種類":"レイアウト切り替え"}
```

名前（要素） 値（データ） 名前（要素） 値（データ）

スクリプト内で、上記の（「名前（要素）と値（データ）」のペアから、値（データ）を取得するには、次の計算式を［変数を設定］スクリプトステップに設定します。

```
JSONGetElement ( Get( スクリプト引数 ) ; "遷移内容" )
```

JSON データ　名前（要素）

取り出す要素を指定しています。「見積リスト」が変数に保持されます。

JSON 関数を使うと、スクリプト引数として指定するデータの内容がわかりやすくなります。

『どのような役割で、このデータが引数として使われているのか』が、すぐにわかるのでスクリプトの変更、修正作業の効率が高くなります。利用を検討してみてください。

「演習 11_15 から 16」フォルダ

演習 11-15 複数の引数を取り扱う：JSON 関数　1

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］を表示し**「画面遷移 _ 演習用（JSON)」スクリプト**を編集します。変数部分以外は GetValue 関数を利用した時と同じなので記述しておきました。

(2)　コメント行「変数を取得（JSONGetElement 関数)」の次行に［変数を設定］スクリプトステップを入力します。

オプションを設定します。

・［名前：］は　「$ 遷移種類」

・［値：］は　　［指定 ...］をクリックして、計算式を入力

```
JSONGetElement ( Get( スクリプト引数 ) ; " 遷移種類 " )
```

(3)　次の行も［変数を設定］スクリプトステップを入力します。

オプションを設定します。

・［名前：］は　「$ 遷移内容」

・［値：］は　　［指定 ...］をクリックして、計算式を入力

```
JSONGetElement ( Get( スクリプト引数 ) ; " 遷移内容 " )
```

(4)　次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します)。

```
#変数を取得（JSONGetElement関数）
変数を設定 [ $遷移種類; 値: JSONGetElement ( Get( スクリプト引数 )
; " 遷移種類 " )]
変数を設定 [ $遷移内容; 値: JSONGetElement ( Get( スクリプト引数 )
; " 遷移内容 " ) ]

#1行目の変数 $遷移種類 で条件分岐
If [ $遷移種類 = " レイアウト切り替え " ]

    #2行目の変数 $遷移内容 を引数としてサブスクリプト実行
    スクリプト実行 [ 「レイアウト切り替え_演習用」; 引数: $遷移内容 ]

Else If [ $遷移種類 = " 関連レコード移動 " ]

    #2行目の変数 $遷移内容 を引数としてサブスクリプト実行
    スクリプト実行 [ 「関連レコード切り替え_演習用（分岐／サブスクリプト)」;
引数: $遷移内容 ]

End If

ウインドウの調整 [ 収まるようにサイズ変更 ]
```

演習ここまで

スクリプト引数を設定します。

演習 11-16 複数の引数を取り扱う：JSON 関数　2

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　ボタンに引数を設定します（いずれも詳細な手順は省略します）。

(2)　「ホーム」レイアウトでレイアウトモードに切り替えます。
真ん中のボタンバー（3 セグメント）の「見積管理」ボタンに「画面遷移 _ 演習用（JSON）」スクリプトを割り当ててあります。**引数を指定します**（引数は下図を参照）。

他の 2 つのボタンには、既に同様の計算式を設定済みです。

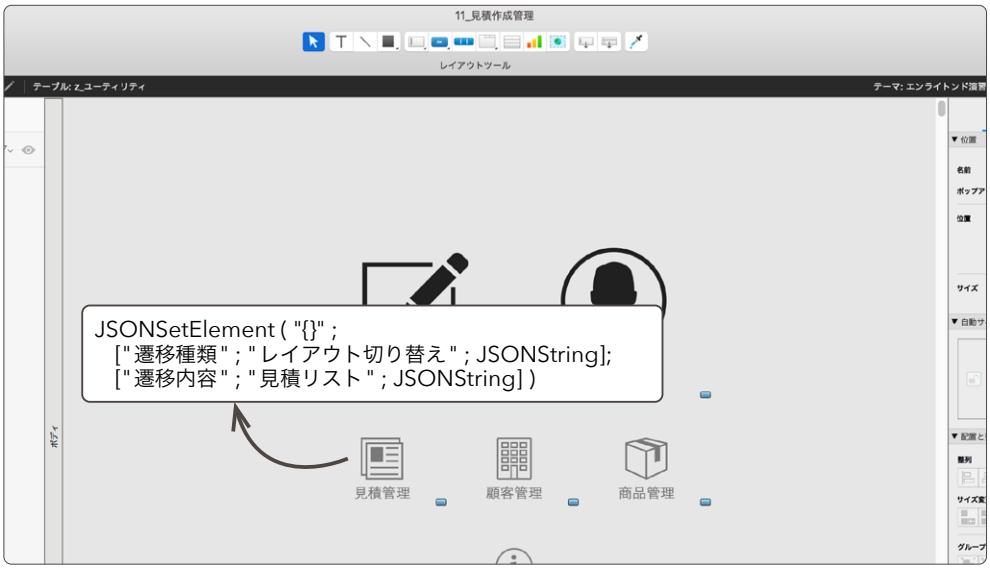


図 11-57「ホーム」レイアウトのボタンへのスクリプト割り当てと引数

(3)　レイアウトを保存します。

(4)　「見積リスト」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（3/4 セグメント）と、ボディの「>」ボタンに、スクリプトと引数が割り当て済みであることを確認してください。

(5)　「見積詳細」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（4/5 セグメント）にスクリプトと引数が割り当て済みであることを確認してください

会社名ボタンに引数を**割り当てます**（引数は次ページ図を参照）。

2 つ目のタブパネルのポータル上の「>」ボタンには、既に同様の計算式を設定済みです。

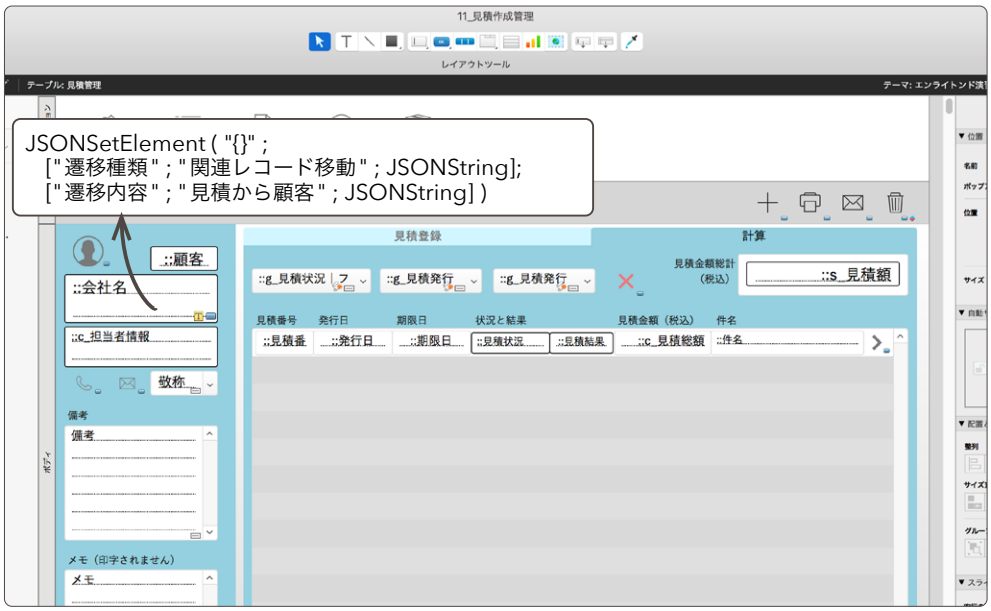


図 11-58「見積リスト」レイアウトのボタンへのスクリプト割り当てと引数 2

(6)　レイアウトを保存します。

(7)　「商品管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（3/4 セグメント）に、スクリプトと引数が割り当て済みであることを確認してください。

(8)　「顧客品管理｜マスタ詳細」レイアウトでレイアウトモードに切り替えます。
「上部ナビゲーションパート」のボタンバー（3/4 セグメント）と、ボディの「>」ボタンに、スクリプトと引数が割り当て済みであることを確認してください

書類アイコンボタン、「見積履歴」タブのポータル上の「>」ボタンにスクリプトと引数が割り当て済みであることを確認してください。

(9)　「ホーム」レイアウトに切り替えます。

(10)　動作を確認します。
(省略します。スクリプトと引数を設定したレイアウトですべてのパターンを試します)。

(11)　デバッガとデータビューアを表示して確認します。
ステップインで 1 行ずつ確認します。変数の内容などを確認してください。

演習ここまで

演習ファイルを閉じて、次のファイルを開いてください。

(ここから、本書に付属している演習ファイルの「演習 11_17 から 19」フォルダの中の「11_見積作成管理 .fmp12」ファイルをベースに進めます)

11.6 ユーザと対話する

これまでの条件分岐は引数や変数のデータを使いましたが、もうひとつの条件があります。

・スクリプトとユーザの対話による分岐（カスタムダイアログでユーザが選ぶ）

です。2 つの要素が必要です。

> ［カスタムダイアログを表示 ］ スクリプトステップ
> Get （最終メッセージ選択) 関数

11.6.1 ［カスタムダイアログを表示］スクリプトステップ

カスタムダイアログを表示する目的は、次のどちらかです。

・ユーザがスクリプトに提示された内容から答えを選択してもらう　→　主に条件分岐へ
・特定のフィールドへ入力してもらう

「答えを選択」してもらうには、オプションのダイアログの「一般」タブで設定します。

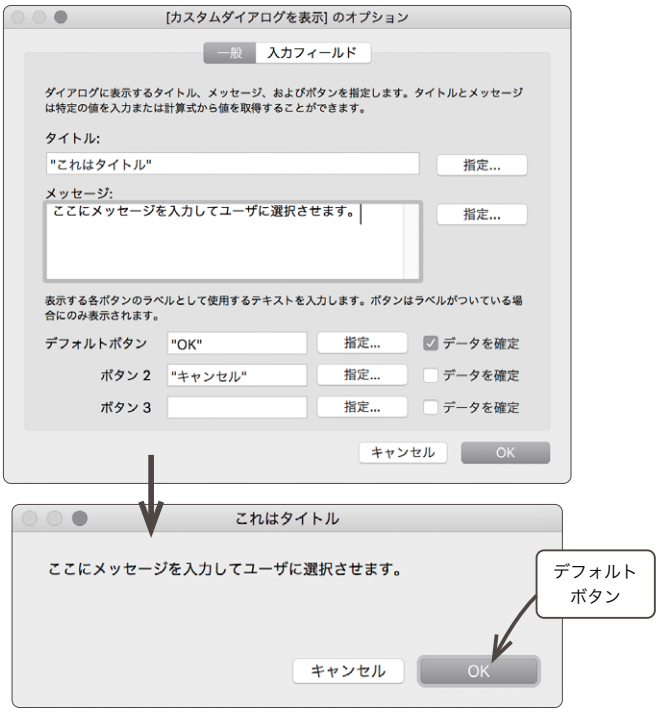


図 11-59「一般」タブ

［タイトル：］［メッセージ：］ボタン名は文字列、または［指定 ...］ボタンをクリックして計算式を指定できます。選択肢は 3 つ設定できます。デフォルトボタンは、カスタムダイアログの表示中に［Enter］または［return］キーを押すと選択されるものです。

ボタンの表示順は左から　［ボタン 3］［ボタン 2］［デフォルトボタン］　です。

「入力」してもらうには、オプションのダイアログの「入力フィールド」タブで設定します。

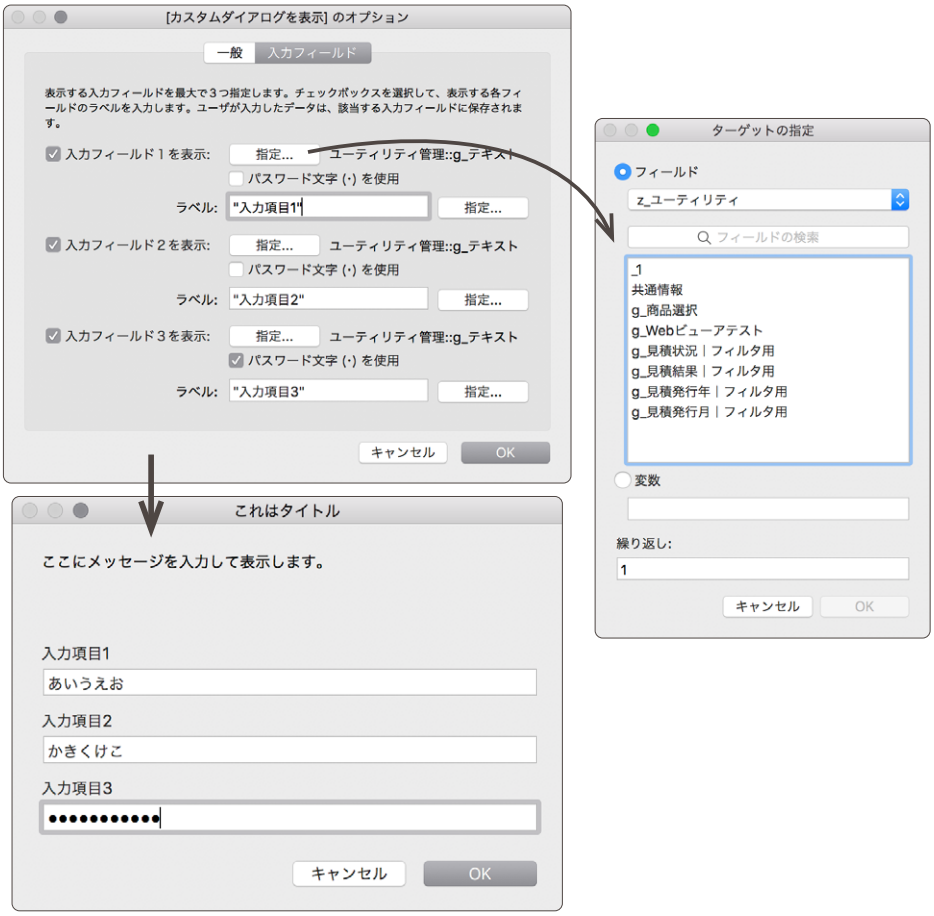


図 11-60「入力フィールド」タブ

フィールドの上に表示する［ラベル：］は文字列、または［指定 ...］から計算式で指定できます。

また、入力フィールドは、フィールド名か、変数名を 3 つ設定できます。

目的のフィールドを指定しても良いですが、前図のようにグローバル格納フィールドを使用したり、変数名を指定すると、入力後にスクリプト内でなんらかの判断をしてから適切なフィールドに入力するなどの動作ができるので便利です。なお、入力された文字を「●」で隠すことが可能です。

サイズや表示位置
マウスでドラッグすると位置やカスタムダイアログのサイズを変更できます。リセットは FileMaker Pro の環境設定から可能です。ただし、この変更はすべてのクライアントには反映されません。

データを確定のチェック
チェックされているボタンをユーザがクリックすると、データが確定されます。

11.6.2 Get (最終メッセージ選択) 関数

「一般」タブで設定できるボタンのうち、どれをユーザがクリックがわかる関数です。

```
Get ( 最終メッセージ選択 ) 関数
```

次の結果（数値）が返ります。
デフォルトボタンは　1
ボタン 2 は　2
ボタン 3 は　3

結果の数字を使用して［If］で条件分岐することが可能です。

11.6.3 スクリプトで印刷する

スクリプトとユーザの対話による分岐の演習として、**印刷用スクリプト**を作成します。スクリプトと対話します。

「見積作成管理 .fmp12」では、要件定義時に、

・見積書と、同様の内容の発注書も印刷する

という要件がありました。そこで、演習ファイルに「見積発注書｜印刷用」レイアウトを作成しておきました。1 レイアウトに「見積書」と「発注書」のレイアウトオブジェクトが配置されています。また、

・印刷される書式をみると明細行が 1 件から 10 件程度のものまで

であったため、ポータルには 15 行までを表示します（多少の余裕を持たせるためです）。想定している用紙サイズは A4 縦です。こんな機能です。

(1) いつ　ユーザの任意
(2) どこで　「見積詳細」レイアウト
(3) 誰が　ユーザ
(4) なにを　現在表示している「見積情報」レコード
(5) なぜ　顧客に提示するために印刷（PDF 化）する
(6) どのように　「見積発注書｜印刷用」レイアウトのレコード（1 件）を A4 縦サイズで印刷（PDF 作成）する

これはスクリプトの流れでは、次のようになります。

(1) ボタンをクリックする
(2)「見積発注書｜印刷用」レイアウトへ切り替える
(3) A4 サイズでレコード（1 件）を印刷（PDF 作成）する
(4) 元のレイアウト（「見積詳細」レイアウト）へ戻る

もう少し細かく考えます。

(1) ボタンをクリックする
(2)「見積発注書｜印刷用」レイアウトへ切り替える
　(2-1) 対象レコードを 1 件にする
(3) 用紙サイズを確認する（A4 サイズ）
(4) 印刷（PDF 作成）する
(5) 元のレイアウト（「見積詳細」レイアウト）へ戻る

「演習 11_17 から 19」フォルダ

演習 11-17 見積書の印刷 1：カスタムダイアログ

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　演習ファイルを確認します。

(2)　「見積発注書｜印刷用」レイアウトへ切り替えます（縦長です）。
　　このレイアウトを印刷します。

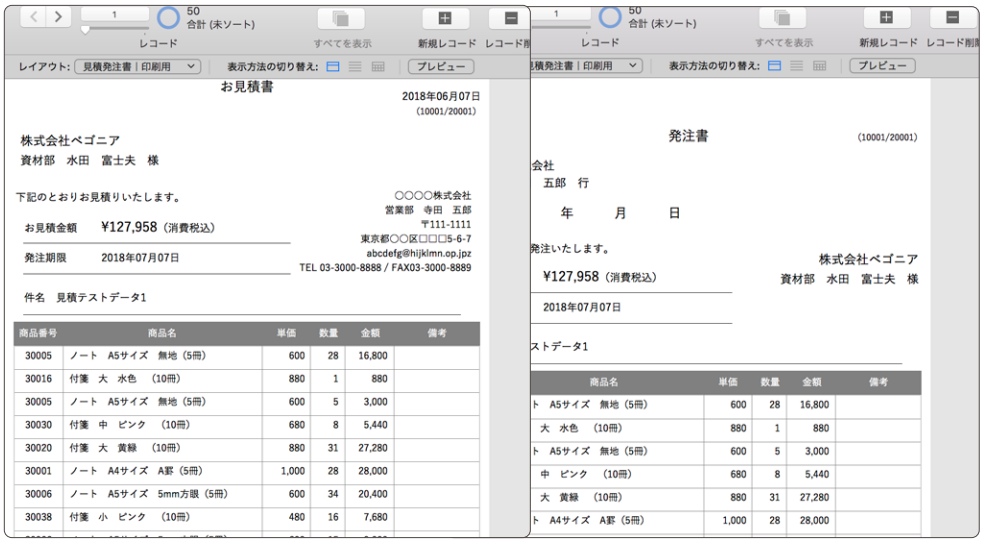


図 11-61 印刷されるレイアウト

(3)　**「印刷 _ 演習用」スクリプト**も用意しました。「見積詳細」レイアウトの右上のボタンバーに割り当てる予定です。

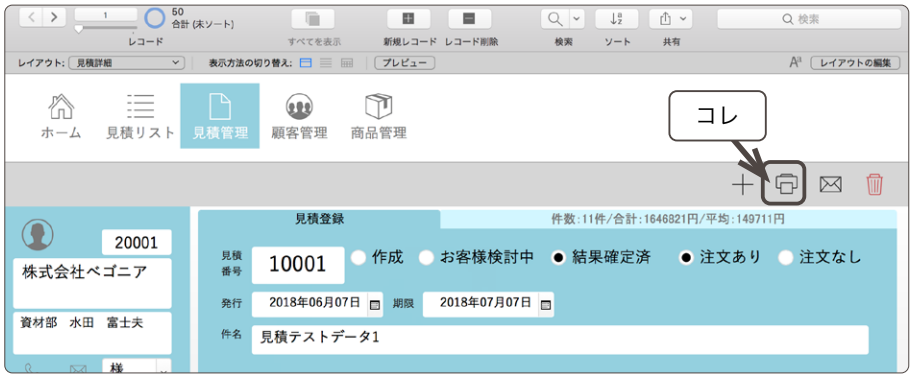


図 11-62 見積発注書を印刷するボタン（予定）

(4)　スクリプトを編集します。

(5)　［スクリプトワークスペース］を表示し**「印刷 _ 演習用」スクリプト**を編集します。
ヘッダだけがあります。

(6)　8 行目に次のスクリプトステップを入力します。

カスタムダイアログを表示

オプションを設定します。
・［タイトル：］は　「印刷」
・［メッセージ：］は　「見積発注書を印刷しますか？」
ボタンは
・［デフォルトボタン］は「する」「データを確定」は　**オン**　そのまま
・［ボタン 2］は　「しない」
・［ボタン 3］は　なにも設定しない



図 11-63「[カスタムダイアログを表示］のオプション」ダイアログ

(7)　次の行にカスタムダイアログの選択結果による条件分岐を入力します。

```
If [Get(最終メッセージ選択) = 1]
Else
End If
```

Get(最終メッセージ選択) = 1　は、「印刷する」という選択です。

(8)　[If] スクリプトステップの下に、次のスクリプトステップを入力します。

レイアウト切り替え

オプションで　「見積発注書｜印刷用」レイアウト　を指定します。

(9)　その次の行に、次のスクリプトステップを入力します。

印刷設定

・[ダイアログ] は　**オフ**

オプションで、
・[用紙サイズ] は　「A4 縦」（プリンタはお手元の物を利用してください）

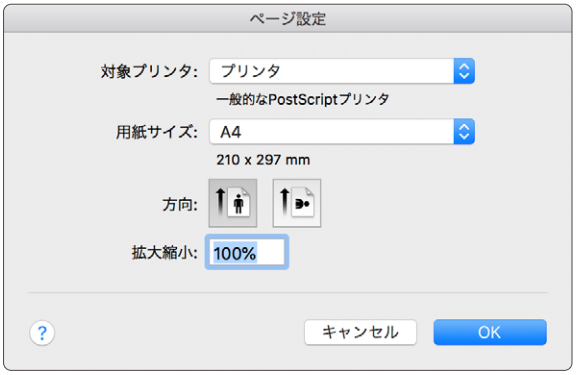


図 11-64 印刷設定のオプション

(10)　その次の行に、次のスクリプトステップを入力します。

印刷

・[ダイアログあり] は　**オン**　　そのまま

オプションで、「現在のレコード」を選択します。
・[印刷]（Windows）ダイアログで［印刷：］からオプションを選択　→　「現在のレコード」
・[プリント]（macOS）ダイアログで［詳細を表示］をクリックしてから、ポップアップメニューから［FileMaker Pro］を選択して　→　「現在のレコード」



図 11-65 印刷のオプション

(11)　その次の行に［レイアウト切り替え］スクリプトステップを入力します。
・オプションは、指定せず「元のレイアウト」のまま

(12) 次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
カスタムダイアログを表示 [ タイトル: "印刷"; "見積発注書を印刷しますか？"]

If [ Get(最終メッセージ選択) = 1 ]

    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]

    印刷設定 [ 記憶する ; ダイアログあり : オフ ]
    印刷 [ 記憶する: (プリンタ名) ; ダイアログあり : オン ]

    レイアウト切り替え [ 元のレイアウト ]

Else
End If
```

(13) 「見積詳細」レイアウトでレイアウトモードに切り替えてボタンにスクリプトを割り当てます。

(14) ブラウズモードに戻ります。動作を確認します。
(プリンタ環境がある場合のみ印刷してください)

a. 1 件目の見積番号「10001」へ移動します。
b. プリンタアイコンボタンを押します。
c. カスタムダイアログが表示されます。

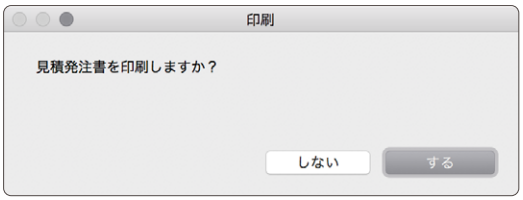


図 11-66 カスタムダイアログの表示

d. [する] をクリックします。
e. [印刷] ダイアログが表示されます。
f. [OK] をクリックします。
g. 印刷されます。
h. 「見積詳細」レイアウトに戻ります。

(15) デバッガとデータビューアを表示して確認します。
ステップインで 1 行ずつ確認します。

演習ここまで

元のレイアウトに戻る

レイアウトを切り替えて印刷する場合、元のレイアウトに戻るようにしなければいけません。印刷実行だけでスクリプトが終了すると、ユーザが戸惑うことになります。

11.7　エラーを処理する

いま「印刷 _ 演習用」スクリプトは、

(1) カスタムダイアログを表示する
(2)「する」がクリックされる
　(2-1) 印刷するレイアウトに切り替える
　(2-2) 用紙設定する（ユーザには表示されない）
　(2-3) 印刷する
　(2-4) 元のレイアウトに戻る
(3) スクリプトが終了する

の、流れです。あるいは、次の流れです。

(1) カスタムダイアログを表示する
(2)「しない」がクリックされる
(3) スクリプトが終了する

しかし［印刷］ダイアログ表示後に［キャンセル］をクリックするとメッセージが表示されます。



図 11-67 印刷ダイアログでキャンセルすると表示される

［キャンセル］をクリックするとスクリプトが「中断」して元のレイアウトに戻れません。

『「印刷」がキャンセルされた』ことを、FileMaker Pro がエラーとして捉えた結果、標準のエラーメッセージが表示されています。このエラーメッセージの内容は［キャンセル］が表示されないように変更することはできません（後述する［ユーザによる強制終了を許可］スクリプトステップで表示されないようにすることはできます）。

エラー後に、適切にスクリプトを実行し続ける（あるいは終了する）ためのスクリプトステップと関数が用意されています。これらも条件分岐に使われます。

［エラー処理］ スクリプトステップ Get (最終エラー) 関数

エラーには、エラーコード（番号）が付与されています。スクリプトデバッガでエラーコードを確認できます。

どのようなエラーコードが用意されているかは、ヘルプを参照してください（ちなみに、今回のエラーコードは「1 ＝ ユーザによるキャンセル」です）。

やってみよう

実際にデバッグしてエラー番号を確認してみましょう。

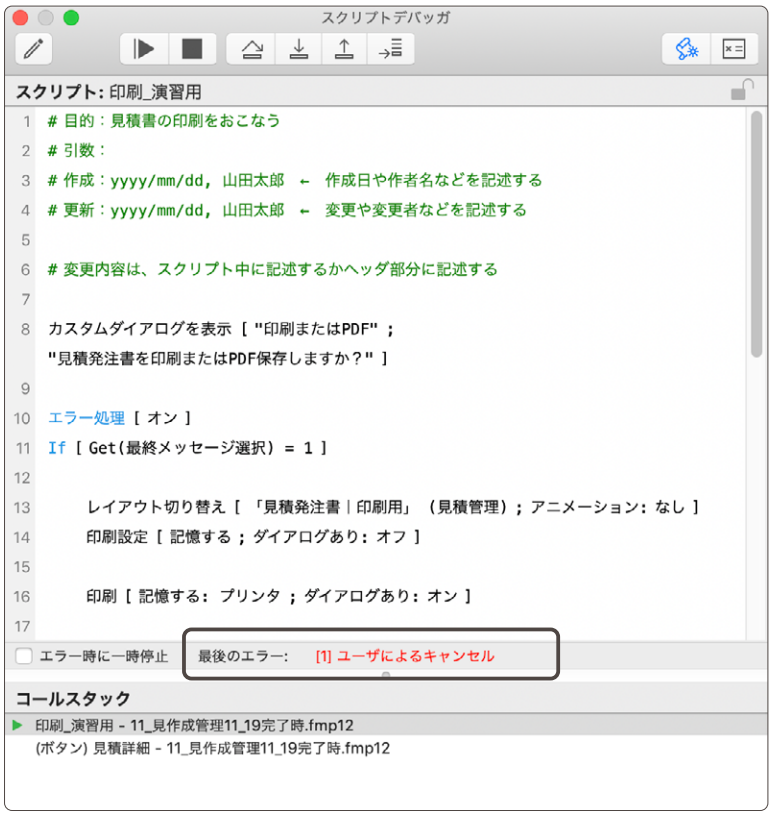


図 11-68 デバッグでエラー番号を確認

「印刷 _ 演習用」スクリプトに「エラー処理」を加える演習を行います。

演習 11-18 見積書の印刷 2：エラー処理

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1) ［スクリプトワークスペース］で**「印刷 _ 演習用」スクリプト**を開きます。

(2) ［印刷］スクリプトステップの前後に［エラー処理］スクリプトステップを入力します。

```
エラー処理 [ オン ]
印刷 [ 記憶する：（プリンタ名） ; ダイアログあり : オン ]
エラー処理 [ オフ ]
```

(3) エラーを捉えてカスタムダイアログを表示します。[If] と［カスタムダイアログ］を使います。
印刷の次行から、

```
エラー処理 [ オン ]
印刷 [ 記憶する : (プリンタ名) ; ダイアログあり : オン ]
If [ Get( 最終エラー ) ]
    カスタムダイアログを表示 [ タイトル : "エラー" ; "印刷がキャンセルされました" ]
End If
エラー処理 [ オフ ]
```

上記のように入力します。

[If] スクリプトステップには、次の計算式を入力します。

```
Get( 最終エラー )
```

(何かエラーがあったら、という条件になります)

[カスタムダイアログを表示] スクリプトステップのオプションは、

・[タイトル :] は 「エラー」
・[メッセージ :] は 「キャンセルされました」
ボタンは
・ [デフォルトボタン] は 「OK」「データを確定」は **オン** そのまま
・ [ボタン 2] は なにも設定しない
・ [ボタン 3] は なにも設定しない

(4) 次のスクリプトの完成形内容になるよう作業し保存します (詳細な手順は省略します)。

```
カスタムダイアログを表示 [ タイトル : "印刷" ; "見積発注書を印刷しますか？" ]

If [ Get( 最終メッセージ選択 ) = 1 ]

    レイアウト切り替え [ 「見積発注書 | 印刷用」 ( 見積管理 ) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ ]

    エラー処理 [ オン ]
    印刷 [ 記憶する : (プリンタ名) ; ダイアログあり : オン ]
    If [ Get( 最終エラー ) ]
        カスタムダイアログを表示 [ タイトル : "エラー" ; "キャンセルされました" ]
    End If
    エラー処理 [ オフ ]

    レイアウト切り替え [ 元のレイアウト ]

Else
End If
```

(5)　動作を確認します。「見積詳細」レイアウトで 1 件目の見積番号「10001」へ移動します。
a. プリンタアイコンボタンを押します。
b. カスタムダイアログが表示されます。
c. ［する］をクリックします。
d. ［印刷］ダイアログが表示されます。
e. ［OK］をクリックします。
f. 印刷されます。
g. 「見積詳細」レイアウトに戻ります。

h. プリンタアイコンボタンを押します。
i. カスタムダイアログが表示されます。
j. ［する］をクリックします。
k. ［印刷］ダイアログが表示されます。
l. ［キャンセル］をクリックします。
m. カスタムダイアログが表示されます。
n. ［OK］をクリックします。
o. 「見積詳細」レイアウトに戻ります。

(6)　デバッガとデータビューアを表示して確認します。

演習ここまで

印刷をキャンセルしても、スクリプトが最後まで動作するようになりました。

11.7.1 ［エラー処理］スクリプトステップ

このスクリプトステップの使い方は、次のようにします。

・エラーを処理したいスクリプトステップの直前に　**オン**
・エラーを処理した直後に　**オフ**

「エラー処理」は、エラーが起きても FileMaker Pro の標準のエラーメッセージを出さずにスクリプトを続行できるようにすることです。そしてエラー処理をする際には、

・エラーが起きた場合にどうするかを開発者が決める

ということが重要です。

・エラー処理だけ
・エラーの内容でスクリプト内で条件分岐
・エラーの内容からユーザとの対話（カスタムダイアログによる選択など）で選択してもらう

などです。そして、重要な点があります。

・エラーを処理したい部分のみで使用すること

これはとても大切です。FileMaker Pro の標準のエラーメッセージを回避することで想定していないエラーの発生を見落とすことを避けるためです。

11.7.2 スクリプトで PDF 保存

「見積作成管理 .fmp12」では、要件定義時に、見積書と、同様の内容の発注書の PDF ファイルの作成も必要でした。

印刷以外に PDF を保存できるようにします。カスタムダイアログの選択を 3 つに増やします。Get(最終メッセージ選択) の結果の想定は以下です。

・「1」は、「印刷する」
・「2」は、「PDF 保存する」
・「3」は、なにもしない

エラー処理も行います。

演習 11-19 見積書の印刷 3：PDF 保存

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］で**「印刷 _ 演習用」スクリプト**を開きます。

(2)　8 行目近辺の［カスタムダイアログを表示］スクリプトステップのオプションを変更します。
・［タイトル：］は　「印刷または PDF」
・［メッセージ：］は　「見積発注書を印刷または PDF 保存しますか？」
ボタンは
・［デフォルトボタン］は　「印刷」「データを確定」は　**オン**　そのまま
・［ボタン 2］は　「PDF」「データを確定」は　**オン**
・［ボタン 3］は　「やめる」「データを確定」は　**オフ**　そのまま

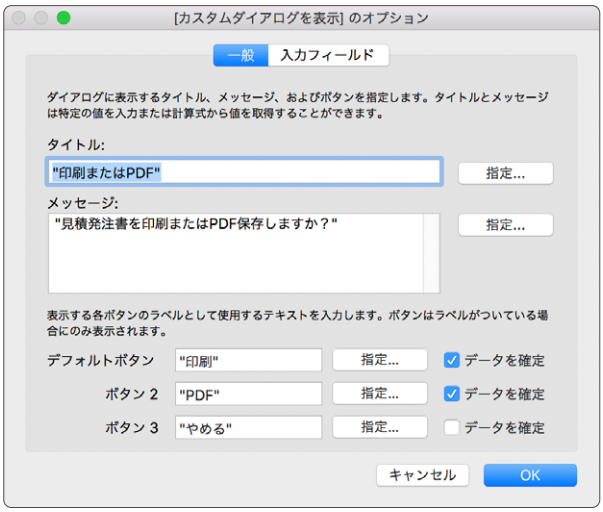


図 11-69［カスタムダイアログを表示］のオプション」ダイアログ

(3)　［If］スクリプトステップからはじまる内容に［ElseIf］を加えて次のように変更します。

```
If [ Get(最終メッセージ選択) = 1 ]

    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ ]

    エラー処理 [ オン ]
    印刷 [ 記憶する:（プリンタ名）; ダイアログあり : オン ]
    If [ Get(最終エラー) ]
        カスタムダイアログを表示 [ タイトル: "エラー"; "キャンセルされました" ]
    End If
    エラー処理 [ オフ ]

    レイアウト切り替え [ 元のレイアウト ]

Else If [ Get(最終メッセージ選択) = 2 ] → このスクリプトステップを加えた

Else
End If
```

(4) 次の行に、スクリプトステップを入力します。

```
レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
印刷設定 [ 記憶する ; ダイアログあり : オフ ]

レコードをPDFとして保存
```

［レイアウト切り替え］と［印刷設定］は、既にある行をコピーと貼り付け（ペースト）して利用します。

［レコードを PDF として保存］スクリプトステップの［オプションの指定］の右の［指定 ...］をクリックし、「現在のレコード」をポップアップメニューから選択します。

(5) ［レイアウトの切り替え］と［エラー処理］の配置に変更が必要です（次ページの完成形を参照）。

(6) 次のスクリプト（完成形）と同じになるよう作業し保存します（詳細な手順は省略します）。

［エラー処理］スクリプトステップ（オン）を最初の［If］スクリプトステップの上に移動するなどしてください。

［エラー処理］スクリプトステップ（オフ）を［End If］スクリプトステップの下に移動するなどしてください。

```
カスタムダイアログを表示 [ タイトル: "印刷またはPDF"; "見積発注書
を印刷またはPDF保存しますか"]

エラー処理 [ オン]
If [ Get(最終メッセージ選択) = 1 ]

    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ]

    印刷 [ 記憶する:(プリンタ名); ダイアログあり : オン]

Else If [ Get(最終メッセージ選択) = 2 ]

    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ]

    レコードをPDFとして保存[ 記憶する ; ダイアログあり : オン ;
現在のレコード ; フォルダを作成 : オフ]

Else
End If

If [ Get(最終エラー) ]
    カスタムダイアログを表示 [ タイトル: "エラー"; "キャンセルされ
ました"]
End If
エラー処理 [ オフ]

レイアウト切り替え [ 元のレイアウト ]
```

(7) 動作を確認します。「見積詳細」レイアウトで 1 件目の見積番号「10001」へ移動します。

a. プリンタアイコンボタンを押します。
b. カスタムダイアログが表示されます。
c. [印刷] をクリックすると [印刷] ダイアログが表示されます。
d. [プリント] をクリックします。
e. 印刷されます。
f. 「見積詳細」レイアウトに戻ります。

g. プリンタアイコンボタンを押します。
h. カスタムダイアログが表示されます。
i. [印刷] をクリックすると [印刷] ダイアログが表示されます。
j. [キャンセル] をクリックします。
k. カスタムダイアログが表示されます。
l. [OK] をクリックします。
m. 「見積詳細」レイアウトに戻ります。

n. プリンタアイコンボタンを押します。
o. カスタムダイアログが表示されます。
p. ［PDF］をクリックすると保存ダイアログが表示されます。
q. 場所と名前を指定して［保存］をクリックします。
r. PDF が保存されます。
s. 「見積詳細」レイアウトに戻ります。

t. プリンタアイコンボタンを押します。
u. カスタムダイアログが表示されます。
v. ［PDF］をクリックすると保存ダイアログが表示されます。
w. ［キャンセル］をクリックします。
x. カスタムダイアログが表示されます。
y. ［OK］をクリックします。
z. 「見積詳細」レイアウトに戻ります。

aa. プリンタアイコンボタンを押します。
ab. カスタムダイアログが表示されます。
ac. ［やめる］をクリックします。

(8) デバッガとデータビューアを表示して確認します。

演習ここまで

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_20」フォルダの中の「11_ 見積作成管理 .fmp12」ファイルをベースに進めます）

11.8 スクリプトの作成 その 2

「印刷 _ 演習用」スクリプトは、PDF を保存するとき、ファイル名と保存場所を指定する必要がありました。状況によっては自動的に指定された方が良いかもしれません。例えば、ファイル名になにも指定しないと「名称未設定 .pdf」という名前で保存されますが、内容がわかりづらいので取り扱いを間違ってしまうかもしれません。

11.8.1 パスを変数で指定する

第 10 章でデスクトップにバイナリファイルを保存したように PDF の保存先やファイル名も関数を使って変数として指定することが可能です。

FileMaker Pro にはパスを作成するために便利な関数があります。

```
Get( デスクトップパス )
```

→ 現在のユーザのデスクトップフォルダへのパスを返します。

```
FileMaker Pro で John Smith というユーザの場合:
Windowsは       /C:/Users/John Smith/Desktop/
macOSは         /Macintosh HD/Users/johnsmith/Desktop/
```

```
Get( ドキュメントパス )
```

→ 現在のユーザのドキュメントフォルダへのパスを返します。

```
FileMaker Pro で John Smith というユーザの場合:
Windowsは
  →  /C:/Users/John Smith/Documents/
macOSは
  →  /Macintosh HD/Users/johnsmith/Documents/

FileMaker Server は
Windowsは
  →  /C:/Program Files/FileMaker/FileMaker Server/Data/Documents/
macOSは
  →  /Macintosh HD/Library/FileMaker Server/Data/Documents/
```

```
Get( テンポラリパス )
```

→ 現在のユーザの FileMaker Pro で使用されている一時フォルダへのパス
または FileMaker Server で使用されるパスを返します。

※ これらの関数の結果は、OS の種類やバージョンによって異なる場合があります。

Get(ドキュメントパス) 関数などを使って PDF のファイル名と保存先パスを自動作成します。

・保存先は　　ドキュメントフォルダ
・ファイル名は　見積番号と保存の日時分秒（ファイル名がユニークになるような工夫です）

「演習 11_20」フォルダ

演習 11-20 変数（保存先のパスとファイル名）を使用する

- **使用ファイル：11_ 見積作成管理** .fmp12

(1)　［スクリプトワークスペース］で**「印刷 _ 演習用」スクリプト**を開きます。

(2)　Elself［Get(最終メッセージ選択) = 2］の 2-3 行下にある、［印刷設定］の次の行にスクリプトステップを入力します。

変数を設定

オプションを指定します。
・［名前：］は　「$PDF 保存場所」
・［値：］は　　［指定 ...］をクリックして、計算式を入力

```
Case (
Get ( システムプラットフォーム ) = 1 ; "filemac:" ;
Get ( システムプラットフォーム ) = -2 ; "filewin:" ;
"" )
 &
Get ( ドキュメントパス ) &
見積管理::見積番号 & "-" &
Filter ( Get(タイムスタンプ) ; "0123456789" ) &
".pdf"
```

Get(タイムスタンプ) は「2018/10/25 2:39:46」のような結果となるので、
Filter 関数で数字だけにしてファイル名に使用します。

(3)　［レコードを PDF として保存］スクリプトステップのオプションを変更します。
オプションは、
・ダイアログを　　**オフ**
・［既存の PDF に追加］は　**オフ**（そのまま）
・［出力ファイルの指定］は　［指定 ...］をクリックして、変数を指定します。

$PDF 保存場所

・［オプションの指定］は　　［指定 ...］をクリックして「現在のレコード」を選択します。
・［ディレクトリの作成］は　**オフ**（そのまま）

(4)　次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
カスタムダイアログを表示 [ タイトル: "印刷またはPDF"; "見積発注書を印刷またはPDF保存しますか"]

エラー処理 [ オン]
If [ Get(最終メッセージ選択) = 1 ]

    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ]

    印刷 [ 記憶する:(プリンタ名); ダイアログあり : オン]

Else If [ Get(最終メッセージ選択) = 2 ]

    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ]

    変数を設定［$PDF保存場所 ; 値:(計算式は省略)］

    レコードをPDFとして保存[ 記憶する ; ダイアログあり : オフ ; $PDF保存場所 ; 現在のレコード ; フォルダを作成 : オフ]

Else
End If

If [ Get(最終エラー) ]
    カスタムダイアログを表示 [ タイトル: "エラー"; "キャンセルされました"]
End If
エラー処理 [ オフ]

レイアウト切り替え [ 元のレイアウト ]
```

(5) 動作を確認します。
   a. 523 ページのようにカスタムダイアログの 3 つのボタンをすべてクリックして動作を確認してください。
   b. PDF の保存先やファイル名を確認してください。

(6) デバッガとデータビューアを表示して確認します。
変数などを確認できます。

演習ここまで

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_21」フォルダの中の「11_ 見積作成管理 .fmp12」ファイルをベースに進めます）

11.8.2 検索をスクリプトで行う

カスタム App では検索は重要な機能です。ユーザが対象レコードを作成するために必要です。もちろん、検索のために以下のようなスクリプトステップが用意されています。

［検索モードに切り替え］
［新規レコード／検索条件］
［検索実行］
［クイック検索］
［対象レコードの絞り込み］
［対象レコードの拡大］
［検索条件を変更］
［レコードを対象外に］
［一致するレコードを検索］

［検索実行］スクリプトステップ
オプションになにも指定しないと検索が実行されます。

検索条件をこのスクリプトステップに直接指定すると、その条件で検索が実行されます。あらかじめ決まった検索条件で検索する場合に使用できます。ただ、パッと見てどのような条件で検索しているのかよくわからないため注意が必要です。なお、検索条件には演算子、文字列、変数名のみを検索条件として指定できます。

これらのスクリプトステップを使ってメニューやステータスツールバーのアイコンをクリックしておこなう検索と同じようにスクリプトで検索を実行できます。

(1) 検索モードに切り替える
(2) 検索条件を入力する
(3) 検索を実行する
(4) 結果が表示される

基本的な流れは上記となりますが、さらにエラー処理が必要です。

(1) 検索モードに切り替える
(2) 検索条件を入力する
(3) 検索を実行する
(4) 結果がある
(5) 結果が表示される（ブラウズモードに戻る）

あるいは、

(1) 検索モードに切り替える
(2) 検索条件を入力する
(3) 検索を実行する
(4) 結果がない（エラー）
(5)（結果がない場合の動作） → 検討対象

です。検索の結果がない場合、どうするべきかを考えます。

入力された検索条件に合致する結果がなければ FileMaker Pro のエラーコードが返ります（「401」= 検索条件に一致するレコードがありません）。どうするべきなのでしょうか。いくつか考えられます。

通常、手作業で検索すると、条件に合致するレコードがない場合、下図のエラーメッセージが表示されます。



図 11-70 手作業で検索をおこなったエラーメッセージ

［キャンセル］をクリックすると、検索前の対象レコードが表示されます。［検索条件変更］をクリックすると、検索モードに切り替ります。

［検索実行］スクリプトステップで検索を実行して、エラー処理をしない場合、下図のエラーメッセージが表示されます。



図 11-71［検索実行］スクリプトステップで検索をおこなったエラーメッセージ

［キャンセル］をクリックすると、スクリプトを「中断」します。検索前の対象レコードが表示されるかどうかは［検索実行］スクリプトステップまでのスクリプト内容により変わります。［検索条件変更］をクリックすると、検索モードに切り替ります。［続行］をクリックすると、スクリプトが続行します。

そして、［検索実行］スクリプトステップで検索を実行して、エラー処理をする場合、開発者が自分でエラーの結果スクリプトでどう対応するか考えなければなりません。

例えば、

（1）検索前に「見積リスト」レイアウトで対象レコードが 3 件
（2）スクリプトでエラー処理をして検索を実行
（3）結果が無し（エラーがあったら）、ブラウズモードに戻るとスクリプトに記述する

この場合、元の対象レコードに戻れません。そのため、

・0 件であるという結果を表示する

という動作になります。なお、「元の対象レコードに戻る」スクリプトステップは用意されていません。検索前に複数のスクリプトステップを組み合わせて、元の対象レコードを保持するための仕組みを作成する必要があります（本書では解説しません）。

必ずしも元の対象レコードに戻らなくてはならないわけではありません。0 件の結果が表示されても良い場合がありますし、見つかるまで検索条件を変えて検索をし続けるようにスクリプトで［Loop］を使って動作を作成することも可能です。ユーザが検索機能を使う場面で、結果が得られなかった（エラーがあった）場合、どのような動作にするか要件定義時などに決めておきます。

「見積作成管理 .fmp12」では、

・0 件であるという結果を表示する

これを採用します。但し「結果が 0 件」ということをユーザにダイアログで知らせます。

(1) 検索モードに切り替える
(2) 検索条件を入力する
(3) 検索を実行する
(4) 結果がない
(5) ブラウズモードに戻る
(6)「結果 0 件」をカスタムダイアログで表示

また、「見積リスト」レイアウトで検索することにします。なお、

・見積先の顧客の電話番号や、見積書の管理番号などで検索したい

という要件もあります。**決まった条件（注文あり）で検索をスクリプトで行います。**

「演習 11_21」フォルダ

演習 11-21 条件の決まった検索

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　演習ファイルを確認します。

(2)　「見積リスト」レイアウトに切り替えます。レイアウトが変更されています。
「検索 _ 演習用（条件あり）」スクリプトを用意して「見積リスト」レイアウトの「注文あり」ボタンに割り当てました。



図 11-72 検索のボタン

(3) **「検索 _ 演習用（条件あり）」スクリプト**を編集します。ヘッダ情報のみがあります。

(4) 8 行目あたりにスクリプトステップを入力します。

```
検索モードに切り替え
```

オプションは　一時停止を　**オフ**　他は設定しません。

(5) 次の行を入力します。

```
レイアウト切り替え
```

オプションは 「見積リスト」レイアウトを指定します。

(6) 次の行を入力します。

```
フィールド設定
```

オプションは
・[ターゲットフィールドの指定] は [指定 ...] をクリックして、「見積管理 :: 見積結果」フィールドを指定
・[計算結果 :] は、[指定 ...] をクリックして、「注文あり」を入力

(7) 次の行を入力します。

```
レコード／検索条件確定
```

オプションは　ダイアログを　**オン**　そのまま

(8) 次に 2 行を入力します。

```
エラー処理 [ オン ]
検索実行
```

[エラー処理] スクリプトステップは、 **オン** にします。
[検索実行] スクリプトステップは、なにもオプションを設定しません。

(9) [If] スクリプトステップを入力します。現在、次のようなスクリプトのはずです。

```
検索モードに切り替え [ 一時停止 : オフ ]
レイアウト切り替え [ 「見積リスト」 ( 見積管理 ) ]
フィールド設定 [ 見積管理 :: 見積結果 ; "注文あり" ]
レコード／検索条件確定 [ ダイアログあり : オン ]
エラー処理 [ オン ]
検索実行
If [ ]
End If
```

(10) [If] スクリプトステップに次の計算式を設定します。
Get(最終エラー) 関数で、エラーを取得して条件分岐をします。

```
Get( 最終エラー )
```

(11) [If] スクリプトステップの次の行を入力します。

```
[カスタムダイアログを表示]
```

オプションは、
・[タイトル：] は 「結果なし」と入力
・[メッセージ：] は 「注文あり」の見積書はありません。 と入力

ボタンは
・[デフォルトボタン] は 「閉じる」「データを確定」は **オフ**
・[ボタン 2] は 設定しない
・[ボタン 3] は 設定しない

(12) [End If] スクリプトステップの下に [エラー処理] スクリプトステップを入力してオプションを **オフ** にします。

(13) 次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
検索モードに切り替え [ 一時停止 : オフ ]

レイアウト切り替え [ 「見積リスト」( 見積管理 ) ]

フィールド設定 [ 見積管理 :: 見積結果 ; " 注文あり " ]
レコード／検索条件確定 [ ダイアログあり : オン ]

エラー処理 [ オン ]
検索実行

If [ Get( 最終エラー ) ]
    カスタムダイアログを表示 [ " 結果なし "; "「注文あり」の見積書はありません " ]
End If
エラー処理 [ オフ ]
```

(14) 動作を確認します。
a. 「見積リスト」レイアウトを表示します。
b. ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。
c. 「注文あり」ボタンをクリックします。
d. 12 件が検索されます。

(15) デバッガとデータビューアを表示して確認します。

(16) エラー状態を確認するには、エラーになるようなテストデータを用意しても良いですし、次のようにエラーになるようにスクリプトステップを加工してテストしても良いかもしれません。ただし、元に戻すのを忘れないでください。

```
フィールド設定 [ 見積管理 :: 見積結果 ; "あいうえお" ]
```

演習ここまで

検索条件と引数

検索条件としてスクリプト内で［フィールド設定］するデータは、Get (スクリプト引数) 関数を使ったり、変数名を指定することも可能です。

例えば、注文あり／注文なしを引数として受け渡せば、1 つのスクリプトで別の検索結果を得られます。その場合、引数を変数として保持しておき、エラー処理後のカスタムダイアログの文言に使用するとわかりやすくなります。

ユーザが検索条件を入力する

ユーザが条件を入力する検索内容も、ほぼ同じです。次は、「検索 _ 見本」スクリプトの内容です。「見積リスト」レイアウトでボタンをクリックすると、検索モードで一時停止して検索条件の入力を待ちます。検索実行は［return］や［Enter］キーを押すことを期待しています。

```
ユーザによる強制終了を許可 [ オフ ]
検索モードに切り替え [ 一時停止 : オン ]
レイアウト切り替え [ 「見積リスト」( 見積管理 ) ]

レコード／検索条件確定 [ ダイアログあり : オン ]

エラー処理 [ オン ]
検索実行

If [ Get( 最終エラー ) ]
    カスタムダイアログを表示 [ "結果なし"; "検索した条件の見積書はありません" ]
End If
エラー処理 [ オフ ]
```



図 11-73 ユーザが条件を入力する検索ボタン

11.8.3 ［ユーザによる強制終了を許可］スクリプトステップ

一時停止などで、ユーザがスクリプトをキャンセルできないようにします。

「検索 _ 見本」スクリプトでは、ユーザが条件を指定する間、スクリプトを一時停止して待っていますが、［検索モードに切り替え］の「一時停止オプション　**オン**」のみの場合、次図のようにレイアウトバーに［続行］［キャンセル］が表示されます。［キャンセル］をクリックすると、検索モードからブラウズモードに戻って、スクリプトも中断してしまいます。



図 11-74 ユーザによる強制終了許可が　無し　か　オフ　の一時停止

［キャンセル］を表示させないようにするには、スクリプトの冒頭に［ユーザによる強制終了を許可］スクリプトステップを入力して「**オフ**」にします。

「印刷 _ 演習用」スクリプトの［印刷］ダイアログで［キャンセル］することを想定するには、次のようなスクリプトでも問題ありません（標準のエラーメッセージが表示されなくなります）。

```
ユーザによる強制終了を許可 [ オフ ]
カスタムダイアログを表示 [ タイトル: "印刷またはPDF"; "見積発注書を印刷またはPDF
保存しますか" ]
If [ Get(最終メッセージ選択) = 1 ]
    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ ]
    印刷 [ 記憶する: (プリンタ名) ; ダイアログあり : オン ]
Else If [ Get(最終メッセージ選択) = 2 ]
    レイアウト切り替え [ 「見積発注書｜印刷用」(見積管理) ]
    印刷設定 [ 記憶する ; ダイアログあり : オフ ]
    変数を設定 [$PDF保存場所 ; 値: (計算式は省略)]
    レコードをPDFとして保存 [ 記憶する ; ダイアログあり : オフ ; $PDF保存場所 ;
現在のレコード ; フォルダを作成 : オフ ]
Else
End If
レイアウト切り替え [ 元のレイアウト ]
```

11.8.4 ［フィールド設定］スクリプトステップ

「検索 _ 演習用（条件あり）」スクリプトでは、検索条件を入力するために［フィールド設定］を使用しました。このスクリプトステップもよく使うものです。オプションは 2 つあります。

ターゲットフィールド
データを設定するフィールド名を指定します。

計算結果
設定するデータを指定します。フィールドを空欄にしたい場合は「""」と指定します。もちろん計算式で指定することも変数名を指定することもできます。

なお、このスクリプトステップには注意点があります。

TO に注意する
ターゲットフィールドや、計算結果は、そのスクリプトが実行されるレイアウトの TO やコンテキストと合致していなければなりません。但しレイアウト上に配置されている必要はありません。

フィールド設定を利用して、グローバル格納フィールドを利用した検索を行うことができます。「検索 _ 見本（グローバル）」スクリプトは次の内容です。

```
検索モードに切り替え [ 一時停止 : オフ ]
レイアウト切り替え [ 「見積リスト」( 見積管理 ) ]
フィールド設定 [ 見積管理::見積状況 ; z_ユーティリティ::g_テキスト 01]
フィールド設定 [ 見積管理::見積結果 ; z_ユーティリティ::g_テキスト 02]
レコード／検索条件確定 [ ダイアログあり : オン ]
エラー処理 [ オン ]
検索実行
If [ Get( 最終エラー ) ]
    カスタムダイアログを表示 [ " 結果なし " ; " 検索した条件の見積書はありません " ]
End If
エラー処理 [ オフ ]
```

このスクリプトでは、ユーザが値一覧選択したグローバル格納フィールドのデータをフィールド設定で検索条件に使用しています。次図のボタンに割り当ててありますので試してみてください。



図 11-75 グローバル格納フィールドに入力されたデータを使用して条件を入力する例のボタン

フィールド設定は、変数や計算式も使用できます。例えば、「今日」が期限の見積書を検索することも可能です。Get (日付) 関数の計算式で検索条件を設定します。「検索 _ 見本（今日が期限）」スクリプトの内容は次のようになっています

```
検索モードに切り替え [ 一時停止 : オフ ]
レイアウト切り替え [ 「見積リスト」 ( 見積管理 ) ]

フィールド設定 [ 見積管理::見積状況 ; "お客様検討中" ]
フィールド設定 [ 見積管理::見積結果 ; "注文なし" ]
フィールド設定 [ 見積管理::期限日 ; "..." & Get ( 日付 ) ]
フィールド設定 [ 見積管理::見積結果 ; z_ユーティリティ::g_テキスト 3 ]
レコード／検索条件確定 [ ダイアログあり : オン ]
エラー処理 [ オン ]
検索実行
If [ Get( 最終エラー ) ]
    カスタムダイアログを表示 [ "結果なし" ; "検索した条件の見積書はありません" ]
End If
エラー処理 [ オフ ]
```

このスクリプトでは、Get (日付) 関数の結果を検索条件に使用しています。「...」は、範囲の演算子です。「今日」を含めた過去からの期限のものが検索されます。下図のボタンに割り当ててありますので試してみてください。



図 11-76 計算式を条件に利用する例のボタン

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_22 から 26」フォルダの中の「11_見積作成管理 .fmp12」ファイルをベースに進めます）

11.8.5 ウインドウでデータを選択

いくつか機能が出来てきました。また別のレイアウトについて考えます。

「見積詳細」レイアウトでは、見積書を提示する先であるお客様や、明細行の商品を選択入力します。「顧客情報」テーブルには 99 レコード、「商品情報」テーブルには 50 レコードが登録されています。ユーザにポップアップメニューで選択してもらおうとするとちょっと不便です。

ユーザにとっては、顧客や商品を選択する操作ですが、機能としては主キーを外部キーに［フィールド設定］して、関連を持たすものです。

「見積詳細」レイアウトで見積先である「顧客情報」を指定する（外部キーにデータを入力する）ための演習をおこないます。ポップオーバーとポータルとスクリプトを使って機能をつくります。

演習ファイルには、ポップオーバー上にポータルを配置するためにリレーションシップグラフに「見積管理 _ 顧客｜選択」TO を作成しました。

関連の定義は､「見積情報」の常に 1 である「_1」フィールドと「顧客情報」の「取引状態」フィールド（「1」は取引中）が照合フィールドとして指定されています。

常に取引中の顧客レコードから選択してもらうためです。

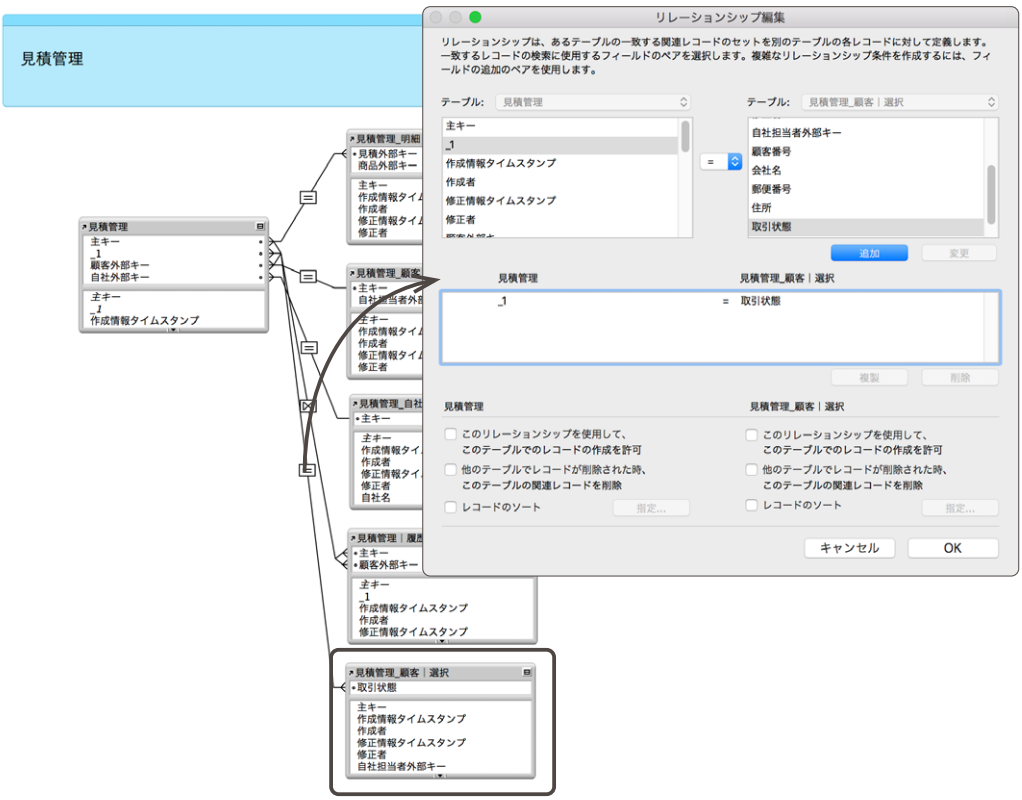


図 11-77 顧客選択のための TO

「見積詳細」レイアウトの人アイコンはポップオーバーボタンとしました。次図のように、ポップオーバー上に「見積管理 _ 顧客｜選択」TO の関連レコードを表示するポータルを作成しました。「←」ボタンに演習で使うスクリプトを割り当て済みです。

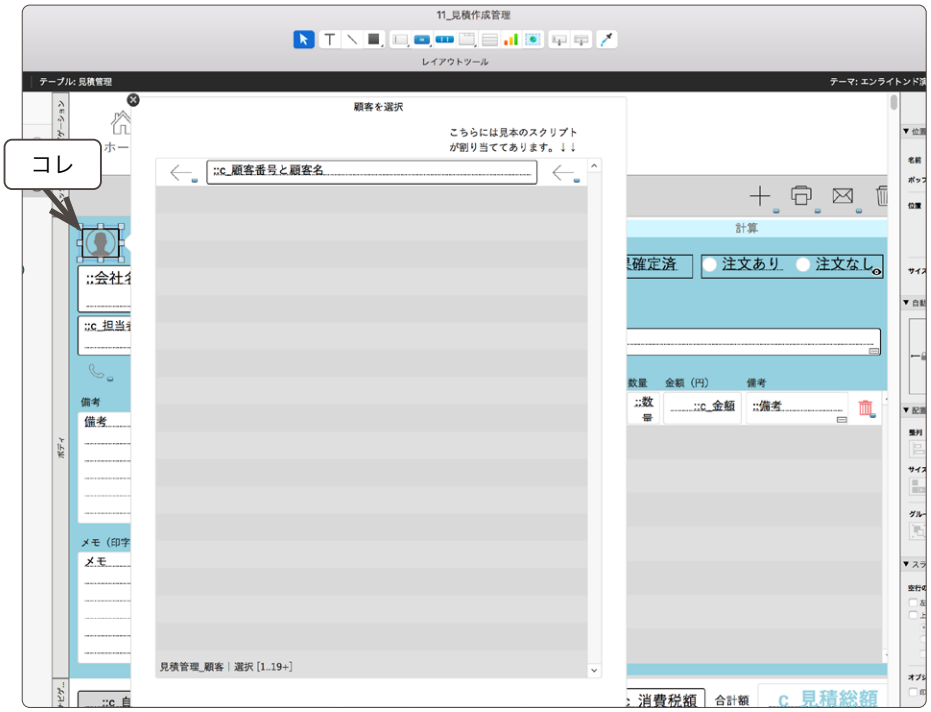


図 11-78「顧客を選択するポップオーバーボタン

スクリプトの流れを考えます。

(1) いつ　　ユーザの任意
(2) どこで　　「見積詳細」レイアウト（「見積管理」TO）
(3) 誰が　　ユーザ
(4) なにを　　現在表示している「見積情報」レコード
(5) なぜ　　見積先の顧客を選択する
(6) どのように　　見積先の顧客をポータルから選択する
「見積管理」TO の「顧客外部キー」フィールドに
「見積管理 _ 顧客｜選択」TO の「主キー」フィールドのデータを入力して
関連を成立させる

これはスクリプトでは、次のようになります。

(1) ボタンをクリックする
(2) ポップオーバーが表示される
(3) ポータルで選択する顧客の「←」ボタンをクリックすると「見積情報」テーブルの「顧客外部キー」フィールドに選択した「顧客情報」レコードの「主キー」フィールドのデータを入力する
(4) ポップオーバーを閉じる

「演習 11_22 から 26」フォルダ

演習 11-22 外部キーを入力する（フィールド設定 1）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］で**「顧客を選択 _ 演習用」スクリプト**を開きます。

(2)　スクリプトにはヘッダ情報だけで、なにも内容はありません。8 行目にスクリプトステップを入力します。

フィールド設定

オプションは
・「ターゲットフィールドの指定」は、[指定 ...] をクリックして、「見積管理 :: 顧客外部キー」フィールドを指定
・［計算結果 : ］は、［指定 ...］をクリックして、「見積管理 _ 顧客｜選択 :: 主キー」を入力

(3)　次の行を入力します。

ポップオーバーを閉じる

(4)　次の行を入力します。

レコード／検索条件確定

オプションは　ダイアログを　**オン**　そのまま

(5)　次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
フィールド設定　[ 見積管理　::　顧客外部キー　;　見積管理_顧客｜選択　::
主キー ]

ポップオーバーを閉じる
レコード／検索条件確定　[ ダイアログあり　:　オン ]
```

(6)　「見積詳細」レイアウトで動作を確認します（デバッガやデータビューアも確認します）。
a.　10 件目のレコードに移動します。見積先顧客名は「バウヒニア株式会社」です。
b.　人アイコンボタンをクリックしてポップオーバーを表示します。
c.　顧客番号「20004」の「株式会社シャクヤク」の「←」ボタンをクリックします。
d.　見積先顧客名は「株式会社シャクヤク」に変わります。

演習ここまで

11.8.6 引数と変数を使ってターゲットフィールドを指定する

［フィールド設定］スクリプトステップは、ターゲットフィールドを設定する必要があるので引数や変数を使えません。

次のスクリプトステップでは使えます。

［フィールドを名前で設定］スクリプトステップ

ターゲットフィールドの設定を計算式、引数や変数などで指定できます（フィールド名は完全修飾名で指定しなければいけません）。

例えば、スクリプト引数を GetFieldName 関数を使ってフィールド名を指定し、それを変数としてターゲットフィールドの指定に利用すると、引数が A の場合は A フィールド、引数が B の場合は、B フィールドにデータを入力、というようなスクリプトが可能になります。

11.8.7 ［レコード／検索条件確定］スクリプトステップ

［フィールド設定］や［フィールドを名前で設定］は、フィールドの内容を変更するだけです。レコードの確定はしません。

スクリプトステップの後、実行したユーザによってレコードは開かれていることになります。他のユーザがそのレコードに入力することができません。しかし、スクリプトの内容によっては、モードが切り替えられたり、次のレコードへ移動したりなど、レコード確定が行われて確定後の状態になるかもしれません。意図せずに『レコードが開きっぱなしになって』他のユーザが使用できない事態が起きるかどうかについて考慮します。

スクリプトの実行中にレコードを開いておく必要があるのならばスクリプトの最後の方に［レコード／検索条件確定］を実行すれば良いことになります。そうでなければ一連のフィールド設定直後に確定し他のユーザが入力できない事態をなるべく回避します。

演習 11-23 外部キーを入力する（フィールド設定 2）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1) ［スクリプトワークスペース］を表示**「顧客を選択 _ 演習用」スクリプト**を開きます。次の内容にスクリプトを書き換えてください。手順は省略します。

```
変数を設定［$設定するフィールド ; 値:（GetValue ( Get(スクリプト引数) ; 1 )］
変数を設定［$設定する値 ; 値:（GetValue ( Get(スクリプト引数) ; 2 )］

フィールドを名前で設定 [ $設定するフィールド ; $設定する値 ]
ポップオーバーを閉じる
レコード／検索条件確定 [ ダイアログあり : オン ]
```

(2)　「見積詳細」レイアウトでレイアウトモードに切り替えて、ポップオーバーの「←」ボタンに引数を設定します。引数の内容は以下です。

```
GetFieldName ( 見積管理::顧客外部キー ) & ¶ &
見積管理_顧客|選択::主キー
```

(3)　レイアウトを保存してブラウズモードに戻って動作を確認します。

演習ここまで

11.8.8　ウインドウ

「見積詳細」レイアウトにはもう一つ選択が必要な項目がありました。商品です。

現在、商品は 50 レコードなので見積先と同じようにポータルからでも充分に選択が可能です。しかし、ポータルにはポータルを配置できません。つまりポータルに配置されたポップオーバーボタンのポップオーバー上にはポータルを配置することができません。

ウインドウは常に 1 つしか表示してはいけないわけではありません。複数のウインドウを表示することが可能です。その管理のために、次のスクリプトステップが用意されています。

[新規ウインドウ]
[ウインドウを選択]
[ウインドウを閉じる]
[ウインドウを調整]
[ウインドウの移動／サイズ変更]
[ウインドウタイトルの設定]

次のスクリプトステップには、新規ウインドウで関連レコードを表示するオプションがありました。

[関連レコードへ移動]

また、ウインドウの大きさは、

・メニューバーやステータスツールバーの表示

も関係することに注意します。ウインドウは流れで利用します。

(1) ウインドウを開く（レイアウトを指定する）
(2) 必要があればレイアウトを切り替える（メニューバーやステータスツールバーの表示も）
(3) ウインドウのサイズを調整する
(4) ウインドウで作業を実施する
(5) ウインドウを閉じる　あるいは　閉じない

ウインドウスタイル

ドキュメントウインドウ

FileMaker Pro の通常のウインドウスタイルです。

フローティングドキュメント
アクティブでもそうでなくても他のウインドウの上に表示されます。そして、このウインドウに表示されているレコードだけが編集可能なのではなく、他のウインドウも編集が可能です。

ダイアログ（モーダル）
ウインドウはユーザが閉じるまで唯一のアクティブなウインドウです。他のウインドウでの操作ができません。アクティブな間は FileMaker Pro を終了したり他のファイルを開いたりすることもできません。

カード
既に解説と演習をしました。

ウインドウのオプション設定

［新規ウインドウ］と［関連レコードへ移動］スクリプトステップでは次図の設定が可能です。

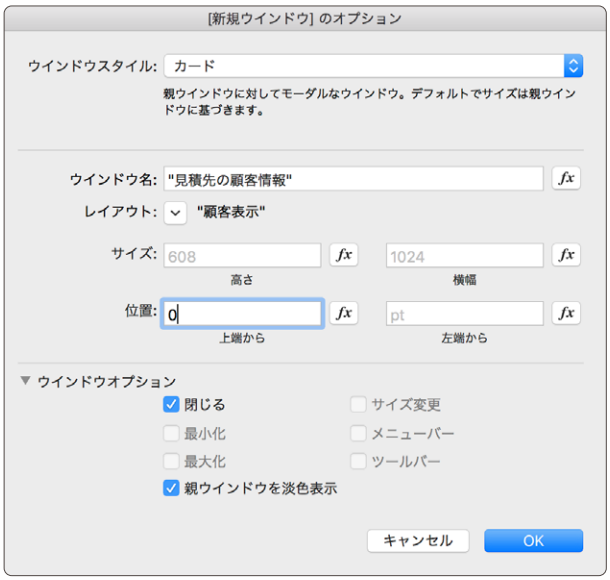


図 11-79［新規ウインドウ］と［関連レコードへ移動］スクリプトステップのウインドウオプションダイアログ

計算式を使用して設定できる
・ウインドウ名
・レイアウト
・サイズ
・位置
これらはすべて、計算式を使って設定することが可能です。

ウインドウオプション
ウインドウの操作についてユーザが、できることと／できないことを設定できます。この設定でチェックを　**オフ**　にしたものはメニューやショートカットも無効になります。

ウインドウを閉じる

ウインドウを開く場合は、必ず閉じる手段について考えます。テストして、閉じられなくなることがないかどうかを確認します。

例えば、ダイアログウインドウ（モーダル）スタイルで「閉じる」を **オフ** にすると、ウインドウを閉じるためのボタンが表示されなくなります。このような設定のウインドウをスクリプトから開いた場合、閉じるための手段が考慮されていないとユーザはウインドウを閉じられずFileMaker Pro の終了もできなくなり困ってしまいます。

商品を選択するウインドウ
これはどのような流れが必要でしょうか。

（1）いつ　　ユーザの任意
（2）どこで　　「見積詳細」レイアウト（「見積管理」TO）
（3）誰が　　ユーザ
（4）なにを　　現在表示している「見積明細」レコード
（5）なぜ　　商品を選択する
（6）どのように　　明細行の商品をウインドウ（「商品管理」TO）から選択する
「見積管理 _ 明細」TO の「商品外部キー」フィールドに
「商品管理」TO の「主キー」フィールドのデータを入力して関連を成立させる

これはスクリプトでは、次のようになります。

（1）ボタンをクリックする
（2）ウインドウが表示される
（3）ウインドウで選択する商品の「←」ボタンをクリックすると「見積明細」テーブルの「商品外部キー」フィールドに選択した「商品情報」レコードの「主キー」フィールドのデータを入力する
（4）ウインドウを閉じる

ポップオーバーのときと、ほぼ同じです。そしてどのスタイルのウインドウを設定したら良いでしょうか。他のウインドウを選択できないほうが良いので

・カードスタイル
・ダイアログ（モーダル）スタイル

どちらかにします。ここではカードスタイルを試します。

演習ファイルには、ウインドウで表示するための「商品選択」レイアウトをリスト表示で作成しました。ボタンも作成してスクリプトも割り当ててあります。あとはスクリプトを作業するだけです。

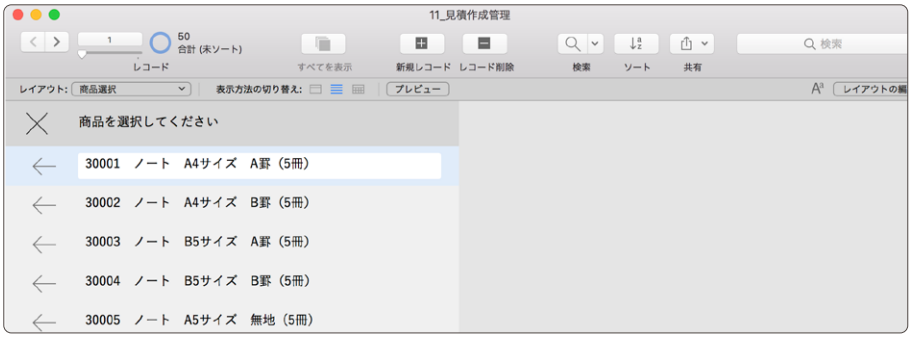


図 11-80 商品を選択するレイアウト

「見積詳細」レイアウトの次図のボタンにも演習で使うスクリプトを割り当て済みです。



図 11-81 ウインドウを開くボタン

ウインドウを開くためのスクリプトを作成します。

演習 11-24 外部キーを入力する（カードスタイルウインドウ）

- **使用ファイル：**11_ **見積作成管理** .fmp12

(1) ［スクリプトワークスペース］を表示して**「商品を選択 _ 演習用（ウインドウを開く）」スクリプト**を開きます。スクリプトにはヘッダ情報だけで、なにも内容はありません。

(2) 8 行目にスクリプトステップを入力します。

```
新規ウインドウ
```

オプションは、
・［ウインドウスタイル：］は「カード」
・［ウインドウ名：］は　「商品選択」
・［レイアウト：］は　「商品選択」レイアウト
・［サイズ：］は　高さ「Get(ウインドウ内容高さ)」 のみ設定
・［位置：］は　上端から、「0」 左端から、「Get(ウインドウ内容幅)–500」
・［ウインドウオプション］は「親ウインドウを淡色表示」だけを **オン**

(3) スクリプトを保存します。次が完成形です。「見積詳細」レイアウトで動作を確認します。

```
新規ウインドウ [ スタイル: カード; 名前: "商品選択"; 使用するレイアウト: 「商品選択」 ( 商品管理 ); 高さ: Get( ウインドウ内容高さ ) ; 上: 0 ; 左: Get( ウインドウ内容幅 )–500 ]
```

演習ここまで

商品を選択してウインドウを閉じるスクリプトを引き続き作業します。

演習 11-25 外部キーを入力する（引数を使う）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　「商品選択」レイアウトでレイアウトモードに切り替え「←」、ボタンに引数を設定します。
引数の内容は以下です。

```
GetFieldName ( 見積管理_明細::商品外部キー ) & ¶ &
商品管理::主キー
```

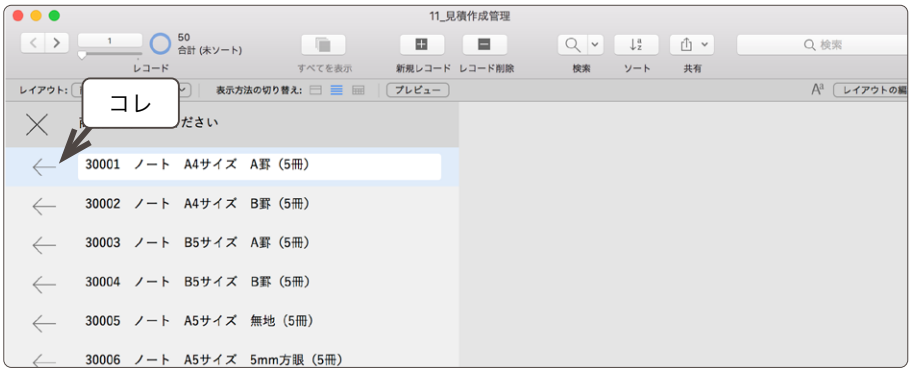


図 11-82 商品を選択するボタン

(2)　レイアウトを保存します。

(3)　「見積詳細」レイアウトで動作を確認します（デバッガやデータビューアも確認します）。
a. 13 件目のレコードに移動します。
b. 明細行は 3 行です。4 行目の箱アイコンボタンをクリックするとカードスタイルウインドウが表示されます。
c. 商品番号「30009」の「←」ボタンをクリックします。
d. 4 行目に商品番号「30009」が入力されます。

演習ここまで

ポップオーバーで顧客を選択したときには、取引中の顧客だけに関連を設定しました。商品も販売中の商品だけにする必要があります。スクリプトトリガとスクリプトで実現します。

トリガは次のものを利用します。

`OnLayoutEnter` スクリプトトリガ

演習ファイルに「商品を販売中のみに絞り込み」スクリプトを作成しておきました（見本フォルダに入っています）。**レイアウトのスクリプトトリガに割り当てます。**

演習 11-26 スクリプトトリガをレイアウトに割り当てる

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　「商品選択」レイアウトでレイアウトモードに切り替えます。

(2)　［レイアウト設定］ダイアログを開きます。

(3)　「スクリプトトリガ」タブを選択します。

(4)　「OnLayoutEnter」を　**オン**　にして［選択 ...］をクリックして、「商品を販売中のみに絞り込み」スクリプトを指定します。

(5)　レイアウトを保存します。

(6)　「見積詳細」レイアウトで動作を確認します（デバッガやデータビューアも確認します）。
商品番号 30002 から 30005 は販売中ではないので、表示されません。

演習ここまで

ウインドウ名

いま演習したスクリプトでは、ウインドウ名を指定しませんでした。それは、常に 1 つのウインドウのみが選択されている状態だったからです。複数のウインドウが表示されいている場合、毎回ウインドウ名を指定した方が確実です。

［ウインドウを閉じる］
［ウインドウを選択］

これらのスクリプトステップにはウインドウ名を指定するオプションがあります。

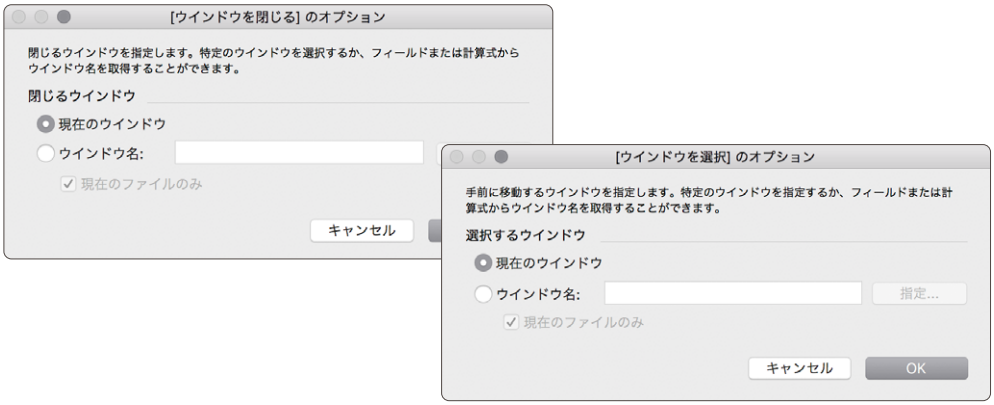


図 11-83　ウインドウ名を指定できる

1 つのカスタム App の中で、ウインドウ名は目的に応じてユニークになるようにすると複数のウインドウを開く場合に、取り扱いしやすくなります。

また、命名ルールを考えても良いでしょう。演習のウインドウ名の場合、

・商品選択

という名前にしました。なにかを選択するウインドウは「○○選択」という命名ルールにするなどと決定することでユニークにしやすく、開発時にも命名しやすくなります。

既に開いていないかどうか

新しいウインドウ名を開く場合に、同じ名前のウインドウが開かれていないか確認することが重要な場合があります。
・開きたい名前のウインドウを選択する
・エラー処理をする

この 2 つで条件分岐をします。ウインドウの選択の前にエラー処理をして、
・既に同じ名前のウインドウが開いている場合は新しく開かない
・同じ名前のウインドウが開いていない場合は新しく開く
この条件には、エラー処理で取得できるエラー番号「112（ウインドウが見つかりません）」を使用します（FileMaker Pro のエラー番号については、ヘルプを参照してください。）

これから開こうとするウインドウ名と同じものが既に開いていないかどうか確かめてから開く場合のスクリプトは、次のようになります。

```
エラー処理 [ オン ]

ウインドウを選択 [ 名前 : 商品選択 ; 現在のファイル ]

If [ Get( 最終エラー ) = 112 ]
    新規ウインドウ [ スタイル : ダイアログ ; 名前 : "商品選択"; 使用するレイアウト :
「商品選択」 ( 商品管理 ); 横幅 : 500; 上 : 5; 左 : Get( ウインドウ内容幅 )/2]
End If

エラー処理 [ オフ ]
```

カードスタイルウインドウとウインドウを選択

上記のように、既に同じ名前のウインドウが開いているか確認すると、カードスタイルウインドウであっても同じ名前であれば合致して、そのカードスタイルウインドウが開いている親ウインドウごと選択されます。

ウインドウのサイズ

クライアントの OS によってウインドウ中の表示領域が違う場合があります。ウインドウの大きさはウインドウの外側の大きさなので、OS や OS のバージョンによって少しずつ大きさが違います。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_27 から 28」フォルダの中の「11_見積作成管理 .fmp12」ファイルをベースに進めます）

11.9 [Loop] で反復

スクリプト内で「ループ処理」をすることも可能です。ループとは『ある条件』で終了するまで処理を反復することです。［Loop］スクリプトステップを使用します。［If］と［End If］のように、

```
Loop
End Loop
```

と、セットで使うスクリプトステップです。ループの終了条件は、

```
Exit Loop IF [ ]
```

で、指定します。これらを組み合わせて、

```
Loop
    繰り返す内容
    Exit Loop IF [ 終了する条件 ]
End Loop
```

このように使います。例えば、次のような処理です。

「見積作成管理 .fmp12」では、

・期限日を 30 日以上過ぎているが、顧客から何も連絡がない（督促にも返事がない）ものは、「注文なし」とする。

という、業務ルールがあることがわかりました。そのための機能は、

（1）期限日を 30 日以上過ぎた見積を検索
（2）検索結果の対象レコードの 1 件目に移動
（3）以下を入力する
　「見積状況」フィールドに「結果確定済」
　「見積結果」フィールドに「注文なし」
（4）入力したら次のレコードに移動
（5）反復する
（6）対象レコードの最後まで入力したら終了

このような内容です。［Loop］を使わなくても全置換でできそうですが、確実に入力するために［フィールド設定］と［レコード確定］を 1 レコードずつに実行します。

※ スクリプト内容を工夫することで入力できなかったレコードを記録することもできますが、本書では演習しません。

前述の例は、[Exit Loop If] スクリプトステップを使わなくてもよいものです。それは次のスクリプトステップを使うからです。

［レコード／検索条件／ページへ移動］

オプションで「次の」で「最後まできたら終了」のチェックを　**オン**　にすると最後のレコードに移動後に反復処理が終了します。

［Loop］の前に最初のレコードに移動することを忘れないようにします。次のように使います。

```
Loop
    繰り返す内容
    ［レコード／検索条件／ページへ移動］
End Loop
```

なお、処理をおこなう内容やレコードの件数にもよりますが、ユーザが見ている画面の中で、レイアウトが目まぐるしく変化したり、レコードが移動する様子が見えてよいかどうかを考えます。

ウインドウの固定
処理中の画面の内容をユーザに表示しないようにする場合は、Loop の前に

［ウインドウの固定］スクリプトステップ

を入力します。スクリプトが終了すると固定が解除されます。スクリプトの途中でウインドウの固定を解除する場合、次のどちらかのスクリプトステップを使います。

［ウインドウ内容の再表示］
［スクリプトの一時停止 / 続行］

本章の最後は、［Loop］を使った反復処理を演習します。

演習 11-27 と 28 のスクリプトは、これまでに作成した

・レイアウト切り替え
・レコード移動
・検索
・フィールド設定
・エラー処理
・ウインドウ固定
・PDF 保存
・変数
・現在のスクリプト終了
・関数と計算式

これらを使います。

演習ファイルに、次のスクリプトを作成しました。

・「返答なし見積の処理 _ 演習用」スクリプト
・「見積一括保存 _ 演習用」スクリプト

「見積リスト」レイアウトにボタンを作成して上記のスクリプトを割り当ててあります。あとは、スクリプトを作業するだけです。



図 11-84 ループ処理をおこなうボタン

『「見積情報」テーブルのレコードで『期限日を 30 日以上過ぎたものを「注文なし」にする』ために、

・「見積状況」フィールドに「結果確定済」
・「見積結果」フィールドに「注文なし」

上記のデータを、条件に合致するレコードに連続入力します。

「演習 11_27 から 28」フォルダ

演習 11-27 ループ処理 1（検索＋フィールド設定）

- **使用ファイル：11_ 見積作成管理 .fmp12**

(1)　[スクリプトワークスペース] を表示して**「返答なし見積の処理 _ 演習用」スクリプト**を開きます。スクリプトにはヘッダ情報だけで、なにも内容はありません。

次の内容にスクリプトを書き換えてください。手順は省略します。

```
検索モードに切り替え [ 一時停止 : オフ ]

レイアウト切り替え [ 「見積リスト」( 見積管理 ) ]

フィールド設定 [ 見積管理::期限日; "..." & Get ( 日付 ) -30 ]

新規レコード / 検索条件
フィールド設定 [ 見積管理::見積状況; "結果確定済" ]
レコードを対象外に
レコード / 検索条件確定

エラー処理 [ オン ]
検索実行

If [ Get( 最終エラー ) ]

    カスタムダイアログを表示 [ "対象なし"; "対象のデータはありません。作業を終了します" ]

Else

End If
エラー処理 [ オフ ]
```

何行か空けておくと次の作業が楽です。

必要なオプションのみ記述します。

[カスタムダイアログを表示] スクリプトステップのオプションは、

・[タイトル：] は　「対象なし」と入力
・[メッセージ：] は 「対象のデータはありません。作業を終了します」 と入力

ボタンは

・[デフォルトボタン] は 「閉じる」「データを確定」は **オフ**
・[ボタン 2] は　設定しない
・[ボタン 3] は　設定しない

(2)　[Else] スクリプトステップの下に、次のスクリプトステップを入力します。

```
ウインドウの固定
```

(3)　次の行は、

```
レコード / 検索条件 / ページへ移動
```

・オプションに「最初の」を指定します。

(4)　次の行に、[Loop] スクリプトステップを入力します。

```
Loop
End Loop
```

(5)　[Loop] の次の行に、[フィールド設定] スクリプトステップを 2 回入力します。

```
Loop
      フィールド設定
      フィールド設定
End Loop
```

1 つめの [フィールド設定] スクリプトステップのオプションは、
・「ターゲットフィールドの指定」は　[指定 ...] をクリックして、「見積管理 :: 見積状況」フィールドを指定
・[計算結果 :] は　[指定 ...] をクリックして、「結果確定済」を入力

2 つめの [フィールド設定] スクリプトステップのオプションは、
・「ターゲットフィールドの指定」は　[指定 ...] をクリックして、「見積管理 :: 見積結果」フィールドを指定
・[計算結果 :] は　[指定 ...] をクリックして、「注文なし」を入力

(6)　2 つ目の [フィールド設定] の次の行は、

```
レコード / 検索条件 / ページへ移動
```

・オプションに「次の」を指定して、最後まで来たら終了を　**オン**　にします。

(7)　[End Loop] の下の行は、

```
レコード / 検索条件確定
```

(8)　次の行は、

```
カスタムダイアログを表示 [ "作業終了"; "作業がおわりました" ]
```

［カスタムダイアログを表示］スクリプトステップのオプションは、
・［タイトル：］は　「作業終了」と入力
・［メッセージ：］は 「作業がおわりました」と入力
ボタンは
・［デフォルトボタン］は　「閉じる」「データを確定」は **オフ**
・［ボタン 2］は　　設定しない
・［ボタン 3］は　　設定しない

(9)　次のスクリプトの完成形内容になるよう作業し保存します（詳細な手順は省略します）。

```
検索モードに切り替え [ 一時停止 : オフ ]

レイアウト切り替え [ 「見積リスト」( 見積管理 ) ]

フィールド設定 [ 見積管理::期限日 ; "..." & Get ( 日付 ) -30 ]

新規レコード / 検索条件
フィールド設定 [ 見積管理::見積状況 ; "結果確定済" ]
レコードを対象外に
レコード / 検索条件確定

エラー処理 [ オン ]
検索実行 [ ]
If [ Get( 最終エラー ) ]

    カスタムダイアログを表示 [ "対象なし" ; "対象のデータはありません。作業を終了します" ]

Else

    ウインドウの固定
    レコード / 検索条件 / ページへ移動 [ 最初の ]
    Loop
        フィールド設定 [ 見積管理::見積状況 ; "結果確定済" ]
        フィールド設定 [ 見積管理::見積結果 ; "注文なし" ]
        レコード / 検索条件 / ページへ移動 [ 次の ]
    End Loop
    レコード / 検索条件確定

    カスタムダイアログを表示 [ "作業終了" ; "作業がおわりました" ]
End If
エラー処理 [ オフ ]
```

(10) 「見積リスト」レイアウトで動作を確認します。

a. 「期限切れ処理」ボタンをクリックします。



図 11-85「期限切れ処理」ボタン

b. 対象レコードがある場合には、そのレコードが処理されて表示されます。
「見積状況」フィールドに「結果確定済」、「見積結果」フィールドに「注文なし」となり、表示がグレーになっているはずです。

もしもスクリプトの内容が間違っていて処理を失敗してしまったときのために、「演習 11-27 のデータを元に戻す」スクリプトを作成してあります。スクリプトメニューに表示されるようになっていますので、実行してください。

演習ここまで

ループ処理の終了について、バリエーションを確認します。

「見積作成管理 .fmp12」では、『指定した発行日の見積』を 1 件ごとに個別の PDF ファイルとして保存する機能も必要なことがわかりました。そこで、この作業のためのボタンと「見積一括保存 _ 演習用」スクリプトを作成しました。

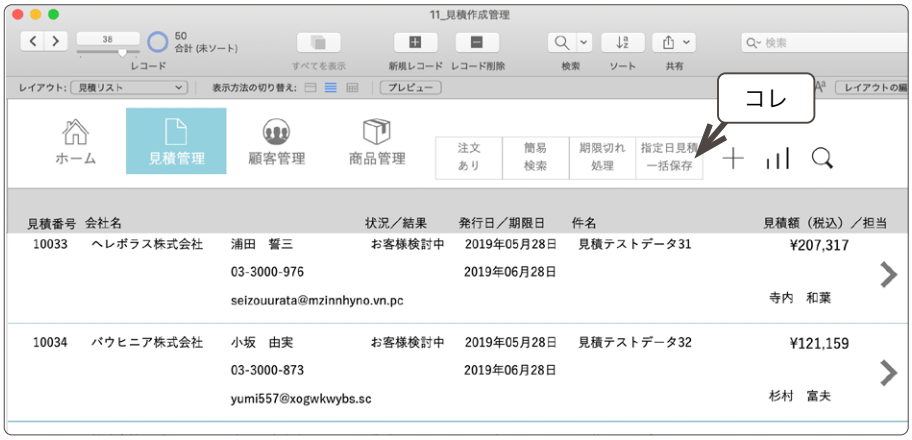


図 11-86 PDF 保存ループ処理を実行するボタン

このスクリプトは長いので、既に作成済みです。「見積リスト」レイアウトで「指定日見積一括保存」ボタンをクリックして動作を確認すると、例えば「2019/5/28」では 7 件のファイルが保存されます。

```
ウインドウの固定

Loop
    Exit Loop If [ not IsEmpty ( $指定発行日 ) ]

    カスタムダイアログを表示 [ "日付を指定してください" ; $指定発行日 ]

    If [ Get(最終メッセージ選択)=2 ]
        現在のスクリプト終了 [ ]
    End If
End Loop

検索モードに切り替え [ 一時停止:オフ ]
レイアウト切り替え [ 「見積リスト」(見積管理) ]
フィールド設定 [ 見積管理::発行日 ; GetAsDate ( $指定発行日 ) ]
レコード/検索条件確定

エラー処理 [ オン ]
検索実行 [ ]
If [ Get(最終エラー) ]

    カスタムダイアログを表示 [ "対象なし" ; "対象のデータはありません。作業を終了します" ]

Else

    レイアウト切り替え [ 「見積発注書|印刷用」(見積管理) ]
    レコード/検索条件/ページへ移動 [ 最初の ]

    Loop
        変数を設定 [ $PDF保存場所; 値:(計算式省略) ]
        レコードをPDFとして保存[ 記憶する ; ダイアログあり : オン ; 現在のレコード ; フォルダ作成 : オフ ]
        レコード/検索条件/ページへ移動 [ 次の ; 最後まできたら終了:オン ]
    End Loop

    カスタムダイアログを表示 [ "作業終了" ; "作業がおわりました" ]

End If

エラー処理 [ オフ ]
レイアウト切り替え [ 元のレイアウト ]
```

ループが 2 箇所で使われています。最初のループは、

```
Loop
    Exit Loop If [ not IsEmpty ( $指定発行日 ) ]

    カスタムダイアログを表示 [ "日付を指定してください" ; $指定発行日 ]

    If [ Get(最終メッセージ選択)=2 ]
        現在のスクリプト終了 [ ]
    End If

End Loop
```

最初の［Exit Loop If］スクリプトステップでは、次の条件でループを終了します。

・変数「$ 指定発行日」に何かデータが保持されている

さらに次の条件では、ループを終了し、スクリプトも終了します。

・カスタムダイアログで「キャンセル」がクリックされた

2 番目のループは、前の演習と同じ内容です。この**ループの終了を他の方法に変えてみます**。

演習 11-28 ループ処理の終了方法

- **使用ファイル：11_ 見積作成管理** .fmp12

(1) ［スクリプトワークスペース］を表示し**「見積一括保存 _ 演習用」スクリプト**を開きます。
次の部分のスクリプトを書き換えてください。手順は省略します。2 番目のループ部分です。

```
レコード / 検索条件 / ページへ移動 [ 最初の ]
変数を設定 [ $カウント ; 値:1 ]

Loop
    Exit Loop If [ $カウント > Get( 対象レコード数 ) ]

    変数を設定 [ $PDF保存場所 ; 値:（計算式省略） ]
    レコードをPDFとして保存 [（省略）]
    レコード / 検索条件 / ページへ移動 [ 次の ; 最後まできたら終了:オフ ]
                                                    ↑ オフにしてください
    変数を設定 [ $カウント ; 値:$カウント + 1 ]
End Loop
```

(2) 「見積リスト」レイアウトで「指定日見積一括保存」ボタンをクリックして動作を確認します。
例えば「2018/5/28」では 7 件のファイルが保存されます。

演習ここまで

レコードを移動しながらループ処理を行うには、[レコード / 検索条件 / ページへ移動] スクリプトステップで「最後まできたら終了を**オン**」に設定するのが有効な方法です。しかし、それ以外のループ終了処理が必要な場合のために、変数で条件分岐する終了方法を試しました。

なお、「見積一括保存 _ 演習用」スクリプトでは、カスタムダイアログのデータ入力にグローバルフィールドではなく変数へ保存する方法を使いました。スクリプトの内容をデバッガやデータビューアで確認してみてください。

本章では、いくつかのスクリプトを作成しました。ユーザが使いやすく、開発者がメンテナンスしやすいスクリプトを目指すことが重要です。

これまでに作成されたり、不要になった画面は下表の通りです。

	画面名	レイアウト種類	パソコン	モバイル
1	見積登録・作業画面　→　見積詳細	フォーム形式	○	×
~~2~~	~~見積検索画面~~　→　見積リストで併用	~~フォーム形式~~	~~⊖~~	~~×~~
3	見積一覧画面　→　見積リスト	リスト形式	○	×
4	見積書（帳票）（印刷／ PDF）　→　見積発注書｜印刷用	フォーム形式	○	×
5	顧客登録・作業画面　→　顧客管理｜マスタ詳細	フォーム形式 （マスタ / 詳細）	○	×
~~6~~	~~顧客検索画面~~	~~フォーム形式~~	~~⊖~~	~~×~~
~~7~~	~~顧客一覧画面　→　顧客リスト~~	~~リスト形式~~	~~⊖~~	~~×~~
8	商品登録・作業画面　→　商品管理｜マスタ詳細	フォーム形式 （マスタ / 詳細）	○	×
~~9~~	~~商品検索画面~~	~~フォーム形式~~	~~⊖~~	~~×~~
~~10~~	~~商品一覧画面~~	~~リスト形式~~	~~⊖~~	~~×~~
11	自社担当者・アカウント情報登録・作業画面	フォーム形式 （マスタ / 詳細）	○	×
12	自社情報管理画面	フォーム形式	○	×
13	ホーム画面　→　ホーム	フォーム形式	○	×
（モバイル画面は省略）				

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 11_29 から 33」フォルダの中の「11_データファイル .fmp12」ファイルをベースに進めます）

11.10 データファイルの取り扱い

FileMaker のファイル（fmp12 形式）以外のファイル（外部データファイル）をスクリプトステップで取り扱うことができます。例えば、操作のログファイルを作成、独自形式のデータファイルをエクスポートして作成、などに利用できます。

データファイル

データファイルを取り扱うための一連のスクリプトステップが用意されています。

［データファイルを作成］
［データファイルを開く］
［データファイルから読み取る］　（開いているデータファイルからデータを読み取る）
［データファイルに書き込む］　（開いているデータファイルにデータを書き込む）
［データファイルの位置を取得］　（開いているデータファイルの読み取り／書き込み位置を取得）
［データファイルの位置を設定］　（開いているデータファイルの読み取り／書き込み位置を設定）
［データファイルを閉じる］

［データファイルを開く］のエラーコード

・同時に開くことができるデータファイルの数は最大 25 です。制限を超えた場合のエラーコードは 2（メモリエラー）です。
・すでに開いているデータファイルを開こうとした場合のエラーコードは 300（ファイルがロックされているか、使用中です）です。

現在開いているデータファイルを確認する関数

```
Get ( オープンデータファイル情報 )
```

開いているデータファイルのファイル ID とパスを返します。
データファイルに関する情報はファイルごとに改行区切になります。次の内容です。
<ファイル ID><タブ><パス><改行>
例：　15　file:/Macintosh HD/Users/ユーザ名 /Documents/change.log

その他

次のスクリプトステップは、FileMaker ファイルや他のファイルなどにも使えます。

［ファイルの存在を取得］
［ファイルを削除］
［ファイルの名前変更］

これらのスクリプトステップでは、次ページ図のような流れを想定しています。

データファイルの操作（作成と開く／閉じる）

データファイルの操作には、次のことが必要です。

・ファイルが存在していること
・ファイルが開いていること

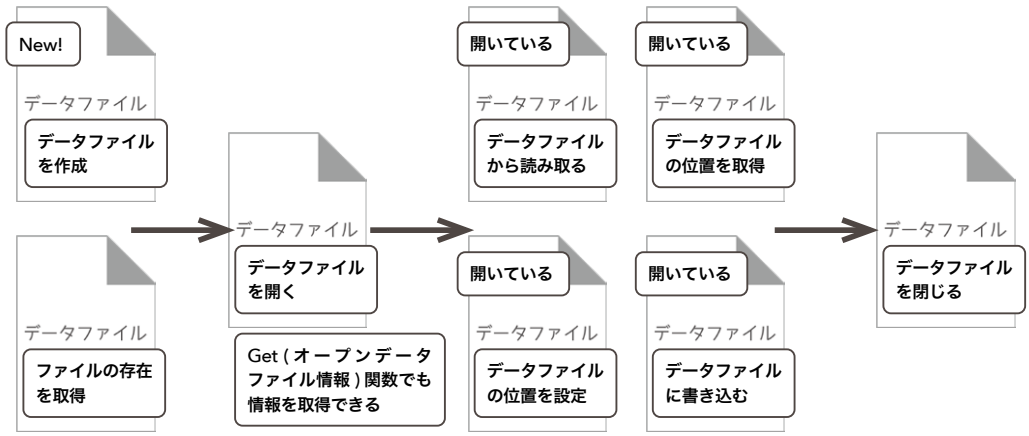


図 11-87 データファイルの取り扱いの流れ

まずは**ファイルの存在を確認し、なければ作成します。**演習ファイルで試します。

「演習 11_29 から 33」フォルダ

演習 11-29 データファイルの作成

- **使用ファイル：**11_ **データファイル** .fmp12

(1) ［スクリプトワークスペース］で**「存在確認と作成」スクリプト**を開きます。次の内容です。

```
レコード / 検索条件確定
#ファイル名がなければ動作しない
If [ IsEmpty ( データファイル例 :: ファイル名 ) ]
   カスタムダイアログを表示 [ "エラー" ; "必要な情報がありません。" ]
   現在のスクリプト終了 [ 結果 : "ファイル名がない" ]
End If

#あらかじめファイル名を想定する
変数を設定 [ $fileName; 値 :"file:" & Get( デスクトップパス ) &
"fmb2019fmb/" & データファイル例 :: ファイル名 & ".txt" ]

#ファイルの存在を確認する
ファイルの存在を取得 [「$fileName」; ターゲット : $resultFileName ]
If [ $resultFileName // ある ]
   カスタムダイアログを表示 ["ファイルあり" ]
Else
   #ファイルがなければ作成
   データファイルを作成 [「$fileName」; フォルダを作成 : オン ]
   カスタムダイアログを表示 [ "ファイルなし"]
End If

#レイアウト上の記号 {{ オープンデータファイル情報 }} を表示するため再表示
ウインドウ内容の再表示
```

この演習ではパソコンの「デスクトップ」に「fmb2019fmb」フォルダを作成、利用します。
すでに同名のフォルダが作成されている場合は、そのフォルダが利用されます。

(2)　実際に動作させます。

a.　「データファイル例」レイアウトでファイル名を入力します。
　・指定ファイル名　　データファイル演習

b.　「ファイル存在確認と作成」ボタンをクリックします。

c.　カスタムダイアログ「ファイルを作成しました」が表示されます。

d.　「OK」をクリックして閉じます。



図 11-88「データファイル例」レイアウト

(3)　デスクトップを確認します。
「fmb2019fmb」フォルダと、その中に「データファイル演習 .txt」ファイルができているはずです。

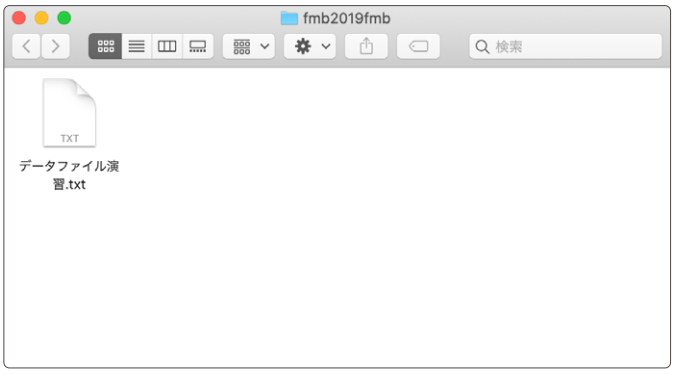


図 11-89 作成したフォルダとファイルの確認

演習ここまで

「データファイル例」レイアウトには、レイアウト上の記号 {{ オープンデータファイル情報 }} を配置しました。この記号は開いているデータファイルを表示します。しかし何も表示されません。

現在は、データファイルを作成しただけです。**データファイルを開きます**。

演習 11-30 データファイルを開く

- **使用ファイル：11_ データファイル .fmp12**

(1) ［スクリプトワークスペース］で**「存在確認、開く」スクリプト**を開きます。次の内容です。

```
レコード / 検索条件確定
#ファイル名がなければ動作しない
If [ IsEmpty ( データファイル例::ファイル名 ) ]
    カスタムダイアログを表示 [ "エラー" ; "必要な情報がありません。"]
    現在のスクリプト終了 [ 結果: "ファイル名がない" ]
End If

#ファイルをすべて閉じておく（これはスクリプト内容をシンプルにするためです）
データファイルを閉じる [ ファイル ID: "*" ]

#あらかじめファイル名を想定する
変数を設定 [ $fileName; 値:"file:" & Get(デスクトップパス) &
"fmb2019fmb/" & データファイル例::ファイル名 & ".txt" ]

#ファイルの存在を確認する
ファイルの存在を取得 [「$fileName」; ターゲット: $resultFileName ]
If [ $resultFileName // ある ]
    #ファイルあれば開く
    データファイルを開く [「$fileName」; ターゲット: $fileOpen ]
End If

#レイアウト上の記号{{オープンデータファイル情報}}を表示するために再表示
ウインドウ内容の再表示

#カスタムダイアログで知らせる
カスタムダイアログを表示 [ "ファイルの状態" ; "fileOpen変数＝" &
$fileOpen & ¶ & "(数字は開いているファイルIDです)"]
```

［データファイルを開く］スクリプトステップの実行後、fileOpen 変数に入力されるデータは「ファイル ID」です。

(2) 実際に動作させます。

a. 「ファイル存在確認、開く」ボタンをクリックします。
b. カスタムダイアログ「ファイルの状態」が表示されます。
fileOpen 変数 = の右に表示されている数字が「ファイル ID」です。
c. 「OK」をクリックして閉じます。
d. 「現在開いているデータファイル」にもファイル情報が表示されます。

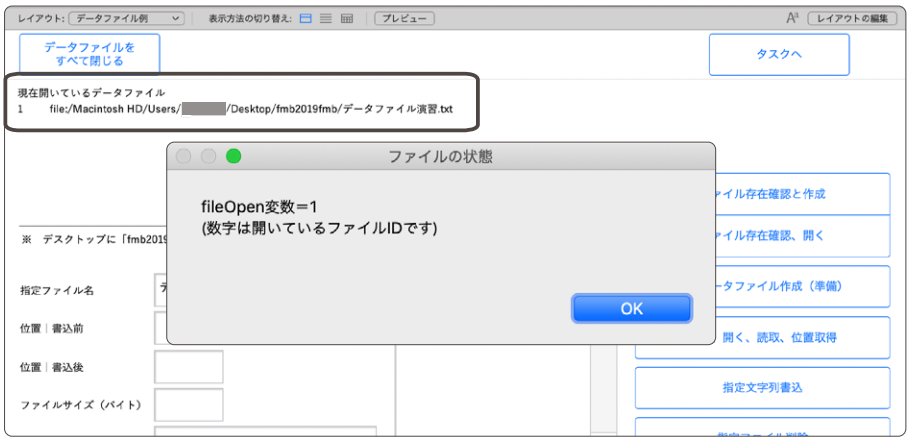


図 11-90「データファイル例」レイアウト

(3)　データファイルが開きました。

(4)　ファイル ID の付き方を確認します。

a.　「複数データファイル作成（準備）」ボタンをクリックします。

b.　6 つのデータファイルが作成されて開かれます。

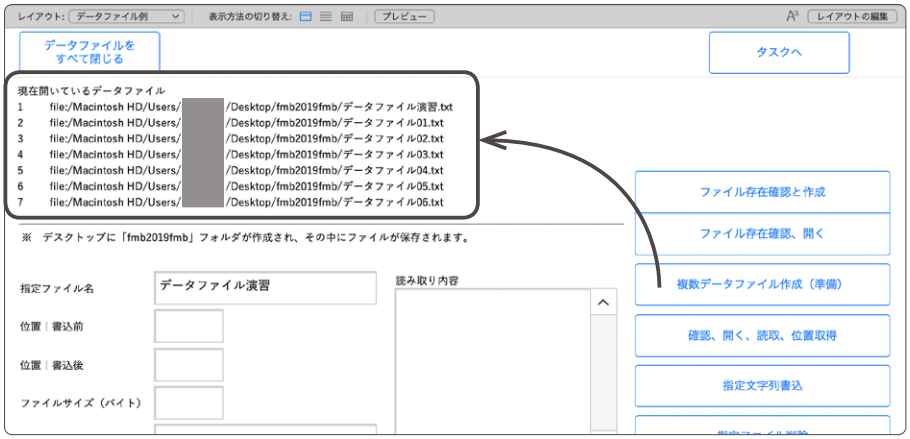


図 11-91「データファイル例」レイアウト

(5)　「データファイルをすべて閉じる」ボタンをクリックします。

(6)　再度、「ファイル存在確認、開く」ボタンをクリックします。
ファイル ID は「8」です。

演習ここまで

ファイル ID

・FileMaker Pro を起動後、データファイルを開く度に「1」からカウントアップします。
・データファイルを開いた FileMaker ファイルを閉じても、データファイルは開いたままです。
・1 つのデータファイルを「開く」度に同じデータファイルでも異なるファイル ID が付与されます。

現在、「データファイル演習 .txt」には何もデータはありません。データを書き込みます。

データファイルの操作（読み取りと位置の確認、位置の指定、書き込み）
開かれているデータファイルから、情報を読み取って、データの位置を確認します。

演習 11-31 データファイルに書き込む

- **使用ファイル**：11_ **データファイル** .fmp12

(1)　［スクリプトワークスペース］で**「存在確認、開く、読取、位置取得」スクリプト**を開きます。

```
レコード / 検索条件確定
#ファイル名がなければ動作しない
If [ IsEmpty ( データファイル例 :: ファイル名 ) ]
    カスタムダイアログを表示 [ "エラー" ; "必要な情報がありません。" ]
    現在のスクリプト終了 [ 結果 : "ファイル名がない" ]
End If

#ファイルをすべて閉じておく（これはスクリプト内容をシンプルにするためです）
データファイルを閉じる [ ファイル ID: "*" ]

#位置を入力するフィールドを初期化
フィールド設定 [ データファイル例 :: 位置 | 書込前 ; "" ]
フィールド設定 [ データファイル例 :: 位置 | 書込後 ; "" ]
レコード / 検索条件確定

#あらかじめファイル名を想定する
変数を設定 [ $fileName; 値 :"file:" & Get( デスクトップパス ) &
"fmb2019fmb/" & データファイル例 :: ファイル名 & ".txt" ]

#ファイルの存在を確認する
ファイルの存在を取得 [「$fileName」; ターゲット : $resultFileName ]
If [ $resultFileName // ある ]
    #ファイルあれば開く
    データファイルを開く [「$fileName」; ターゲット : $fileID ]
End If

#レイアウト上の記号 {{ オープンデータファイル情報 }} を表示するため再表示
ウインドウ内容の再表示

#読み取り　最後の位置を確認するため
データファイルから読み取る [ ファイル ID: $fileID ; ターゲット :
データファイル例 :: 読み取り内容 ; 読み取る : UTF-8 ]

#位置を取得
データファイルの位置を取得 [ ファイル ID: $fileID ; ターゲット :
データファイル例 :: 位置 | 書込前 ]
```

書き込むための位置を確認するには、先に読み取りを行ってデータの最後の位置を知る必要があります。このスクリプトではレイアウト上で確認しやすいようにフィールドへ読み取ったデータを保存していますが、変数に保存しておけば十分です。

(2) 実際に動作させます。
a. 「確認、開く、読取、位置取得」ボタンをクリックします。
b. 「位置｜書込前」に「0」、「読み取り内容」「ファイルサイズ」には何も入力されません。

(3) データファイルを確認します。
a. デスクトップの「fmb2019fmb」フォルダの「データファイル演習 .txt」をテキストエディタ（メモなど）で開きます。
b. 何も入力されていないことを確認したらファイルを閉じてください。

(4) データを書き込みます。まず書き込む文字列を指定します。
a. 「データファイル例」レイアウトで「指定文字列内容」フィールドに「あいうえお」と入力してください。
b. 「指定文字列書込」ボタンをクリックします。
c. 「位置｜書込後」に「16」、「ファイルサイズ」に「16」、「読み取り内容」に「あいうえお」が読み取られます。

データファイルへ「指定文字列内容」フィールドの内容を書き込み後、再度読み取って「位置｜書込後」フィールドへ入力しています。「指定の文字列を書込」スクリプトは次の内容です。

```
レコード / 検索条件確定
#ファイル名と書き込み内容がなければ動作しない
If [ IsEmpty ( データファイル例::ファイル名 ) or IsEmpty ( データファイル例::書き込み内容 ) ]
    カスタムダイアログを表示 [ "エラー";"必要な情報がありません。"]
    現在のスクリプト終了 [ 結果: "ファイル名、書込内容がない" ]
End If

#ファイルをすべて閉じておく（これはスクリプト内容をシンプルにするためです）
データファイルを閉じる [ ファイル ID: "*" ]

#位置を入力するフィールドを初期化
フィールド設定 [ データファイル例::位置｜書込前 ; "" ]
フィールド設定 [ データファイル例::位置｜書込後 ; "" ]
レコード / 検索条件確定

#あらかじめファイル名を想定する
変数を設定 [ $fileName; 値:"file:" & Get(デスクトップパス) & "fmb2019fmb/" & データファイル例::ファイル名 & ".txt" ]

#ファイルの存在を確認する
ファイルの存在を取得 [「$fileName」; ターゲット: $resultFileName ]

If [ $resultFileName // ある ]
```

```
    #ファイルあれば開く
    データファイルを開く[ 「$fileName」 ; ターゲット: $fileID ]

    #レイアウト上の記号{{オープンデータファイル情報}}を表示するために再表示
    ウインドウ内容の再表示

    #読み取り　最後の位置を確認するため
    データファイルから読み取る [ ファイル ID: $fileID ; ターゲット: $fileText ; 読み取る: UTF-8 ]

    #位置を取得
    データファイルの位置を取得 [ ファイル ID: $fileID ; ターゲット: データファイル例::位置｜書込前 ]

    #位置を設定
    データファイルの位置を設定 [ ファイル ID: $fileID ; 新しい位置: データファイル例::位置｜書込前 ]

    #書き込み
    データファイルに書き込む [ ファイル ID: $fileID ; データソース: データファイル例::書き込み内容 ; 書き込み: UTF-8 ; 改行を追加: オン ]

    #位置を取得　書き込み後の最後の位置を確認するため
    データファイルの位置を取得 [ ファイル ID: $fileID ; ターゲット: データファイル例::位置｜書込後 ]

    #読み取り　レイアウト上に表示するため
    データファイルから読み取る [ ファイル ID: $fileID ; ターゲット: データファイル例::読み取り内容 ; 読み取る: UTF-8 ]

    #ファイルサイズを取得
    ファイルサイズを取得 [ $fileName ; ターゲット: データファイル例::ファイルサイズ ]

End If
```

(5)　読み取ったデータの最後の位置を、書き込みの位置として利用します。
手順（4）ではデータファイルになにも入力されていなかったためファイル冒頭の「0」からデータが書き込まれました。再度、手順（4）を別の文字列で行ってください。

「位置｜書込前」が「16」となり、データファイル内の前回の書き込み直後の位置から書き込まれていることがわかります。

演習ここまで

データファイルへ書き込みや読み取りをしました。簡単なスクリプトで演習を行いましたが、連続して書き込む場合はデータファイルは毎回閉じずに「ファイル ID」をなんらかの方法（スクリプト中で取得するか、ファイルを開いた時にグローバル変数で持つなど）で取得して利用する方法でもよいでしょう。

また、書き込みや読み取りは任意の位置指定が可能です。任意の位置から文字列を読み取ります。

演習 11-32 データファイルから読み取る

- **使用ファイル：11_ データファイル** .fmp12

(1) データを読み取るための長文が入力されたデータファイルを作成します。「読み取り用データファイル作成（準備）」ボタンをクリックします。

(2) データファイルを確認します。
   a. デスクトップの「fmb2019fmb」フォルダの「fmb2019testtext.txt」をテキストエディタ（メモなど）で開きます。
   b. 入力されていることを確認したらファイルを閉じてください。

(3) データを読み取ります。
   a. 「指定の位置から読み取り（UTF-8）」ボタンをクリックします。
   b. 「指定読み取り内容」に内容がすべて読み取られます。
      冒頭は、次のような内容です。

```
1
1234567890 FileMaker Pro
データインポート用の新しいユーザインターフェース
新しい [ フィールドデータのインポート順 ] ダイアログボックスで……略
```

(4) 引き続き、データを読み取りをいくつか試します。
   a. 条件を指定します。
      ・指定読込み開始位置　0
      ・指定読み取り数　12
   b. 「指定の位置から読み取り（UTF-8）」ボタンをクリックします。
   c. 「指定読み取り内容」に「1（改行）1234567890」が読み取られます。

   d. 条件を指定します。
      ・指定読込み開始位置　13
      ・指定読み取り数　22
   e. 「指定の位置から読み取り（UTF-8）」ボタンをクリックします。
   f. 「指定読み取り内容」に「FileMaker Pro」が読み取られます。

演習ここまで

読み取りの位置を指定しなれば、すべてが読み取られます。位置がわかっていれば、あらかじめ指定して読み取ることが可能です。どこを読み取れば目的のデータが得られるかわからない場合は、すべてを読み取って、スクリプトなどで処理します。

最後に、新規レコードが作成された時にデータファイルへ記録するスクリプトを確認します。

演習 11-33 新規レコード作成時にデータファイルに記録する

- **使用ファイル：11_ データファイル .fmp12**

(1)　「タスク」レイアウトへ切り替えます（「タスクへ」ボタンをクリックしてください）。



図 11-92「タスク」レイアウト

(2)　新規レコードを作成します。
- a.　「新規レコード作成」ボタンをクリックします。
- b.　「確認」ダイアログが表示されます。
- c.　「OK」をクリックして閉じます。

(3)　データファイルを確認します。
- a.　デスクトップの「fmb2019fmb」フォルダの「新規記録 .txt」をテキストエディタ（メモなど）で開きます。
- b.　簡単な記録が入力されています。
  レイアウト名 | テーブル名 | 主キー | 作成 TS| 作成者　の項目です。
  作成したレコードと見比べます（「タスク」レイアウトの表形式の一番下の行）。
- c.　「新規記録 .txt」を閉じます。

(4)　もう一度「新規レコード作成」ボタンをクリックし「新規記録 .txt」にデータが書き込まれているか確認します。

演習ここまで

このスクリプトでは、一度開いた「新規記録 .txt」データファイルはそのままにしています。$$newDataID グローバル変数に、「新規記録 .txt」の「ファイル ID」を保存して利用しています。

ファイルに関するスクリプトステップを使ったスクリプトで、新規レコード作成、編集、削除などの操作を外部データファイルに記録することができます。

やってみよう

「データファイル例」レイアウトの「指定ファイル削除」ボタンで、「指定ファイル名」に入力されている名前のデータファイルを削除できるのでやってみましょう。

第 12 章 集計とグラフ

集計やグラフが必要な場合があります。蓄積したデータの活用方法の 1 つです。

「**環境設定**」の［**新規フィールドを現レイアウトに加える**］チェックを　**オフ**　にしてください。

（ここから、本書に付属している演習ファイルの「12 章開始時」フォルダの中の「12_ 見積作成管理 .fmp12」ファイルをベースに進めます）

12.1　どのような集計やグラフが必要か

データ蓄積後の利用方法として、集計やグラフについて考えます。

カスタム App の企画や要件定義において「集計」や「グラフ」についての要望や意見を集める場合、これまでも表計算ソフトなどで定期（日、週、月、年など）作成されていれば、ヒアリングメンバーの頭に具体的なイメージや改善点が思い浮かびます。

しかし、既存の書類がなかったり、まったく新しい集計やグラフが必要かどうかを検討する場合、なかなか意見がでないことがあります。あるいは、多種多様な集計やグラフをヒアリングのメンバーの立場ごとに希望され『なんでもできるようにして』という要望がでるかもしれません。

12.1.1　設計が必要

FileMaker Pro 単体では、ユーザが任意のタイミングで任意の集計やグラフを『なんでも自由に作成できる』ようなカスタム App は容易ではありません。機能としての集計やグラフにも、

・要件
・データモデリング
・テーブル、フィールド、関連
・レイアウト
・スクリプト

が必要であるためです。さらに次のことを考える必要もあります。

・タイミング（作業者）
・集計結果の保存／非保存
・グラフの結果表示までの時間（パフォーマンス）

内容によっては、他の機能よりも作業時間がかかる可能性があります。つまり、

・集計やグラフには充分な準備が必要

ということです。データと結果の関係によって作業時間が変わります。

※ 他のアプリケーションと統合すると、カスタム App に蓄積されたデータを任意に加工して集計やグラフを作成できるようになるかもしれません。例えば、DataAPI の機能を利用することです。本書では解説しません。詳しい情報はクラリス・ジャパン株式会社の Web サイトからどうぞ。

追加の機能とする場合がある

集計やグラフの内容や種類は、企画や要件定義の段階で決定できることがベストです。

しかし、何が必要なのかわからないのであれば、無理に決める必要はありません。『あとで決めよう』とか『できあがってからデータを見て決めよう』という意見に落ち着くこともあります。

重要度、優先度、難易度

機能にはいくつかの側面があります。

・企画時の目的の実現に対してどれだけ重要な機能か（重要度）
・業務上、またユーザにとってどれだけ必要な機能か（優先度）

これら以外に

・機能を作成する難易度

があります。開発者にとって難易度が低い機能は『すぐにできる』ものです。それが重要度も優先度も高い機能であればユーザにとっても良いことです。逆に、それほど重要度も優先度も高くない機能が難易度が高い場合、ユーザ的には『思ったよりも時間がかかる』ものになります。作成の難易度は、ユーザがその機能を使う状況と関係しません。

12.1.2 集計やグラフの要件定義と設計

基本的に、他の機能と同じです。

(1) いつ
(2) どこで
(3) 誰が
(4) なにを
(5) どのように
(6) どうやって
(7) 何故
(8) 備考

意識したいのは、次の内容です。

・(1) の「いつ」　→　タイミングと頻度、定期的　なのか　任意　なのか
・(4) の「なにを」　→　出力形式（画面のみ、印刷、PDF、エクスポート）と内容（項目と順も）
・(7) の「何故」　→　会議のため　などだけではなく、結果を何のために使用するのか
（営業担当ごとの注文数や注文金額の合計で成績を見るなどの具体的な目的）

また（8）「備考」として、

・出力された結果の受け渡し方法（レイアウトにアクセスしてもらう、印刷して人が配布、メール文中に記述、メールに PDF などで添付）

を確認します。他にもいくつか考えることがあります。

必要なデータが揃っているか
集計やグラフに必要な結果から､元になるデータを考えます。これはとても重要です。どのテーブル、どのような対象レコード、から結果を出すのかを設計すると、

・テーブルに、必要な項目（フィールド）がない

ということに気付くことがあります。フィールドを増やし、入力用レイアウトを変更する必要があるかもしれません。既に運用しているカスタム App の集計やグラフ機能であれば、蓄積されたデータへ項目が追加されることになります。さらに、初期値を入力するかどうか、これまでのデータは集計やグラフの対象外にするかどうか、などを考えます。

フィールドタイプ
集計やグラフの元データとして、複数のデータからの計算結果を利用する場合、

・統計関数の計算式が指定された計算タイプフィールド
・集計タイプフィールド

のどちらを使用するかを考えます。
※ 統計関数は、主に関連レコードの情報について使用されることが多いものです。

例えば､「見積情報」と「見積明細」テーブルの関連から考えると「見積情報」から「見積明細」テーブルの関連レコードの情報を集計するのであれば統計関数を使うことが多いかもしれません。

実行するアプリケーション
(3) 誰が　としては、処理を実行するのが、

・FileMaker Pro、 FileMaker Go などのクライアント
・FileMaker Cloud ／ FileMaker Server などのサーバー（ホスト）

どちらにするかを考えます。FileMaker Cloud ／ FileMaker Server を利用しているのであれば、ユーザが任意に実行するか、スケジュールで定期的に実行するかも決定します。後述のパフォーマンスとの兼ね合いも含めて検討します。

同じ結果のバリエーションに注意
同じデータを元に複数の集計やグラフを表示する、例えば、

・商品番号ごとの 12 ヶ月の内訳の売上件数

を結果として表示している場合に、

・月ごとに商品番号の内訳の件数見たい

という要望がでることがあります。テーブル設計によって単純にソート順を変更するボタンを作成すれば良いかもしれませんし、別の集計機能を 1 つ追加することになるかもしれません。一見、ソート違いのように思えても、必ずテストをして確認します。

集計結果の見た目
集計結果では「表」的な見た目が希望されることが多くなります。希望通りの表示あるいは出力が可能かについて必ず確認します。見た目のために設計を工夫する必要があるかもしれません。

グラフのタイプ
グラフの場合、目的に応じてわかりやすいタイプを選択することが必要です。やはり、希望通りの表示が可能かについて必ず確認します。

セキュリティ
データに応じて検討します。例えば、

・個人の情報や成績などに関係する場合、「誰が結果を見られるか」
・重要な業務の情報が含まれる場合、「誰が印刷やエクスポートできるか」

集計対象を誰が決めるか
集計やグラフの対象レコードは、検索や関連を使用して作成します。この決定方法です。主に、次のどちらかです。

・あらかじめ決まった範囲を決定する
・ユーザが任意に決定する

後者であれば、集計やグラフの処理実行前にユーザが範囲を指定できるレイアウトが必要です。

パフォーマンス
これはすぐにわからないかもしれません。データの蓄積が少ないうちは、集計の結果がすぐに表示されていても、蓄積データ数が増えると結果の表示に時間がかかるようになってしまうことがあります。

可能であれば、テスト時に 1 年分や 2 年分のデータ数で結果が出るまでの時間を計測できるのがベストです。経験値で時間がかかるかどうかを予想できるようになることもあります。もしも、『時間がすごくかかってユーザが待てない』という状況になったら、

・FileMaker Cloud ／ FileMaker Server が夜間にスケジュールを実行
・一度実行した集計結果を保存しておく場所を作成する（何度も負荷の高い処理を実行しない）
・集計タイプフィールドではなく、数字タイプフィールドに結果を格納するように工夫する

などを考えます。夜間に実行すると業務時間に負荷が掛かるのを防げますし、毎回最新データを処理しなくても過去の結果を確認するだけで済むユーザもいます（結果を格納しておくと、集計タイプフィールドがレイアウト上で表示される度に働くよりも結果の表示が早くなります）。

などなど、たくさんの事柄を考える必要があります。

12.2 集計の作成

本章では、複数の集計タイプフィールドと集計パートを使った集計の確認と作業を行います。

（レイアウトウイザード使った集計用レイアウト簡易な作成方法は FileMaker Master Book：初級編を参照してください。本書では、既に作成されているレイアウトに集計用のパートなどを追加して集計結果を表示するような設計をして作成する方法の演習を行います。）

12.2.1 準備

範囲や対象などが決定したら、機能を作成します。集計に必要なものは、主に次の 2 つです。

・「集計タイプ」のフィールド
・「集計」パート

集計タイプフィールド

集計タイプフィールドが集計するのは、「対象レコード」です。

レイアウトパートやパートに設定されるソート対象（区分けフィールド）と組み合わせて、対象レコードの総計と区分けグループごとの小計が集計されます。計算方法の設定の種類は下表を参照してください。

集計タイプのフィールドの計算方法	
方法	**内容**
合計	指定したフィールド内の値の合計を計算します。
平均値	指定したフィールド内の値の平均値を計算します。
カウント	指定したフィールドに値が入力されているレコードの数をカウントします。
最大値	指定したフィールドの最大値、または最も新しい日付、時刻、タイムスタンプを検索します。
最小値	指定したフィールドの最小値、または最も古い日付、時刻、タイムスタンプを検索します。
標準偏差	指定したフィールド内の値が対象レコードでどのくらい分散しているかを計算します。指定したフィールド内の値の平均値から標準偏差が計算されます。
合計に対する比	指定したフィールド内のある値が全体の値の合計に占める割合を計算します。
一覧	空白以外の値の改行で区切られた一覧をフィールド内に作成します。

そして、集計タイプフィールドと集計パートと一緒に準備するものは、次の 3 つです。

・関連（リレーションシップ）
・ソート
・レイアウト

「見積作成管理 .fmp12」に次の 2 つの集計機能を作成しました（テストデータも入力しました）。

・見積先の顧客ごとの件数（受注数と総数）と合計金額（受注金額と総合計）
・商品ごとの見積件数（見積数と割合）と合計金額

12.2.2 集計の仕組み 1

「見積先の顧客ごとの件数（受注数と総数）と合計金額（受注金額と総合計）」について動作を確認します。「見積リスト」レイアウトに切り替えて、画面右上のボタンをクリックしてください。

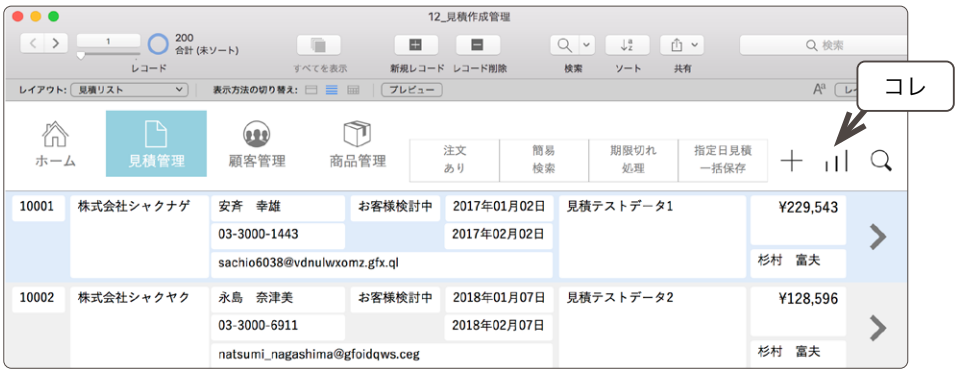


図 12-1 集計を確認するためのソートボタン

「見積先の顧客ごとの件数（受注数と総数）と合計金額（受注金額と総合計）」が集計パート（グレーの部分）に表示されます（表示されていない場合は、画面を上下にスクロールして確認してください）。



図 12-2 集計結果

どのような準備をしたかを確認します

フィールドの確認

集計結果のために 4 つのフィールドを「見積情報」テーブルに用意しました。[データベースの管理] ダイアログを開いて「フィールド」タブで確認します。

- 見積先ごとの見積作成数　「s_ 見積件数」フィールド
- 見積先ごとの見積金額合計　「s_ 見積額」フィールド
- 見積先ごとの「注文あり」の見積作成数　「c_ 見積受注件数」フィールド
- 見積先ごとの「注文あり」の見積金額合計　「c_ 見積受注金額」フィールド



図 12-3 集計に使うフィールド

「s_ 見積額」、「s_ 見積件数」フィールドは「集計」タイプです。

「集計」タイプフィールドの集計対象として指定可能なのは、同テーブルのフィールドのみです。それぞれ、次の内容で設定されています。

・「s_ 見積額」フィールド　→　金額　→　合計　→「c_ 見積総額」フィールド
・「s_ 見積件数」フィールド　→　件数　→　カウント　→「主キー」フィールド



図 12-4「s_ 見積額」フィールドの設定

「s_ 見積件数」フィールドのカウント対象に「主キー」フィールドを使用しているのは、空欄がありえないため確実にカウントできるという理由からです。

「c_ 見積受注件数」、「c_ 見積受注金額」フィールドは「計算」タイプです。「見積管理｜集計｜自己」TO（次図を参照）の関連レコードデータを統計関数を使用して計算しています。

・「c_ 見積受注件数」フィールド
　→　件数　→　カウント（Count 関数）→
　→　Count (見積管理｜集計｜自己 :: 主キー)
・「c_ 見積受注金額」フィールド
　→　金額　→　合計（Sum 関数）
　→　Sum (見積管理｜集計｜自己 ::c_ 見積総額)

関連の確認

受注数や受注金額を計算するために、照合フィールドとして「c_ 注文有無」フィールドも作成しました（計算タイプフィールド（索引あり））。

・「見積結果」フィールドが「注文あり」だったら「1」そうでなければ「0」

[データベースの管理] ダイアログの「リレーションシップ」タブで確認します。「見積管理」TO と「見積管理｜集計｜自己」TO の関連を定義しました。これは「見積情報」テーブル同士の自己リレーションシップです（次図の TOG を参照してください）。

・「見積管理」TO「顧客外部キー」　＝　「見積管理｜集計｜自己」TO「顧客外部キー」
・「見積管理」TO「_1」　＝　「見積管理｜集計｜自己」TO「c_ 注文有無」

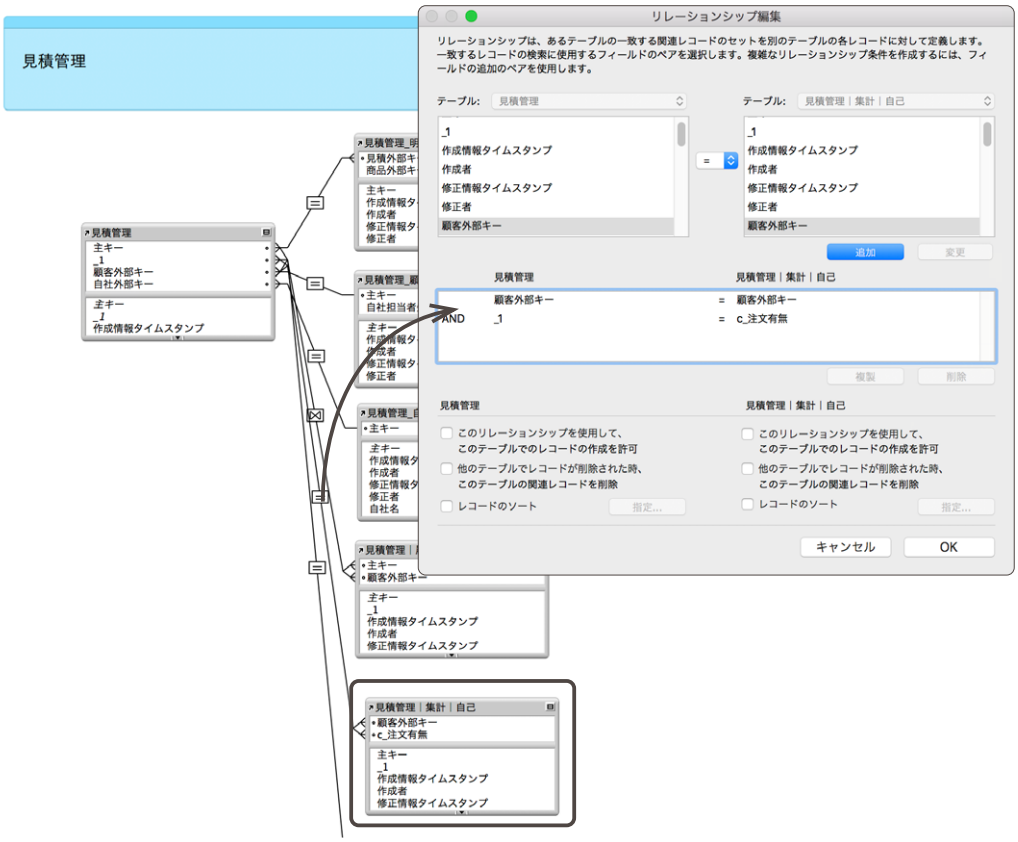


図 12-5 注文ありの件数を計算するための TO

構造が用意されていることを確認しました。[キャンセル] をクリックして [データベースの管理] ダイアログを閉じてください。

集計のためのフィールドは用意されていました。これらは、「ボディ」パートなどに配置すると、対象レコードの集計結果や各レコードの関連レコードの計算結果を表示します。

「見積先の顧客ごと」は、どこで指定するのでしょうか。これはレイアウトのパートで行います。

レイアウトの確認

あたらめて「見積リスト」レイアウトを確認します。ソートしていない状態では「小計」パートは表示されていませんでした。レイアウトモードに切り替えます。

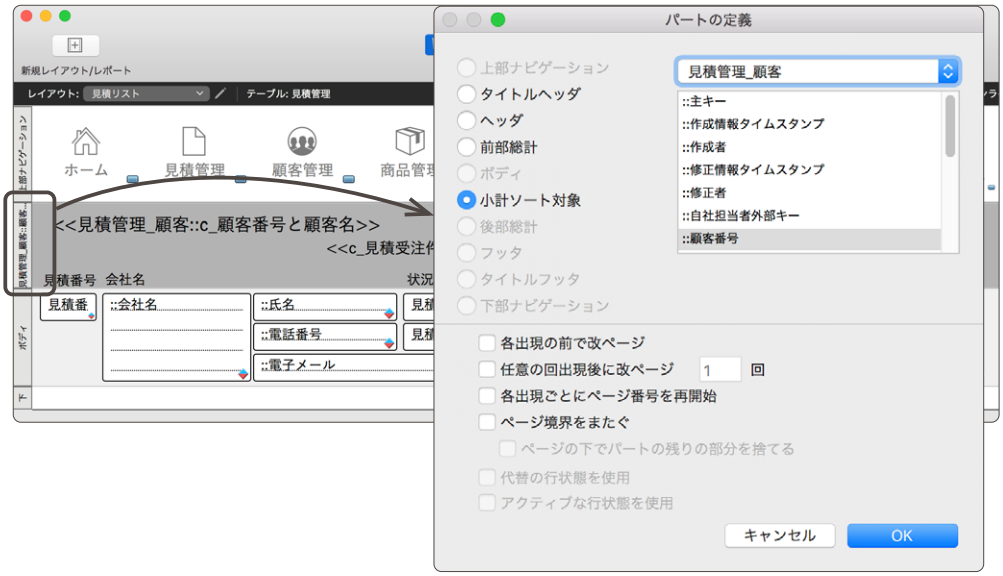


図 12-6「見積リスト」レイアウト

背景がグレーの「小計」パートが作成されています。マージフィールドが置かれています。パートラベルをダブルクリックすると［パートの定義］ダイアログが表示されます（上図を参照）。指定したフィールドが含まれるソートが実行されると結果を表示します。現在、次の設定です。

・「見積管理 _ 顧客」TO の「顧客番号」フィールドでソートした集計結果を出す

ブラウズモードに戻ります。画面右上のボタンには「集計用ソート見積リスト _ 演習用」スクリプトが割り当てられています。これは、ソートを実行するためのスクリプトです。

```
レコードのソート [ 記憶する; ダイアログなし ]
レコード / 検索条件 / ページへ移動 [ 最初の ]
```

指定したフィールド以外がソート順に含まれていても構いません。ソート順は次の内容です。

(1)「見積管理 _ 顧客」TO の「顧客番号」フィールド	昇順	→「見積先の顧客」
(2)「見積番号」フィールド	降順	
(3)「発行日」フィールド	降順	

集計のためのフィールドを用意し、レイアウトでパートを作成しソートを指定することで「小計」パートが表示されて希望した内容の結果が表示されることがわかります。

さらに全体の結果を表示するために**「見積リスト」レイアウトに「前部総計」パートを作成**します。

「12 章開始時」フォルダ

演習 12-1　集計に総計を加える

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「見積リスト」レイアウトで［レコード］–［レコードのソート ...］メニューを選択し、ソートを解除します。

(2)　パートを作成して設定します。

a.　レイアウトモードに切り替え、［レイアウト］–［パート設定 ...］メニューを選択します。

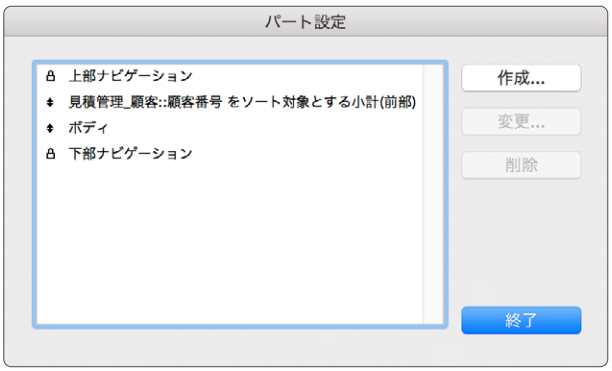


図 12-7［パート設定］ダイアログ

b.　［作成 ...］をクリックします。

c.　［パートの定義］ダイアログが表示されます。

d.　「前部総計」をクリックします。

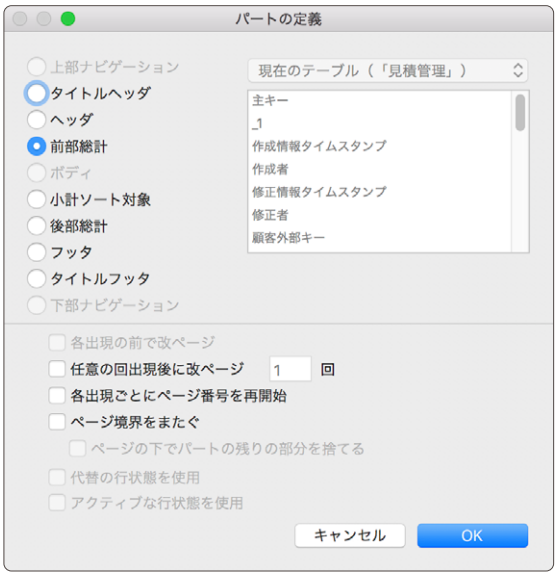


図 12-8［パートの定義］ダイアログ

e. ［OK］をクリックして［パートの定義］ダイアログを閉じます。
f. ［終了］をクリックして［パート設定］ダイアログを閉じます。

(3) パートが加わっています。

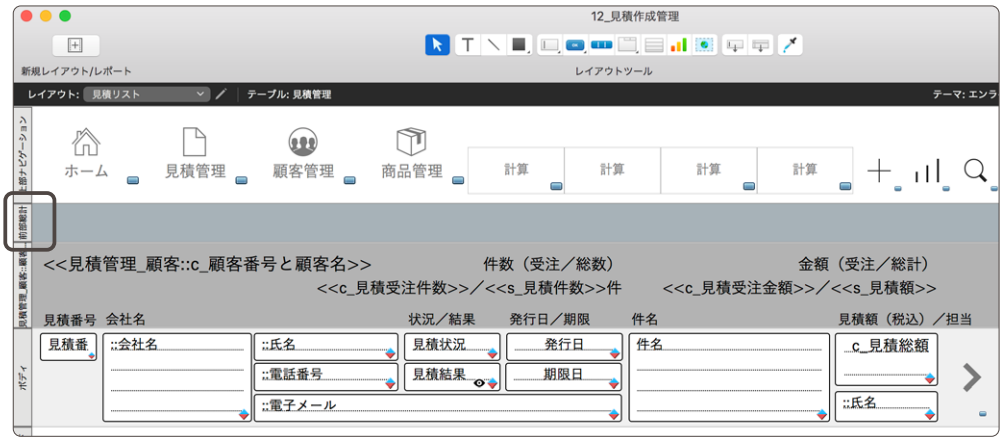


図 12-9 パートが加わった

a. 「前部総計」パートのパートラベルをクリックします。
b. 右パネルのインスペクタの「スタイル」タブで「デフォルト」が割り当てられていることを確認します。
c. インスペクタの「位置」タブで、「前部総計」パートの高さを「60pt」に設定します。

(4) 「前部総計」パートにマージフィールドを配置します。
a. ［挿入］–［マージフィールド ...］メニューを選択します。
b. 「s_ 見積件数」フィールドを選択します。
c. ［OK］ボタンをクリックします。
d. マージフィールドが置かれます。
e. <<s_ 見積件数 >> をクリックします。
f. インスペクタの「スタイル」タブで「タイトルテキスト 2」を割り当てます。
g. インスペクタの「外観」タブの［段落設定］で、「右揃え」にします。
h. 次のように「件」を追加します。

<<s_見積件数>>件

i. 同じ手順で「s_ 見積額」フィールドもマージフィールドで配置します。

<<s_見積額>>

j. 「データ」タブの［データの書式設定］で「通貨」を選択します。「3 桁区切りを使用」を **オン** にします。

(5) 次図のように「前部総計」パートに配置します。



図 12-10 総計パートに集計タイプフィールドを配置

(6) レイアウトを保存してブラウズモードに戻ります。

(7) 前部総計が機能しているはずです。

(8) 右上のボタンをクリックしてソートを実行します。前部総計は変わりません。



図 12-11 ソートの有無では前部総計の結果は変わらない

(9)　クイック検索エリアに「結果確定済」と入力して検索を実行します。

(10)　前部総計の数字が対象レコードに合わせて変わります。

(11)　画面右上のボタンをクリックしてソートを実行し、小計パートの内容を確認してください。



図 12-12 対象レコードで前部総計の結果が変わる

演習ここまで

「集計」タイプフィールドは、主に結果のタイプは数値です。レイアウト上では余裕のあるフィールド幅を設定します。また、小数点以下の端数が発生するような数値では、インスペクタで書式設定をおこないます。

なお、「一覧」を選択すると改行区切りのデータ（テキスト）になります。この場合、フィールドの高さの設定にも気をつけます。



図 12-13「s_ 見積額」フィールドの設定

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 12_2 から 3」フォルダの中の「12_ 見積作成管理 .fmp12」ファイルをベースに進めます）

12.2.3 集計の仕組み 2

続いて「商品ごとの見積件数（見積数と割合）と合計金額」について、どのような準備をしたかを確認します。この集計は作業途中です。

フィールドの確認

3 つの「集計」タイプフィールドを「見積明細」テーブルに用意しました。[データベースの管理] ダイアログを開いて「フィールド」タブで確認します。

- 商品ごとの見積数の金額合計　「s_ 商品合計」フィールド
- 商品ごとの見積数　「s_ 商品件数」フィールド
- 商品ごとの見積数の割合　「s_ 商品内訳」フィールド

さらに、商品数のカウント用に「計算」タイプのフィールドを用意しました。

これは「見積明細」レコードではあるものの、値引きなど「商品外部キー」が入力されないことがあることを想定したものです。

- 「c_ 商品カウント用」フィールド（「商品外部キー」が入力されていたら「1」を返す）

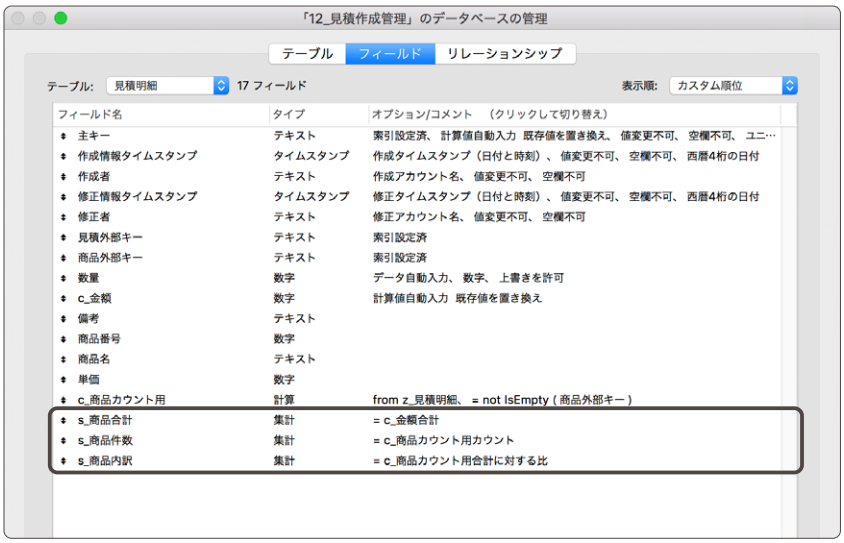


図 12-14 集計に使うフィールド

「集計」タイプフィールドは、それぞれ、次の内容で設定されています。

・「s_ 商品合計」フィールド
　→　金額　→　合計　　→「c_ 金額合計」フィールド
・「s_ 商品件数」フィールド
　→　件数　→　カウント　　→「c_ 商品カウント用」フィールド
・「s_ 商品内訳」フィールド
　→　割合　→　合計に対する比　　→「c_ 商品カウント用」フィールド

関連の確認
「見積管理 _ 明細」TO だけを使用します。新たな関連は必要ありません。

［キャンセル］をクリックして［データベースの管理］ダイアログを閉じてください。

では、「商品ごとの」を指定するレイアウトのパートはどうでしょうか。

レイアウトの確認
「ホーム」レイアウトで「演習グラフ表示見積明細集計一覧」ボタンをクリックすると、「商品ごと明細」レイアウトが表示されます。これが集計用に用意したレイアウトです。

レイアウトは

作業途中です。レイアウトモードに切り替えます。「前部総計」パートはあります。後は、**「小計」パートが必要**です。

このレイアウトの右上にもソート用のボタンを用意しました。「集計用ソート商品明細 _ 演習用」スクリプトが割り当てられています。しかし、**ソート用のスクリプトの中身は空**です。

ここで、表示について考えます。「見積明細」テーブルのレコードはどんどん作成されます。「見積先の顧客ごとの件数集計」のように、

・「小計」パート（集計の結果） ＋ 「ボディ」パート（実際のデータ）

の、すべてを表示すると目的の結果を確認することが難しくなります。そのため、

・「前部総計」と「小計」パートだけ

のような形で表示することがあります。この方法の演習をします。

「演習 12_2 から 3」フォルダ

演習 12-2　パートを変更して集計を表示する

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「商品ごと明細集計」レイアウトで、レイアウトモードに切り替えます。
「ボディ」パートに商品名と集計タイプフィールドが配置されていることを確認してください。

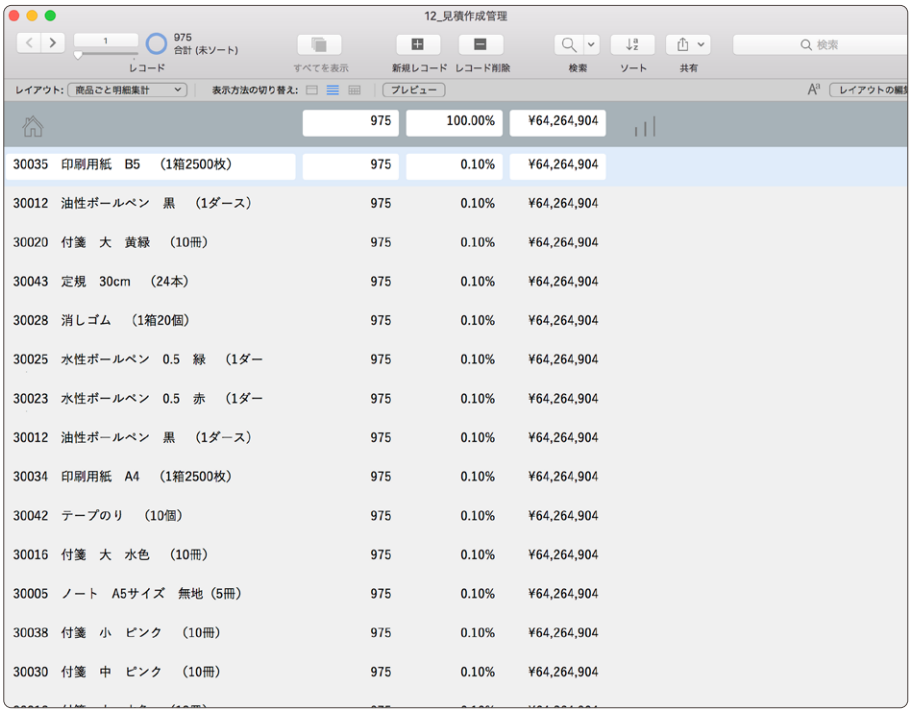


図 12-15 作業前のレイアウト（レイアウトモード）

(2)　パートを変更して集計結果を表示します。
a.　「ボディ」パートのパートラベルをダブルクリックして［パートの定義］ダイアログを表示します。
b.　「小計ソート対象」をクリックします。
c.　「商品外部キー」フィールドをクリックします。
d.　［OK］をクリックして［パートの定義］ダイアログを閉じます。

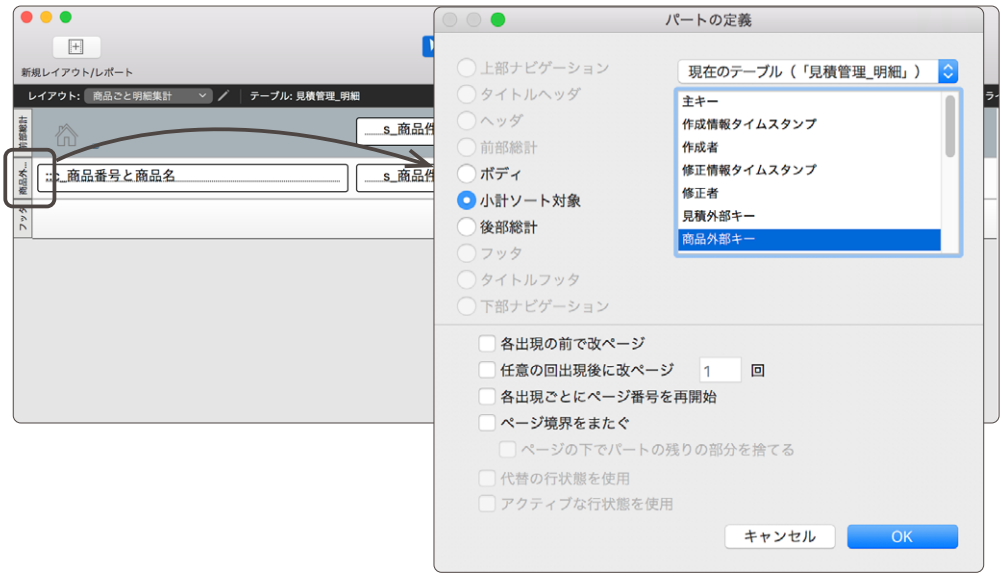


図 12-16［パートの定義］ダイアログ

e.　メッセージが表示されます。「上に印刷」をクリックします。



図 12-17「小計」パートに変更したときのメッセージ

(3)　パートが変更されました。レイアウトを保存してブラウズモードに戻ります。「前部総計」パート以外は表示されていません。

(4)　ソートを実行します（次図も参照）。
・「見積管理 _ 明細 _ 商品」TO の「商品番号」フィールド　　昇順
・「見積管理 _ 明細」TO の「商品外部キー」フィールド　　昇順

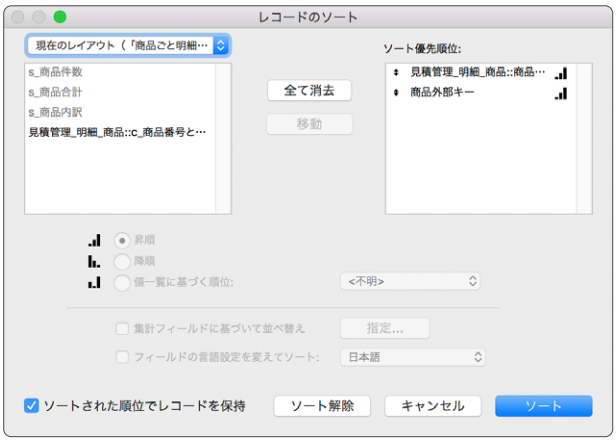
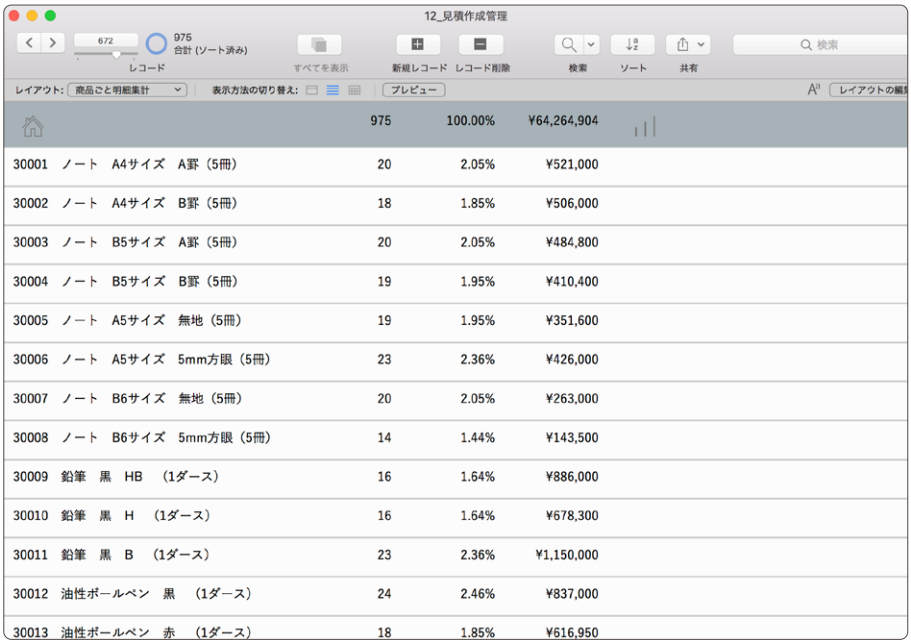


図 12-18［レコードのソート］ダイアログ

(5)　集計されます。



	975	100.00%	¥64,264,904
30001　ノート　A4サイズ　A罫（5冊）	20	2.05%	¥521,000
30002　ノート　A4サイズ　B罫（5冊）	18	1.85%	¥506,000
30003　ノート　B5サイズ　A罫（5冊）	20	2.05%	¥484,800
30004　ノート　B5サイズ　B罫（5冊）	19	1.95%	¥410,400
30005　ノート　A5サイズ　無地（5冊）	19	1.95%	¥351,600
30006　ノート　A5サイズ　5mm方眼（5冊）	23	2.36%	¥426,000
30007　ノート　B6サイズ　無地（5冊）	20	2.05%	¥263,000
30008　ノート　B6サイズ　5mm方眼（5冊）	14	1.44%	¥143,500
30009　鉛筆　黒　HB　（1ダース）	16	1.64%	¥886,000
30010　鉛筆　黒　H　（1ダース）	16	1.64%	¥678,300
30011　鉛筆　黒　B　（1ダース）	23	2.36%	¥1,150,000
30012　油性ボールペン　黒　（1ダース）	24	2.46%	¥837,000
30013　油性ボールペン　赤　（1ダース）	18	1.85%	¥616,950

図 12-19 作業後に集計結果が表示されたレイアウト

(6)　スクリプトを編集します。「集計用ソート商品明細 _ 演習用」スクリプトを次の内容としてください。詳細な手順は省略します。

```
レコードのソート　[ 記憶する　;　ダイアログあり　;　オフ ]
レコード／検索条件／ページへ移動　[　最初の　]
```

［レコードのソート］スクリプトステップは、先ほどと同じソート順を指定します。

・「見積管理 _ 明細 _ 商品」TO の「商品番号」フィールド　　昇順
・「見積管理 _ 明細」TO の「商品外部キー」フィールド　　昇順

(7)　スクリプトを保存します。

(8)　動作を確認します。

a.　「商品ごと明細集計」レイアウトに切り替えます。
b.　ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。
c.　ソートを解除します。
d.　画面右のソートのボタンをクリックします。
e.　「前部総計」、「小計」パートと集計結果が表示されているはずです。

演習ここまで

さらに**「商品ごと明細集計」レイアウトのソート順を見積件数が多い順に変更**します。

演習 12-3　集計結果でソート順を設定する

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　スクリプトを編集します。「集計用ソート商品明細 _ 演習用」スクリプト変更します。

a. ［レコードのソート］スクリプトステップで、歯車アイコンをクリックします。
b. ［指定 ...］をクリックします。
c. ［レコードのソート］ダイアログが表示されます。
d. ［ソート優先順位：］の、「見積管理 _ 明細 _ 商品」TO の「商品番号」フィールドをクリックします。
e. ダイアログの下の方にある「集計フィールドに基づいて並べ替え」をチェックします。
f. ［フィールド指定］ダイアログが表示されます。
g. 「s_ 商品件数」フィールドを指定します。

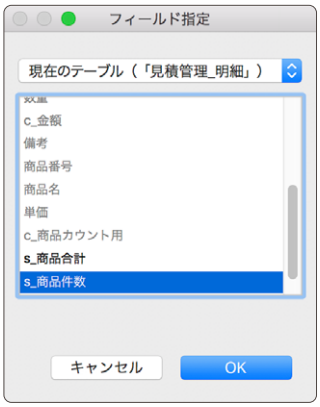


図 12-20［フィールド指定］ダイアログ

h. ［OK］をクリックします。
i. ［レコードのソート］ダイアログで、「降順」をクリックします（次図も参照）。



図 12-21［レコードのソート］ダイアログ

(2)　スクリプトを保存します。

(3)　動作を確認します。

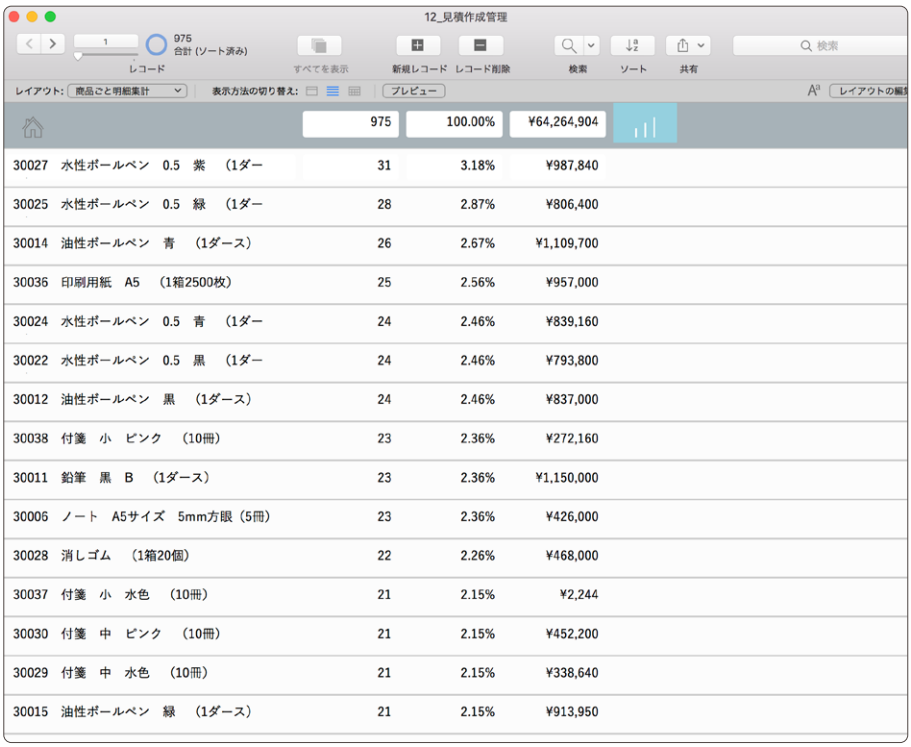
a.　「商品ごと明細集計」レイアウトに切り替えます。

b.　ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。

c.　ソートを解除します。

d.　画面右のソートのボタンをクリックします。

e.　見積数が多い順に集計が表示されます。



	975	100.00%	¥64,264,904
30027　水性ボールペン　0.5　紫　（1ダー	31	3.18%	¥987,840
30025　水性ボールペン　0.5　緑　（1ダー	28	2.87%	¥806,400
30014　油性ボールペン　青　（1ダース）	26	2.67%	¥1,109,700
30036　印刷用紙　A5　（1箱2500枚）	25	2.56%	¥957,000
30024　水性ボールペン　0.5　青　（1ダー	24	2.46%	¥839,160
30022　水性ボールペン　0.5　黒　（1ダー	24	2.46%	¥793,800
30012　油性ボールペン　黒　（1ダース）	24	2.46%	¥837,000
30038　付箋　小　ピンク　（10冊）	23	2.36%	¥272,160
30011　鉛筆　黒　B　（1ダース）	23	2.36%	¥1,150,000
30006　ノート　A5サイズ　5mm方眼（5冊）	23	2.36%	¥426,000
30028　消しゴム　（1箱20個）	22	2.26%	¥468,000
30037　付箋　小　水色　（10冊）	21	2.15%	¥2,244
30030　付箋　中　ピンク　（10冊）	21	2.15%	¥452,200
30029　付箋　中　水色　（10冊）	21	2.15%	¥338,640
30015　油性ボールペン　緑　（1ダース）	21	2.15%	¥913,950

図 12-22 件数の多い順に集計結果が表示されたレイアウト

演習ここまで

この演習のように「集計」タイプフィールドの結果はソートにも利用できます。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 12_4 から 5」フォルダの中の「12_ 見積作成管理 .fmp12」ファイルをベースに進めます）

さらに、「見積リスト」レイアウトの「見積先の顧客ごとの件数と合計金額」集計について、次の結果も表示できるようにします。

・発行月ごとの結果

集計に条件が加わりました。「集計」タイプフィールドの追加は必要ありませんが、ソートのためのフィールドが必要です。「見積情報」テーブルには「発行日」フィールドがあります。これを利用してフィールドを 1 つ作成しました。

・「c_ 発行月」フィールド（「計算」タイプ）

```
Date ( Month ( 発行日 ) ; 1 ; Year ( 発行日 ) )
```

→ 「発行月の 1 日」の日付です。

常に発行日当月の 1 日が結果として保存されるため、これを月ごとのソート条件として利用できます。集計では、このようなフィールドを作成することがあります。

発行月ごとの結果のために「小計」パートをレイアウトに追加します。

「演習 12_4 から 5」フォルダ

演習 12-4　小計パートを 2 つにする

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「見積リスト」レイアウトで、レイアウトモードに切り替えます。

(2)　［レイアウト］–［パート設定 ...］メニューを選択します。

(3)　「小計」パートを追加して設定ます。

a.　［作成 ...］をクリックします。
b.　［パートの定義］ダイアログが表示されます。
c.　「小計ソート対象」をクリックします。
d.　「c_ 発行月」フィールドをクリックします。
e.　［OK］をクリックして［パートの定義］ダイアログを閉じます。
f.　メッセージが表示されます。「上に印刷」をクリックします。
g.　［終了］をクリックして［パート設定］ダイアログをとじます。

(4)　パートが加わっているのを確認して［終了］をクリックします。

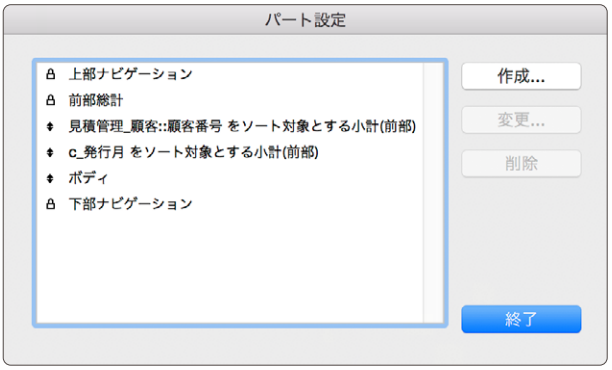


図 12-23 パートが加わった

(5)　レイアウトを整えます。

「発行月をソート対象とする小計（前部）」パートに次の作業をします。

a.　右パネルのインスペクタの「スタイル」タブで「デフォルト」を割り当てます。
（パート作成時に既に割り当てられているはずです）

b.　インスペクタの「位置」タブで、パートの高さを「70pt」に設定します。

c.　上の「見積管理 _ 顧客 :: 顧客番号をソート対象とする小計（前部）」パートから、フィールドラベルを移動します（グループ化してありますので簡単に作業できます）。



図 12-24 高さを変更してフィールドラベルを移動

「見積管理 _ 顧客 :: 顧客番号をソート対象とする小計（前部）」パートに次の作業をします。

d.　パートの高さをインスペクタの「位置」タブで「70pt」に設定します。

さらに作業します。

e.　「前部総計」パートから、マージフィールドをコピーして、「発行月をソート対象とする小計（前部）」「小計」パートにコピーと貼り付け（ペースト）をします。



図 12-25 高さを変更して、マージフィールドを配置

f. 「発行月をソート対象とする小計（前部）」パートにマージフィールドを配置します。
g. ［挿入］–［マージフィールド ...］メニューを選択します。
h. 「c_ 発行月」フィールドを選択します。
i. インスペクタの「スタイル」タブで「タイトルテキスト 2」を割り当てます。
j. インスペクタの「外観」タブの［段落設定］で、「右揃え」にします。

<<c_ 発行月 >> 表示「YYYY 年 MM 月」に変更します。

k. <<c_ 発行月 >> のテキストオブジェクトを選択します。
l. 「データ」タブの［データの書式設定］でカレンダーマークをクリックします。
m. ［書式］でポップアップメニューから「その他」を選択します。
n. 右図を参照してください。日付と曜日の表示をなくします。



図 12-26 高さを変更して、マージフィールドを配置してインスペクタで日付の表示を変更する

(6) レイアウトを保存してブラウズモードに戻ります。

演習ここまで

目的の結果を表示するために、ボタンに割り当てられているスクリプトのソート順を変更します。

演習 12-5　ソート順を設定する

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　スクリプトを編集します。「集計用ソート見積リスト _ 演習用」スクリプト変更します。
次のスクリプトステップのオプションを変更します。手順は省略します。

レコードのソート　[　記憶する　;　ダイアログ　:　**オフ**　]

・「見積管理 _ 顧客 :: 顧客番号」フィールド　昇順
・「c_ 発行月」フィールド　降順
・「発行日」フィールド　降順
・「見積番号」フィールド　降順

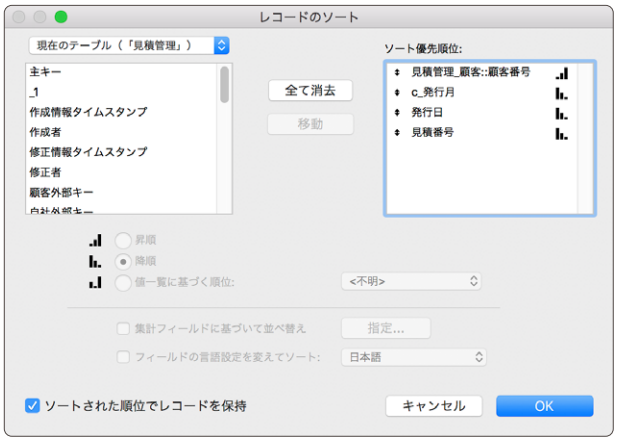


図 12-27　設定後のソート順

(2)　スクリプトを保存します。

(3)　動作を確認します。
a.　「見積リスト」レイアウトに切り替えます。
b.　ステータスツールバーで「すべてを表示」をクリックして全レコードを表示します。
c.　ソートを解除します。
d.　画面右のソートのボタンをクリックします。
e.　見積先の顧客ごとの発行月の新しい順に結果が表示されるはずです。

演習ここまで

集計には、関連、テーブル、フィールド、レイアウト、ソート、ボタンなど、一通りの準備と作業がありました。さらに複雑な集計や、印刷、PDF 保存、エクスポートなどの出力までを考えると、設計やテストが大事であることがわかります。

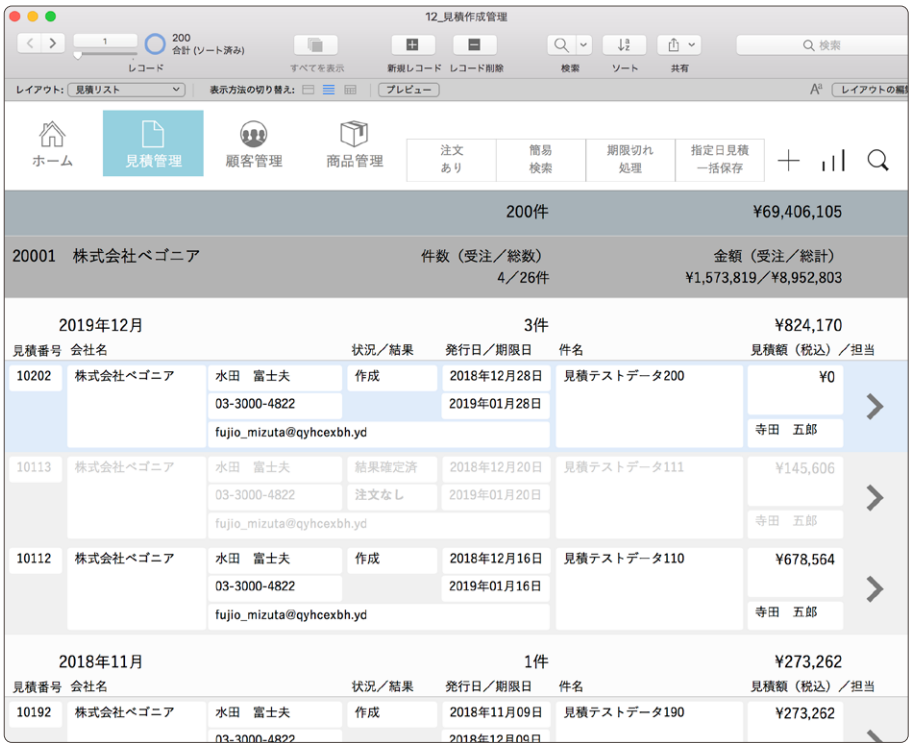


図 12-28 見積先の会社ごとで、発行月ごとの結果

年、月、週の小計

集計する対象の日付タイプフィールドに対して前の演習のように計算式で別の集計用フィールドを作成します。年のみ、月のみ、週のみでは以下の計算式も使います。

```
WeekOfYear ( 日付 ) 関数
Month ( 日付 ) 関数
Year ( 日付 ) 関数
```

集計が上手くいかないとき

集計が想定した結果にならない場合、以下を確認します。

- ソート項目（集計タイプフィールドのオプションの指定、小計パートの指定、レイアウトでおこなうソートの指定に同じフィールドが入っているか、など）を確認する
- 集計タイプフィールドのオプションの指定、小計パートの指定の区分けフィールドが空欄だと集計結果に空欄表示されるので、空欄が不要であれば対象レコードから除く
- 集計タイプフィールドのオプションの指定、小計パートの指定の区分けフィールドへ入力されているデータの表記揺れがある場合は統一する
- レイアウト、フィールドの TO、コンテキストを確認する
- 正しいパートに集計タイプフィールドが配置されているか確認する

集計結果表示のためのレイアウト

集計結果を表示するレイアウトでは、ユーザがデータを入力できないようにすることをお勧めします。ソートによってユーザが入力途中でデータを見失うことがあるかもしれないからです。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 12_6 から 8」フォルダの中の「12_ 見積作成管理 .fmp12」ファイルをベースに進めます）

※ クイックグラフについては FileMaker Master Book：初級編をご覧ください。
本書ではレイアウトオブジェクトとしてのグラフについて解説します。

12.3　グラフの作成

集計は数値や文字列でデータの情報を表示しますが、グラフは図示することで視覚的にデータの情報がわかるものです。一目でわかりやすい内容を目指します。

10 種類のグラフが用意されています。グラフも集計タイプフィールドを使用します。次表のような考え方でグラフの種類を選択します。

・縦棒グラフ
・積み重ね棒グラフ
・正負縦棒グラフ
・横棒グラフ
・積み重ね横棒グラフ
・円グラフ
・線グラフ
・面グラフ
・散布図
・バブル



図 12-29 グラフの種類

縦棒グラフ	時系列（年月など）や場所（地域や区域など）、人（担当者）などの、件数や金額などデータの比較を確認したいとき
積み重ね縦棒グラフ	時系列（年月など）や場所（地域や区域など）、人（担当者）などの、件数や金額などデータの比較とともに内訳も確認したいとき
正負縦棒グラフ	縦棒グラフと同じ確認をしたいが中点（ゼロであることが多い）よりも大きい、小さいも表現したいとき
横棒グラフ	時系列（年月など）や場所（地域や区域など）、人（担当者）などの、件数や金額などデータの比較を確認したいとき
積み重ね横棒グラフ	時系列（年月など）や場所（地域や区域など）、人（担当者）などの、件数や金額などデータの比較とともに内訳も確認したいとき
円グラフ	割合を確認したいとき
線グラフ	時系列（年月など）や場所（地域や区域など）、人（担当者）などの、件数や金額などデータの変化の比較を確認したいとき
面グラフ	時系列（年月など）や場所（地域や区域など）、人（担当者）などの、件数や金額などデータの変化の比較を確認したいとき
散布図	2 つのデータ系列の関係を確認したいときで、次のように考えます データ値の傾向が左から右へ上がっていれば強い相関関係 左から右へ下がっていれば相関関係が弱い はっきりとしない場合は相関がない
バブル	散布図と同じでデータに応じて円の大きさが変わるものです

12.3.1 グラフにどのようなデータを使うか

データをグラフにするには「**データ系列**」という考え方をします。どのような系列としてグラフにするか、がポイントです。

次表のデータがあるとして、横見出の「月ごとのグラフ」が必要なのか、縦見出の「お客様ごとのグラフ」が必要なのかを考えます。

	4 月	5 月	6 月	7 月	8 月	9 月
お客様 A	10	20	30	40	50	60
お客様 B	10	20	30	40	50	60
お客様 C	10	20	30	40	50	60
お客様 D	10	20	30	40	50	60
お客様 E	10	20	30	40	50	60

「月」ごとの「データ系列」には月ごとラベル（4 月など）で 5 つの「要素」データが含まれています。

```
10
10
10
10
10
```

「お客様」ごとの「データ系列」は顧客ごとラベル（お客様 A など）に 6 つの「要素」データです。

```
10
20
30
40
50
60
```

データを表のような形で考え、どの系列のラベルやデータを使うのか考えます。上の表から月ごとのグラフを作成するには、

・ラベル系列（4 月、5 月、6 月、7 月、8 月、9 月）　→　テキスト、日付、数字
・データ系列（10、20、30、40、50、60）　→　数字のみ

と捉えます。ラベル系列、データ系列は、次のいずれかの方法で設定できます。

・フィールドや変数、関数で作成される改行区切りのデータ
・対象レコードの特定のフィールドか計算式
・関連レコードの特定のフィールドか計算式

グラフの種類により次のようになります。

	X 軸	Y 軸
縦棒グラフ	ラベル	データ　1 つまたはそれ以上
積み重ね縦棒グラフ	ラベル	データ　2 つ以上
正負縦棒グラフ	ラベル	データ　1 つ
横棒グラフ	ラベル	データ　1 つまたはそれ以上
積み重ね横棒グラフ	ラベル	データ　2 つ以上
円グラフ	カテゴリラベル	データ　1 つ（スライス）
線グラフ	ラベル	データ　1 つまたはそれ以上
面グラフ	ラベル	データ　1 つまたはそれ以上
散布図	1 つまたはそれ以上のデータ系列（各データ系列に X 軸と Y 軸が必要）	
バブル	1 つまたはそれ以上のデータ系列（各データ系列に X 軸と Y 軸が必要）	

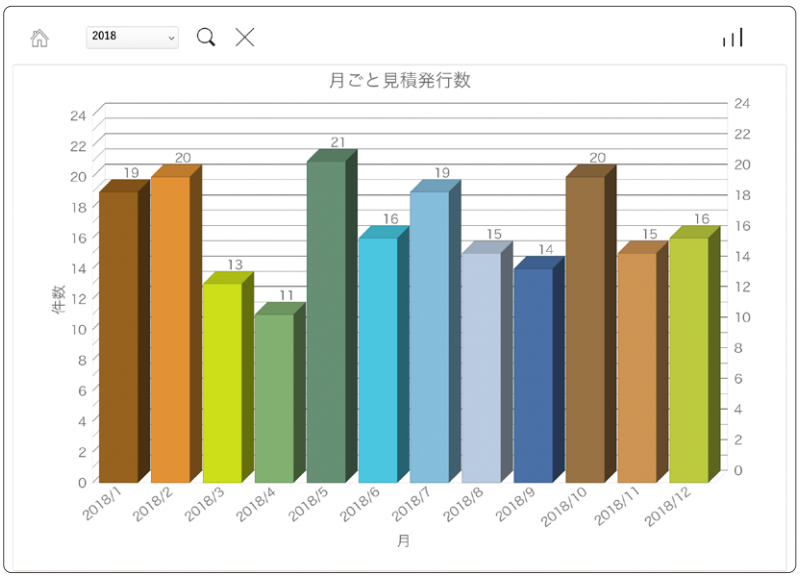


図 12-30 棒グラフの例

グラフデータの選択

どの対象レコードのデータにするかを、次の 3 種類から選択します。

現在の対象レコード

対象レコードについてグラフが表示されます。例えば、棒グラフの数は、対象レコードの数と同じ、ということになります。ラベル系列をテキスト、データ系列を数字で設定します。

現在の対象レコード（集計レコードグループ）

対象レコードを集計した結果についてグラフが表示されます。

現在のレコード（区切データ）

1 レコードについてグラフが表示されます。ラベル系列もデータ系列も、改行区切りのフィール名や計算式を指定します。改行区切りは List 関数、ValueListItems 関数や、集計タイプフィールドの一覧オプション、またはスクリプトで作成できます。

グラフオブジェクトの設定は、［グラフ設定］ダイアログを使います。



図 12-31［グラフ設定］ダイアログ

右側で設定を行うと、結果がプレビューされます。

［グラフ設定］のプレビュー
設定しながら結果を確認できるので、［グラフ設定］のプレビューはとても便利です。しかし、対象とするレコードが多いとプレビュー自体に時間がかかることがあります。そのようなときは、プレビューを一時停止し、確認したいときのみプレビューをするなどの操作をしてください。

［グラフ設定］のデータ
［実際のデータを使用］のプレビューでは表示に時間がかかることがあります。そのようなときは［サンプルデータを使用］に切り替えるなどの操作をしてください。

演習ファイルにいくつかのグラフを作成しておきました。まず、見積書の発行年月の棒グラフを確認します。次の手順で表示します（前ページの図の棒グラフです）。

(1)　演習ファイルの「ホーム」レイアウトの「演習用グラフ表示見積件数棒グラフ」ボタンをクリック
(2)　左上にあるポップアップメニューで「2018」を選択して虫眼鏡ボタンをクリック
(3)　右上のソートボタンをクリック

これは、棒グラフの例です。レイアウトモードで［グラフ設定］ダイアログの設定を確認してみてください。「見積管理」テーブルのフィールドが設定されています。

・データの選択は「現在の対象レコード」
・X 軸は「c_ 発行月」フィールド（テキスト）
・Y 軸は「s_ 見積件数」フィールド（集計／数値）

対象レコードが変更されると、そのデータを反映してグラフも変わります。

12.3.2 グラフ表示の例

別のグラフを確認します。「積み重ね横棒」グラフです。

「顧客情報」テーブルにダミーデータ用のフィールドを作成し「顧客積重横棒グラフ」レイアウト（「顧客管理」TO）とグラフを作成しました。

どのようにデータが表示されるか確認します。グラフに必要なデータの 1 例です。

「演習 12_6 から 8」フォルダ

演習 12-6　データとグラフ

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「レイアウトメニュー」から、「開発者用」フォルダにある「z_ 顧客情報」レイアウトに切り替え（表形式のはずです）、データを確認します。



自社担当者外部キー	顧客番号	会社名	ダミー1	ダミー2	ダミー3	ダミー4	ダミー5	郵便
096D69E6-154B	20001	株式会社ベゴニア	6	8	3	?	?	303-0062
AC82AA0B-3E3A	20002	ホオズキ株式会社	8	1	9	?	?	018-1414
AC82AA0B-3E3A	20003	ヘレボラス株式会社	3	7	8	?	?	018-2619
2B7BCC73-DCDF	20004	株式会社シャクヤク	9	7	10	?	?	326-0807
2B7BCC73-DCDF	20005	パウヒニア株式会社	3	7	3	?	?	999-7621
096D69E6-154B	20006	株式会社アサガオ	3	6	7	?	?	840-0004
26CC8BA1-AF7A	20007	シラン株式会社	5	1	8	?	?	808-0064
2B7BCC73-DCDF	20008	株式会社シャクナゲ	2	1	2	?	?	930-2334

図 12-32 ダミーデータ内容

a.　「ダミー 1」「ダミー 2」「ダミー 3」フィールドには数字が（1 つ）入力されています。
b.　「ダミー 4」フィールドには数字の改行区切りテキスト（3 行）が入力されています。
c.　「ダミー 5」フィールドには文字の改行区切りテキスト（3 行）が入力されています。

(2)　対象レコードによる表示の違いを確認します。

(3)　「顧客積重横棒グラフ」レイアウトに切り替えます。
a.　すべてのレコードを表示します。
b.　ソートを解除します。

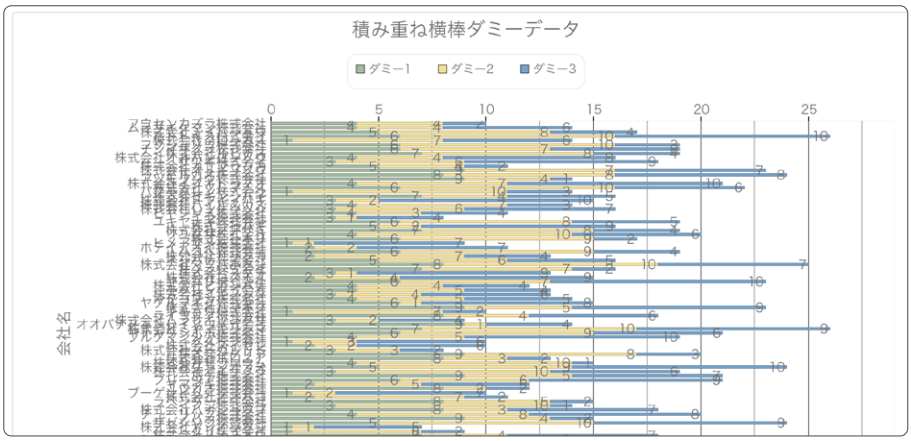


図 12-33「顧客積重横棒グラフ」レイアウトのグラフオブジェクト

(4)　すべての会社名が表示されているため、うまく読めません。
グラフオブジェクトを縦長に伸ばせば読み取れるかもしれませんが、ここでは対象レコードを絞り込んでみます。

(5)　レイアウトには「会社名」フィールドを検索するためのグローバル格納フィールドとボタン（スクリプト割り当て済み）が配置されています。
（簡易な仕組みとして配置されています）
a.　家アイコンの右のフィールドに「ア」と入力します。
b.　虫眼鏡ボタンをクリックします。

(6)　対象レコードが 14 件になりました。見やすくなりました。データが横に積み重なっていることを確認してください。

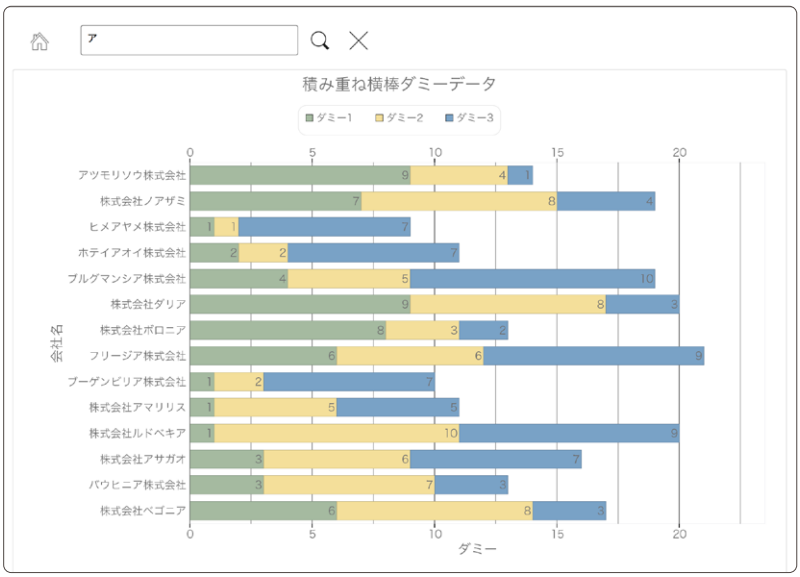


図 12-34 検索後の「顧客積重横棒グラフ」レイアウトのグラフオブジェクト

演習ここまで

積み重ね横棒グラフは、例えば、顧客ごと（ラベル系列）に月々の売り上げ金額や販売件数（データ系列）を積み上げることで、それらの合計した結果と内訳の両方を可視化できます。

なお、一度にたくさんのラベル系列を表示する場合、データの傾向を見やすくするためにソートが必要かもしれません。例えば、この演習のグラフではデータ系列の数値の合計数でソートすることで合計値による順位がわかりやすくなります。

グラフの設定を確認します。

演習 12-7　グラフの設定を確認

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「顧客積重横棒グラフ」レイアウトでレイアウトモードに切り替えます。

(2)　グラフオブジェクトをダブルクリックすると、設定内容を確認できます。

「グラフ」
X 軸の設定で、積み重なっているデータは次の 3 つのフィールドのデータだとわかります。
「ダミー 1」
「ダミー 2」
「ダミー 3」

Y 軸には「会社名」フィールドが設定されています。

「スタイル」
(省略します。配色などを設定する部分です)

「データソース」
「現在の対象レコード」が選択されています。
「集計レコードグループ」「個々のレコードデータ」のどちらを選択しても表示に変化はありません。

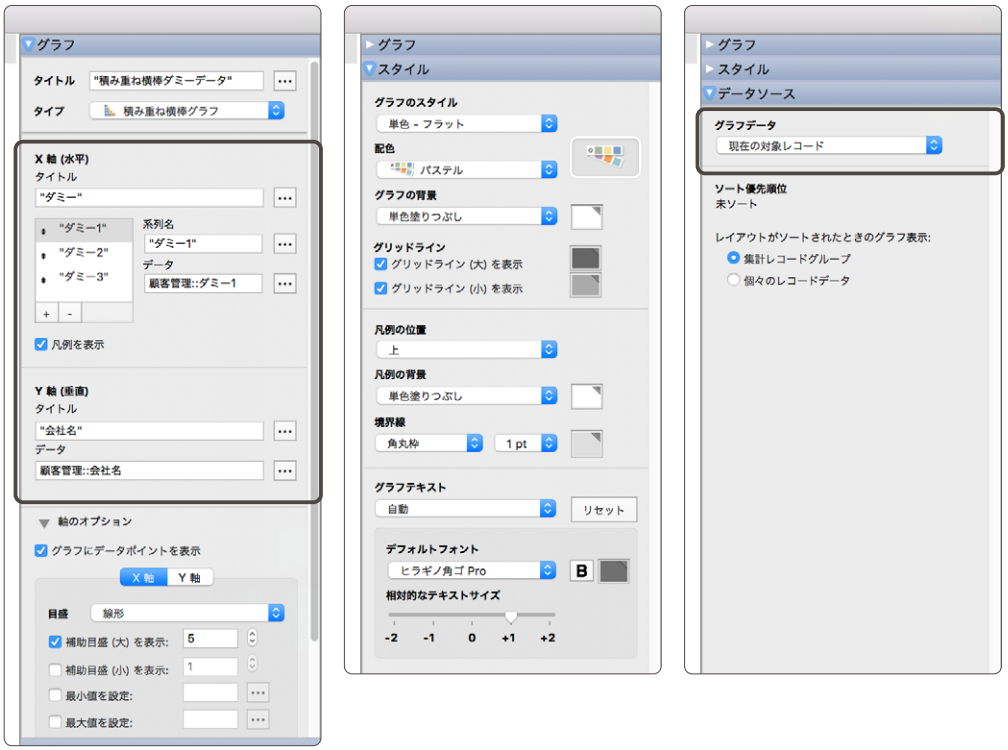


図 12-35［グラフ設定］ダイアログの設定内容

演習ここまで

やってみよう

他の設定を確認したり、設定を変更して表示の確認をしてみましょう。

現在の「積み重ね横棒グラフ」は、次の設定でグラフを表示していることがわかりました。「顧客管理」テーブルのデータです。

・現在の対象レコード
・データ系列は、1 レコードの 3 つのフィールド（数値）
・ラベル系列は、「会社名」フィールド（テキスト）

設定を変更します。

演習 12-8　グラフの設定による違いを確認する

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「顧客積重横棒グラフ」レイアウトでレイアウトモードに切り替えます。

(2)　グラフオブジェクトをダブルクリックすると、設定内容が表示されます。変更します。
データソースの設定を変更します。

a.　「データソース」で「現在のレコード（区切りデータ）」を選択します。
b.　［終了］をクリックして［グラフ設定］を閉じます。
c.　レイアウトを保存して、ブラウズモードに戻ります。
d.　現在のレコードだけのグラフになります。

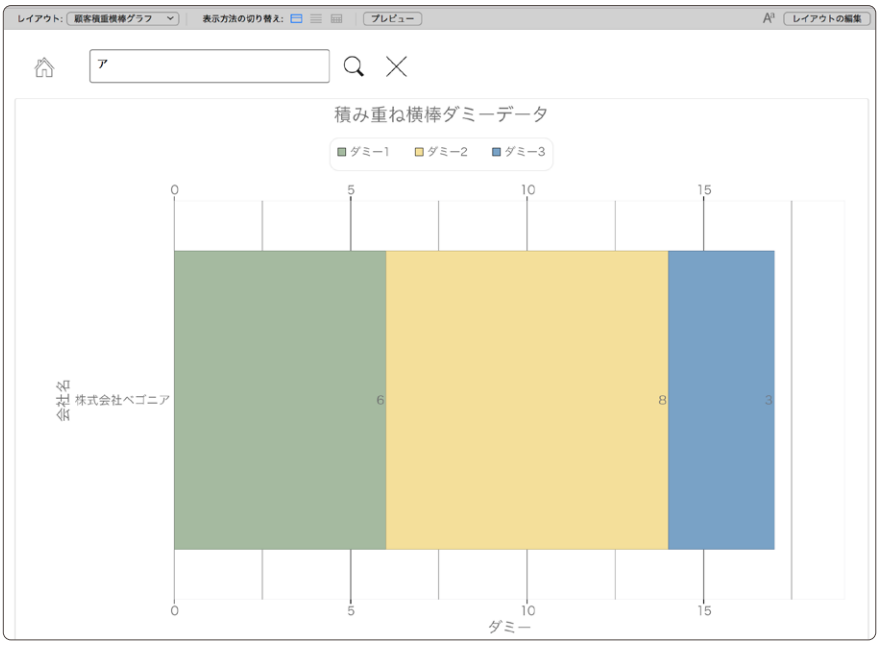


図 12-36　現在のレコードだけ

(3)　更に変更します。レイアウトモードに切り替えます。
データ系列の設定を変更します。

a. ［グラフ設定］を表示して「グラフ」を表示します。
b. X 軸の設定の系列名「ダミー 3」と「ダミー 2」を［–］で削除します。
c. 「データ」を「顧客管理 :: データ 4」フィールドを指定します。
d. 「凡例を表示」を **オフ** にします。
e. Y 軸の設定の「タイトル」を「ダミー」と変更します。
f. 「データ」を「顧客管理 :: データ 5」フィールドを指定します。
g. ［終了］をクリックして［グラフ設定］を閉じます。
h. レイアウトを保存して、ブラウズモードに戻ります。
i. 現在のレコードだけで、3 つの棒グラフになります。

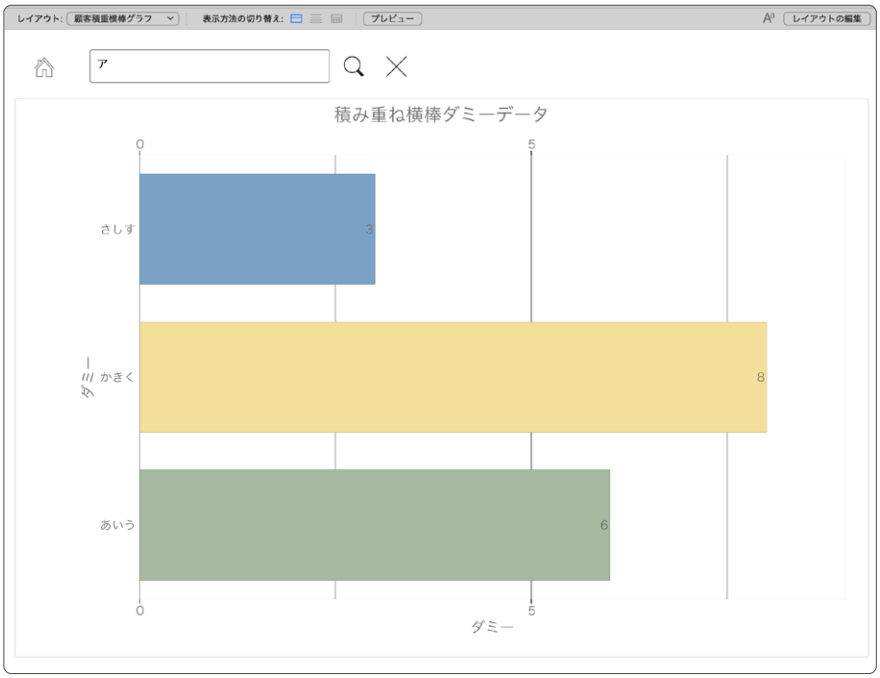


図 12-37 現在のレコードだけど、棒グラフが 3 つ

積み重ねではなくなりました。次のデータを利用しました。

・X 軸の系列は 1 系列で 3 行の改行区切（データ） → 「データ 4」フィールド
・Y 軸の系列も 3 行の改行区切（ラベル） → 「データ 5」フィールド

演習ここまで

この演習の 2 番目の設定変更では、「顧客管理」テーブルのデータ例を表示しています。

・現在の対象レコードの任意の 1 件のレコード
・データ系列は、1 つのフィールド（改行区切）
・ラベル系列は、1 つのフィールド（改行区切）

グラフには、集計タイプフィールドや数値や計算式が入力されたフィールド、変数なども使います。

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 12_9」フォルダの中の「12_ 見積作成管理 .fmp12」ファイルをベースに進めます）

別のグラフを確認します。「円」グラフです。

「見積明細」テーブルから「見積明細円グラフ」レイアウトにグラフを作成しました。**集計タイプフィールドを使っているグラフ**を確認します。

「演習 12_9」フォルダ

演習 12-9　集計タイプフィールドのグラフ

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「ホーム」レイアウトで、「演習用グラフ表示見積明細円グラフ」をクリックします。

(2)　「見積明細円グラフ」レイアウトに切り替わります。
a.　すべてのレコードを表示します。
b.　ソートを解除します。

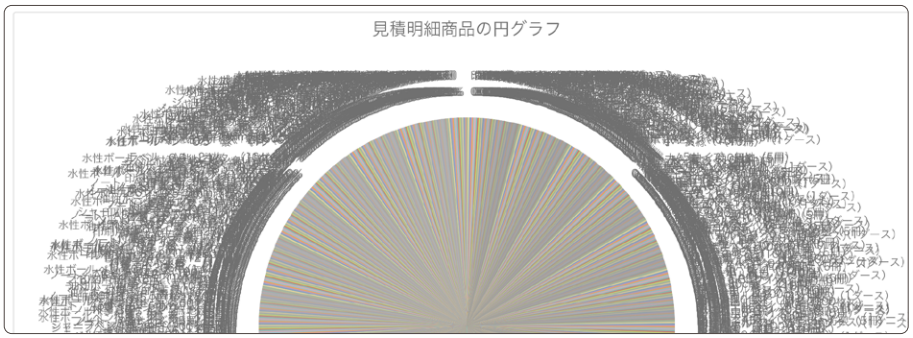


図 12-38「見積明細円グラフ」レイアウトのグラフオブジェクト

(3)　現在は、すべての商品名が表示されているために読めません。
「見積明細」テーブルは 975 件登録があります。集計されていないとこのような感じです。

(4)　レイアウトモードに切り替えます。

(5)　グラフオブジェクトをダブルクリックし［グラフ設定］を表示し、設定を確認します。
・「データソース」では「現在の対象レコード」を選択している
・「集計レコードグループ」を選択している。　……など。

(6)　ブラウズモードに戻ります。

(7)　このままで集計されるとどうなるかを確認します。集計結果を表示するには、ソートが必要でした。レイアウトの右上のボタンには、次のソートが設定されています。

・「見積管理 _ 明細 _ 商品 :: c_ 商品番号と商品名」フィールドの　昇順

a. レイアウト右上のボタンをクリックします。
b. 集計されて円グラフが変化します。少し見やすくなりました。

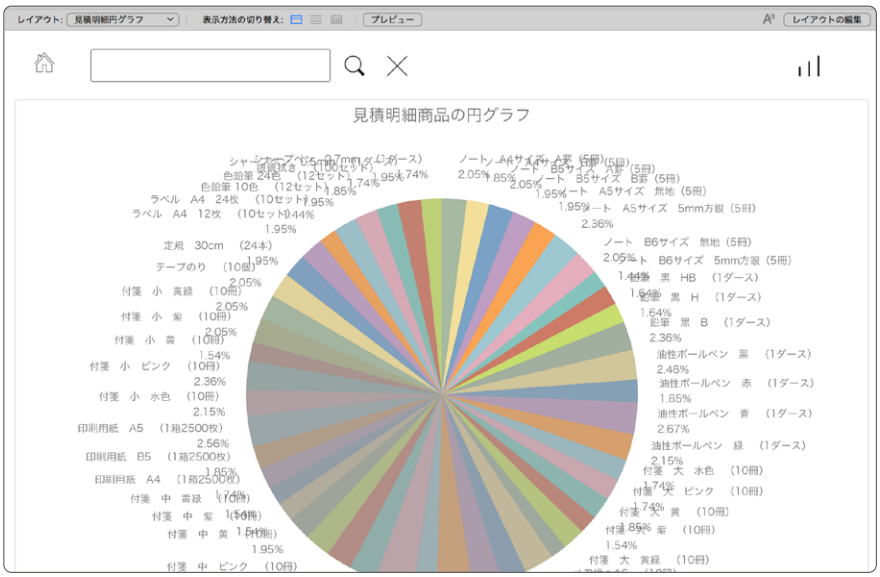


図 12-39 ソートで集計結果が表示された

(8) さらに件数を絞り込みます。レイアウトには、一部のレコードに限定するための簡易な仕組みが配置されています。
a. フィールドに「鉛筆」と入力して虫眼鏡ボタンをクリックします。
b. 対象レコードが 92 件になります。
c. レイアウト右上のボタンをクリックします。集計されて円グラフが変化します。

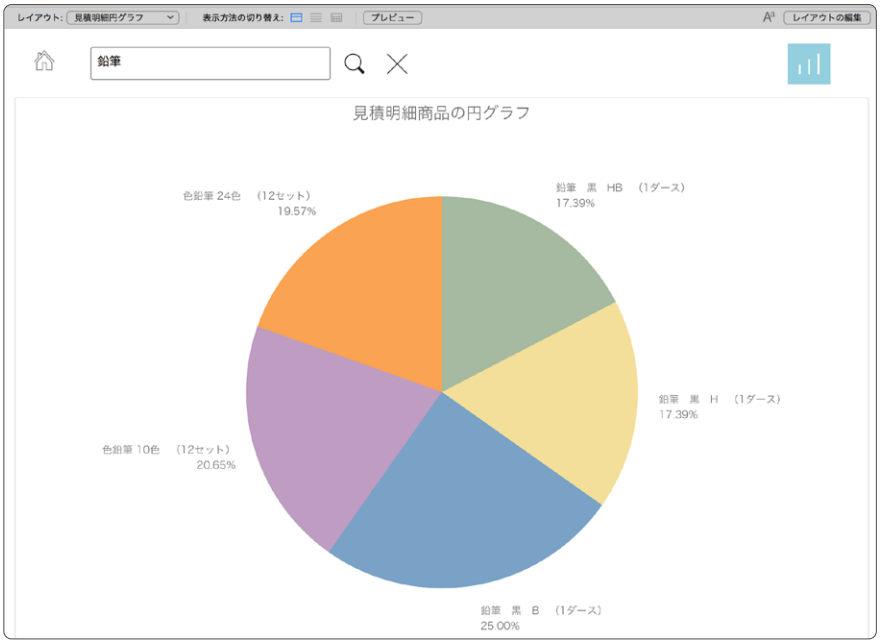


図 12-40 検索とソートで集計結果が表示された

演習ここまで

この演習では「見積明細」テーブルのデータ例を表示しています。

・現在の対象レコード
・データ系列は、「s_ 商品内訳」フィールド（集計／数値）
・ラベル系列は、「商品名」フィールド（テキスト）

このグラフは、データ系列に集計結果を指定していますのでソートしないと正しい結果が表示されません。さらに円グラフでは、対象とするラベル系列の件数が多すぎるとスライス（切れ目）も同様となってグラフの内容を読み取れません。

別のグラフを確認します。**グラフの種類を変更してみましょう。**

演習 12-10 グラフの種類の変更

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1)　「ホーム」レイアウトで「演習用グラフ表示見積件数棒グラフ」ボタンをクリックします。

(2)　「月ごと見積件数棒グラフ」レイアウトが表示されます。

(3)　件数を絞り込んでからソートします。
a.　ポップアップメニューで「2018」を選択ます。
b.　虫眼鏡ボタンをクリックします。対象レコードが 199 件になります。
c.　レイアウト右上のボタンをクリックします。

(4)　レイアウトモードに切り替えて、グラフタイプを変更します。
a.　グラフオブジェクトをダブルクリックして［グラフ設定］を表示します。
b.　「グラフタイプ」を「線グラフ」に変更します。
c.　［終了］をクリックして［グラフ設定］を閉じます。

(5)　レイアウトを保存してブラウズモードに戻ると線グラフが表示されます。

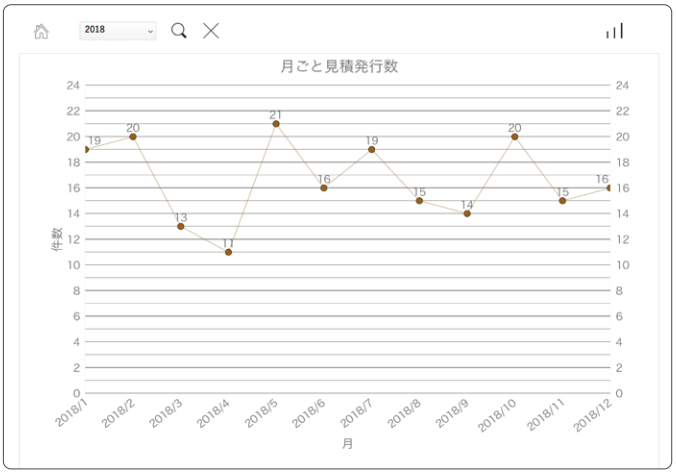


図 12-41　検索とソートで結果が表示された

演習ここまで

次の例です。「顧客管理」TO から、「顧客管理 _ 見積」TO の関連レコードの見積状況の内訳を円グラフで表わします。次の設定です。

・関連レコード（「顧客管理 _ 見積」TO）の任意の 1 レコード
・データ系列は、「顧客管理 _ 見積」TO の「見積状況件数」フィールド（数値）
・ラベル系列は、「顧客管理 _ 見積」TO「s_ 見積状況」フィールド（テキスト）

［グラフ設定］を確認／変更します。

演習 12-11 関連レコードのグラフ

- **使用ファイル：12_ 見積作成管理 .fmp12**

(1) 「ホーム」レイアウトで「顧客管理」ボタンをクリックします。

(2) 「演習用グラフ表示顧客別状況円グラフ」ボタンをクリックします。
（「顧客ごと見積状況円グラフ表示」スクリプトが割り当てられています）

(3) 「顧客別見積状況円グラフ｜関連」レイアウトが表示されます。レイアウトにはグラフオブジェクトが置いてありますが、なにも設定されていません。

(4) 丸に × のボタンをクリックします（カードスタイルウインドウでグラフが表示されるようにスクリプトを作成しました）

(5) 「顧客別見積状況円グラフ｜関連」レイアウトで、レイアウトモードに切り替えてます。

(6) グラフをダブルクリックして［グラフ設定］で設定をおこないます。

a. 「グラフ」
「タイトル」に以下の計算式を指定します。

```
"顧客別見積状況内訳　" & 顧客管理::c_顧客番号と顧客名
```

「タイプ」は、「円グラフ」を選択します。

「カテゴリラベル」は、「顧客管理 _ 見積 :: 見積状況」フィールドを指定します。
「スライスデータ」は、「顧客管理 _ 見積 ::s_ 見積状況件数」フィールドを指定します。

「オプション」で、「グラフに値を表示」を **オン** 「実際の値」を **オン** にします。

b. 「スタイル」は、自由に設定してください。

c. 「データソース」
「グラフデータ」は、「関連レコード」を選択します。

「関連テーブル」を「顧客管理 _ 見積」を選択します。

「関連レコードのソート順」の［指定 ...］をクリックして、ソートを指定します。

・「見積状況」フィールドの「見積状況」値一覧順

(7)　［終了］をクリックして［グラフ設定］を閉じます。

(8)　レイアウトを保存してブラウズモードに戻ります。

(9)　表示を確認します。

a.　「ホーム」レイアウトで「顧客管理」ボタンをクリックします。

b.　顧客番号「20001」のデータを確認します。

見積状況は、次の件数内訳です。

・作成は　　10 件
・お客様検討中は　9 件
・結果確定済は　　7 件

c.　「演習用グラフ表示顧客別状況円グラフ」ボタンをクリックします。

d.　内訳が円グラフで表示されています（確認したものと同じです）。

（テストのための関連レコードは「顧客情報」テーブルの、最初の 10 件のレコードのみに入力されています／各種設定などは、次図も確認してください）

演習ここまで



図 12-42　設定内容

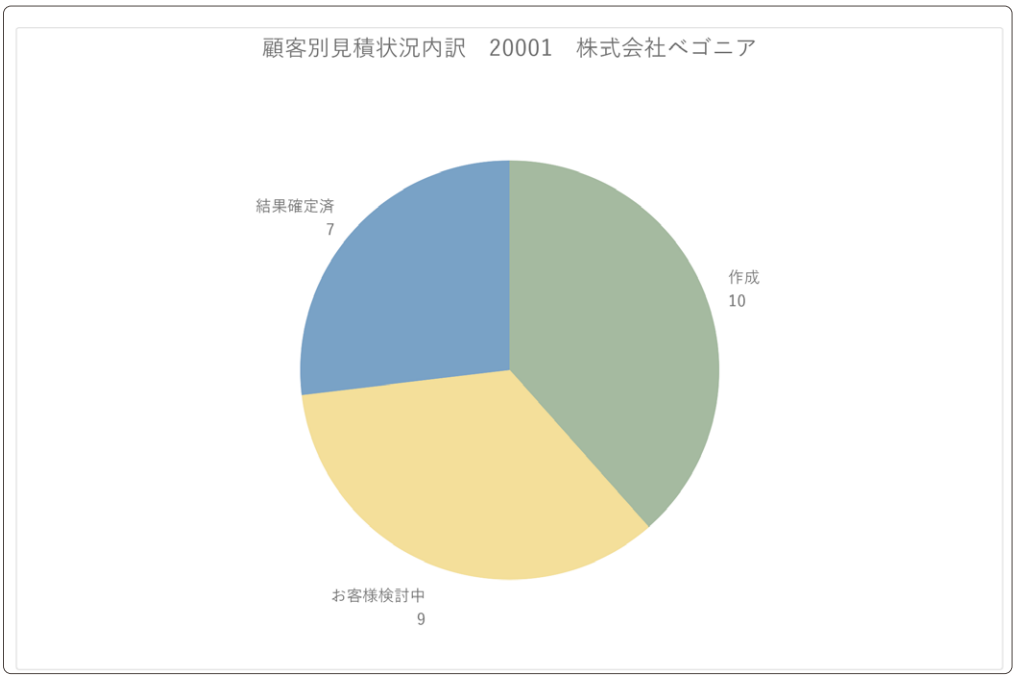


図 12-43 グラフ表示

演習ファイルを閉じてください。

第 13 章 モバイル／ FileMaker Go

クライアントのハードウェアはパソコンだけではなく iPad や iPhone を使用することが可能です。
FileMaker Go ならではの独自の機能があります。

13.1 FileMaker Go のクライアント

クライアントのハードウェアとして iPad や iPhone を使用することは特別なことではなくなりました。外出先から接続するから、という理由以外でも iPad、iPhone などからカスタム App を利用することが可能です。

なお、パソコンだけを想定して作成されたレイアウトを iPad や iPhone で表示すると、フォントやボタンが小さくて見づらかったり使いづらいかもしれません。

FileMaker Go には独自の機能があります。それを活かし、同じ作業でもパソコンとは違う方法で設計することが可能です。iPad や iPhone から使うための専用画面を計画し作成することをお勧めします。

13.1.1 カスタム App ファイルをどこに持つか

2 つの方法があります。

共有ファイルに接続する（デバイスにファイルを持たない）
ネットワークや Wi-Fi 環境が整備されるようになったことで、FileMaker Cloud や FileMaker Server で公開されているカスタム App ファイルに企業や団体の中や外出先からでも同じように接続して使うことができるようになりました（オンライン）。

オンライン接続できない場所も存在します。または、セキュリティ上の制限で外出先から接続できないルールがあるかもしれません。iPad や iPhone にカスタム App ファイルを保存してオフラインで使う方法があります。

デバイスにファイルを持つ
iPad や iPhone にファイルを保存することは、ネットワーク環境に左右されずにいつでもカスタム App が使えるという利点があります。

考慮することは、アカウントでログインすることや、iPad や iPhone の OS（iOS/iPadOS）の設定でパスコードロックをかけるなど、置き忘れ、紛失などの場合に簡単に他の人が内容を見ることができないようにセキュリティを設定することです。

オフライン使用では、データの同期の有無も重要です。データの一元化によるデータの蓄積とその使用についてどのように考えるか、です。

・商品カタログの表示

このような場合、定期的に iPad や iPhone にカスタム App ファイルを保存する作業だけで、常に最新情報は FileMaker Cloud や FileMaker Server で管理している情報であると位置づければ良いのです。

・営業担当者が顧客訪問時に訪問記録を外出先で作成する

同期が必要です。帰社時に接続して同期ができるような仕組みを考える必要があります。

デバイスへカスタム App ファイルを送る方法
具体的な方法については本書では取り扱いません。FileMaker Pro のヘルプやクラリス・ジャパン株式会社の Web サイトのナレッジベース「iPad、iPhone、および iPod touch 上で、Dropbox および Box.net に保存された FileMaker ファイルを開く方法」（https://support.claris.com/s/answerview?language=ja&anum=000024031）などを参照してください。

デフォルトでは、デバイスのローカルに保存したファイルは [マイ App] タブの [iPhone 上]、[iPad 上]、または [iPod 上] に表示され、ドキュメントフォルダに保存されます。

データの同期
具体的な方法については本書では取り扱いません。FileMaker Pro のヘルプやクラリス・ジャパン株式会社の Web サイトのナレッジベース「データベースの同期 - アプローチの概要」（https://support.claris.com/s/answerview?language=ja&anum=000024001）を参照してください。

13.1.2 FileMaker Pro との違い

一番大きな違いは、FileMaker Go ではカスタム App のテーブルやフィールドをつくるなどの開発作業ができないことです。その他、動作には多少の違いがあります。例えば、次のような内容です。

・プレビューモード、レイアウトモードがありません。

・カスタムメニューの表示に違いがあります。

・キーボードを接続するとショートカットが使えます。

・ポータル内では、垂直スクロールバーではなく上下にスワイプすることによってスクロールできます。

・スライドコントロールでは、［スワイプジェスチャを有効にする］が有効な場合、スワイプして前または後ろに移動できます。

・FileMaker Pro でファイルに設定した印刷オプションは、FileMaker Go の同じファイルの印刷オプションには適用されません。逆の場合も同様です。

・iOS/iPadOS でサポートされていないフォントは、サポートされているフォントに変換して表示されます。

・テキストスタイルは、太字、斜体、下線、単語下線、二重下線、字消し線、タイトル、大文字、小文字に対応しています。

・タブ順が設定されている場合、［Tab］や［return］キーで移動できるのはフィールドだけです。レイアウトオブジェクトへの設定や、［Enter］キーでの移動は対応していません。

・FMP12 または XML 形式へのレコードのエクスポートはできません。

・［デバイスから挿入］スクリプトステップを使用して、コンテンツをオブジェクトフィールドに挿入することが可能です。
ミュージックライブラリ
フォトライブラリ
カメラ
ビデオカメラ
マイク
署名

・また、［デバイスから挿入］スクリプトステップを使用して、バーコードソースからコンテンツをオブジェクトフィールドまたはテキストフィールドに挿入

・レイアウト上のオブジェクトタイプフィールドでは、インスペクタの［データの書式設定］オプション［イメージ］および［インタラクティブコンテンツ］の違いに対応していません。

・デバイスにファイルが保存されている場合、オブジェクトタイプフィールドのデータを外部に格納することはできません。

・iOS/iPadOS の「通知」を利用できます。

・iOS/iPadOS の「センサー情報」を取得できます。

・ファイルオプションの［保存されている資格情報による認証を許可］のオプション［要 iOS または iPadOS パスコード］をオンにすると、Touch ID（指紋認証）や Face ID（顔認証）を利用できます（デバイスがサポートしている場合のみ）。

・実行中のスクリプトを停止するには、画面の任意の場所をタップしてスクリプトの停止を確認します。停止を確認しないと、スクリプトは実行を続けます。

・別のアプリケーションに切り替えると、FileMaker Go はオーディオまたはビデオファイルが再生中でない限り中断状態になります。

主なものを紹介しました。初めからすべてを覚え、念頭において開発する必要はありません。

さらに詳しい情報は、FileMaker 製品のマニュアルやその他の公開されている文書、FileMaker Go のヘルプ、「FileMaker Go デベロップメントガイド」を参照してください。

13.2 レイアウト

FileMaker Go のレイアウトは、パソコン用と同じものを使える場合もあります。しかし入力項目やボタンのサイズなどが小さくなりますし、だからといって画面が小さい iPhone、iPad に合わせるとパソコンでは使いづらいかもしれません。

パソコン、iPad、iPhone それぞれで使いやすいレイアウトを作成すると管理が楽になります。もちろん、フィールドの追加などが必要な場合はそれぞれ作業をしなければなりませんが、レイアウトに関する作業をパソコン、iPad、iPhone のどれでも丁度良いようにサイズや位置を常に調整するよりも、それぞれ最適な作業をする方がユーザの使いやすさも作業にかかる時間も短縮できます。これは FileMaker WebDirect でも同じです。

縦か横か

iPad、iPhone のためのレイアウトでは、ユーザが使いやすい画面の方向を決定します。

・縦向き（縦長の画面）
・横向き（横長の画面）
・両方

両方の場合、縦横それぞれにレイアウトを作成することをお勧めします。向きによってスムーズにすべてのレイアウトオブジェクトの配置やサイズが適切になるような調整に時間がかかったり、画面の表示領域をうまく使えないことがあるかもしれません。

iOS / iPadOS

iPad、iPhone は、いわゆるデスクトップパソコンより画面が小さいことが多いです。操作はマウスではなく指でタッチしたりスワイプしたりします。前述したように縦横の向きもあります。デスクトップパソコンのためのレイアウトとは、考え方を変える必要があります。

画面は小さいけれど文字は読める大きさでなければなりませんし、ボタンは指で操作できるサイズにしなければなりません。一度に表示できる項目の数が限定されます。ただし縦に長いリスト形式などのレイアウトのスクロールは指で簡単にできます。

・指でタッチされる想定のレイアウトオブジェクトは 44×44pt 以上の大きさにする
・可能な限り、キーボードによる入力よりも指でタッチして選択入力できるようにする
・キーボード入力が必要な項目はタッチキーボードタイプを適切に指定して入力を補助する
・画面が小さいのでデータに視線がすぐに行くようにたくさんの色は使わない
・グラデーションを多用すると画面表示が遅くなることがあるので気をつける
・ウインドウは常に 1 つになるように工夫する（ポップオーバーやカードスタイルウインドウを使うなど）

FileMaker Pro にはタッチ用のテーマが複数用意されています。これらも活用してください。また、実際に使用するために、次のようなことも試してみてください。

・画面を実際に見る場所で、どのように見えるか確認してみる

外出先、昼、夜、乗物の中など一番使われそうな場所と時間帯と画面の見え方によって選択するテーマや色が変わるかもしれません。Apple の「ユーザインターフェイスのデザインのヒント」や「Human Interface Guidelines」（英語）も参考にしてみてください。

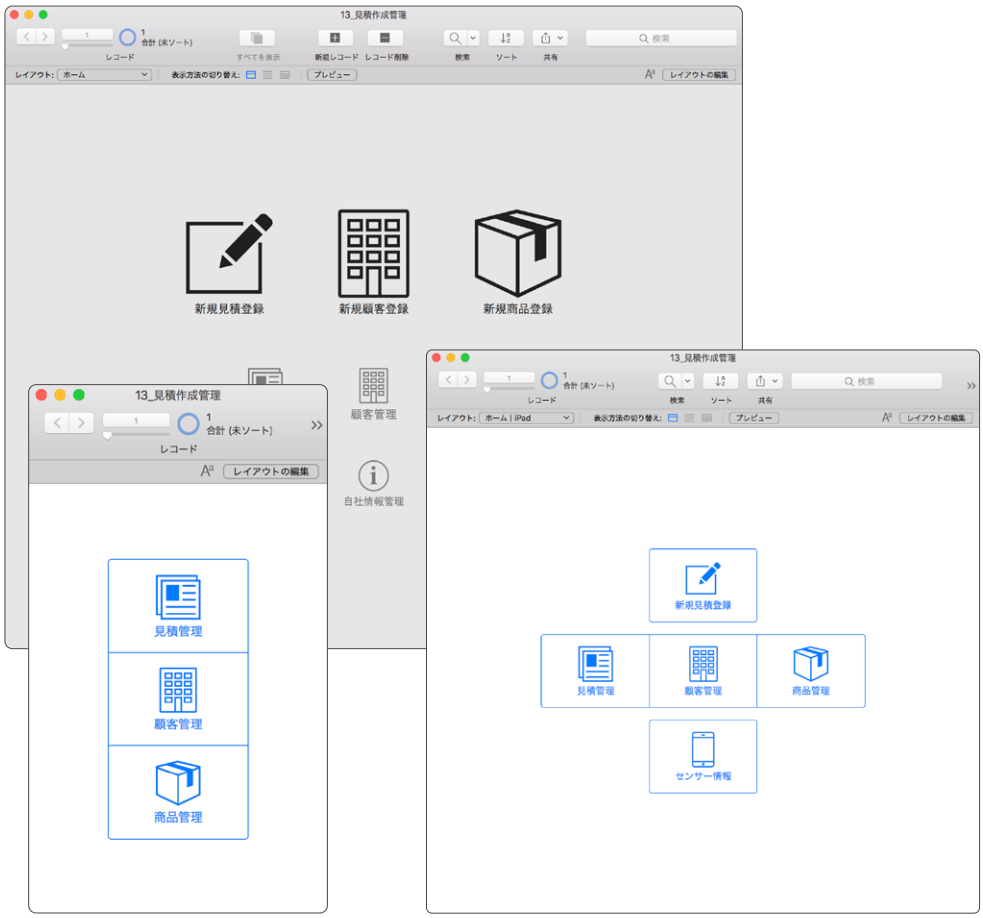


図 13-1 トップ画面の違い例

すべての作業ができる必要があるか

状況によってユーザがクライアントを自由に選べるのであれば、画面が小さい iPhone で画面が大きいパソコンと同等の作業をすることを必ずしも前提にしなくても良いかもしれません。必要な機能や項目を限定することで、小さい画面に上手く項目を配置できます。

13.3　スクリプト

FileMaker Go は FileMaker Pro とほとんど同じスクリプトステップが使用できます。しかし、レイアウトでも前述したように『パソコンではない』ということを念頭においてスクリプトを作成することが必要な場合もあります。

iPad、iPhone の処理能力、Wi-Fi の通信速度が一定ではなく遅くなるかもしれないことも考えなければなりません。FileMaker Cloud ／ FileMaker Server でホストされている場合、複雑で長時間掛かる作業は、iPad、iPhone で行うよりもパソコンやサーバーで行うことを検討します。例えば、ユーザの待ち時間を減らすことが可能です。

［サーバー上のスクリプト実行］スクリプトステップ

を使用して FileMaker Cloud ／ FileMaker Server で作業を実行すると待ち時間が少なくなったり、Wi-Fi 上の通信量を少なくすることが可能です。

サポートされていないスクリプトステップの確認
［スクリプトワークスペース］で確認できます。サポートされていないスクリプトステップは FileMaker Go でエラーになります。

エラー処理をして、

・エラー番号　3（コマンドが使用できません）
・エラー番号　4（コマンドが見つかりません）

の場合の条件分岐によるスクリプトの作業を行います。FileMaker Pro のエラー番号の詳細については、ヘルプを参照してください。

ユーザがスクリプトを停止する
［エラー処理］と［ユーザによる強制終了］スクリプトステップが　**オフ**　またはスクリプトに使用されていない場合、エラーメッセージがユーザに表示されます。ユーザはそのメッセージの選択肢により、スクリプトを途中終了することができます。

スクリプトが途中で終了することで、想定していないことが起きないよう、エラー処理とエラーによる条件分岐をおこないます。

スクリプトの違い
動作が少し違う場合があります。例えば、［メールを送信］スクリプトステップの、「ダイアログなしで実行」オプションは指定しても無視されるため、必ず手動でメールを送信する必要があります。

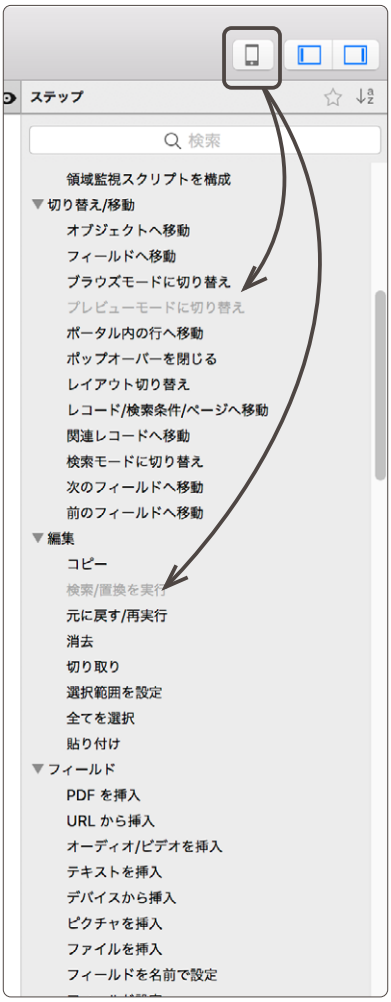


図 13-2［スクリプトワークスペース］の表示

［URLを開く］スクリプトステップ

Safari に切り替えずに表示できます。[Safari で開く]オプションのあるポップアップが表示されます。

ウインドウの扱い
FileMaker Go も複数ウインドウは可能です。ただし FileMaker Pro と違いがあります。常に 1 つのウインドウのみが表示されます。また［ウインドウの調整］スクリプトステップは、「隠す」オプションを選択すると、ウインドウの順番が変わる（一番後ろになる）ことになります。

やってみよう
次の演習ファイル「13_ 見積作成管理 .fmp12」ファイルの「ホーム｜ iPad」レイアウトには「URL を開く」ボタンを置きました。iPad や iPhone でこのファイルを開いて試すことができます。

演習ファイルを開いてください。

（ここから、本書に付属している演習ファイルの「13 章開始」フォルダの中の「13_ 見積作成管理 .fmp12」ファイルをベースに進めます）

13.3.1 起動時に専用のレイアウトに切り替える

現在、演習ファイルのユーザ用の主なレイアウトは下表の通りです。

	画面名	レイアウト種類	パソコン	モバイル
1	ホーム	フォーム形式	○	×
2	見積詳細	フォーム形式	○	×
3	見積リスト	リスト形式	○	×
4	見積発注書｜印刷用	フォーム形式	○	×
5	顧客管理｜マスタ詳細	フォーム形式（マスタ / 詳細）	○	×
6	商品管理｜マスタ詳細	フォーム形式（マスタ / 詳細）	○	×
7	ホーム｜ iPad	フォーム形式	×	○
8	見積リスト｜ iPad	リスト形式	×	○
9	見積詳細｜ iPad	リスト形式	×	○
10	顧客管理｜マスタ詳細｜ iPad	フォーム形式（マスタ / 詳細）	○	×
11	商品管理｜マスタ詳細｜ iPad	フォーム形式（マスタ / 詳細）	○	×
12	ホーム｜ iPhone	フォーム形式	×	○
13	見積リスト｜ iPhone	リスト形式	×	○
14	顧客管理｜ iPhone	リスト形式	○	×
15	商品管理｜ iPhone	リスト形式	○	×

薄く塗った部分が、モバイルデバイス用に作成したレイアウトです。これらの画面を遷移できるようにします。

クライアントのデバイスのタイプによって、それぞれのために用意された「ホーム」レイアウトに切り替えるスクリプトです。

「13 章開始」フォルダ

演習 13-1　起動時にクライアントにあわせて画面を表示する：スクリプト

- **使用ファイル：13_ 見積作成管理 .fmp12**

(1)　［スクリプトワークスペース］を表示し「起動処理 _ 演習用」を開きます。
現在は 1 行のみ記述されています

(2)　次の内容のスクリプトを作成してスクリプトを保存します（詳細な手順は省略します）。

```
レイアウト切り替え [ 「起動画面」(z_ユーティリティ ) ]

If [PatternCount ( Get ( アプリケーションバージョン ) ;
"Go" )]

  If [Get( デバイス )=3]
     レイアウト切り替え [ 「ホーム｜iPad」(z_ユーティリティ ) ;
       アニメーション : 下からスライドイン ]

  Else if [Get( デバイス )=4]

     レイアウト切り替え [ 「ホーム｜iPhone」(z_ユーティリティ ) ;
       アニメーション : 左からスライドイン ]

  End If
Else

  レイアウト切り替え [ 「ホーム」(z_ユーティリティ ) ]

End If
ウインドウの調整（収まるようにサイズ変更）
```

演習ここまで

このスクリプトは、「見積作成管理 .fmp12」ファイルの起動時に動作するようにすることで、適切な「ホーム」レイアウトに切り替えることができるようになります。

「起動 _ 演習用」スクリプトをファイルを起動時に実行するために、スクリプトトリガを設定します。

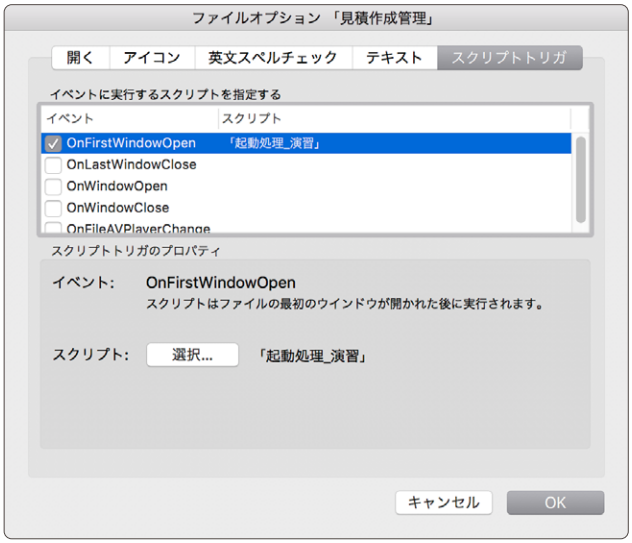
演習 13-2　起動時にクライアントにあわせて画面を表示する：ファイルオプション

- **使用ファイル：13_ 見積作成管理 .fmp12**

(1)　[ファイル]–[ファイルオプション ...] メニューを選択し「スクリプトトリガ」タブをクリックします。

(2)　「OnFirstWindowOpen」にチェックし、「起動処理 _ 演習用」スクリプトを指定します。

図 13-3 起動時のスクリプトに指定



(3)　[OK] をクリックしてファイルを終了します。

(4)　パソコンと iOS/iPadOS デバイスの両方で起動して表示される画面の違いを確認してください（iOS/iPadOS デバイスは、手元にある場合のみ確認してください）。

演習ここまで

このスクリプトは、FileMaker Pro か、FileMaker Go かを条件分岐しています。

```
Get ( アプリケーションバージョン )
```

次に、FileMaker Go だったら、どのデバイスかを条件分岐しています。

```
Get ( デバイス )
```

これらの関数の結果については、ヘルプも参照してください。想定される結果通りになるか、必ずテストします。

［レイアウトの切り替え］スクリプトステップのオプション
FileMaker Go の場合のみ、「アニメーション」でレイアウト切り替え時のアクションを指定可能です。右図のバリエーションが用意されていますので、試してみてください。

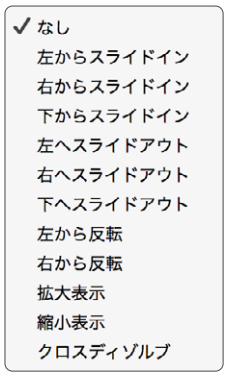


図 13-4 アニメーションの種類

iPhone や iPad から演習ファイルを開くと専用のレイアウトを表示することができるようになりました。いくつか動作を確認します。

13.3.2 デバイスからの挿入

FileMaker Go で開いているファイルへ iOS / iPadOS から特別な入力を行うために、

［デバイスからの挿入］スクリプトステップ

が用意されています。下図のオプションが用意されています（歯車アイコンをクリックして［次から挿入］の右のポップアップメニューから選択します）。

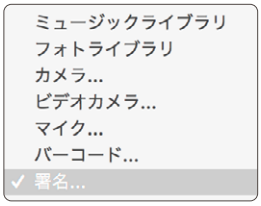


図 13-5［デバイスからの挿入］オプション

既にデバイスのライブラリに保存されている音楽（ミュージック）、写真（フォト）は、各ライブラリに移動してファイルを選択します。

それ以外の挿入するバイナリファイルについては、さらにオプションを設定します。例えば、上図で［カメラ ...］を選択すると、下図の設定が可能です。



図 13-6 カメラを選択した場合

「見積作成管理 .fmp12」ファイルに用意した、署名についてのスクリプトの動作を確認します。

iPhone の場合は次のとおりです（iOS デバイスが、手元にある場合のみ確認してください。手軽に確認するには、AirDrop などでデバイス上にファイルを保存してください）。

(1) FileMaker Go で演習ファイルを開きます。
(2)「見積管理」ボタンをタップします。
(3) 見積リストに切り替ります。

このレイアウトでは、「見積状況」フィールドが「結果確定済」になると「見積結果」フィールドとボタンが表示されます。

(4) 見積番号「10001」のレコードで「お客様検討中」をタップしリストから「結果確定済み」を選択します。



図 13-7 結果確定済みに変えた

(5)「注文あり／注文なし」のラジオボタンが表示されます。
(6)「注文あり」に変更すると、カスタムダイアログで署名の意思が確認されます。
(7)［OK］をタップすると、署名挿入インターフェースが表示されます。



図 13-8 ダイアログ

(8) 署名をもらいます。
(9) 署名後［承認］をタップします。
(10) 署名が保存されます。

(11) 署名後は、鉛筆アイコンのボタンをタップすると署名内容を確認できます。



図 13-9 署名挿入インターフェース

この動作は**「デバイスからの署名挿入 _ 見本」スクリプト**を使っています。次の内容です。

```
レコード / 検索条件確定

If [ 見積管理::見積状況 = "結果確定済" and 見積管理::見積結果 = "注文あり" ]

    カスタムダイアログを表示 ["署名いただけますか？" ; "ご発注の署名をいただけますか？" ]

    If [ Get ( 最終メッセージ選択 ) = 1 ]

        エラー処理 [ オン ]
        デバイスから挿入 [ 見積管理::発注署名; タイプ: 署名; タイトル: "サインをお願いいたします" ; メッセージ: "ご発注誠にありがとうございます" ; プロンプト: "ここにサイン" ; 表示: フルスクリーン ]

        If [ Get(最終エラー) ]
        Else

            カスタムダイアログを表示 [ "ありがとうございました。"]
            フィールド設定 [ 見積管理::見積状況 ; "結果確定済" ]
            フィールド設定 [ 見積管理::見積結果 ; "注文あり" ]
            エラー処理 [ オフ ]

        End If

    Else
    End If

    ポップオーバーを閉じる
    レコード / 検索条件確定
End If
```

ラジオボタン（「見積結果」フィールド）のオブジェクトのスクリプトトリガに設定されています。入力内容が変更される（OnObjectModify）されると、その内容に従って署名挿入インターフェースを表示します。実際には、もう少し考えなければなりません。例えば、

・一旦「注文あり」にしたが「注文なし」にする、あるいはその逆の作業
・すでに署名をもらっている場合に、再度署名をもらえる方が良いかどうか

などです。

13.3.3 オブジェクトタイプフィールドの操作

［デバイスから挿入］などでオブジェクトタイプフィールドに保存されたバイナリデータを、

・スクリプトステップ
・関数
・スクリプトトリガ

などを使用して、FileMaker Go で操作することができます。次のようなものがあります。

スクリプトステップ：

```
AVPlayer再生
AVPlayerオプション設定
AVPlayer再生状態設定
```

スクリプトトリガ：

```
OnObjectAVPlayerChange
OnFileAVPlayerChange
OnExternalcommandReceived
```

関数：

```
GetAVPlayerAttribute
Get ( トリガ外部イベント )
```

「見積作成管理 .fmp12」ファイルに用意した、動画の再生について**「動画の状態 _ 見本」スクリプト**の動作を確認します。

このスクリプトは、シンプルな内容です。

```
If [ GetAVPlayerAttribute ( "playbackState" )=1 ]
    フィールド設定 [ z_ユーティリティ::g_動画再生タイムスタンプ ; Get(スクリプト
引数) ]
End If
```

iPhone の場合は次のとおりです（iOS デバイスが、手元にある場合のみ確認してください。手軽に確認するには、AirDrop などでデバイス上にファイルを保存してください）。

(1) FileMaker Go で演習ファイルを開きます。
(2)「商品管理」ボタンをタップします。
(3) 商品リストに切り替ります。
(4) 商品番号「30001」のレコードで絵画アイコンのボタンをタップすると、ポップオーバーが表示されます。

・「再生ボタン」
・黄色いグローバル格納フィールド
・オブジェクトタイプフィールド

が配置されています。

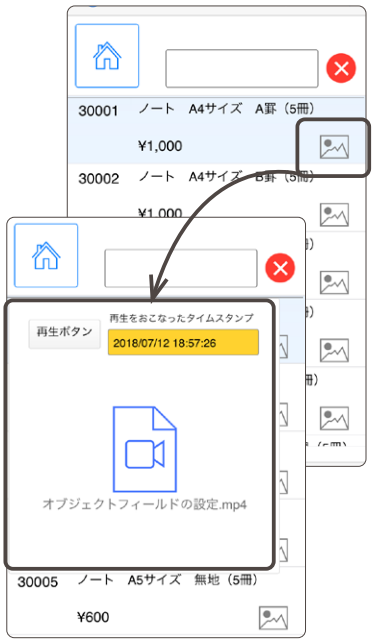


図 13-10 ポップオーバーボタン

(5) オブジェクトタイプフィールドをタップします。
(6) コントローラーが表示されますので確認してください。

オブジェクトタイプフィールドに動画などが入力されていてブラウズモードで入力が可になっていると、FileMaker Go では右図のようなコントローラーが表示されます（動画は商品番号「30001」に保存）。



図 13-11 動画のコントローラー

(7)「▷再生」をタップすると、動画が再生されます。
(8) 再生後のタイムスタンプが記録されます。

「▷再生」で動画を再生するとタイムスタンプが記録されます。再生のたびに更新されるはずです。

ポップオーバーの中のオブジェクトタイプフィールドに格納されているバイナリデータに再生などのイベントが起きると（OnObjectAVPlayerChange）、そのタイムスタンプが記憶されます。

また、再生と再生時の状態をあらかじめ設定しておくことも可能です。**「動画の再生 _ 見本」スクリプト**の動作を確認します。

このスクリプトもシンプルな内容です。

```
AVPlayer再生 [ フィールド： 商品管理::商品画像 ； 表示: フルスクリーンのみ ； インタラクションを無効にする： いいえ ； コントロールを隠す： いいえ ； 位置： 商品管理::商品画像 ]
```

「再生ボタン」をタップすると、右図のようにフルスクリーンで再生されます。動画以外の写真などもフルスクリーンで表示されます。

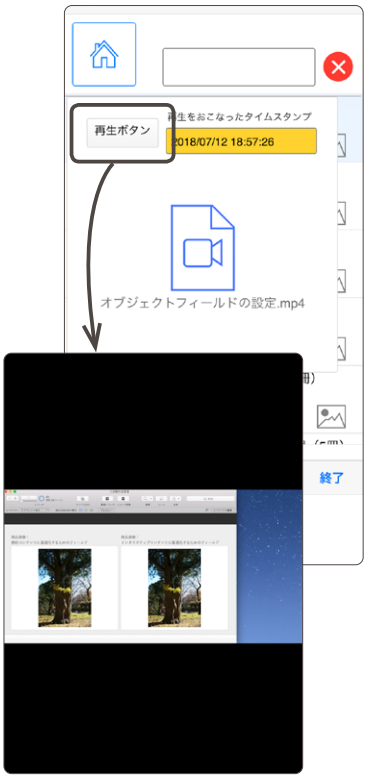


図 13-12 フルスクリーンで再生

13.3.4 センサー情報とローカル通知

iOS または iPadOS の機能を利用することもできます。

センサーの情報

iPad や iPhone には「センサー」が搭載されています。そして、様々な情報を検知、記録しています。これらの情報を取得する関数が用意されています。

```
GetSensor ( センサー名 {; オプション 1 ; オプション 2 } )
```

センサー名に、サポートされているセンサーの名前を指定します。例えば、

```
GetSensor ( "stepCount" ; 600 )
  →  600秒、つまり10分間の歩数（数値）を取得します。
```

取得できるセンサー名についてはヘルプを参照してください。
なお iOS または iPadOS の「設定」の「FileMaker Go <バージョン>」で、「位置情報」や「モーションとフィットネス」の設定の許可が必要な場合があります。

［OnTimer スクリプトをインストール］を使ったスクリプトと組み合わせて、定期的にセンサー情報を取得して利用すれば『今日の歩数（1 時間ごと）』などの記録が可能です。

ローカル通知

［OnTimer スクリプトをインストール］の場合、FileMaker Go の画面を表示していないといけません。しかし、「一定の時間を待つ」のであれば、その間、他のアプリケーションで作業したいこともあります。

FileMaker Go が、

・実行されていない
・バックグラウンドで実行されている

これらの状況でも、指定した時間の経過後に「通知」するスクリプトステップが用意されています。

［ローカル通知の構成］スクリプトステップ

iOS / iPadOS の「設定」の「FileMaker Go <バージョン>」で、「通知」を許可する必要があります。

「見積作成管理 .fmp12」ファイルに用意した通知についての**「ローカル通知 _20 秒後」スクリプト**の動作を確認します。

実行後、「20 秒後」に、「600 秒以内の歩数を記録するか確認」するものです。

iPhone の場合は次のとおりです（iOS デバイスが、手元にある場合のみ確認してください。手軽に確認するには、AirDrop などでデバイス上にファイルを保存してください）。

(1)　FileMaker Go で演習ファイルを開きます。
(2)　「ローカル通知」ボタンをタップします。
(3)　ホームボタンなどをタップして FileMaker Go をバックグラウンドにします。
(4)　（概ね 20 秒待ちます）
(5)　通知が表示されます（右図）。

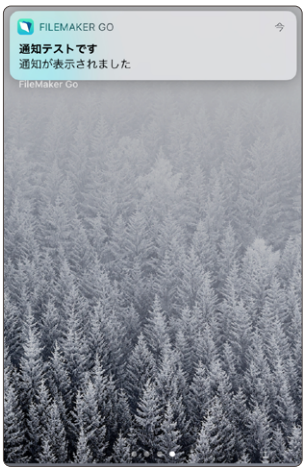


図 13-13 通知が表示される

(6)　通知部分を長押しすると、選択肢が表示されます。
(7)　「再通知」をタップします。
(8)　（概ね 20 秒待ちます）
(9)　通知が表示されます。

繰り返し通知できることがわかります。

(10) 通知部分を長押しして「記録する」をタップします。



図 13-14 長押しで表示されるボタン

(11) FileMaker Go に切り替わって、センサー記録用に作成した「デバイスの情報」レイアウトが表示されます。
(12) グレーに塗った部分「歩数」の結果のみが入力されています。



図 13-15 情報を記録した

「デバイスの情報」レイアウトで、他のボタンをタップするとセンサー情報が取得できます。これは、**「センサーからの情報」スクリプト**で動作しています。

FileMaker Go ならではの機能を一部分のみ紹介しました。

演習ファイルには、他にも「顧客」または「顧客管理」ボタンをタップして顧客がリスト表示されるレイアウトに、ポップオーバーボタンで、電話をかける、または、メールを作成するボタンも配置しました。モバイルデバイスならではの使い方を模索して開発を行ってください。

13.3.5 その他

FileMaker Go は、iPad や iPhone 上で、バックグラウンドに切り替わったり、ネットワークの状態によって突然接続が中断することがあります。

拡張アクセス権

FileMaker Cloud ／ FileMaker Server で共有されるカスタム App に、FileMaker Go のクライアントからの使用を想定する場合、ユーザのアクセス権に「fmreauthenticate[x] 拡張アクセス権」を設定します。

fmreauthenticate[x] 拡張アクセス権

iPad や iPhone で、FileMaker Go がバックグラウンドに切り替わると、開いているすべてのカスタム App ファイルの状態が保存されます。そして、手前に切り替わる時点で、「fmreauthenticate[x]」拡張アクセス権に指定された制限時間 [x] 分が経過していたら、再ログインが必要です。

例えば「fmreauthenticate[10]」では、ユーザは最大 10 分まで FileMaker Go の使用を中断しても、再ログインを求められません。また、FileMaker Go が手前に戻るたびにユーザにログインさせるには「fmreauthenticate[0]」を設定します。

レコードの確定

FileMaker は、レコードの確定を一定のデータのかたまりとして取り扱っています。確定前の状態で接続が中断したり強制終了するとこのかたまりごと破棄します。

FileMaker Go でユーザが実施する作業について、小まめにレコード確定されるように工夫したり、想定される場所のネットワークの状態によっては前述の同期を検討します。

演習ファイルを閉じてください。

第 14 章 運用を計画しよう

カスタム App 開発の初期段階から運用時のことを考えることは重要です。

14.1 運用準備

業務で使用するようなカスタム App には、セキュリティやパフォーマンスが考慮された開発作業を実施する「**開発担当者**」と、安定、安全、信頼がある運用のために管理作業を実施する「**運用管理者**」が必要です。

業務に関するシステムを、日々、使えるように動かし続けることを「**運用**」と呼びます。システムの運用には次の点が重要です。

・安定性
・安全性
・信頼性

ハードウェア、アプリケーション、ネットワークなど、システム全体を考慮します。上記の 3 点は、うまくいっている間はユーザが意識することはほとんどありません。

本章では、運用の計画と準備について考えます。なお、新しいカスタム App のための内容とします。

14.1.1 カスタム App の使われ方

システムの運用で重要なことは、

・カスタム App のファイルを物理的にどこに配置するか
・どのような方法でユーザが使用するか

を、決定することです。これらは、可能な限り企画、要件定義時に決定します。ポイントは、

・複数のユーザから使用されるかどうか
・カスタム App とクライアントは、どこから、どのように接続（アクセス）するか
・ユーザがカスタム App にアクセスするクライアントのハードウェアはなにか
・ユーザがカスタム App にアクセスするアプリケーションはなにか（FileMaker Pro、FileMaker Go、Web ブラウザのいずれか）

……などです。これらが曖昧なままだと設計や開発が進められません。

しかし、徐々にユーザの希望や企業や団体のルールが決まり、運用準備を開始する頃には決定しているはずです。

※本書では、Claris FileMaker Data API については取り扱いません。FileMaker Server のヘルプ、Data API ガイド、本書の上級編などを参照してください。

14.1.2 ユーザが単独で使用（スタンドアロン）

共有せず、クライアントのハードウェアにカスタム App のファイルを保存する方法です。例えば、

・複数のユーザから使用されるかどうか → 1 人のみ
・カスタム App とクライアント → 同ハードウェア上
・ユーザがカスタム App にアクセスするクライアントのハードウェア → iPad
・ユーザがカスタム App にアクセスするアプリケーションはなにか → FileMaker Go

と、なります。次のような場合に、この方法を選択します。

・ネットワーク環境がない場所での使用が必要
・データの共有が不要である

クライアントハードウェア、アプリケーション、カスタム App ファイルがあれば使用できます。ネットワークを利用してアクセスするよりも動作が早くなる期待ができます。なお、ユーザ数が増えるたび、そのユーザ用のクライアントハードウェアにアプリケーションをインストールし、カスタム App ファイルのコピーを保存して使用を開始させる必要があります。

注意点は、ユーザ 1 人ずつがファイル、データを管理しなければならないことです。そのため、なかなかまとまった運用になりません。不慮の事故でファイルが壊れてしまいバックアップをとっていなかったらデータが無くなってしまいます。また、後からデータの一元化をしたくなったとしても簡単にはできません。あらかじめ同期について考えておく必要があります（簡単に後述します）。また、新しい機能を追加することに備えて、新しくリリースされたファイルを差し替える手順（データ移行が必要かどうかなど）も考える必要があります。

スタンドアロンを想定する場面は、限られた用途のもの、例えば商品カタログ、あるいは使用期間が短い催しの受付用、調査のための情報収集ツールなどがあります。

14.1.3 ユーザがホストにアクセスして使用（クライアント／サーバー）

サーバーにカスタム App ファイルを保存して共有する方法です。FileMaker Cloud や FileMaker Server がホストして共有し、クライアントからのアクセスを可能にします。例えば、

・複数のユーザから使用されるかどうか → 2 人以上
・カスタム App とクライアントのアクセス → ネットワーク（有線 LAN または Wi-Fi）
・ユーザがカスタム App にアクセスするクライアントのデバイス
→ パソコン または iPhone や iPad
・ユーザがカスタム App にアクセスするアプリケーションはなにか
→ FileMaker Pro または FileMaker Go または Web ブラウザ

となります。主に、次のような場合です。

・ネットワーク環境がある
・データの共有が必要

ネットワーク環境があれは通常はこちらを選択します。

ホスト
カスタム App ファイルへ同時に 2 人以上のユーザがネットワークからアクセスして使用できるようにすることです。

クライアント
ホストにアクセス（接続）するのがクライアントです。

FileMaker Pro のホスト
FileMaker Pro もファイルをホストできますが、検証作業を主な目的としています。本番運用環境では使用できません。FileMaker Cloud や FileMaker Server を利用してください。

例えば、次の使われ方を想定する場合、

・複数のユーザから使用されるかどうか　→　10 人
・カスタム App とクライアントのアクセス　→　ネットワーク（有線 LAN または Wi-Fi）
・ユーザがカスタム App にアクセスするクライアント　→　パソコン
・ユーザがカスタム App にアクセスするアプリケーション　→　FileMaker Pro

クライアント／サーバーとして決定します。

カスタム App の使われ方は、どのような選択をするとしても運用の安定性、安全性、信頼性をできるかぎり確保できるように計画します。

14.1.4　同期

ユーザがネットワークの無い場所でも使用できるようにスタンドアロンのカスタム App の使い方を選択したけれど、使用後のデータはクライアント／サーバーでホストされているカスタム App に反映したいという場合があります。

必要なのは「**データの同期**」です。大きく分けて 2 種類があります。

一方向
モバイルデバイスのカスタム App から、サーバーでホストされているカスタム App へデータを送信するだけ、あるいはその逆という形です。

双方向
モバイルデバイスのカスタム App と、サーバーでホストされているカスタム App の両方でデータを送信するという形です。

前者は、データの集約を目的にすることが多く、後者は、データを集約し双方が同じ内容にすることを目的とします。

同期にはいくつかのポイントがあります。

同期する項目
同期するカスタム App 同士が同じファイルでデータが違うだけであれば、同期項目の合致が容易です。違う場合は、どの項目がどの項目へ同期されなければならないのか（マッピング）を考える必要があります。

競合
1 つのレコードが同期する両方のカスタム App に含まれていて両方が修正された場合「**競合**」と呼びどのように競合を解決するのかについてルールを決定します。「**競合の解決**」です。

主キーは UUID が便利
『あるレコードと他のレコードが同じ情報の内容違いであるかどうか』を照合するためにレコードの「主キー」を使います。このとき、UUID を使った主キーを採用していると照合がスムーズです。

例えば、同じ UUID のレコードで『どちらが新しいものか』について比較するには「修正タイムスタンプ」を比較すれば良い、と考えることができます。

削除
レコードの削除についてもルールが必要です。

一方向と双方向の混在
マスタ情報は、一方向（サーバーでホストされているカスタム App からモバイルデバイスへ）、イベント情報は、双方向というような 1 つのカスタム App の同期に 2 種類が混在することがあります。

同期のタイミング
常に行うのか、一定期間で行うのか、などです。

様々なことを考慮して機能を作成したり専用の製品を導入するなどして同期を実行します。同期の有無は、原則としてカスタム App 作成初期から決定しているべきです（「主キー」などの設計に関わります）。ただし、そのテスト、動作確認はカスタム App ファイルが完成してからでないと確実に行うことができません。

詳細は、ヘルプやクラリス・ジャパン株式会社のナレッジベース「データベースの同期 - アプローチの概要」（https://support.claris.com/s/answerview?language=ja&anum=000024001）を参照してください。

14.1.5 FileMaker Cloud または FileMaker Server によるホスト

クライアント／サーバー方式で FileMaker Cloud または FileMaker Server を利用する場合、共有方法がいくつかあります。アクセスできるクライアントのアプリケーションが決まっています。

・通常　→　FileMaker Pro（macOS または Windows）、FileMaker Go（iOS/iPadOS）
・FileMaker WebDirect（Web 公開）→　Web ブラウザ
・カスタム Web（Web 公開）　→　Web ブラウザ（FileMaker Cloud は非対応）
・ODBC ／ JDBC からのアクセス　→　ODBC ／ JDBC クライアント
（FileMaker Cloud は ODBC インポートのみ）

カスタム App の使われ方と一緒に決定します。設計、開発に影響があります。どの種類を前提にしているかにより FileMaker Cloud または FileMaker Server の設定内容を決定します。

他のカスタム App
すでに FileMaker Cloud または FileMaker Server を利用している場合、基本的な設定が既に済んでいるかもしれません。その場合、変更する必要があるかを検討します。共有されるすべてのカスタム App ファイルについて有用な設定になることを目指します。

複数の共有方法
1 つのカスタム App に複数のクライアントタイプからのアクセス方法を選択できます。マルチユーザと呼びます。1 つでもすべてでもどの組み合わせでも可能です。

FileMaker クライアント
FileMaker Pro、FileMaker Go、そして、FileMaker WebDirect への Web ブラウザ接続の 3 種類を「**FileMaker クライアント**」と呼びます。それぞれの特徴があります。

FileMaker Pro
パソコンのスペックの高さに比例して処理能力が期待できます。複数のウインドウ、豊富なフォントの表示が可能であること、また、ほぼすべてのスクリプトステップが利用できます。

FileMaker Go
iPhone や iPad は、画面サイズがパソコンよりも小さく、処理能力もパソコンよりも劣る場合があります。しかし、署名キャプチャ、バーコードや NFC（近距離無線通信）タグのスキャン、iBeacon、TouchID / FaceID の対応、センサー情報の取得など独自の機能を利用したカスタム App を提供することができます。また、画面サイズが小さいことを逆手にとり、必要最低限の項目などに集約された誰にでもわかりやすいレイアウトデザインと操作性を実現することができます。

FileMaker WebDirect
ほとんどのパソコンやデバイスには Web ブラウザがインストールされています。そのため、アプリケーションをインストールせずに使用を開始できます。ただし機能が限定されます。

なお、それぞれ最適なレイアウトや画面遷移を用意することをお勧めします。

その他のアクセス
FileMaker Server を利用する場合は、カスタム Web 公開と ODBC ／ JDBC クライアントからのアクセスも可能です。

ファイルへの設定が必要
カスタム App の使われ方の計画にあわせてホストするファイルにアクセスできるように設定します。ただし、実際に、

・なんというアカウントがどのクライアントでファイルへアクセスできるのか

は、ファイルに登録されるアカウントに割り当てられるアクセス権セットの拡張アクセス権で決定します。

14.1.6 ハードウェアとアプリケーション

別の面から運用の準備をします。ハードウェアとアプリケーションです。FileMaker Cloud のコンピュート（処理能力）タイプ、FileMaker Server のホスト専用機有無や、接続する FileMaker クライアントの種類などは、カスタム App の使われ方により、何を準備する必要があるかわかります。

これらをいつ準備するのかを計画しなければなりません。次の流れにどのぐらいの期間が必要なのかを前もって見積もる必要があります。

（1）ハードウェアとアプリケーションを購入する
（2）ハードウェアをセットアップする
（3）ハードウェアにアプリケーションをインストールする
（4）アプリケーションを設定する

特に（1）と（2）は、所属している企業や団体により、購入やセットアップをおこなうのが専用の部門、部署である場合には、開発・運用担当者だけでは進められません（システムに必要なネットワーク環境は整備されている前提です）。

購入
必要な品について決定します。

・システムを運用するために必要なハードウェアの台数と OS、スペック（仕様や性能）
・システムに必要なアプリケーションの名称（種類）、バージョンと本数

購入を専用の部門や部署が担当する場合は、必要なことを正確に伝えます。この内容はカスタム App の使われ方が決定した時点から検討を開始します。そして運用開始までの間にこれらに変更がないかを考えます。

ハードウェアのスペックに最低限必要な内容は、インストールするアプリケーションと関係します。アプリケーションが動作するために必要な推奨環境が指定されています。既存のものを利用する場合『いまあるもので条件を満たしているかどうか』を確認します。推奨環境については、クラリス・ジャパン株式会社の Web サイト（https://www.claris.com/ja/）を参照してください。

必要なセットアップ内容
ハードウェアとアプリケーションの入手について決定したらセットアップについて検討します。

・システムを運用するために必要なハードウェアへのセットアップ内容

です。企業や団体で利用されるハードウェアには、

・ネットワークの設定
・OS のアカウントの設定
・OS のセキュリティの設定（ファイアウォールなど）

などのセットアップが必要になります。セットアップを専用の部門や部署が担当する場合は、必要なことを正確に伝えます。次はセットアップ時に必要な内容の 1 つです。

必要なポート番号の許可
ポート番号の決定は、カスタム App の使われ方により違います。それ以外にもカスタム App によって必要な設定を決定します。

ポート番号については、FileMaker Pro のヘルプやクラリス・ジャパン株式会社の Web サイトのナレッジベース「FileMaker Server によって使用されるポート番号」（https://support.claris.com/s/answerview?anum=000025632&language=ja）で、最新情報を参照してください。

インストールと設定
次にアプリケーションのインストールと設定を誰が実施するかを計画します。

これらの作業、つまりどれぐらいの数のハードウェア、アプリケーションを発注し、納入後にセットアップ、インストールと設定を行うのか、その作業時間を予測して期間を計画し、実行します。

きちんと計画して運用準備作業を行うことで、運用の安定性、安全性、信頼性を高めます。

14.2 データの準備

ハードウェア、アプリケーション、ネットワークの他に、リリースされるカスタム App にも準備が必要です。主にセキュリティとデータの準備です。

14.2.1 ユーザ

カスタム App にアカウントとパスワードでログインするユーザを登録します。

14.2.2 マスタ情報

各種マスタ情報を登録します。例えば「見積作成管理システム」なら次のようなデータです。

・顧客情報
・顧客担当者情報
・商品情報
・自社情報
・自社担当者情報

これらの情報がないとイベント情報（例えば見積情報）が作成することができません。マスタ情報が登録されていない場合、ユーザは新しいシステムの使用開始当初に次のような手順が必要になります。

(1) 見積情報を登録作業を開始する
(2) 顧客の登録が無いので、顧客を登録する
(3) 改めて見積情報を登録する。顧客を選択して必要な情報を入力する
(4) 明細行で商品を選ぼうとすると、登録が無いので商品を登録する
(5) 改めて明細行で商品の選択して必要な情報を入力する

本来は、

(1) 見積情報を登録作業を開始する
(2) 顧客を選択して必要な情報を入力する
(3) 商品の選択をして必要な情報を入力する

この手順のはずです。

ユーザは『前より大変になった』と感じるかもしれません。実際には、よく使うマスタの情報が一通り登録されれば通常の手順になるのですが、このやり方では、それはいつになるかはわかりません。つまり、新しいシステムの利用開始前に、主なマスタ情報は登録済みにしておくほうがユーザがストレスを感じません。この情報の登録作業を「データ移行」と呼びます。

値一覧
カスタム値の値一覧の内容も運用前に見直します。テスト用の文言などが含まれているかもしれません。

14.2.3 データ移行

もしも既にマスタ情報（顧客情報や商品情報など）がスプレッドシートファイル、他のシステムに存在しているのであれば、それを利用してデータの「移行」を実施します。大まかに 2 種類があります。

手入力
まとまった情報がどこにもなく、ばらばらで紙に書いたものしかないなどの場合、手入力で情報を登録します。手入力であっても、運用開始前にまとめて作業した方が、五月雨式にユーザの登録に任せるよりもお勧めです。

インポート
いくつかに分散していても、どこかにまとまっていても入力されているデータがある場合は、専用のインポート機能をカスタム App に作成してデータ移行を実施します。概ね、以下の流れになります。

(1) データを集める

(2) 集めてきたデータの項目を揃える

(3) データの項目が、カスタム App の項目が合致するか確認する
例えば、郵便番号が 123-4567 と 1 項目になっているか、123 と 4567 の 2 項目かなど

(4) 合致させたデータ項目の内容とカスタム App で想定している内容に合致するか確認する
例えば、住所の数字の全角半角など

(5) これまでの作業で分かった内容を元に、必要があれば 1 箇所にデータを集めて加工する（加工作業用のカスタム App を作ったりスプレッドシートを使ったりする）

(6) 複数の種類のデータをインポートする場合の順番を考える
例えば、顧客情報をインポートしたあと振られた主キーを顧客担当者情報に外部キーとして入力するなど

ここまでの作業で、カスタム App に必要なインポートと、インポート後の処理がわかります。

(7) データ移行機能を作成する

すぐに実行してはいけません。この移行は失敗すると大変です。手順を作成することで何度でもデータ移行作業を実施できるようにすることはとても重要です。

(8) データ移行の具体的で詳細な手順を考え、手順書を作成する

(9) テスト用のファイルを準備してリハーサルを行う

リハーサルがうまくいったら、

(10) データ移行を実行する

この作業はインポートするデータの項目のばらつきやデータ数が多いほど、作業時間がかかります。また、慎重に実施する必要があります。充分な作業期間を確保します。

データの運用

マスタ情報の運用方法にはポイントがあります。他のシステムの情報で置き換える必要があるか、です。テーブルやフィールド項目に影響があるからです。

例えば、企業や団体の基幹システムに顧客情報や商品情報があってカスタム App に必要な場合、両方に同じようなマスタ情報を持って更新するよりも連携した方が効率的です。

・直接他のシステムの情報を利用する
・定期的なインポートで最新情報に置き換える

のどちらが良いかを考えます。FileMaker Pro で作成したカスタム App 同士であれば簡単に外部データソースとして複数のファイルで参照することが可能です。あるいは、ODBC ドライバ経由、FileMaker Data API などが可能なデータベースであればデータ連携が可能です。そうでなければ、

・運用開始時の最新情報をデータ移行で入力し、その後はカスタム App で独自に運用する

ということになります。

イベント情報のデータ移行

イベント情報（例えば見積情報）のデータ移行は、マスタ情報よりも作業期間がかかります。ほとんどの場合、新しいカスタム App と移行したいと考えるデータの構造が違うからです。

それ以外の面では、データ移行の対象となるイベント情報は、過去の情報であることを考慮しなければなりません。例えば見積結果をすべての過去の見積情報に正しく付与しなければなりません。マスタ情報の移行にも影響が出ます。現在は取引していない顧客や販売していない商品の情報を過去の見積情報のために登録しなければなりません。

イベント情報のデータ移行について、本当に必要かどうか、検討してください。

すべての過去のイベント情報だけではなく、例えば今年度のもの、運用開始当月のもの、あるいは運用開始週のものだけを集計やグラフのために移行するということもあります。その場合、件数が少なく、マスタ情報へ過去の情報を登録する頻度が低いため比較的作業しやすいかもしれません。

また、イベント情報は、マスタ情報の登録後にデータ移行作業を実行する必要があります。そのため、さらに充分な作業期間を確保するようにします。

14.3 管理の準備

管理運用の準備も必要です。

14.3.1 管理者を決定する

カスタム App を運用していくには、開発担当と運用担当の両方が必要です。ハードウェアとネットワークの管理には、専門の知識が必要となる場合があります。所属している企業や団体に専門の部門や部署がある場合には任せる方が良いでしょう。

開発担当

主に、カスタム App 自身について作業します。運用開始後に発見されたバグを解決したり、ユーザの意見を検討し、定期的、あるいは任意のタイミングで修正や機能の改善や追加をします。

運用管理担当

運用について作業します。バックアップが定期的に実行されているか、不具合が起きていないかを確認し、問題があった場合に対処したり、あるいは専門の部門や部署へ連携します。

2 種類の管理を 1 人で実施することもありますが、規模によっては全体の作業完了まで時間がかかるかもしれません。複数の人で手分けする方が早くて安全です。

セキュリティ

管理者についてもセキュリティを考えます。開発や運用管理が複数の人で行われる場合、［完全アクセス］アクセス権セットのアカウントは共有せず、1 人に 1 つ設定します。

14.3.2 運用の 1 日を考える

管理者が決定したら、運用開始後の予定も立てる必要があります。1 日の間になにが起きるかというと、主に以下のようなことが考えられます。

・カスタム App をユーザが使用できるように継続してホスト
・定期的なバックアップ
・定期的なバッチ処理

バックアップとバッチ処理が同時に実行されないように考える必要があります（バッチ処理については後述します）。もう少し長い期間で考えると、1 週間、1 ヶ月、1 年の間には、

・メンテナンスのためにカスタム App のホストを一時停止する

ということが定期的に必要かもしれません。または、緊急時の対応についても考えておかなければなりません。つまり、

・通常の、運用の 1 日の中で行うこと
・あらかじめ計画して運用を停止／再開すること
・計画になく運用が中断すること

この 3 点について準備します。

次のことを行うことをお勧めします。

・FileMaker Cloud または FileMaker Server で設定した内容を記録する
・設定や作業の手順書を作成する

運用管理は開発とは違う側面を考えることが多くなります。運用を全く初めて行う場合は、戸惑うことも起きるかもしれません。経験値を蓄積していきましょう。

14.3.3 運用開始までの期間を考える

ハードウェア、アプリケーション、カスタム App へのデータ移行、これらの作業期間を考えることで、運用開始までの計画を立てることができます。運用までの準備のうち、

(1) ハードウェアとアプリケーションを購入する
(2) ハードウェアをセットアップする
(3) ハードウェアにアプリケーションをインストールする
(4) アプリケーションを設定する
(5) カスタム App にマスタ値をデータ移行する
(6) 運用の予定と対応を準備する

ここまでを解説しました。これ以外にも

(7) カスタム App を共有する
(8) カスタム App への接続テスト

が必要です。そして運用開始後のことを考えます。

・運用のルールを考える

これらとともに、

・運用開始日を決定する

となります。ユーザに対してカスタム App の使用方法について説明することも必要かもしれません。運用開始までにこれだけの準備と作業が必要なことがあらかじめわかっていれば、それなりの期間を考えておくことが可能です。

運用のルール
カスタム App をユーザが使っていく上で、ユーザがどのようなルールで管理をするかです。開発者が担当する、あるいは、別途に担当者を設定することがあります。例えば、2 点についての管理ルールです。

・マスタ値の管理
・アカウントの管理

14.4　複数のファイル

カスタム App の設計で、ファイルの構成を考えることが必要になる場合があります。

14.4.1　データ分離モデル

複数のファイルで作成されたカスタム App は、1 つのファイルにすべて（データ、テーブル、関連、スクリプト、レイアウト、アクセス権限など）が入っているために取り扱いが簡単です。

これをあえて複数のファイルに分けて 1 つのカスタム App とする場合があります。これは設計時に決断します。開発作業が完了した後からファイルを分けることは作業をやり直すに等しい労力が必要です。

複数のファイルを外部データソースで参照することで、1 つのカスタム App として取り扱います。大まかに言うと、次のように分離します。

・レイアウトとカスタム App の論理（スクリプト）　→　インターフェースファイル
・データファイル

インターフェースファイル
ユーザに表示されるもの、ユーザが操作する機能を集めます。外部データソースのテーブルの TO をリレーションシップグラフに配置します。

データファイル
データ、つまりテーブルを集めます。フィールド定義に必要な TO と関連をリレーションシップグラフに配置します。

分離すると良いところ
運用開始後にリリースする必要がある場合、インターフェースファイルの差し替えだけで済む場合があります。リリース作業が短時間で済むために、運用停止期間も短くなります。

普通のリリース作業
1 つのファイルで運用している場合、リリース時には、現在のデータを新しくリリースされるファイルに移し替える必要があります。この作業は、ユーザ使用中はできません。計画的に作業をします。運用停止する日付あるいは時間帯を設定し、リリース作業を実施します。

テーブルやフィールドの変更をおこなった場合のリリース作業にはデータの移し替えが必要です。
※本書では Data Migration Tool については取り扱いません。詳細はクラリス・ジャパン株式会社の Web サイト（https://www.claris.com/ja/）を参照してください。

分離で注意が必要なところ
開発作業が複雑になる部分があります。主に、インターフェースファイルを中心としてファイル構成を考えます。そしてインターフェースファイルのスクリプトのための TOG とデータファイルのフィールド定義で指定する計算式の TOG は、それぞれのファイルに作り分けられ別のものになります。

インターフェースファイルの起動、終了時に外部データソースのファイルがすべて開く／閉じるかを確認する必要があります。また開発中にフィールド定義について設定したくなっても、データファイルへ移動してからでないと［データベースの管理］ダイアログを開くことはできません。グローバル変数はファイルごとであることを意識する、機能の内容によっては、データファイル上にスクリプトを作成しインターフェースファイルから呼び出す必要があるかもしれません。カスタム値の値一覧についても注意をしなければなりません。

カスタムメニュー、カスタム関数、アカウントとアクセス権セットなどファイルごとに必要なものがあります。

特にアカウントとアクセス権セットはインターフェースファイルとデータファイルの両方に登録されていないといけません。アカウント管理の作業は常に両方のファイルにおこなわなければいけないのです。ユーザによるパスワードの変更はスクリプトステップで機能として作成しなければ両方のファイルに反映できません。そしてインターフェースファイルを差し替えるときに、アカウントを一旦初期設定パスワードでリセットするなどの作業が必要です。

データ分離モデルを採用する場合には、通常よりも開発期間を長めに設定するなどの充分な対策を考えてください。

データファイルをさらに分割
例えば、顧客マスタが 100 万件あるなど、マスタ情報のデータが大容量である場合、データファイルを分けてバックアップの頻度を下げるという選択肢もあります。これはイベント情報のデータよりも、マスタ情報のデータの方が静的である、つまり新規作成、変更、削除作業が少ないという考え方によるものです。

インターフェースファイルだけユーザ側に
共有されているカスタム App を FileMaker Go から使用する場合、インターフェースファイルをモバイルデバイスのローカルに保存するという方法があります。サーバーとの通信がほとんどデータ内容だけになるため処理が早くなる期待ができます。ただしモバイルデバイスはパソコンと比べてスペックが低いため、シンプルな画面、処理を心がけるようにします。

14.5　複数のカスタム App

同じ企業や団体で使用する複数のカスタム App がある場合、共通のマスタ情報を使用する方が開発面、管理面ともに効率が高いかもしれません。また、次のようなこともあります。

・1 人が複数のカスタム App のユーザである

共通のアカウントで使用できるようにするかどうかを検討する必要があります。カスタム App ごとにアカウントが異なるとユーザが戸惑います。

・アカウントが同じ内容でカスタム App ごとにアクセス権セットが違う

という運用が望ましくなります。このような場合、外部認証サーバーを導入できることがベストな運用です。外部認証が利用できない場合、アカウント管理用のカスタム App を作成するなどを検討しても良いかもしれません。

第 15 章 セキュリティ

セキュリティの目的はシステム全体の保護です。想定しない外部からのアクセス、内部からのユーザによる誤操作、悪意のある行為（盗用など）などからデータやシステムを守ります。

15.1 セキュリティの対象

本書では、カスタム App のセキュリティについて解説します。大きくわけて 2 種類です。

・ファイルに対して行う設定
・FileMaker Cloud や FileMaker Server で行う設定

他にも、次のようなことも考えることが必要です。

ネットワークやハードウェア
ネットワークとハードウェア、OS への外部からの進入については、

・ネットワーク（外部からのアクセスの経路を防ぐ）
・ハードウェアや OS（外部、あるいは想定していない内部からのシステムへの進入を防ぐ）

などの対策が必要です。これらは専門の知識が必要になります。ネットワーク、ハードウェア、OS のセキュリティ対策について、所属している企業や団体にシステム部門や部署がある場合はそのルールに従い、協力してセキュリティの設定を行います。

本書では、ファイルに対して行う設定について記述します。

15.2 アカウント

データ保護の目的の 1 つは、ファイルにアクセスする権利を持っているユーザの操作によってデータの整合性が損なわれることを防ぐことです。

15.2.1 ユーザを決定

カスタム App を使用できるユーザを決定します。ユーザを決めることはセキュリティ設定の第一歩です。ユーザのセキュリティ設定はファイルごとに行います。

・アカウント名とパスワードによりログインするための「**アカウント**」
・アカウントの権限（できること／できないこと）を設置する「**アクセス権セット**」

アカウントでログインし、成功するとファイルへの接続が許可されてアクセス権セットで表示や操作が制限されます。アカウントとアクセス権セットの管理には［完全アクセス］アクセス権セットが割り当てられたアカウントか、［完全アクセスのないアカウントを管理］のチェックがオンのアクセス権セットが付加されたアカウントが必要です。

15.2.2 デフォルトアカウント

FileMaker Pro で新しいファイルを作成すると次のアカウントが作成されます。「デフォルトアカウント」です。

・アカウント名は　　Admin
・パスワードは　　空欄
・アクセス権セットは　[完全アクセス]

ファイルオプションで、このアカウントによる自動ログインが設定されています。[完全アクセス] アクセス権セットのアカウントは、データ、レイアウト、スクリプト、各種設定のすべてを行うことができる、開発に必須の「**権限**」です。

開発者のセキュリティ
カスタム App の運用開始時までに、Admin アカウントの自動ログインは解除するべきです。その上で、Admin アカウントに推測されづらいパスワードを設定したり、別のアカウント名の [完全アクセス] アクセス権セットのアカウントを登録して利用します。

・開発作業中は、必要時以外は開発担当者のみファイルへアクセスする
・運用開始後は、ユーザが与えられたアカウントとアクセス権セット以外でのアクセスをできないようにする

このことを守ります。

「Admin」アカウント
「Admin」というアカウント名は推測しやすいので、他の [完全アクセス] アクセス権セットのアカウントを作成後、「Admin」アカウントのアクティブを　**オフ**　にするか削除するか検討してください（すべての [完全アクセス] アクセス権セットのアカウントを削除してしまわないように注意します）。

[ゲスト] アカウント
新しいファイルには [ゲスト] アカウントも登録されています。このアカウントは特別です。削除できません。「アクティブ」にするかどうか（有効／無効）の設定が可能です。デフォルトではチェック　**オフ**（無効）　になっています。

[ゲスト] アカウントが有効になっている場合、ユーザはアカウントがなくてもファイルにアクセスできます。つまり、誰でもアクセスを可能とする場合にこのアカウントを利用します。例えば、アンケートや閲覧のみの画面や機能を提示する FileMaker WebDirect へのアクセスなどです。

ほとんどの場合、セキュリティを考慮するとデフォルトアカウントだけではカスタム App は運用できません。ユーザと開発者に必要なアカウントとアクセス権セットを作成して運用します。

演習ファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 15_1 から 3」フォルダの中の「15_ 見積作成管理 .fmp12」ファイルをベースに進めます）

「見積作成管理 .fmp12」に、次のアカウント（とアクセス権セット）を登録しておきました。

アカウント		アクセス権セット
アカウント名	パスワード	
Admin	1111	完全アクセス
ユーザテスト	2222	営業ユーザ
管理テスト	3333	管理ユーザ

15.2.3　自動ログインの解除

新しいファイルでは、ファイルオプションで自動ログインが設定されています。そのため、ファイルを開いたとき、ログインのためのダイアログが表示されていません。

自動ログインに設定されている以外のアカウントでのログイン

［次のアカウントを使用してログイン］のチェックが　**オン**　の場合に、設定されているアカウント以外でログインしたい場合は、次のように操作します。

・［Option］キー（macOS）を押しながらファイルを開く
・［Shift］キー（Windows）を押しながらファイルを開く

資格情報の保存

［ファイルオプション］ダイアログの［保存されている資格情報による認証を許可］を選択すると、macOS、iOS、iPadOS のキーチェーンまたは Windows の資格情報マネージャにパスワードが保存され、ユーザはファイルを開くたびに資格情報を入力する必要がなくなります。FileMaker Cloud によってカスタム App を共有している場合は、最初に認証された後、Claris ID または外部 IdP の資格情報が保存されます。

ただし、この設定には、十分な検討が必要です。



図 15-1［ファイルオプション］ダイアログ

演習ファイルの自動ログインの解除を「ファイルオプション」で行います。

「演習 15_1 から 3」フォルダ

演習 15-1　自動ログインを解除

- **使用ファイル：15_ 見積作成管理 .fmp12**

(1)　[ファイル]–[ファイルオプション ...] を選択し [ファイルオプション] ダイアログを開きます。

(2)　[次のアカウントを使用してログイン] のチェックを　**オフ**　にします。

(3)　ファイルを閉じます。

(4)　ファイルを開いて「Admin」アカウントでログインしてください。
・アカウント名　Admin
・パスワード　　1111

(5)　自動ログインが解除できました。

演習ここまで

15.2.4　アクセス権セット

ユーザのセキュリティは、アクセス権セットでの権限設定が基本です。次のように運用します。

・1 つの役割に 1 アクセス権セットを作成して運用する

担当や部門、部署での作業の違いを管理します。1 アクセス権セットには複数のアカウントを結びつけることが可能です。新しいファイルにはデフォルトのアクセス権セットが 3 つが登録されます。

・[完全アクセス]
・[データ入力のみ]
・[閲覧のみアクセス]

この「デフォルトのアクセス権セット」は「拡張アクセス権」の設定以外の変更はできません。



図 15-2 デフォルトのアクセス権セット（囲み部分）

完全アクセス	・すべての開発機能を含めたファイルへのすべてのアクセス ・複製できない ・ファイルにこのアクセス権セットが付加されているアクティブなアカウントが 1 つ以上必要
データ入力のみ	・レコードの作成、削除、変更、インポート、エクスポートが可能 ・レイアウトモードをはじめとした、開発機能へはアクセスできない
閲覧のみアクセス	・レイアウトとレコードの表示、エクスポートが可能 ・レイアウトモードをはじめとした、開発機能へはアクセスできない

デフォルトのアクセス権セットだけではカスタム App は運用できない場合は、必要なアクセス権セットを作成して運用します。次の内容を検討します。

機密にする情報を考える
どのデータの何をユーザに許可するかです。人に関する情報を取り扱うカスタム App であれば、従業員の給与や、患者の病歴などは必要な場合のみ表示するべきです。

・表示／編集
・削除

非表示にするのか、あるいは表示するのか、表示だけか、編集もできるようにするか、ユーザが判断して削除できるようにするか、などを考慮します。

整合性を考える
どのデータの何をユーザが実行できるかです。

・作成
・削除
・編集（関連に関することなど）

設計時に想定したデータの関係と一緒に考慮します。例えば、ユーザが共通に利用する商品マスタへ、誰でもデータを追加できるようにするか、データの関連の変更や削除をできるようにするか、などです。

整合性
データを想定している通りに正確で「一貫性」があるように管理することです。

フィールドオプションやスクリプトステップ、カスタムメニューやアクセス権セットなども、一定の制限をかけることができます。これらの設定を行わないと、
・フィールドに想定されていないデータが入力される
・スクリプトを［Esc］や［Enter］キーで作業をキャンセルしたり想定しない実行がなされる
・ショートカットキーでレコードの削除や複製などを実行する
・ステータスツールバーで想定していない操作を実行する
・ポータルで、作成や削除を行った結果、関連が崩れる
・計算式が変更される、データビューアを経由して、変数やフィールド内容の変更を操作する
などが起こり得ます。

例えば、フィールドオプションの入力値の制限で日付は西暦 4 桁と設定することで『西暦 4 桁で入力されている』という一貫性を持たせます。

スクリプトと完全アクセス
スクリプトに［完全アクセス権を付与］して、ユーザに割り当てられたアクセス権セットではできない処理を実行させることが可能です。

例えば、アクセス権セットでレコードの削除が許可されていないアカウントでも、スクリプトでレコード削除が可能になります。

一方、［完全アクセス権を付与］が設定されたスクリプトの取り扱いには注意が必要です。実行される作業を把握し、しっかりテストします。なお、設定を行ったスクリプトのみ有効です。サブスクリプトを実行しても引き継がれません。

この設定は「［完全アクセス］アクセス権セットのアカウントのみが可能です（第 16 章で演習を行います）。

アクセス権セットによるログイン後の違い
アクセス権セットを使い分けることで、ログイン後の作業の可否を切り分けることが可能です。現在、演習ファイルには、2 種類のアクセス権セットに、1 つずつのアカウントが作成済みです。

・「ユーザテスト」　→　営業ユーザアクセス権セット
・「管理テスト」　→　管理ユーザアクセス権セット

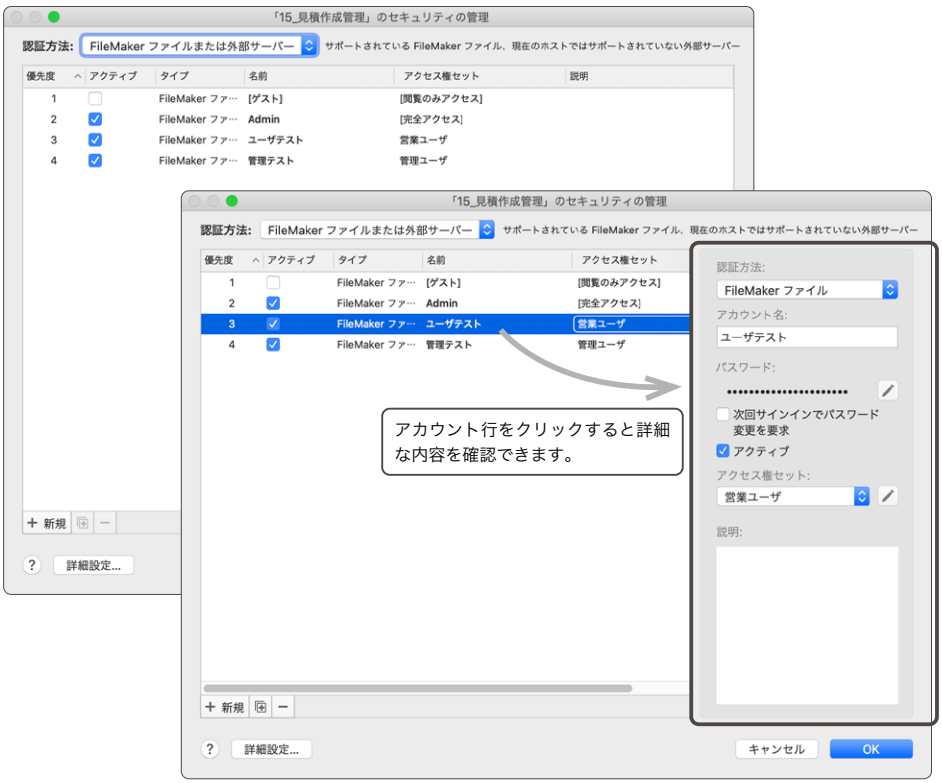


図 15-3 作成したアクセス権セット

どちらも開発に関する作業（構造、スクリプト、レイアウトの変更など）はできません。しかし、「管理ユーザ」アクセス権セットは、「値一覧」の管理が可能になっています。

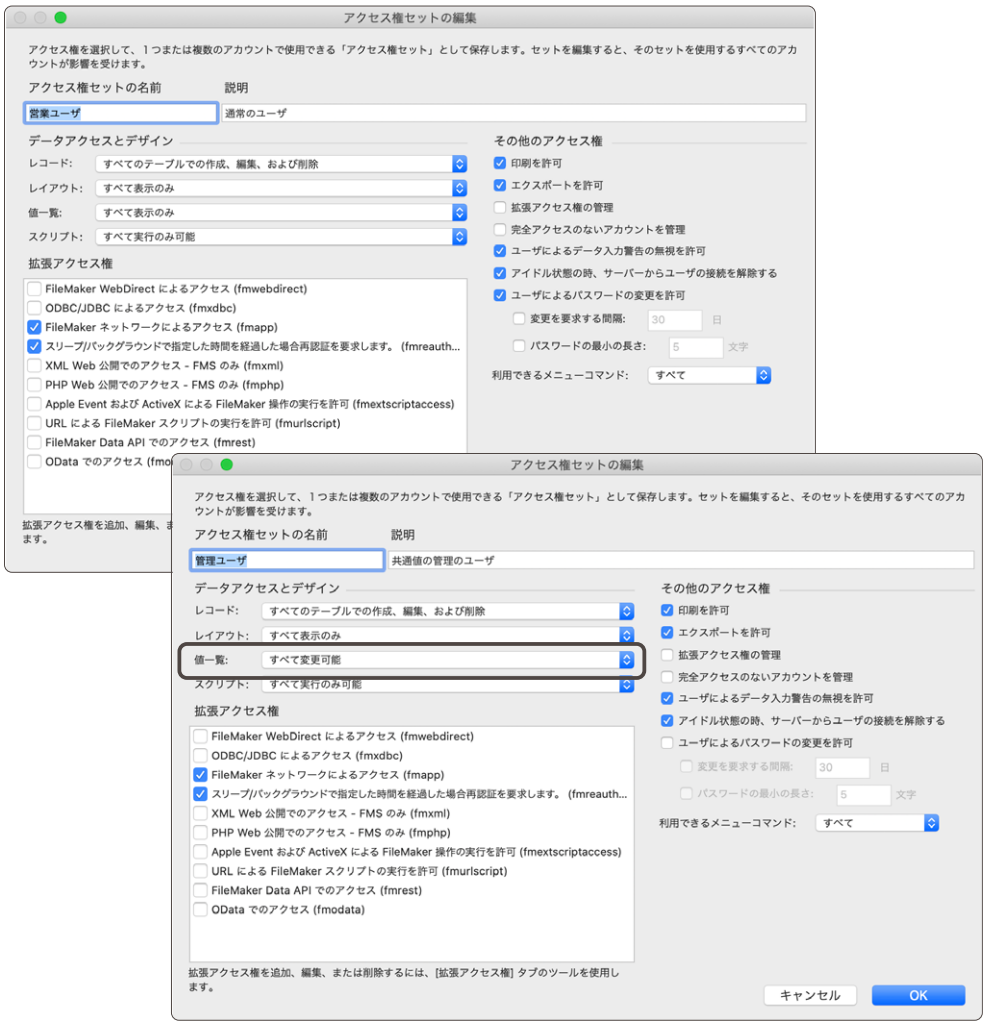


図 15-4 各アクセス権セットの内容

アクセス権セットは、権限として設定するだけではなく、アクセス権セット名を取得して同じ権限のアカウントに制限をかけることが可能です。次の関数を利用します。

```
Get ( アカウントアクセス権セット名 )
```

「ホーム」レイアウトの「自社情報管理」ボタンに [次の場合にオブジェクトを隠す] を設定しました。

```
Get ( アカウントアクセス権セット名 )  =  " 営業ユーザ "
```

「管理ユーザ」アクセス権セットは、自社情報管理の作業は不可なので、ボタンを非表示にしています。なお、○○さんは表示する、□□さんは表示しない、のような、個別のアカウントごとに指定すると管理が複雑になるため、特別な場合を除いて行わないことをお勧めします。

「ユーザ用メニュー」という名称の「**カスタムメニュー**」(第 16 章を参照) も作成してあります。ファイルが開かれた時にログイン中のユーザのアクセス権セットにより違うカスタムメニューをインストールして使ってもらうためです。

メニューのインストールのために、**「起動処理 _ 演習」スクリプト**の内容を書き換えました。次の内容です。

```
レイアウト切り替え [ 「起動画面」 (z_ユーティリティ ) ]
If [ PatternCount ( Get(アプリケーションバージョン) ; "Go" ) ]

  If [ Get(デバイス) = 3 ]
    レイアウト切り替え [ 「ホーム | iPad」 (z_ユーティリティ ) ]
  Else If [ Get(デバイス) = 4 ]
    レイアウト切り替え [ 「ホーム | iPhone」 (z_ユーティリティ ) ]
  End If

Else

  レイアウト切り替え [ 「ホーム」 (z_ユーティリティ ) ]

End If
```

↓ここから下が追加した部分です。

```
If [ 開発者かどうか = 1 ]

  ツールバーの表示切り替え [ 表示する ]
  メニューセットのインストール [ 「[FileMaker の標準メニュー ]」 ] [ ファイルデフォルトとして
  使用 : オン ]

Else

   ツールバーの表示切り替え [ 隠す ]
   メニューセットのインストール [ 「ユーザ用メニュー」 ] [ ファイルデフォルトとして使用 : オン ]

End If

ウインドウの調整 [ 収まるようにサイズ変更 ]
```

また、動作確認のための**「再ログイン」スクリプト**も作成しました。

```
エラー処理 [ オン ]
再ログイン [ ダイアログあり : オン ]

If [ Get ( 最終エラー ) ]

  カスタムダイアログを表示 [ "エラー" ; "再ログインできません" ]

Else

  スクリプト実行 [ 指定 : 一覧から ; 「起動処理 _ 演習」 ; 引数 : ]
End If
```

15.2.5 アクセス権セットの設定

アクセス権セットは、次のように設定します。

手順：

(1) [ファイル]–[管理]–[セキュリティ ...] メニューを選択
(2) [セキュリティの管理] ダイアログが表示される
(3) [詳細設定 ...] をクリック
(4) 新規作成、編集、複製、削除などを実行する

[新規 ...] をクリックすると、[アクセス権セットの編集] ダイアログが表示されます。

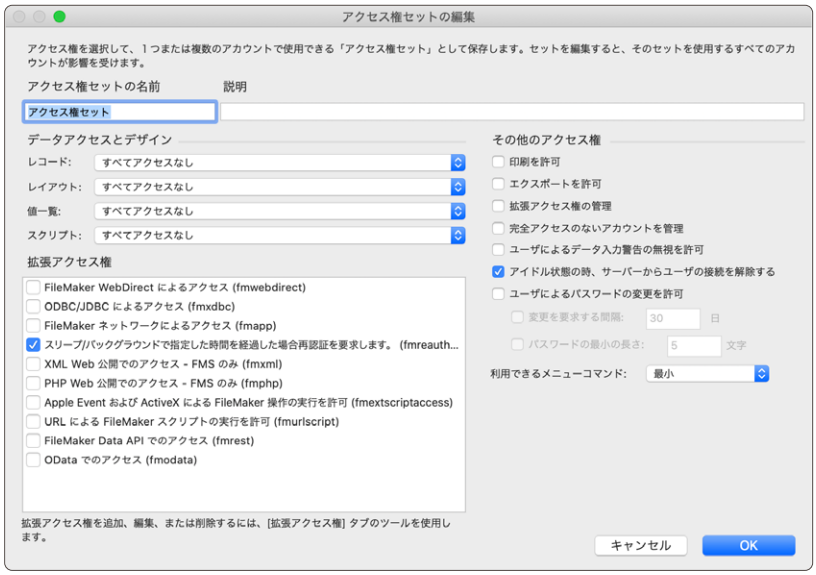


図 15-5 アクセス権セットの新規作成時

[アクセス権セットの名前]
わかりやすい名前を入力します。この名前が、Get (アカウントアクセス権セット名) の結果になります。

[データアクセスとデザイン]
レコード、レイアウト、値一覧、スクリプトについての権限を設定します。

[レコード:] は、レコードの表示、編集、作成、削除をファイル全体に対して設定するか、個々のテーブル、あるいはフィールドについて設定します。

[レイアウト] は、レイアウトの表示、作成、変更、削除をファイル全体に対して設定するか、個々のレイアウトについて設定します。レイアウトへの制限を設定する場合、注意が必要です。ユーザへ表示されるレイアウトに制限するとスクリプト利用しているレイアウトが表示できなくなってエラーが起きるかもしれません。注意してください。

[値一覧] は、アクセスなし、表示のみ、変更可能をファイル全体に対して設定するか、個々の値一覧について設定します。

［スクリプト］は、アクセスなし、実行のみ、変更可能をファイル全体に対して設定するか、個々のスクリプトについて設定します。アクセスなしとするとスクリプトが実行できなくなります。

［拡張アクセス権］
ファイルへのクライアントからのアクセスについての設定をおこないます。

［その他のアクセス権］
印刷、エクスポートなどいくつかの FileMaker Pro の機能について設定を行います。情報の出力を許可することなので　**オン**　設定には注意します。

［拡張アクセス権の管理］を　**オン**　にするとそのファイルの拡張アクセス権を設定することが可能になります。初期値は　**オフ**　です。

［完全アクセスのないアカウントを管理］は、そのファイルに設定されているアクセス権セットの内容を確認したり、アカウントへのアクセス権セットの付与と変更が可能になります。［完全アクセスのないアカウントを管理］を　**オン**　にして、［利用できるメニューコマンド :］を「すべて」に設定します。そして［ファイル］-［管理］-［セキュリティ］メニューが選択できることを確認します。初期値は　**オフ**　です。

［ユーザによるデータ入力警告の無視を許可］を　**オン**　にするとフィールドオプションで設定された［入力値の制限］によるデータ入力やインポートでの警告を無視できます。初期値は　**オフ**　です。

［アイドル状態の時、サーバーからユーザの接続を解除する］は、ログイン後、何も作業していない状況で一定時間が経過したら、接続の解除をする設定です。例えば、画面をそのままにしてユーザが席を離れて一定時間が経過したら、接続が解除され、他の人がファイルを操作してしまうことを防げます。接続時間は、FileMaker Cloud や FileMaker Server の［Admin Console］で設定します。初期値は　**オン**　です。

［ユーザによるパスワードの変更を許可］は、アカウントを同じパスワードで使い続けることを制限するかどうかです。　**オン**　にする場合は、
(1)　アカウント管理担当者がアカウントを登録する
(2)　ユーザにアカウントを知らせる
(3)　ユーザが初回ログイン時に、登録時のパスワードと違うものにする
という手順にします。オプションとして指定間隔ごとの変更、最低文字数の設定が可能です。

［利用できるメニューコマンド］は、
・すべて
・編集のみ
・最小
から、選択します。実際にメニューに表示される内容は［データアクセスとデザイン］で設定した内容により変わります。なお、カスタムメニューセットの設定内容が優先されます。

［アクセス権セットの編集］ダイアログで設定した内容は、ユーザの作業を制限します。設定後は必ずテストを実施します。

アクセス権セットによる画面表示コントロールとメニューの違いを確認します。

演習 15-2　アクセス権セット設定で違いをみる 1

- **使用ファイル：15_ 見積作成管理 .fmp12**

(1)　演習ファイルを閉じます。

(2)　ファイルを開きます。次のアカウントでログインします。

・アカウント名　Admin
・パスワード　　1111

(3)　「ホーム」レイアウトが表示されます。画面左上にアクセス権セットとアカウント名が表示されています。

(4)　これまでと同じく開発作業に必要なすべてのアクセスが可能です。

・ステータスツールバーが表示されている
・FileMaker の標準メニューセットが表示されている
・［データベースの管理］ダイアログなどが表示できる

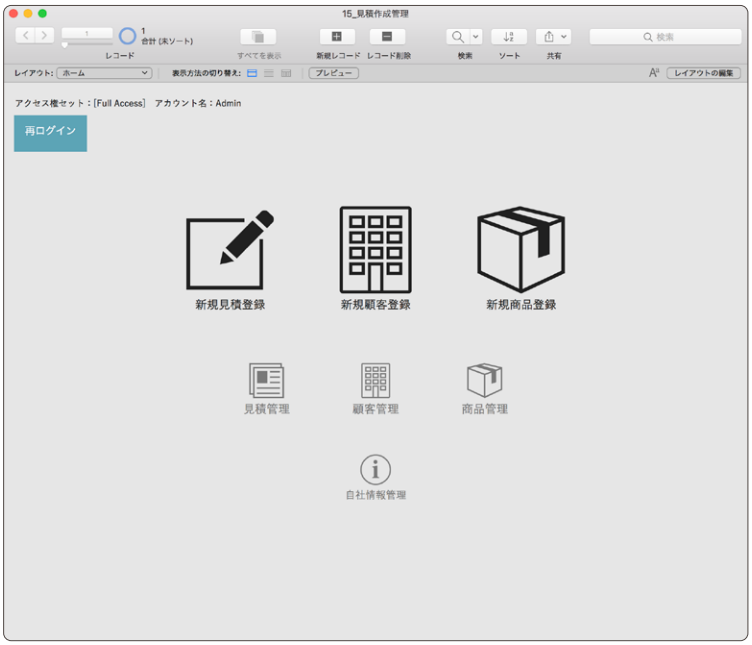


図 15-6［完全アクセス］アクセス権セットの起動後の画面

(5)　画面左上の「再ログイン」ボタンをクリックしてログインします。

・アカウント名　ユーザテスト
・パスワード　　2222

このアカウントには、「営業ユーザ」アクセス権セットが割り当てられています。

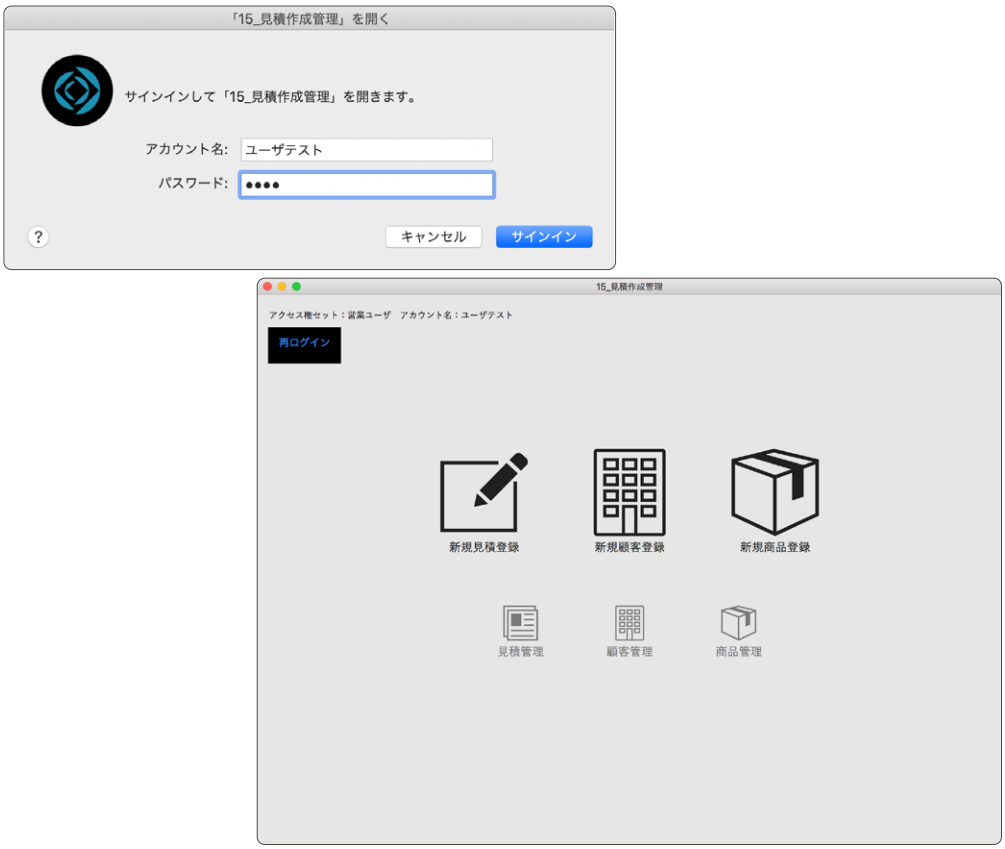


図 15-7 ユーザテスト（営業ユーザアクセス権セット）で再ログイン後の画面

違いを確認します。

アクセス権セットで制限を設定したことはできません。
例えば、
・[ファイル]–[管理] メニューが選べない
・レイアウトモードへ切り替えできない

[次の場合にオブジェクトを隠す] が設定された「自社情報管理」ボタンが表示されません。

「起動処理 _ 演習」スクリプトで指定した、次の 2 点も確認します。
・ステータスツールバーが表示されない
・ユーザ用メニューセット

(6) さらに画面左上の「再ログイン」ボタンをクリックしてログインします。
・アカウント名　管理テスト
・パスワード　3333

このアカウントには、「管理ユーザ」アクセス権セットが割り当てられています。

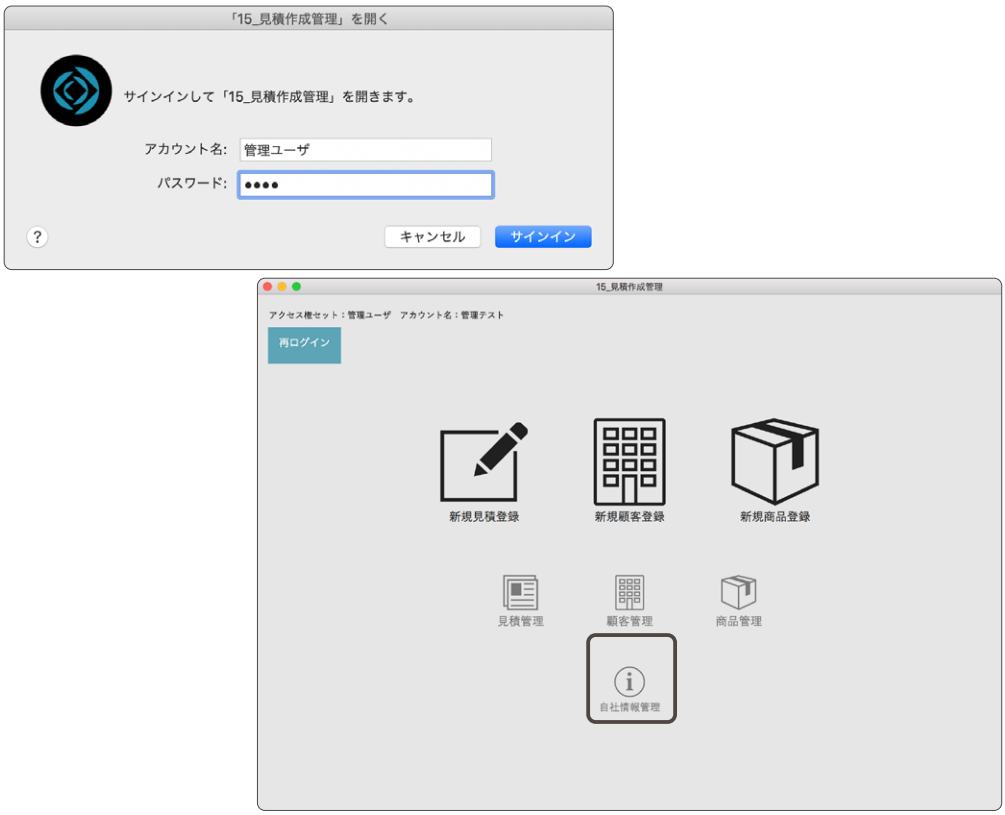


図 15-8 管理テスト（管理ユーザアクセス権セット）で再ログイン後の画面

違いを確認します。

アクセス権セットで制限を設定したことはできません。

例えば、

・レイアウトモードへ切り替えできない

[ファイル]–[管理] メニューは［値一覧 ...］だけを選択できます。値一覧は「すべて変更可能」と設定したからです。

「自社情報管理」ボタンが表示されています。

「起動処理 _ 演習」スクリプトで指定したので、次の 2 点も確認します。

・ステータスツールバーが表示されない

・ユーザ用メニューセット

(7)　Admin で再ログインしてください。

演習ここまで

アクセス権セットを使うと、様々な制限ができることを確認しました。続いて、**アクセス権セットを変更して結果を確認**します。制限を行うとどのようになるでしょうか。

演習 15-3　アクセス権セット設定で違いをみる 2

- **使用ファイル：15_ 見積作成管理 .fmp12**

(1)　［セキュリティの管理］ダイアログを開いて［詳細設定 ...］をクリックします。
(2)　「営業ユーザ」をダブルクリックして［アクセス権セットの編集］ダイアログを開きます。
現在、［レコード：］は、「すべてのテーブルでの作成。、編集、および削除」です。

(3)　［データアクセスとデザイン］の［レコード:］のポップアップから［カスタムアクセス権 ...］を選択します。［カスタムレコードアクセス権］ダイアログが表示されます。

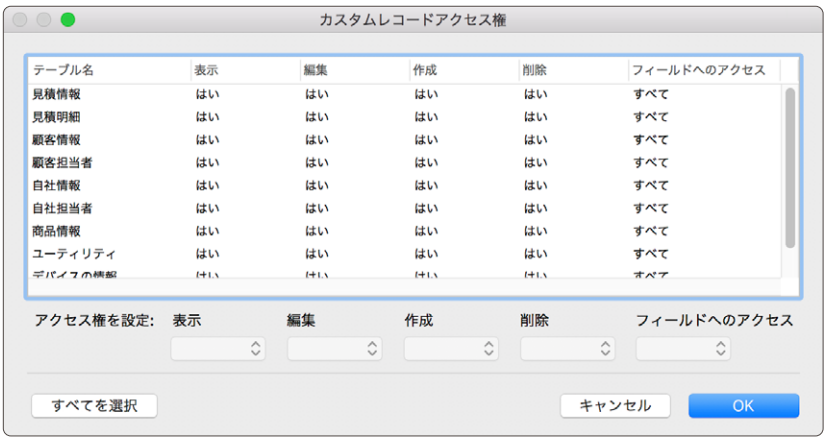


図 15-9［カスタムレコードアクセス権］ダイアログ

(4)　「見積情報」テーブルの内容を制限します。
a.　「見積情報」をクリックします。
b.　表示を「いいえ」にします。

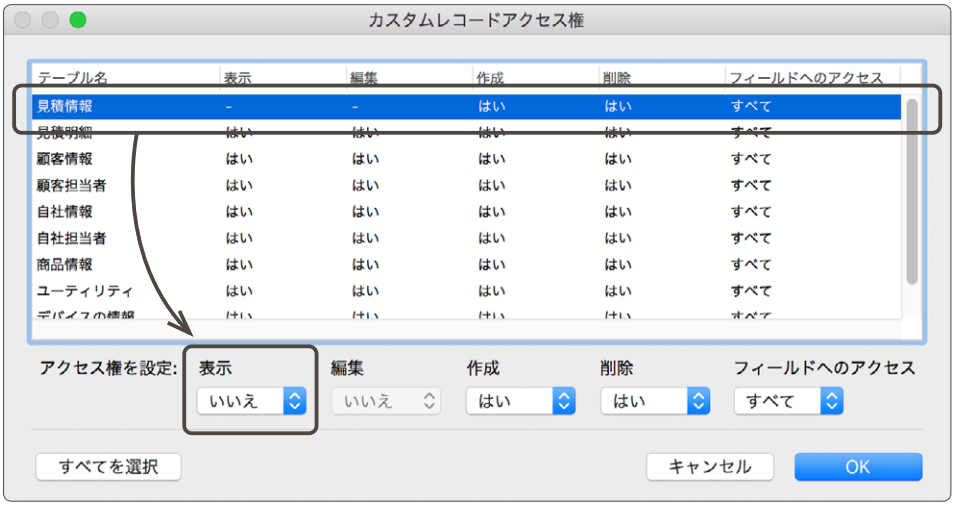


図 15-10［カスタムレコードアクセス権］ダイアログ　変更後

c.　［OK］をクリックします。
d.　［レコード：］が、「カスタムアクセス権 ...」に変わっています。

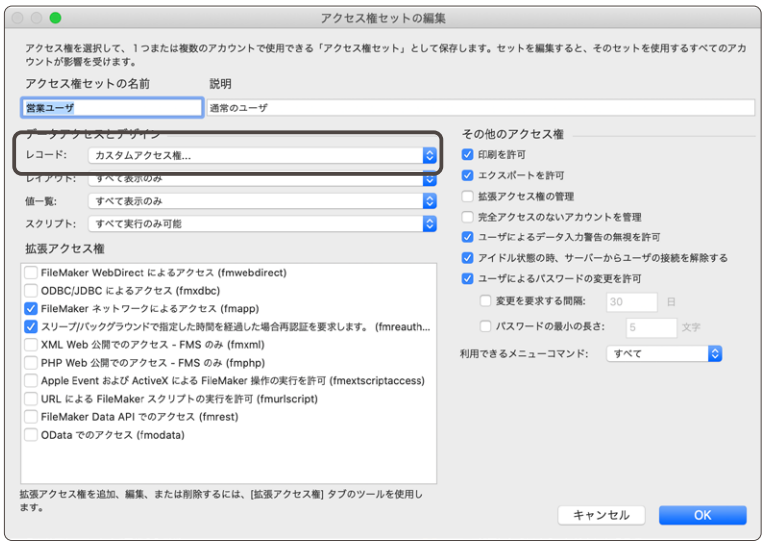


図 15-11［アクセス権セットの編集］ダイアログ　変更後

e.　さらに［OK］をクリックして、［セキュリティの管理］ダイアログを閉じます。

(5)　アクセス権セットが編集できました。

(6)　「再ログイン」ボタンをクリックしてログインします。

・アカウント名　ユーザテスト
・パスワード　　2222

(7)　「見積管理」ボタンをクリックして「見積リスト」レイアウトへ切り替えると、「< アクセス権がありません >」と表示されています。データが見えません。

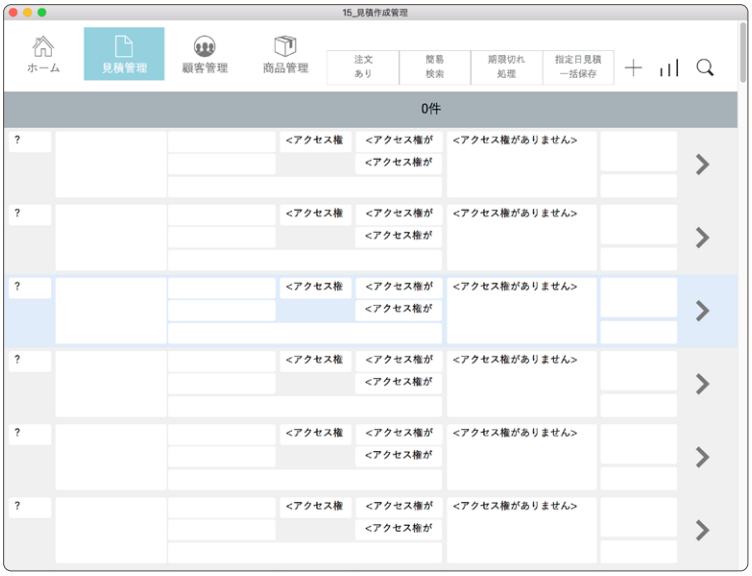


図 15-12 表示されない

(8)　「ホーム」ボタンをクリック、「再ログイン」ボタンから Admin でログインします。

(9)　さらにアクセス権セットを編集します。

a. ［セキュリティの管理］ダイアログを開きます。
b. ［詳細設定 ...］をクリックします。
c. 「営業ユーザ」の［アクセス権セットの編集］ダイアログを開きます。
d. ［データアクセスとデザイン］の［レコード：］のポップアップで［カスタムアクセス権 ...］を選択します。

e. 「見積情報」をクリックします。
f. 表示を「はい」にします。
g. 編集を「はい」にします。
h. フィールドへのアクセスで「制限 ...」を選択します。
i. ［カスタムフィールドアクセス権］ダイアログが表示されます。
j. 「件名」フィールドを「表示のみ」に設定します。

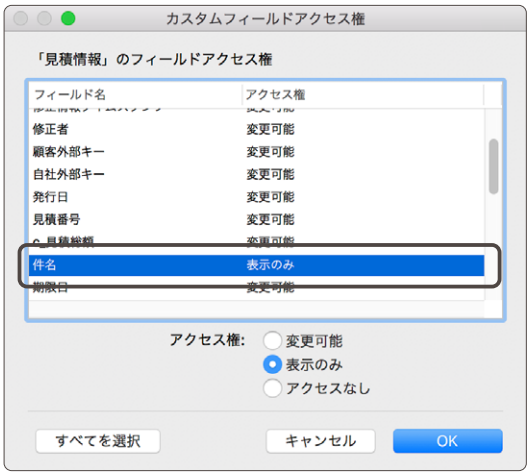


図 15-13［カスタムフィールドアクセス権］ダイアログ　変更後

k. ［OK］をクリックします。

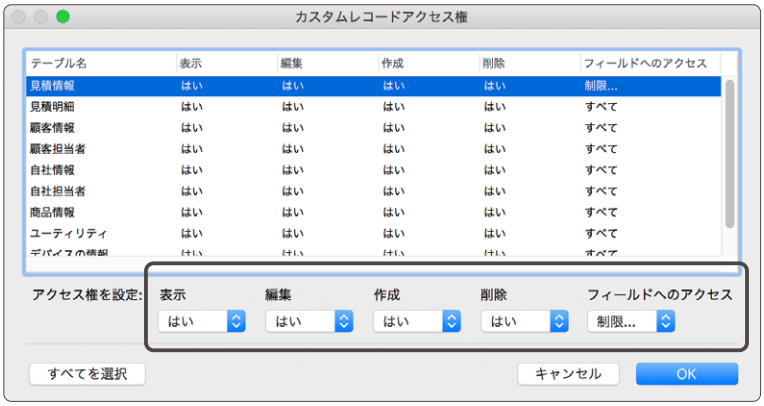


図 15-14［カスタムレコードアクセス権］ダイアログ　変更後

l. 何度か［OK］をクリックして［セキュリティの管理］ダイアログを閉じます。

(10) 「再ログイン」ボタンをクリックしてログインします。

・アカウント名　ユーザテスト
・パスワード　　2222

(11) 「見積管理」ボタンをクリックして「見積リスト」レイアウトへ切り替えると、データが表示されています。



図 15-15 表示されるようになった

(12) 任意の行の［>］ボタンをクリックして「見積詳細」レイアウトへ切り替えます。

(13) 「件名」フィールドにカーソルをいれて編集しようとするとメッセージが表示されます。

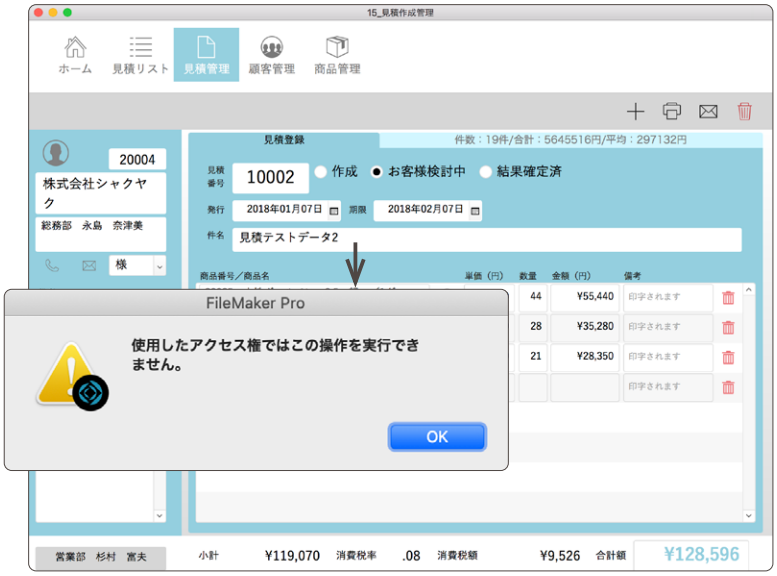


図 15-16 編集できない

(14) 「ホーム」ボタンをクリック、「再ログイン」ボタンから Admin でログインします。

(15) かなり細かな設定が可能なことがわかりました。

(16) 設定を元に戻します。

a. 「営業ユーザ」の［アクセス権セットの編集］ダイアログを開きます。
b. ［データアクセスとデザイン］の［レコード：］のポップアップで、「すべてのテーブルでの作成。、編集、および削除」を選択します。
c. 何度か［OK］をクリックして［セキュリティの管理］ダイアログを閉じます。

(17) これまでと同じ手順で、フィールドが表示され、「件名」フィールドが編集可能なことを確認します。

演習ここまで

他の［データアクセスとデザイン］のカスタムアクセス権のダイアログは次図です。

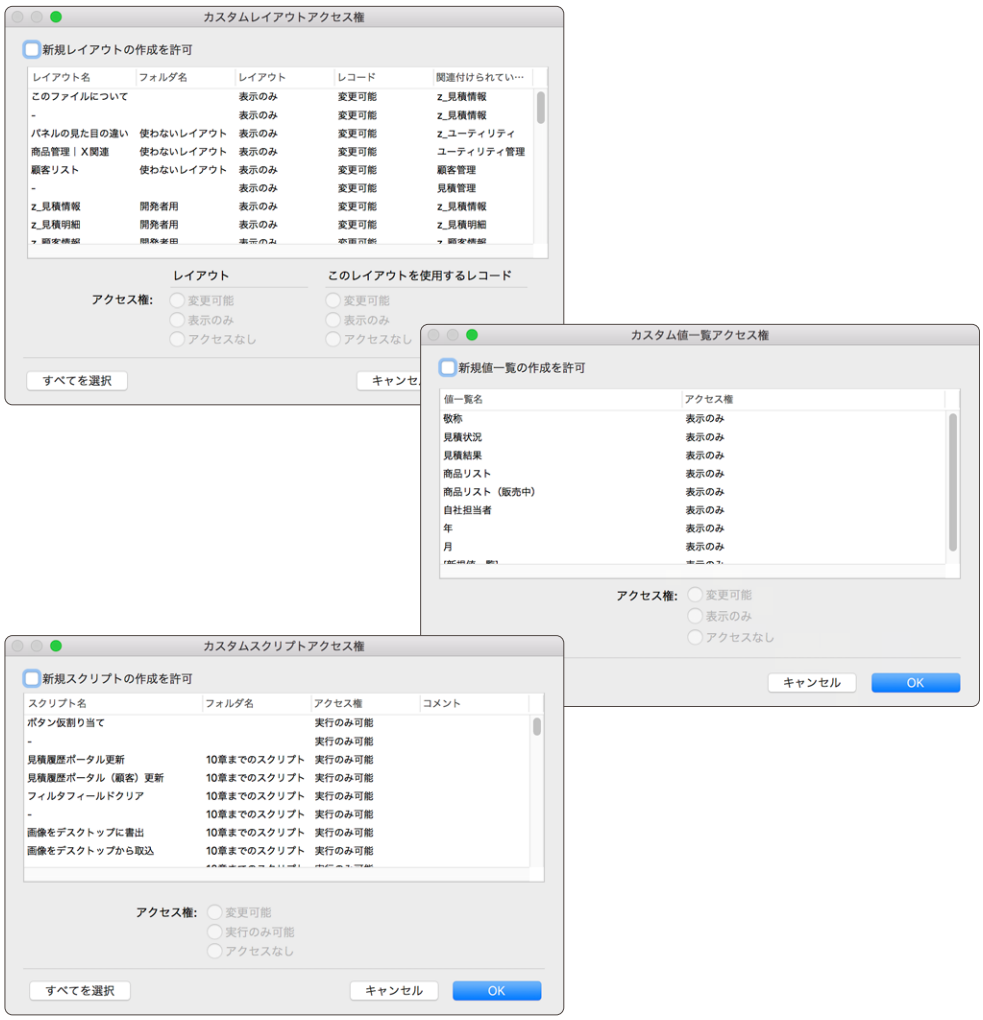


図 15-17 カスタムアクセス権のダイアログ

15.2.6 拡張アクセス権の設定

アクセス権セットには、「拡張アクセス権」も設定できます。次の方法で設定します。

手順：

(1) [ファイル]–[管理]–[セキュリティ ...] メニューを選択
(2) [セキュリティの管理] ダイアログで [詳細設定 ...] をクリック
(3) 「拡張アクセス権」タブをクリック
(4) 新規作成、編集、削除などを実行する



図 15-18 拡張アクセス権の設定

この内容は、[アクセス権セットの編集] ダイアログで設定（次図四角形部分）する名称です。

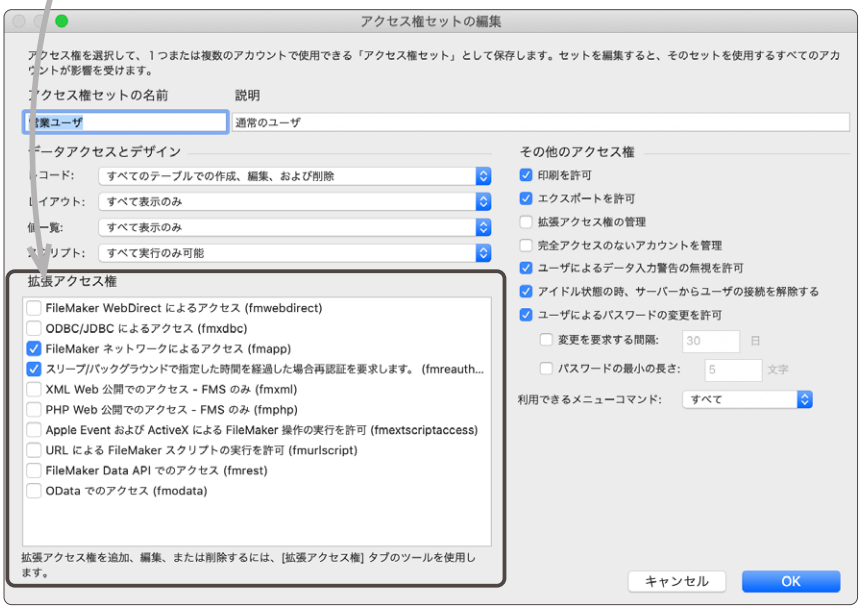


図 15-19 [アクセス権セットの編集] ダイアログ

拡張アクセス権は、次のようなものです。

・アクセス権セットに割り当てて利用できる接続経路を限定する
・1 アクセス権セットに複数の拡張アクセス権を割り当てられる

以下の関数で設定内容がわかります。

```
Get ( アカウント拡張アクセス権 )
```

デフォルトの拡張アクセス権
新しいファイルには次表の 9 つのデフォルト拡張アクセス権があります。

拡張アクセス権	内容
fmwebdirect	FileMaker WebDirect によるアクセスが可能
fmxdbc	ODBC/JDBC データソースとしてのアクセスが可能
fmapp	FileMaker Pro、FileMaker Go で FileMaker ネットワークによるアクセスが可能
fmreauthenticate10	FileMaker Pro、FileMaker Go で再ログインをおこなわなくても良いスリープ／バックグラウンドの最長時間を指定します。初期値は 10 分です。
fmxml	XMLWeb 公開でのアクセスが可能
fmphp	PHPWeb 公開でのアクセスが可能
fmextscriptaccess	AppleScript および ActiveX による FileMaker スクリプトの実行が可能
fmurlscript	URL からのスクリプトの実行が可能
fmrest	FileMaker Data API での Web サービスからのアクセスが可能

拡張アクセス権の設定を変更してもすぐに有効にはなりません。

・FileMaker Pro や FileMaker Go を終了
・ファイルを閉じる
・ログアウトする

これらによりセッションを終了後、次のセッションから有効になります。

FileMaker Go と拡張アクセス権
iOS / iPadOS はマルチタスクなので、常に FileMaker Go を使っているとは限りません。他のアプリを使用している間は FileMaker Go はバックグラウンド状態になります。

［fmreauthenticate10］は、10 分間までの間、FileMaker Go の使用を中断できるということになります。10 分が過ぎると、戻ってきたときにログインを要求します。この設定は最大「10080」分（一週間）まで可能です（第 13 章も参照）。

やってみよう
「見積作成管理 .fmp12」のアクセス権セットには、［fmreauthenticate10］は設定されていません。設定を行った前後で FileMaker Go でどのように変わるかを確認しましょう。

拡張アクセス権を作成する
デフォルトの９つとは別の拡張アクセス権を作成して利用できます。新しく作成する拡張アクセス権は、デフォルトのものとは少し種類が違います。アクセス権セットに任意のキーワードを付加するものです。

例えば「見積書印刷」という拡張アクセス権を作成して「営業ユーザ」「事務員ユーザ」「経理ユーザ」アクセス権セットに割り当て、見積書を印刷するスクリプトで **Get(アカウント拡張アクセス権)** 関数の結果に FilterValues 関数で拡張アクセス権名があったら実行するという条件分岐を入れる、などの使い方ができます。

15.2.7 アカウント情報

原則として 1 ユーザに 1 アカウントで運用することで人の識別が可能です。そして、1 アカウントごとに必ず 1 アクセス権セットを割り当てなければなりません。

本書では、1 ユーザに 1 アカウントの運用ルールを強く推奨します。アカウント情報の共有は、認証情報漏洩の危険が高くなるので、特別な事情がない限り、してはいけません。また、1 ユーザが 1 アカウントを使用することにより、何か問題が起きたときのユーザの作業調査を可能にします。

アカウント作成の注意点としては、次のことがあります。

・アカウント名の大文字と小文字は区別されない
・パスワードは大文字、小文字が区別される

認証方法
これまで演習でおこなったログインは「FileMaker ファイル」認証です。FileMaker Pro では、カスタム App の共有手段に応じてそれぞれ次の方法が用意されています。

・FileMaker ファイル
・外部サーバー（Apple Open Directory、Windows ドメイン）
・OAuth アイデンティティプロバイダ（Amazon、Google、Microsoft Azure AD）
・Claris ID
・外部 IdP (Okta、Microsoft Active Directory (AD))

「FileMaker ファイル」の認証
FileMaker ファイル内に保持しているアカウントでログインする方法です。アカウントを登録すると、パスワードは、不可逆の暗号化（ハッシュ化と呼ばれます）が施されてファイルに保存されます。

ユーザがログインダイアログでアカウントを入力すると、パスワードは同じようにハッシュ化されて、保存されている内容と比較されています。

複数のファイルで構成されるカスタム App とアカウント
FileMaker Pro のセキュリティ設定はファイルごとに実施する必要があり、設定内容のインポートなどができません。アクセス権セットや拡張アクセス権は、手作業でファイルごとに設定します。

アカウントの登録は、スクリプトで実施することも可能です。その場合、アカウントの登録がエラーになったらスクリプトを続行するかどうかなどエラー処理を考慮するべき点がいくつかあります。

［完全アクセス］アクセス権セットのアカウントの削除
例えば、なにか調査をするとします。協力者のパソコンなどでカスタム App をスタンドアロンで使用して調査結果の入力することにします。そして入力された結果はエクスポートされたデータとして受け取ります。このような場合、カスタム App のファイルを複数のユーザに配布する必要があります。

このとき、FileMaker Pro の［Developer ユーティリティ］を利用すると、ファイルに登録されているすべての［完全アクセス］アクセス権セットのアカウントを削除できます。このことにより、ファイルの構造変更などの作業を禁止することができます。

この作業は元に戻せないので、慎重に行ってください。［完全アクセス］アクセス権セットのアカウントを削除する前のファイルを別途保存しておくことをお勧めします。
※ 高度なツールを **オン** にしていると［Developer ユーティリティ］を使用可能です（方法は 9 ページを参照）。

「外部サーバー」
FileMaker Server でカスタム App を共有する場合に、ネットワーク上の認証サーバーをカスタム App のログイン時の認証に利用する方法です。

ファイルにアカウント情報は登録しません。代わりに認証グループ名を指定します。このグループ名に対して、1 つのアクセス権セットを割り当てます。また、外部認証をおこなうには、FileMaker Server でファイルがホストされている必要があります。

企業や団体などの場合、複数のシステムを 1 つのアカウントとパスワードで利用できるように、システム部門がルールを決めてネットワーク上に認証サーバーを設置している場合があります。その仕組みを利用するのが外部認証です。ユーザはカスタム App と他のシステムを同じアカウントで使用できます。外部認証は、認証サーバーを管理している部門や部署と連携して設定してください。特にグループ名の変更は双方で連絡を取りあって同じになるように運用します。

アカウント情報を入力してログインを実行すると、認証サーバーに情報が渡され、確認されます。認証が OK なら、そのアカウントが所属しているグループ名がファイルに設定されているか確認し、最初に合致したアクティブなグループ名のアクセス権セットに決定します。

ユーザが全員、外部認証でファイルにアクセスする場合も、必ず［完全アクセス］アクセス権セットのアカウントを 1 つ以上作成してください。

「OAuth アイデンティティプロバイダ」の認証
FileMaker Server でファイルがホストされている場合に利用可能な認証方法です。OAuth アイデンティティプロバイダに基づいてユーザが認証されます。

「Claris ID」および「外部 IdP」による認証
FileMaker Cloud でファイルがホストされている場合に利用可能な認証方法です。Claris ID は Claris International Inc. の製品とサービス全般にわたってユーザを認証するための総合的なアカウントシステムです。

外部サーバー、OAuth アイデンティティプロバイダ、Claris ID あるいは外部 IdP アカウントによる認証について、詳しくはヘルプやクラリス・ジャパン株式会社の Web サイト（https://

www.claris.com/ja）を参照してください。

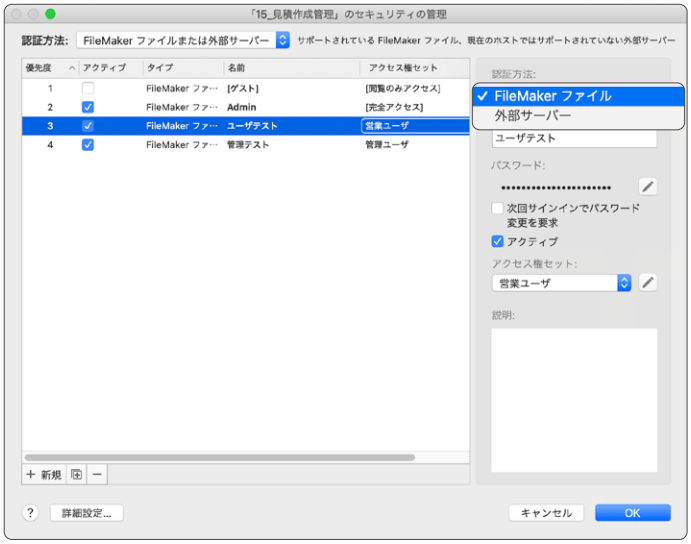


図 15-20 認証の種類（ローカル環境で開いたとき）

認証方法がわかる関数

次の関数が用意されています。

```
Get ( アカウントタイプ )
```

現在のアカウントの認証タイプが返ります。次表の通りです。

アカウントタイプ	返される値
FileMaker ファイル : ゲスト	Guest
FileMaker ファイル	FileMaker File
外部サーバー	External
OAuth: Amazon	Amazon
OAuth: Google	Google
OAuth: Microsoft Azure AD	Azure
Claris ID または外部 IdP	Claris ID < チーム名 > < チーム名 > はユーザが属しているチームの名前

演習ファイルを閉じて、次のファイルを開いてください。

（ここから、本書に付属している演習ファイルの「演習 15_4 から 5」フォルダの中の「15_ 見積作成管理 .fmp12」「目録 .fmp12」「会議 .fmp12」ファイルをベースに進めます）

15.3 ファイルアクセス

これまではユーザとファイルについてのセキュリティのための解説でしたが、ファイル間のセキュリティについても設定することが可能です。

外部データソース

FileMaker ファイルは、他のファイルを参照できます。普通に外部データソースを設定するには、

手順：

(1) [ファイル]–[管理]–[外部データソース ...] メニューを選択
(2) [外部データソースの管理] ダイアログが表示される
(3) 設定を行う

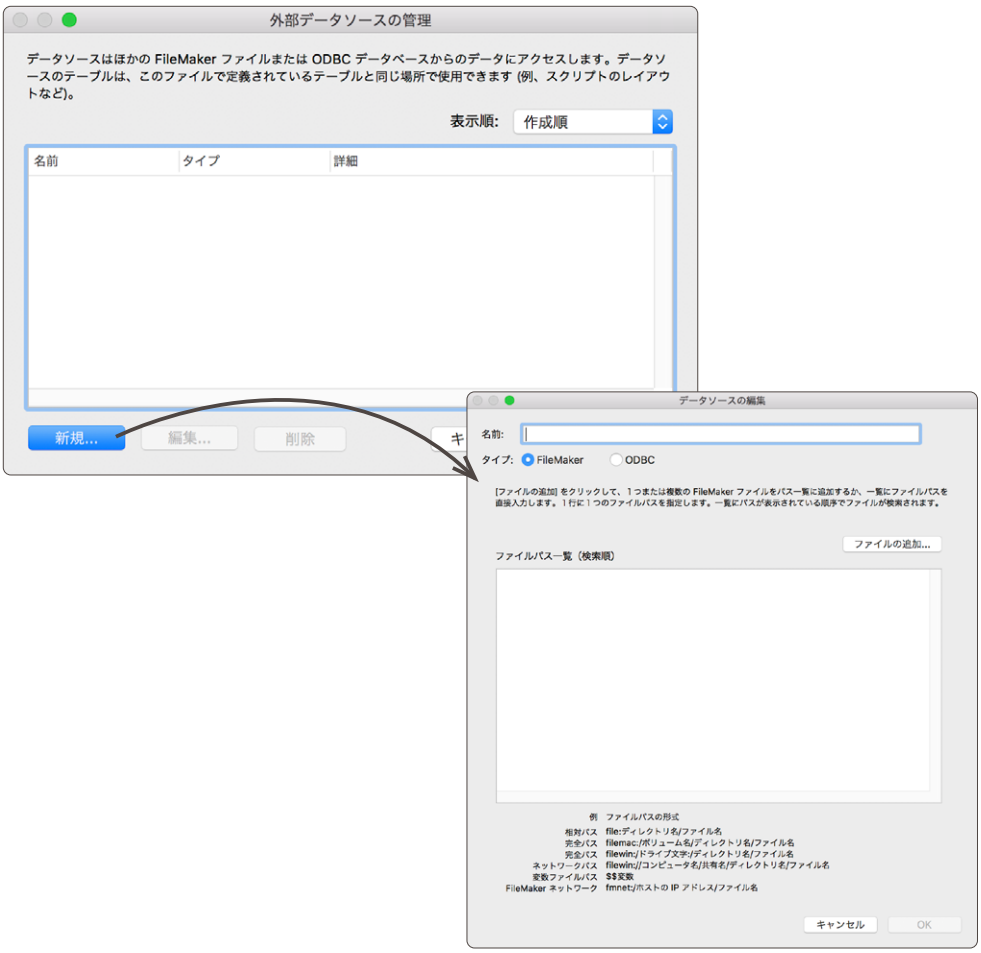


図 15-21 外部データソースの設定

例えば、「演習 15_4」フォルダに「見積作成管理 .fmp12」と「目録 .fmp12」ファイルがある場合、 file: 目録 と登録すると、参照できます。

外部データソースで参照可能なファイルは、そのファイルのテーブルをリレーションシップグラフで TO を作成し、スクリプトなどで利用できます。しかし、ファイルによっては外部からの参照を

制限したいことがあるかもしれません。

他のファイルからの参照について「**ファイルアクセス**」で制限を設定します。

ファイルアクセスの制限
外部参照時に、そのファイルの［完全アクセス］アクセス権セットのアカウントでログインしなければならないという制限が設定可能です。通常は　**オン**　になっています。次の方法で確認と設定を行います。

手順：
(1)　［ファイル］–［管理］–［セキュリティ ...］メニューを選択
(2)　［詳細設定 ...］をクリック
(3)　「ファイルアクセス」タブを選択
(4)　設定内容が表示される



図 15-22 ファイルアクセスの設定（新規ファイル作成後）

［このファイルへの参照の使用に完全アクセス権を要求する］のチェックが　**オン**　では、

・「保護されたファイル」（別のファイルから参照されるためにはログインが必要）
・未認証のファイルからは［ファイルを開く］スクリプトステップで開けない

という状態になります。

なおデフォルトでは、そのファイル自身の認証が含まれます（上図参照）。ファイル自身を参照できるので、コピーされたファイルまたはクローンファイルが相互に連携できます。この状態が望ましくなければ、自己認証を取り除きます。

ファイルの識別子
認証されたファイルには英数字によるユニークな識別子が付与され、保護されたファイルが把握しています。認証されたファイルの名前の変更や複製された場合でも確実に安全な状態を維持できるためです。認証されたファイルをまったく別の内容の同名のファイルと差し替えようとしてもできません。

「演習 15_4 から 5」フォルダには、「見積作成管理 .fmp12」（ファイルアクセスが　オフ）の他に、外部データソースとして利用するための「目録 .fmp12」と「会議 .fmp12」ファイルを作成しておきました。

3 つのファイルには、次の［完全アクセス］アクセス権セットのアカウントが登録されています。

アカウント		アクセス権セット
アカウント名	パスワード	
fmb	fmb	完全アクセス

外部データソースについて確認します。
「目録 .fmp12」ファイルは演習用に次のように設定されています。

・［このファイルへの参照の使用に完全アクセス権を要求する］のチェックは　**オフ**

「演習 15_4 から 5」フォルダ

演習 15-4　外部データソースの確認

- **使用ファイル：15_ 見積作成管理 .fmp12　別のファイル .fmp12**

(1)　「見積作成管理 .fmp12」を開きます。ログインは Admin、fmb、どちらのアカウントでも構いません。

(2)　［データベースの管理］ダイアログを表示して「リレーションシップ」タブを選択します。
a.　左下の［テーブルを追加］ボタンを押します。
b.　［FileMaker データソースの追加 ...］を選択します。
c.　ファイルを選択するダイアログが表示されます。
d.　「15 章」フォルダの中の「演習 15_4 から 5」フォルダの中の「目録 .fmp12」ファイルを指定して［開く］をクリックします。
e.　fmb アカウントでログインします。
f.　［テーブルを指定］ダイアログが表示されます



図 15-23 ファイルを指定後に表示される［テーブルを指定］ダイアログ

g. 「製品」をクリックします。
h. ［名前］を「外部データソース製品」と入力します。
i. ［OK］をクリックします。
j. リレーションシップグラフに「外部データソース製品」TO が出来上がります。
(外部データソースの TO は、TO 名が斜体で表示されます)



図 15-24 TO が追加された

k. ［OK］をクリックして、［データベースの管理］ダイアログを閉じます。
（この方法は、他のファイルの TO を作成する作業中に外部データソースも登録されるものです）

(3) 「見積作成管理 .fmp12」の外部データソースの内容を確認します。
a. ［ファイル］–［管理］–［外部データソース ...］メニューを選択します。
b. ［外部データソースの管理］ダイアログが表示されます。



図 15-25［外部データソースの管理］ダイアログ

c. 「目録」をダブルクリックすると次のように登録されています。

file: 目録

d. ［OK］をクリックして、［外部データソースの管理］ダイアログを閉じます。

(4) ［ウインドウ］–［ウインドウを表示］メニューを選択すると、次のように表示されます。
これは、「見積作成管理 .fmp12」から「目録 .fmp12」ファイルが参照されているためです。

（目録）

(5) 「見積作成管理 .fmp12」を閉じます。参照されているファイルも閉じるので［ウインドウ］メニューを選択しても、何も表示されません。

演習ここまで

続いて、**ファイルアクセスについて確認します。**
「会議 .fmp12」ファイルは

・［このファイルへの参照の使用に完全アクセス権を要求する］のチェックは　**オン**

に、なっています。

演習 15-5　ファイルアクセスの確認

- **使用ファイル：15_ 見積作成管理 .fmp12　別のファイル .fmp12**

(1) 「見積作成管理 .fmp12」を開きます。fmb アカウントでログインします。

(2) ［データベースの管理］ダイアログを表示して「リレーションシップ」タブを選択します。
a. 左下の［テーブルを追加］ボタンを押します。
b. ［FileMaker データソースの追加 ...］を選択します。
c. ファイルを選択するダイアログが表示されます。
d. 「15 章」フォルダの中の「演習 15_4 から 5」フォルダの中の「会議 .fmp12」ファイルを指定して［開く］をクリックします。
e. メッセージが表示されます。これが保護されたファイルへの認証です。



図 15-26 保護されたファイルについてのメッセージ

f. ［はい］をクリックします。
g. 「会議」をクリックします。

h. ［名前］を「ファイルアクセス会議」と入力します。
i. ［OK］をクリックします。
j. リレーションシップグラフに「ファイルアクセス会議」TO が出来上がります。



図 15-27 TO が追加された

(3) ［OK］をクリックして［データベースの管理］ダイアログを閉じます。

(4) 「見積作成管理 .fmp12」の外部データソースの内容を確認します。
a. ［ファイル］–［管理］–［外部データソース ...］メニューを選択します。
b. ［外部データソースの管理］ダイアログが表示されます。



図 15-28［外部データソースの管理］ダイアログ

c. 「会議」をダブルクリックすると次のように登録されています。

file: 会議

d. ［OK］をクリックして、［外部データソースの管理］ダイアログを閉じます。

(5) 「見積作成管理 .fmp12」でファイルアクセスを確認します。

a. ［詳細セキュリティ設定］ダイアログを表示し「ファイルアクセス」タブを選択します。

b. 何も表示されていません。



図 15-29 認証された側には表示はない

c. ［OK］をクリックしてダイアログを閉じます。

(6) 「会議 .fmp12」で認証ファイルになっているか確認します。

a. ［ウインドウ］–［ウインドウを表示］メニューの（会議）を選択します。

b. 「会議 .fmp12」のウインドウが表示されます。

c. ［詳細セキュリティ設定］ダイアログを表示し「ファイルアクセス」タブを選択します。

d. 認証されたファイル名、認証日、認証したアカウントが表示されています。



図 15-30 認証した側に認証ファイルとして表示される

e. ［OK］をクリックしてダイアログを閉じます。

(7)　ファイルを閉じます。2 つのファイルの間にファイルアクセスが設定されました。

演習ここまで

認証を解除する
ファイルを選択して［認証の解除］をクリックすると、解除されます。解除後、ファイルから再度参照しようとするとメッセージが表示されて認証を行う必要があります。

［詳細セキュリティ設定］ダイアログで認証を行う
演習手順とは別の方法も可能です。［詳細セキュリティ設定］ダイアログの「ファイルアクセス」タブの、［認証 ...］をクリックして、ファイルの認証を行うことが可能です。

リセットする
保護されたファイルを複製すると同じ識別子になります。一方のファイルの識別子をリセットし、重複しないようにします。

リセットは、ファイルの［詳細セキュリティ設定］ダイアログの「ファイルアクセス」タブの、［すべてをリセット ...］をクリックすると、次図のメッセージが表示されます。［はい］をクリックします。

リセット後は、そのファイルに認証されているすべてのファイルの再認証と、リセットされたファイルのアクセスが認証されていた他の保護されたファイルへの再認証を行う必要があります。



図 15-31　リセットのメッセージ

15.4 暗号化によるセキュリティ対策

アカウントやファイル参照以外にもセキュリティ対策をすることができます。

ネットワーク、サーバーやクライアントのハードウェアへの外部あるいは想定していない内部からのアクセスについて対策が必要です。しかし、アクセスされてしまった場合についても対策しておくことができます。

15.4.1 ファイルを暗号化

ハードウェアにアクセスされてしまう事態が起きたら、その悪意がある人はアカウントがわからなくても他の方法でファイルのデータを読み取ることができるかもしれません。その対策としてファイルを暗号化することができます。暗号化後、次のようになります。

・暗号化パスワードを知らない人はファイルを使用することができない

暗号化パスワードは「暗号化のキー」です。

暗号化と復号

「復号」は、暗号化、圧縮化などがされているデータを元のデータに復元することです。暗号化されているデータの復号は、キー（秘密の暗号キー）で行います。

通常、ファイルは暗号化されていません。必要な場合、FileMaker Pro の［Developer ユーティリティ］で暗号化を行います。以下の作業が可能です（FileMaker Cloud ではホストするファイルには暗号化が必須です）。

※ 高度なツールを **オン** にしていると［Developer ユーティリティ］を使用可能です（方法は 9 ページを参照）。

・暗号化する、暗号化キーを変更する
・暗号化をやめる

暗号化は、指定したファイルのコピーを作成して行われます。つまりファイルのバックアップが保持されます。

暗号化ファイルの使用

FileMaker Pro、FileMaker Go、FileMaker Cloud、FileMaker Server は暗号化されたファイルを読み書きすることができます。FileMaker Pro および FileMaker Go で暗号化されているファイルを開こうとすると、まず暗号化パスワードの入力を求められます。暗号化パスワードが正しく入力されると、通常のアカウントログインを求められます。FileMaker Cloud や FileMaker Server でホストされている場合は、展開時に暗号化パスワードを入力して保存することが可能です。この後、クライアント側では暗号化パスワードを入力することなくカスタム App を利用することができます。

暗号化パスワードの管理

暗号化して安心していても簡単に読み取れる状態（テキストファイルなど）で暗号化時に設定したパスワード情報と暗号化されたファイルを一緒に保存していては意味がありません。暗号化パスワードの情報は物理的に別の場所で管理するようにしてください。例えば書き留めて鍵のかかる場所に保管することです。

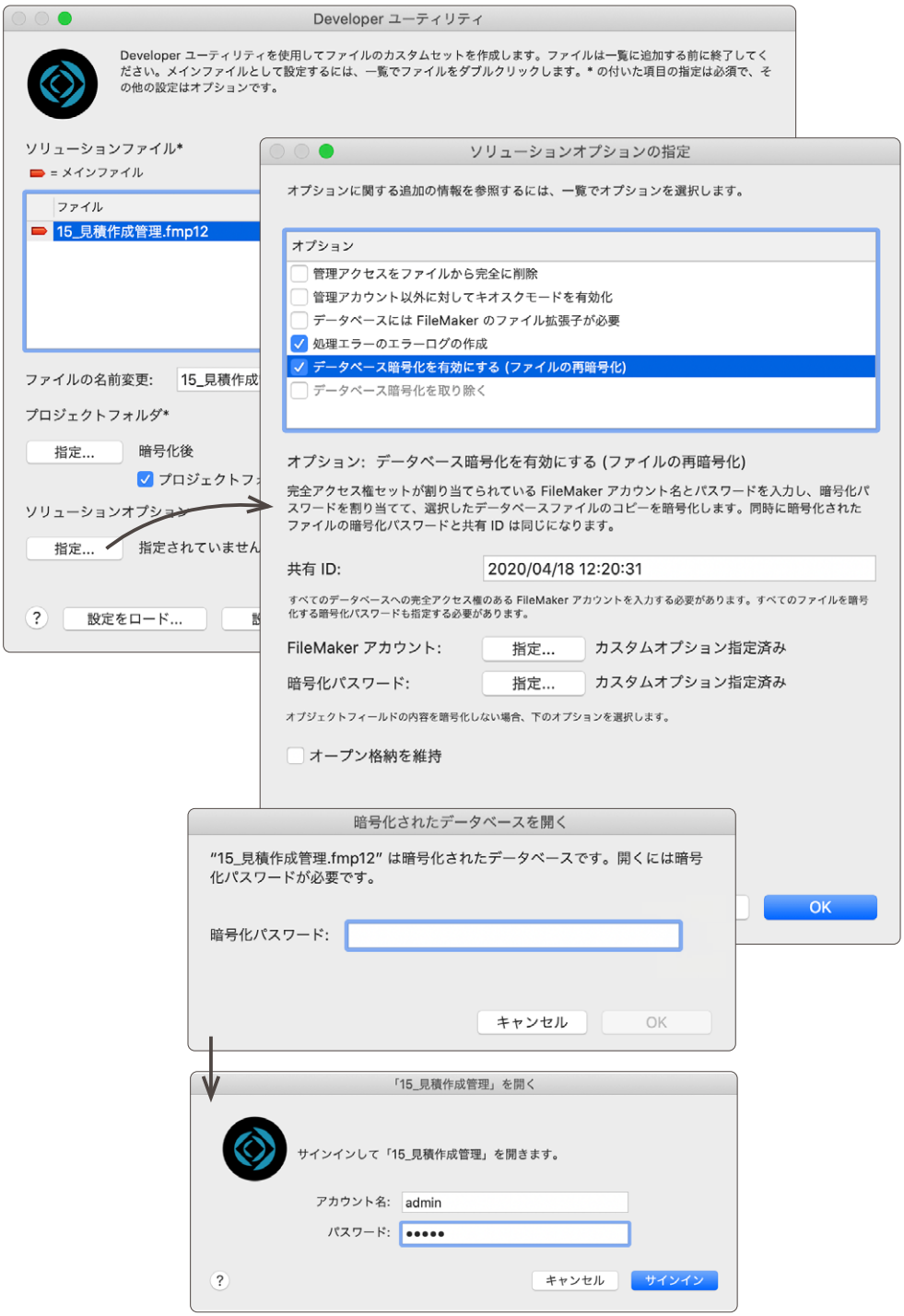


図 15-32 暗号化のダイアログ（一部）と暗号化後のファイル使用時の例

15.5 ファイルの表示と非表示

ホストされているファイルを開く場合、［ホスト］ダイアログにファイルアイコンを表示するかどうかについて設定が可能です。ファイルが見えると『何のファイルなんだろう……』とユーザの興味を引くことがあります。些細なことではありますが、ファイルの存在を知られないようにすることもセキュリティ対策として有効です。

15.5.1 ファイルを常に非表示にする

ファイルを開くための［ホスト］ダイアログに表示しない設定が可能です。これは、ファイルごとに設定します。なお、次の手順では、ファイルを非表示にする設定に加え、アクセス権セットによってファイルにアクセスできるユーザを限定しています。

手順：

(1)　［ファイル］–［共有設定］–［FileMaker クライアントと共有］メニューを選択
(2)　［FileMaker ネットワーク設定］ダイアログが表示される
(3)　次の図のように設定を行う

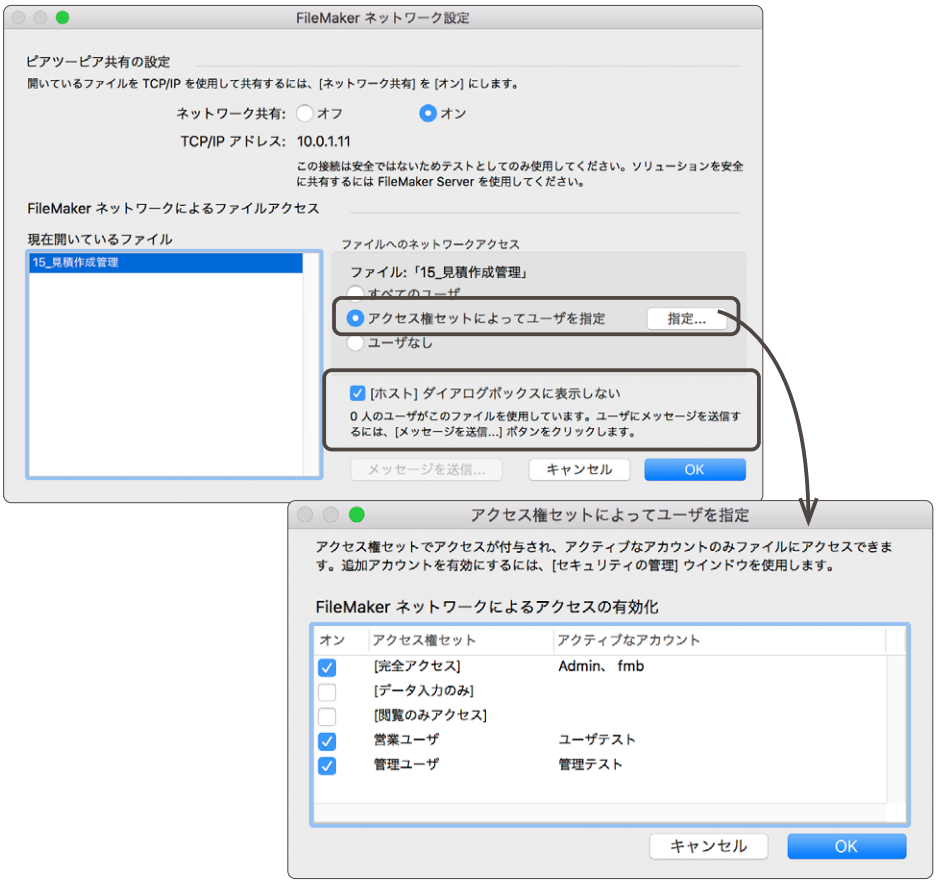


図 15-33［FileMaker ネットワーク設定］ダイアログ

「[ホスト］ダイアログボックスに表示しない」のチェックを　**オン**　にすると、文字通り、ダイアログに表示されなくなります。その場合、どうしたら良いのでしょうか。

「**ネットワークファイルパス**」を使います。

手順：

(1)　[ファイル]–[ホスト]–[ホストを表示 ...] メニューを選択
(2)　[ホスト] ダイアログが表示される
(3)　右上の書類アイコンをクリックする
(4)　ネットワークファイルパスが表示される
(5)　次の形式でパスを入力します。
　　`fmnet:/` [サーバーの IP アドレスや名前] / [ファイル名]
(6)　ログインしてファイルを開きます。

ネットワークパス
ファイルの表示／非表示によるの対策は [ホスト] ダイアログにファイルが表示されないだけなので、サーバーの IP アドレスやファイル名が分かっているとログインダイアログを出すまでは簡単です。しかし、不用意なアクセスを招かないためには手軽でわかりやすい対策です。

15.5.2　FileMaker WebDirect の設定

FileMaker WebDirect でも、ホストのファイル一覧に表示しない設定が用意されています。これは、[ファイル]–[共有設定]–[FileMaker WebDirect の構成] メニューを選択し [FileMaker WebDirect の設定] ダイアログを開いて設定します。

なお、次の図では、ファイルを非表示にする設定に加え、アクセス権セットによってファイルにアクセスできるユーザを限定しています。

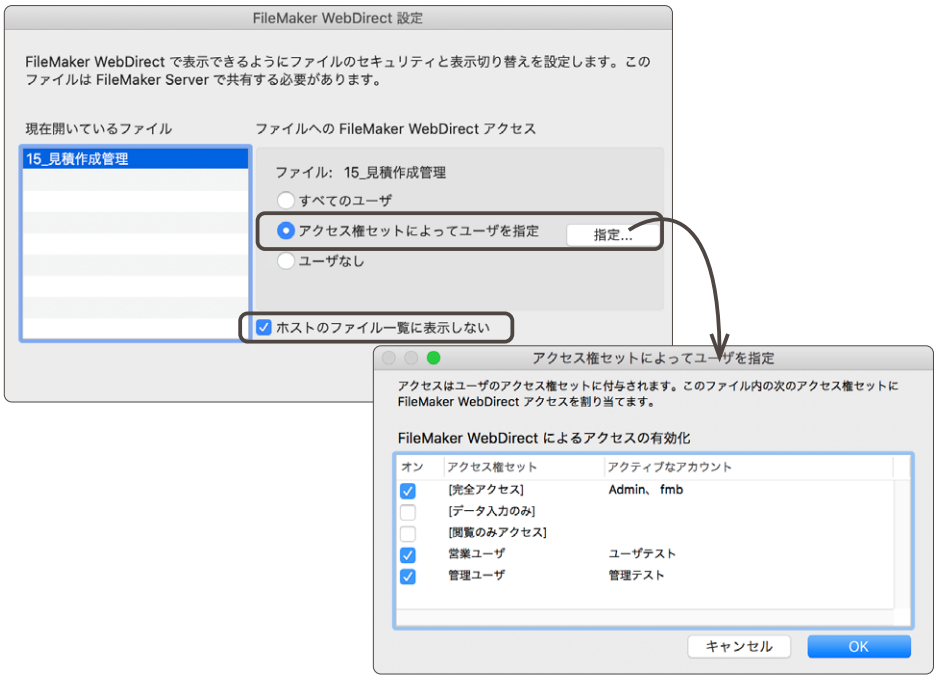


図 15-34 [FileMaker WebDirect の設定] ダイアログ

15.6 セキュリティを計画する

セキュリティの計画は、運用開始前までに行ってください。カスタム App の取り扱うデータの重要性によりどのような対策を設定するかについて取捨選択をします。

まず、内外からの不正なアクセスなどからデータの保護を考えます。

- ネットワーク、サーバーとクライアントのハードウェア、OS のセキュリティ対策がどのように行われているかを確認する
- 必要があれば、上記の対策の変更を担当者と検討する

ファイルのセキュリティでは、

- 他のファイルからの外部参照についてファイルアクセスの設定が必要があるかどうか
- ファイルを暗号化するかどうか（暗号化パスワードの保管場所をファイルとは別の場所にする）
- 共有ファイルを開く際に、ファイル名を表示するかどうか

［完全アクセス］アクセス権セットのアカウントについては、

- 開発担当者以外が使用できないように管理する
- 役割によって複数作成する
- 緊急時のアカウントを、ファイルとは別の場所に保管する

アカウントは、

- カスタム App のユーザを決定
- ユーザによるアカウントのログイン方法を、外部認証か、FileMaker ファイル認証か決定
- ユーザの権限を必要に応じて 1 つあるいは複数決定
- アクセス権セットを権限を決定内容に応じて作成しアカウントに割り当て

外部認証を使用する場合は運用ルールは担当者と検討して決定し FileMaker ファイル認証を使用する場合は、

- アカウントを 1 人 1 つにする
- パスワードはユーザのみが知り得るようにする
- 以下の作業についてルールを決定する
    - アカウントの管理担当者の決定
    - アカウントの新規登録
    - アカウントの停止
    - アカウントの再開
    - アカウントの削除
    - パスワードを忘れたときの対応

FileMaker Cloud / FileMaker Server では SSL 暗号化通信を行いますが、これに際して FileMaker Server では次の作業が必要です（FileMaker Cloud では、Claris International Inc. の保守チームが行います）。

・SSL 暗号化通信のための証明書を購入しインストールする

デフォルトの証明書
FileMaker Server では、Claris International, Inc. により署名された標準の SSL 証明書が提供されていますが、サーバー名の検証は行われません。この FileMaker デフォルト証明書はテスト用にのみ利用できます。

さらに、

・運用中のファイルだけではなく、バックアップファイルの保存場所についても不正なアクセスから守る対象として決定

また、物理的なアクセスにも備えます。ネットワークやハードウェア、ファイルに対策していたとしても、誰でも物理的にアクセスできて簡単に持ち出せたりコピーできる場所にファイルや暗号化パスワードが保管されているとセキュリティが低くなります。

これらに加えてユーザがデータをきちんと扱えるように整合性を考えます。カスタム App で、アクセス権セットやカスタムアクセス権によりデータの表示や実行できる機能を制限する必要があるかを検討します。

例えば、

・ステータスツールバーの表示／非表示
・カスタムメニューの使用／不使用

これもセキュリティのための対策の 1 つです。

企業や団体によっては、カスタム App のデータの作成、変更、削除の履歴（ログ）を取得して保存する必要があるかもしれません。FileMaker Cloud / FileMaker Server でも、いくつかの種類のログを記録することができます。

充分な作業期間を設けて計画、設定、確認を行います。

演習ファイルを閉じてください。

第 16 章 カスタム App の最適化

ユーザに必要な機能以外の運用に必要な最適化や機能を考えます。さらに、メンテナンスについても考慮します。

16.1 ユーザについて

カスタム App に必要な機能は、使う人の役割に応じて変わることがあります。

「見積作成管理システム」では、ユーザの運用について次表のように決定しました。

ユーザ	**使用方法**	作業に応じて必要なアカウントとパスワードでログインする
	アカウント	1 人に 1 つ（種類によって操作できる機能が違う）
	アカウント種類	共通値の管理（アカウントの管理も含む）
		通常利用

さらに詳しく、機能ごとの　できる／できない　を決定します。開発者用のアカウントも必要です。

例えば次表のように、使う人の役割に応じてユーザの種類とできることを決定します。

ユーザの種類	**見積**			**顧客**			**商品**			**アカウント**		**自社情報管理**
	登録	**編集**	**削除**	**登録**	**編集**	**削除**	**登録**	**編集**	**削除**	**登録**	**削除**	
共通値の管理	×	×	×	○	○	○	○	○	○	○	○	○
通常利用	○	○	○	×	○	×	×	○	×	×	×	×
開発者	○	○	○	○	○	○	○	○	○	○	○	○

「共通値管理ユーザ」は、見積書は作成せず、マスタ（顧客、商品、アカウント、自社情報）の管理をおこなうだけという考え方です。そして「開発者ユーザ」はすべての作業が可能となっています。

動作のテストをするため、開発者ユーザは『なんでもできる』方が便利です。ただし、以下の 2 つの立場を持っているしたら、

・開発者
・ユーザ

立場ごとにアカウントを作成して使い分けることをお勧めします。なお、原則としてアカウントの管理は、担当ユーザ（上表では、共通値の管理ユーザ）に任せます。

この、ユーザの種類は「アクセス権セット」と関係します。

ユーザの種類を念頭において、あらためてカスタム App に機能を加えます。起動と終了に関係することなど、いくつかの事柄です。

16.2　ステータスツールバーの表示／非表示

カスタム App の作成中は、ブラウズモードでステータスツールバーが表示されていますが、運用開始後も表示したままにするかどうかは検討する必要があります。

スクリプトやボタンで機能を作成せずに、標準のコマンドを使ってユーザが操作する前提であれば、そのまま表示しておきます。

機能を作成し、レイアウト上のボタンを使って欲しい場合は非表示をお勧めします。ユーザに 2 つの表示（ステータスツールバーとレイアウト上のボタン）を見せるよりも、ひとつに絞った方がわかりやすくなります。

また、想定外のことが起きるかもしれません。例えば、レイアウト上に配置されている「新規○○レコード作成」ボタンに割り当てられているスクリプトの動作がレコード作成後にフィールド設定をする内容であっても、ステータスツールバーの「新規レコード」ボタンを押した場合は同様になりません。あるいは「レコード削除」ボタンのスクリプトがレコード削除の前に条件分岐をしていても、ステータスツールバーの「レコード削除」ボタンでは条件分岐をすることなくレコード削除を実行できてしまいます。

標準のコマンドは、カスタムメニューの設定でコントロールすることもできますが、ユーザ、レイアウト、メニューセット、スクリプトの組み合わせを設計し綿密な作業とテストが必要です。

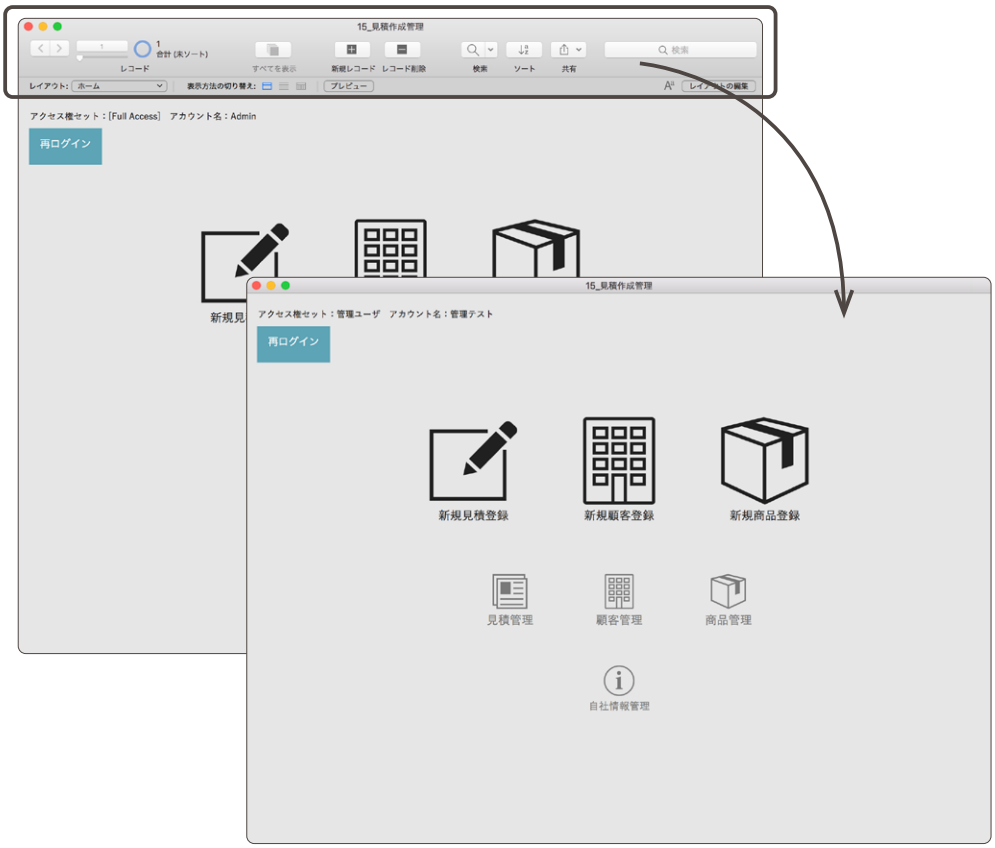


図 16-1　ステータスツールバーの表示と非表示

16.3 カスタムメニュー

（高度なツールが　**オン**　の場合に、カスタムメニューの作成が可能です（方法は 9 ページを参照））。

ステータスツールバーと同じように、どのようなメニューをユーザが実行しても良いかを考えます。ステータスツールバーだけを隠してもメニューから実行できてしまったら同じことです。



図 16-2 FileMaker Pro の標準のメニュー（macOS の場合）

標準のメニューは、上図のような並びです。このメニューやメニュー項目を加工することができます。

・標準のメニュー名、メニュー項目名を変更する
・メニュー項目を無効にする
・メニュー項目を削除する
・メニュー項目に、スクリプトを割り付ける
・メニュー項目のキーボードショートカットを変更する
・新しいメニュー、メニュー項目を作成する

カスタマイズしたメニューをセットとし、次のことに応じて変わるように設定することができます。

・ファイル
・レイアウト
・モード
・クライアントハードウェアの OS

あるいは以下も可能です（FileMaker WebDirect はカスタムメニューに対応していません）。

・スクリプトでメニューセットを変える

アクセス権セットごとに設定できるメニュー
カスタムメニューと別に、アクセス権セットの［利用できるメニューコマンド］の設定で標準メニューの中で利用できるコマンドをいくつかのパターンに限定することができます。

カスタムメニューの準備
カスタムメニューの作成、設定などの定義作業には、以下が必要です。

・高度なツールが　**オン**　の、FileMaker Pro
・［完全アクセス］アクセス権セット

演習ファイルを開いてください。次のアカウントでログインします。

・アカウント名　　Admin
・パスワード　　1111

（ここから、本書に付属している演習ファイルの「演習 16_1 から 2」フォルダの中の「16_ 見積作成管理 .fmp12」ファイルをベースに進めます）

16.3.1 「メニュー」とは

メニューについて改めて考えます。メニューは階層化されています。次の順です。

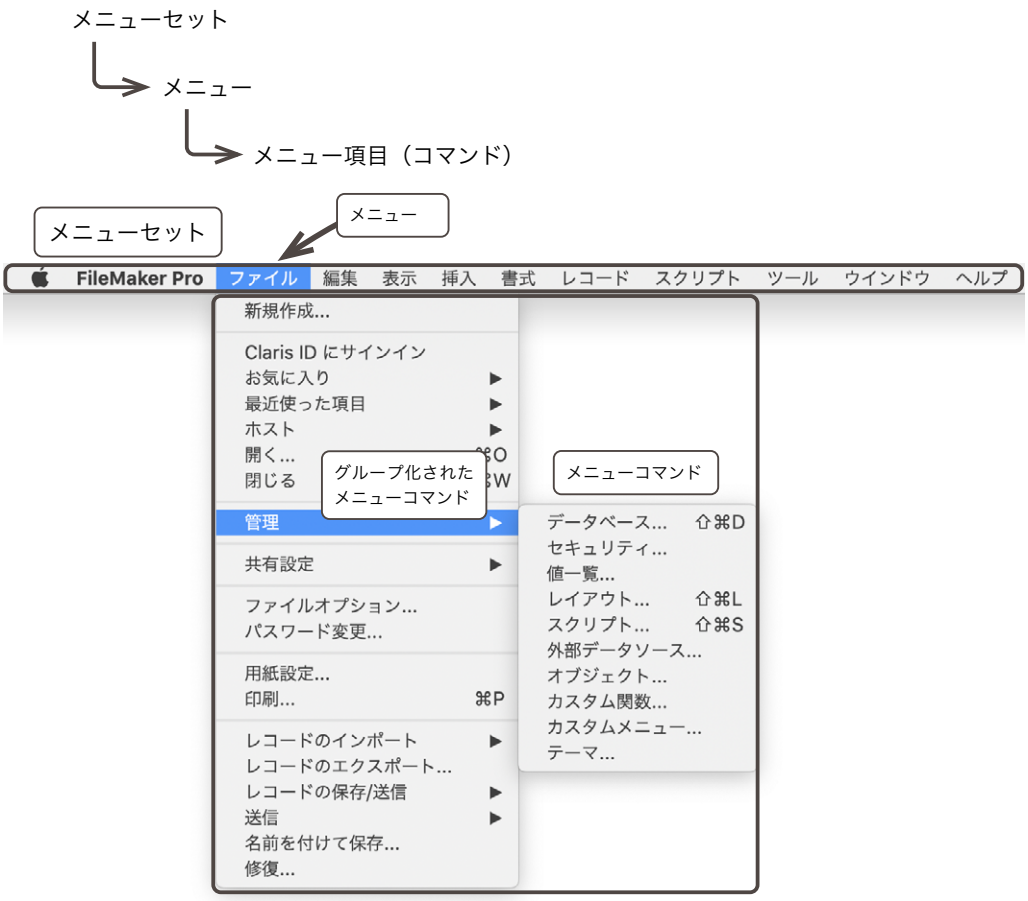


図 16-3 メニューの階層

macOS の場合、標準のメニューセットは、次のメニューで構成されています。

(1) FileMaker Pro などのアプリケーションとしてのメニュー
(2) ファイル
(3) 編集
(4) 表示
(5) 挿入
(6) 書式
(7) レコード → 検索モードの場合は、検索条件
(8) スクリプト
(9) ツール
(10) ウインドウ
(11) ヘルプ

16.3.2 カスタムメニューセットを切り替える

カスタムメニューセットを切り替える方法は

・ファイルのデフォルトメニューセットで設定する
・レイアウト設定で設定する
・メニューセットを［ツール］–［カスタムメニュー］メニューから切り替える
・［カスタムメニューのインストール］スクリプトステップ

で可能です。

16.3.3 カスタムメニューの作成

カスタムメニューは、［カスタムメニューの管理］ダイアログから次の作業で作成します。

(1) カスタムメニューを作成する
(2) カスタムメニューのオプションや、項目の編集、作成などのカスタマイズをする
(3) カスタムメニューセットを作成する
(4) カスタムメニューセットにカスタムメニューを割り当てる

演習ファイルに、新しいカスタムメニューセットをつくります。

「演習 16_1 から 2」フォルダ

演習 16-1　カスタムメニューセットをつくる：標準メニューセットのコピー

- **使用ファイル：16_ 見積作成管理 .fmp12**

(1)　カスタムメニューを作成します。
a.　［ファイル］–［管理］–［カスタムメニュー ...］を選択します。
b.　［カスタムメニューの管理］ダイアログが表示されます。
c.　「カスタムメニュー」タブをクリックします。

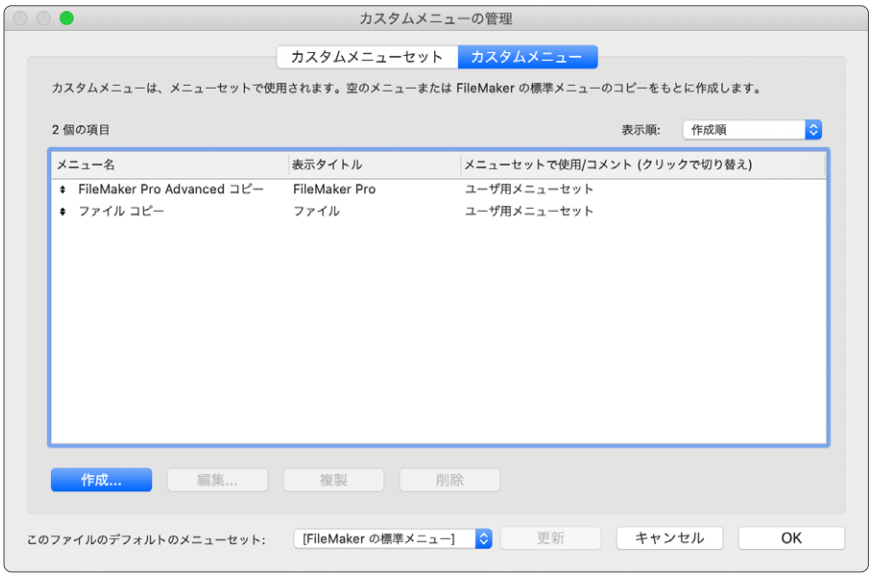


図 16-4［カスタムメニューの管理］ダイアログの「カスタムメニュー」タブ

d. ［作成 ...］をクリックします。
e. ［カスタムメニューの作成］ダイアログが表示されます。
f. ［FileMaker の標準メニューから作成する］が **オン** のまま「FileMaker Pro」をクリックして［OK］をクリックします。

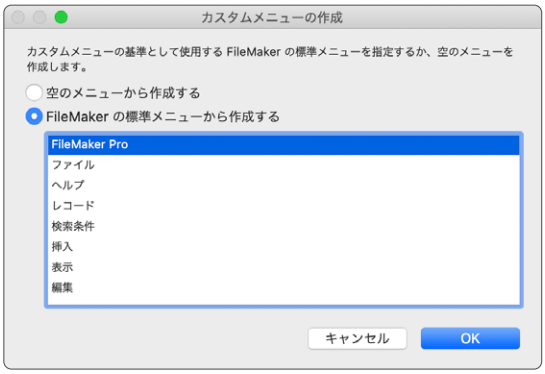


図 16-5［カスタムメニューの作成］ダイアログ

g. ［カスタムメニューの編集］ダイアログが表示されます。
カスタムメニュー名が「FileMaker Pro コピー」になっています。つまり標準のメニューがコピーされたものです。
h. カスタムメニュー名は変更せず［OK］をクリックしてダイアログを閉じます。

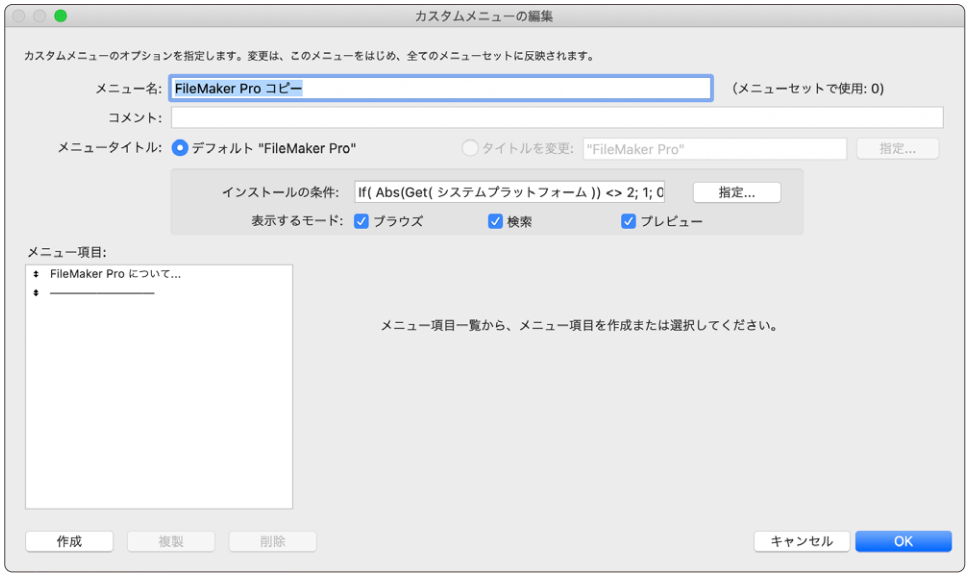


図 16-6 カスタムメニューを作成

i. そのまま他の FileMaker の標準メニューをカスタムメニューに作成します。
j. 作成すると［カスタムメニューの管理］ダイアログが次図のようになります。

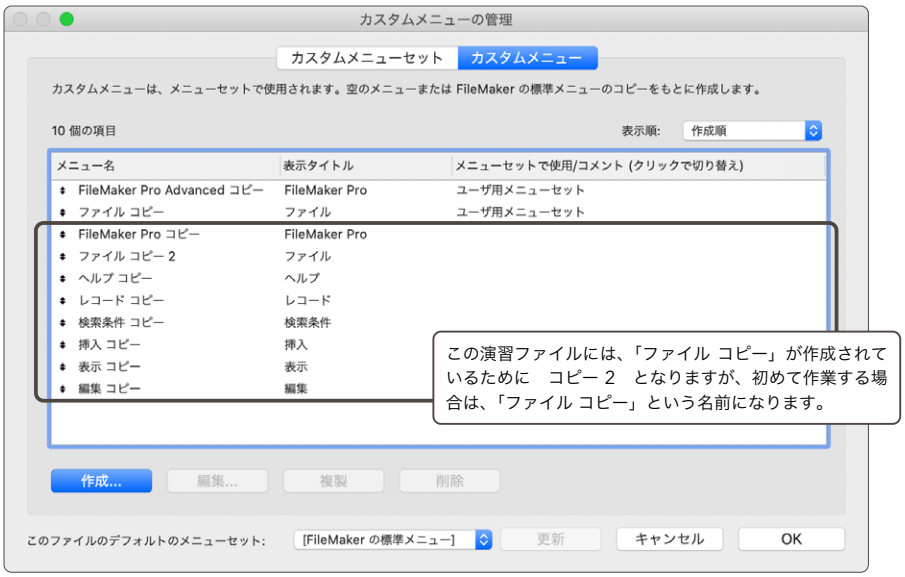


図 16-7 カスタムメニュー作成後

(2)　カスタムメニューが作成されました。

(3)　次にカスタムメニューセットを作成します。

a. ［カスタムメニューの管理］ダイアログで「カスタムメニューセット」タブをクリックします。
b. ［作成 ...］をクリックします。
c. ［カスタムメニューセットの編集］ダイアログが表示されます。
d. ［メニューセット名］を「ユーザ用メニュー（演習）」とします。

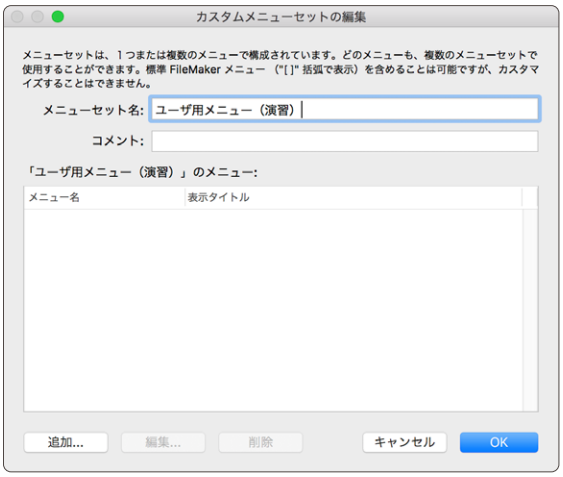


図 16-8［カスタムメニューセットの編集］ダイアログ

(4)　作成したカスタムメニューを「ユーザ用メニュー（演習）」メニューセットに定義します。

a. ［追加 ...］をクリックします。
b. ［メニューの選択］ダイアログが表示されます。

c. 右のスクロールバーで一番下までスクロールすると、作成したカスタムメニューが表示されますので、作成したカスタムメニューをすべて指定します。

d. 「FileMaker Pro コピー」をクリックして、[Shift] キーを押しながら「編集コピー」をクリックすると範囲で指定できます。



図 16-9［メニューの選択］ダイアログで作成したカスタムメニューを選択

e. ［選択］をクリックします。選択したカスタムメニューが追加されます。



図 16-10 カスタムメニューがカスタムメニューセットに定義された

f. ［OK］をクリックしてダイアログを閉じます。

(5) 「ユーザ用メニュー（演習）」メニューセットが出来上がりました。



図 16-11「ユーザ用メニュー（演習）」メニューセットが完成

(6)　もう一度［OK］をクリックして［カスタムメニューの管理］ダイアログを閉じます。

(7)　メニューを切り替えて表示を確認します。

a. ［ツール］–［カスタムメニュー］を選択すると、現在は［FileMaker の標準メニュー］です。
b. これを、［ユーザ用メニュー（演習）］を選択します。
c. メニューが切り替わります。
d. 「ツール」などのメニュー項目が減ります。
この演習で作成していない「書式」などのメニューが非表示となります。
e. 再度、［FileMaker の標準メニュー］へ戻します。

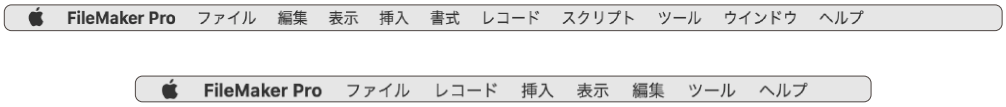


図 16-12 メニューセットによる違い

演習ここまで

演習で作成したカスタムメニューセットは、標準メニューとほぼ変わりませんが、次の項目は［カスタムメニューの作成］ダイアログの［FileMaker の標準メニューから作成する］では表示されません。

・書式
・スクリプト
・ウインドウ

※高度なツールが　オン　の状態では常に「ツール」メニューが表示されます。

16.3.4 メニューとメニュー項目

カスタムメニューのメニューとメニュー項目は、複数のメニューセットで共有することができます。

新しいカスタムメニューセットを作成しても、すべてを新しく作成しなければならないわけではありません。例えば、新しいカスタムメニューセットに標準のレコードメニューと同じ内容のレコードメニューを定義可能です。

カスタムメニューで定義する内容は、メニューセットごとに取り扱うことが可能です。

カスタム App ファイルに初めてカスタムメニューセットを作成する場合には、標準のメニューを利用する場合であっても、カスタムメニューを作成する必要があります。そしてこの標準メニューのコピーは、他のカスタムメニューセットにも割り当てできます。

演習で作成したカスタムメニューセットの内容を確認します。

手順：

(1)　［カスタムメニューの管理］ダイアログを表示します。
(2)　「ユーザ用メニュー（演習）」メニューセットをクリックます。
(3)　［編集 ...］をクリックします。
(4)　［カスタムメニューセットの編集］ダイアログが表示されます。

メニューの順番

演習で作成した「ユーザ用メニュー（演習）」メニューセットは、特にメニューの順番を気にせずに作業しました。よく見てみると、「FileMaker の標準メニュー」と違います。

ユーザは、複数のカスタム App を利用しているかもしれませんので、標準メニューを同じ順番でメニューが表示される方が使いやすいはずです。メニューの順番はメニュー名をドラッグすることで自由に動かすことができます。



図 16-13 順番を標準メニューに合わせた例

表示順を動かせないメニュー
「FileMaker Pro コピー」と「ヘルプコピー」には、左に錠マークがついています。この 2 つのメニューは順番を変えることはできません。

メニューの中身

演習 16-1 で作成した「FileMaker Pro コピー」をクリックし [編集 ...] をクリックすると、メニュー内容を確認できます。

このダイアログの内容を確認します。

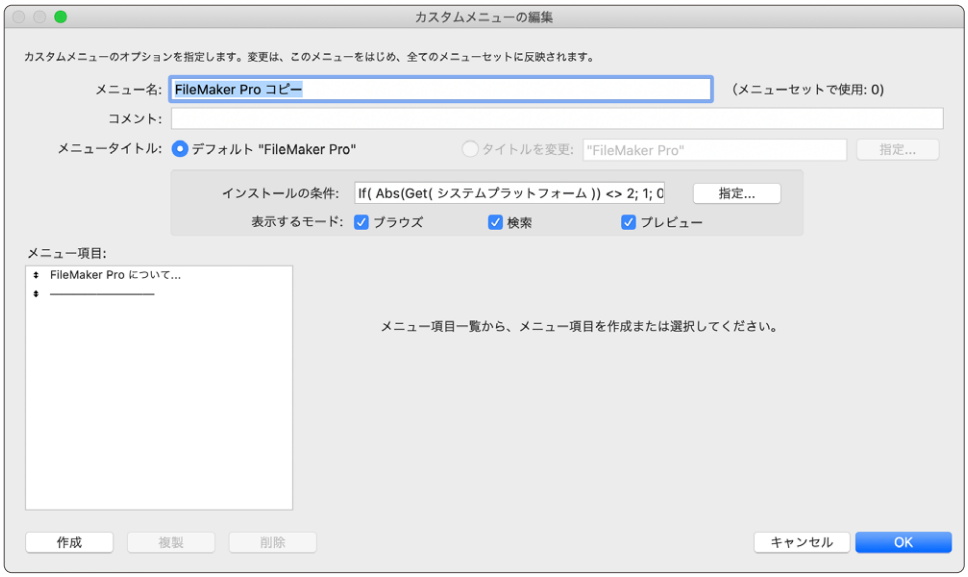


図 16-14「FileMaker Pro コピー」の中身

「カスタムメニュー名：」「コメント：」は管理用の項目です。

メニュータイトル

この名前が実際にユーザに表示されるものです。これを変更できます。計算式も利用できます。$$ からはじまるグローバル変数で動的な変更も可能です。

インストールの条件
このメニューでは［インストールの条件］に計算式が入っています。

```
If( Abs ( Get ( システムプラットフォーム ) ) <> 2 ; 1 ; 0 )
```

Get(システムプラットフォーム) 関数の結果が 2 あるいは -2 以外なら「1（真 = True)」となって、メニューがインストール（表示）されるという条件分岐です。

`Get( システムプラットフォーム )` 関数

→ 現在のプラットフォームを評価した値を返します。
- 1 → 現在のプラットフォームが Intel ベースの macOS の場合
- -2 → 現在のプラットフォームが Windows の場合
- 3 → 現在のプラットフォームが iOS または iPadOS の場合
- 4 → プラットフォームが FileMaker WebDirect の場合
- 5 → プラットフォームが Linux の場合

つまり、次のような計算式でクライアントハードウェアの OS のプラットフォームごとにメニューをインストールして表示をコントロールできるということです。

Windows のみにインストールするには
```
If ( Abs ( Get ( システムプラットフォーム ) ) = 2 ; 1 ; 0 )
```

macOS のみにインストールするには
```
If ( Abs ( Get ( システムプラットフォーム ) ) = 1 ; 1 ; 0 )
```

常にインストールされるようにするには「1」を指定します。

［キャンセル］をクリックして、ダイアログを閉じます。

別のメニューの内容を確認します。演習 16-1 で作成した「ファイルコピー 2」をクリックして［編集 ...］をクリックします。

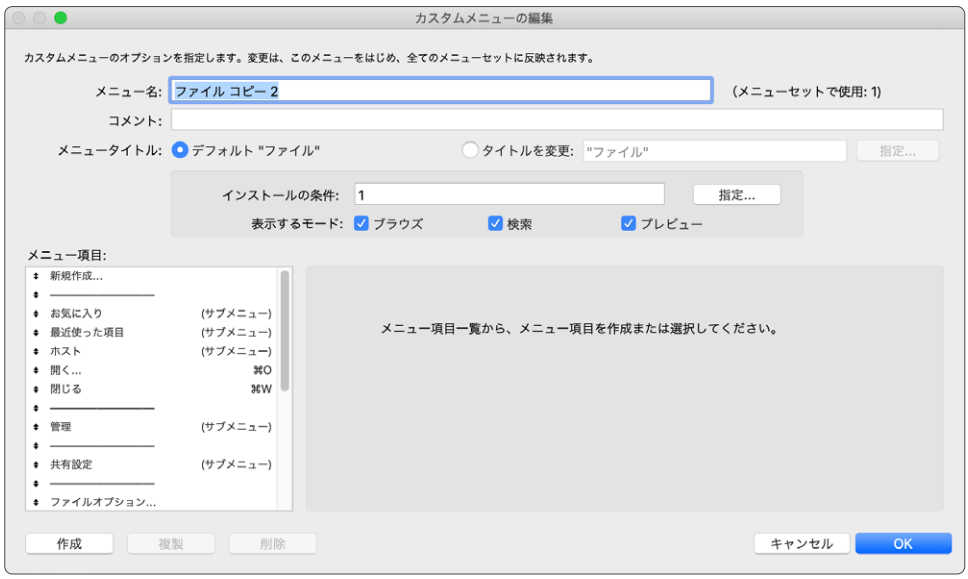


図 16-15「ファイルコピー」の中身

いつも使っているメニュー項目ばかりです。「管理」メニュー項目の右に「(サブメニュー)」と表示されています。さらに下の階層があることを表しています。

「開く ...」メニュー項目をクリックします。次図のようになります。

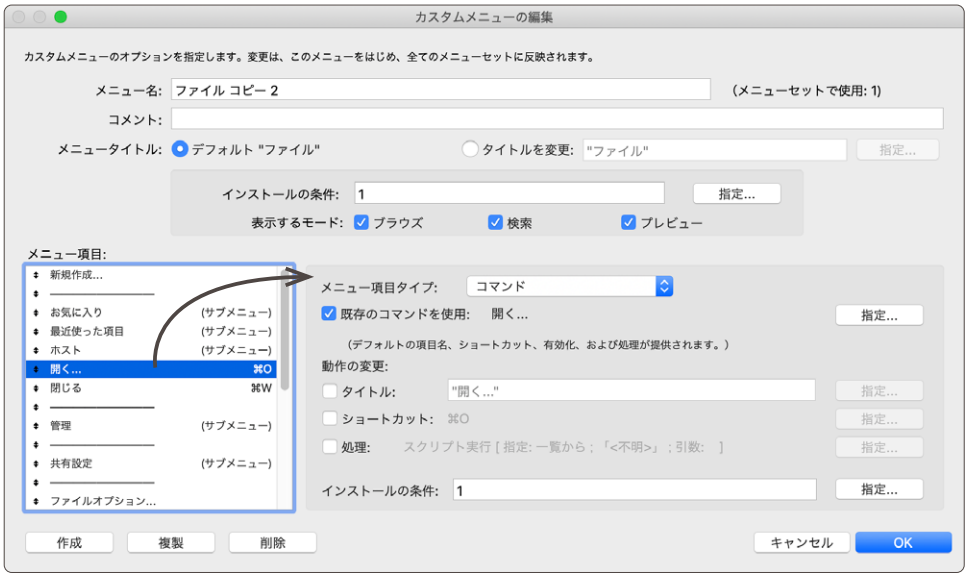


図 16-16「開く ...」の設定

細かく設定できることがわかります。メニューのインストール条件とは別に、メニュー項目にもインストール条件があります。

[既存のコマンドを使用]

「既存のコマンドを使用：」の右にある［指定 ...］から FileMaker Pro に用意されているコマンドを指定して別のコマンドを実行するようにできます。

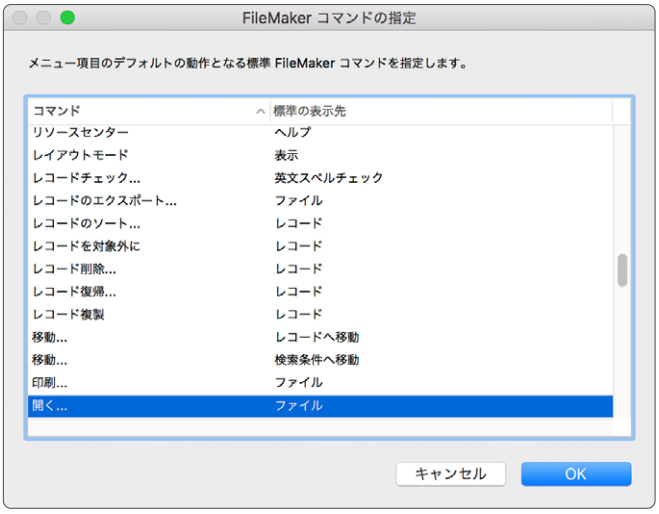


図 16-17 コマンドの指定

［既存のコマンドを使用］でコマンドを変更すると、標準のコマンドが適用されるすべての場所（ステータスツールバーや右クリックのショートカットメニュー、スクリプトステップなど）にもそれが反映されます。この設定は充分にテストしてください。

[処理]
「処理：」のチェックを　**オン**　にしてスクリプトを指定して標準のコマンドとは違う動作にすることもできます。

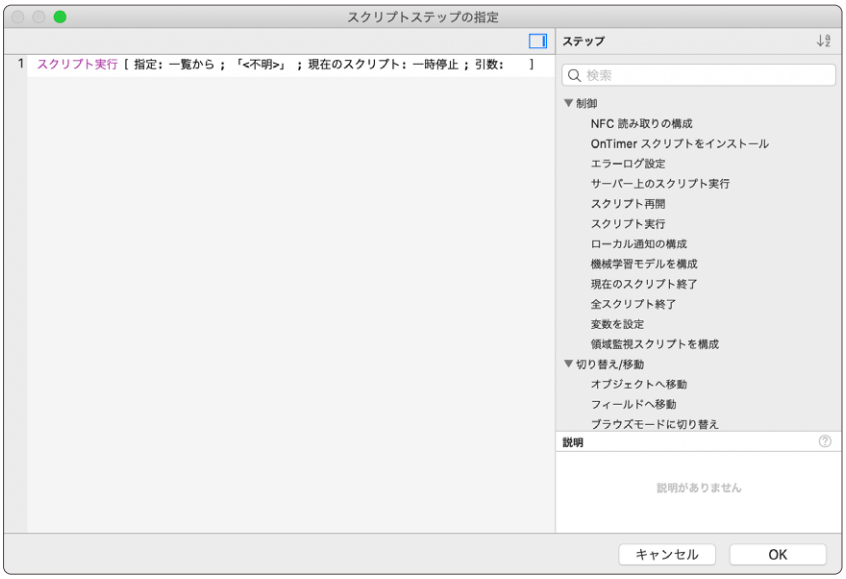


図 16-18 スクリプトの指定

スクリプトを指定すると標準のコマンドがスクリプトで上書きされます。標準のコマンドが適用されるすべての場所にもそれが反映されます。この設定もやはり充分にテストしてください。

カスタムメニューでは多様な設定が可能です。うまく利用するとユーザが操作しやすく、誤操作を防ぐことにつながります。例えば、

・全レコード削除
・ウインドウを閉じる
・フィールド内容の全置換

の操作をできないようにすることです。しかし、編集や印刷など基本的なメニューを非表示にすることは、ユーザが不便に感じるのでお勧めしません。

［キャンセル］をクリックして、ダイアログを閉じます。

テストする
すべてのアクセス権セット、すべてのクライアントのハードウェア OS で確認します。

ファイルごとに作成する
カスタムメニューは、カスタム App ファイルごとに定義します。他のファイルのものを利用することはできません。他のファイルから定義をコピーと貼り付け（ペースト）することができます。

FileMaker Go
FileMaker Pro と違う点があります。

・カスタムメニューから削除した内容は非表示ではなく、無効（選べない）になります。
・FileMaker Go には、[クイック検索]や[レコード終了]など、カスタムメニューのメニュー項目で置き換えられないものがあります。
・メニュー項目名を変更する場合「&（アンパサンド）」は表示されません。
・FileMaker Go 上で利用できないカスタムメニューやメニューコマンドは無視されます。

カスタムメニューセットに書式メニューを追加します。

演習 16-2　カスタムメニューをつくる：メニューの追加

- **使用ファイル：16_ 見積作成管理 .fmp12**

(1)　［ファイル］–［管理］–［カスタムメニュー ...］を選択します。

(2)　［カスタムメニューの管理］ダイアログが表示されます。

(3)　「ユーザ用メニュー（演習）」メニューセットをクリックして［編集 ...］をクリックします。

(4)　［カスタムメニューセットの編集］ダイアログが表示されます。

(5)　メニューを追加します。
a.　［追加 ...］をクリックします。
b.　［メニューの選択］ダイアログが表示されます。
c.　［書式］を選択して［選択］をクリックします。
d.　選択したカスタムメニューが追加されます。

(6)　演習 16-1 で作成したメニューを、［FileMaker の標準メニュー］と同じ並びにします。以下です。錠マークのメニューの位置は変更できません。

1)　ファイルコピー 2
2)　編集 コピー
3)　表示 コピー
4)　挿入 コピー
5)　［書式］
6)　レコード コピー
7)　検索条件 コピー
8)　ヘルプ コピー

(7)　［OK］をクリックしてダイアログを閉じます。

(8)　もう一度［OK］をクリックして［カスタムメニューの管理］ダイアログを閉じます。

(9)　切り替えて確認します。

（10） メニューを切り替えて表示を確認します。

a. ［ツール］–［カスタムメニュー］を選択すると、現在は［FileMaker の標準メニュー］です。
b. ［ユーザ用メニュー（演習）］を選択します。
c. メニューが切り替わります。
d. 「スクリプト」などのメニュー項目が減ります。
e. 再度、［FileMaker の標準メニュー］へ戻します。



図 16-19 並び替えて表示を確認

演習ここまで

開発時の作業効率のために

［メニューセットのインストール］スクリプトステップで［FileMaker の標準メニュー］を選択し、「ファイルのデフォルトとして使用」のチェックを **オン** にしておくと「FileMaker の標準メニュー」以外がファイルのデフォルトとして設定されていても［完全アクセス］アクセス権セットの付加されたアカウントでログインしてファイルを開いた後に常に標準のメニューを利用できるので開発時にカスタムメニューを毎回切り替えなければならないということがありません。

16.3.5 カスタムメニューを他のファイルにコピー

カスタムメニューは、他のファイルにコピーすることができます。次の手順でコピーします。

（1）カスタムメニューを、コピー元ファイルから、貼り付け（ペースト）する。
（2）カスタムメニューセットを、コピー元ファイルから、貼り付け（ペースト）する。

つまり、メニューセットで参照されているメニューはコピーされません。メニューセットが参照しているメニューを先に貼り付けします。

やってみよう

［ファイル］–［新規作成...］メニューで新しいファイルを作成して「ユーザ用メニュー（演習）」のコピーと貼り付けしてみましょう。

演習ファイルを閉じて、次のファイルを開いてログインしてください。

（ここから、本書に付属している演習ファイルの「演習 16_3 から 4」フォルダの中の「16_ 見積作成管理 .fmp12」ファイルをベースに進めます）

・アカウント名　　Admin
・パスワード　　1111

なお、2 つのスクリプトが演習ファイルに作成されています。

・アカウントの新規登録　→　「アカウント登録 _ 演習用」スクリプト
・アカウントの停止　　→　「アカウント停止」スクリプト

16.4　アカウントの管理

アカウントを管理する作業は［セキュリティの管理］ダイアログを使用します。このダイアログで作業できることはとても重要な部分です。

そして、基本的な設定を完了後、アカウント管理用のスクリプトを作成して運用時に利用することをお勧めします。間違ってアクセス権セットを削除してしまうなどのミスが起きないようにするためです。

・アカウントの新規登録
・アカウントの停止（無効化）

これらをスクリプトで実行できるようにすると、例えば、運用開始直前に改行区切りテキストと Loop を利用して連続したアカウントの登録や停止を行うことができます。

あるいは、良くあることとして、

・パスワードを忘れたとき

があります。これもスクリプトを作成して対応することができます。他には、

・パスワードの変更
・アカウントの有効化
・アカウントの削除

があります。

アカウントの新規登録のスクリプトを作成します。

「演習 16_3 から 4」フォルダ

演習 16-3　アカウントの登録

- **使用ファイル：16_ 見積作成管理 .fmp12**

(1)　「見積作成管理 .fmp12」を開きます（Admin ／ 1111 でログインします）。

(2)　［スクリプトワークスペース］を表示し「アカウント登録 _ 演習用」スクリプトを次の内容に書き換えてください。必要なオプションのみ記述します。手順は省略します。

```
変数を設定 [ $アカウント名 ; "" ]

Loop
  Exit Loop If [ ( Get(最終メッセージ選択) = 1 and not
    IsEmpty ( $アカウント名 )) ]

  カスタムダイアログを表示 [ "アカウント登録"; "アカウント登録をおこないますか？「はい」の場合はアカウント名を入力してください。" ;
   $アカウント名 ]

  If [ Get(最終メッセージ選択)=2 ]
    現在のスクリプト終了 [ ]
  End If
End Loop

エラー処理 [ オン ]
アカウントを追加 [ $アカウント名; パスワード: 1111 ]

If [ Get(最終エラー) ]

  カスタムダイアログを表示 [ "エラー" ; "アカウントの登録でエラーが発生しました。"]

Else

  カスタムダイアログを表示 [  "作業完了" ; "アカウントの登録が完了しました。" ]

End If
エラー処理 [ オフ ]
```

＜スクリプトオプション　1 回目の［カスタムダイアログを表示］スクリプトステップ＞
「一般」タブ
・［タイトル：］は　　「アカウント登録」
・［メッセージ：］は
　「アカウント登録をおこないますか？「はい」の場合はアカウント名を入力してください。」

ボタンは
・［デフォルトボタン］は　「はい」「データを確定」は　**オン**　そのまま
・［ボタン 2］は　　「いいえ」「データを確定」は　**オフ**　そのまま
・［ボタン 3］は　　（設定なし）で　そのまま

「入力フィールド」タブ

・［入力フィールド 1 を表示：］の［指定 ...］ で変数欄に　$ アカウント名
・［ラベル：］に「作業するアカウント名」
・他は設定しません。

＜スクリプトオプション　2 回目の［カスタムダイアログを表示］スクリプトステップ＞
「一般」タブ

・［タイトル：］は　　「エラー」
・［メッセージ：］は　　「アカウントの登録でエラーが発生しました。」

ボタンは

・［デフォルトボタン］は　「OK」「データを確定」は　**オン**　そのまま
・［ボタン 2］は　　（設定なし）で　そのまま
・［ボタン 3］は　　（設定なし）で　そのまま

「入力フィールド」タブは設定しません。

＜スクリプトオプション　3 回目の［カスタムダイアログを表示］スクリプトステップ＞
「一般」タブ

・［タイトル：］は　　「作業完了」
・［メッセージ：］は　　「アカウントの登録が完了しました。」

ボタンは

・［デフォルトボタン］は　「OK」「データを確定」は　**オン**　そのまま
・［ボタン 2］は　　（設定なし）で　そのまま
・［ボタン 3］は　　（設定なし）で　そのまま

＜スクリプトオプション　［アカウントを追加］スクリプトステップ＞

・［アカウント名：］は　　「$ アカウント名」
・［パスワード：］は　　「1111」
・［アクセス権セット］は　「営業ユーザ」を選択
・「ユーザは次回サインイン時にパスワードの変更が必要」は　**オン**

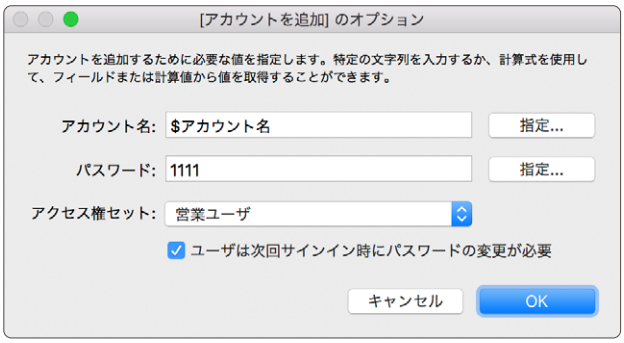


図 16-20［アカウントを追加］のオプションダイアログ

(3)　スクリプトを保存します。

(4)　[完全アクセス] アクセス権セットで実行する設定をします。

a.　スクリプトパネルでアカウント登録 _ 演習用」スクリプトをクリックします。

b.　右クリックで表示されるショートカットメニューから、「完全アクセス権を付与」を選択します。

c.　付与されると、スクリプトパネルでスクリプト名の右に人マークが表示されます。



図 16-21 ショートカットメニュー

演習ここまで

このスクリプトでは、簡易なエラー処理だけを行いました。実際には、もう少し考えなければいけません。例えば、同じアカウント名がすでに登録されているかもしれません。

アカウント登録時のパスワードはスクリプトに直接入力していますが、ランダムな文字列を登録しユーザ本人にのみ伝えて、さらに初回ログイン時にユーザが変更する方法の方がよりセキュリティが高くなります。

「アカウント停止」スクリプトも、演習したスクリプトの内容と似た内容です。[アカウントを追加] スクリプトステップの代わりに、次のものを使っています。

[アカウントの有効化] スクリプトステップ

アカウントの登録の動作テストをします。

演習 16-4　アカウントの登録テスト

- **使用ファイル：16_ 見積作成管理 .fmp12**

(1)　「ホーム」レイアウトで、「自社情報管理」ボタンをクリックします。

(2)　「自社情報管理」レイアウトが表示されます。

(3)　右上にアカウントについてのボタンが配置されています。

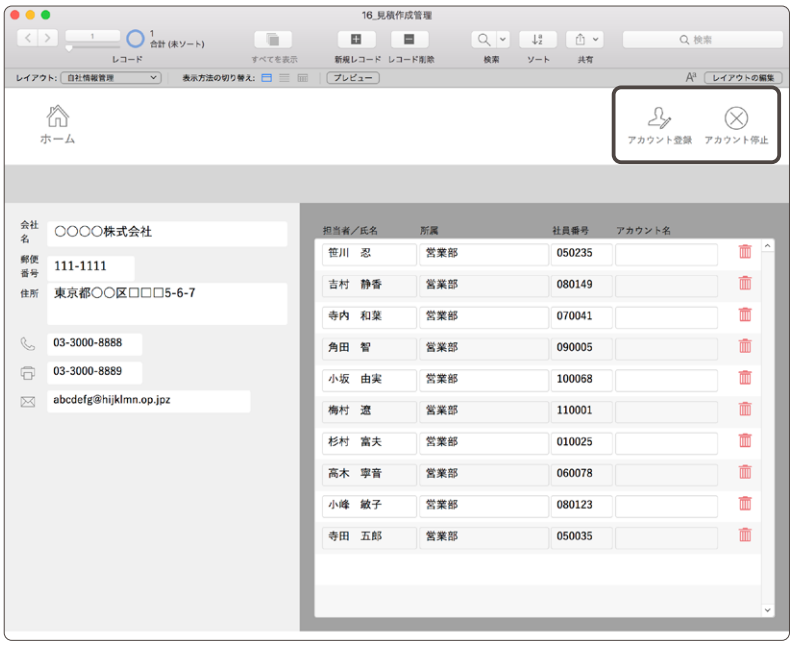


図 16-22「自社情報管理」レイアウト

(4) 「アカウント登録」ボタンをクリックして次のアカウント名を登録します。

・アカウント名 「演習テスト」

(5) 登録が成功したら、[セキュリティの管理] ダイアログで登録を確認します。

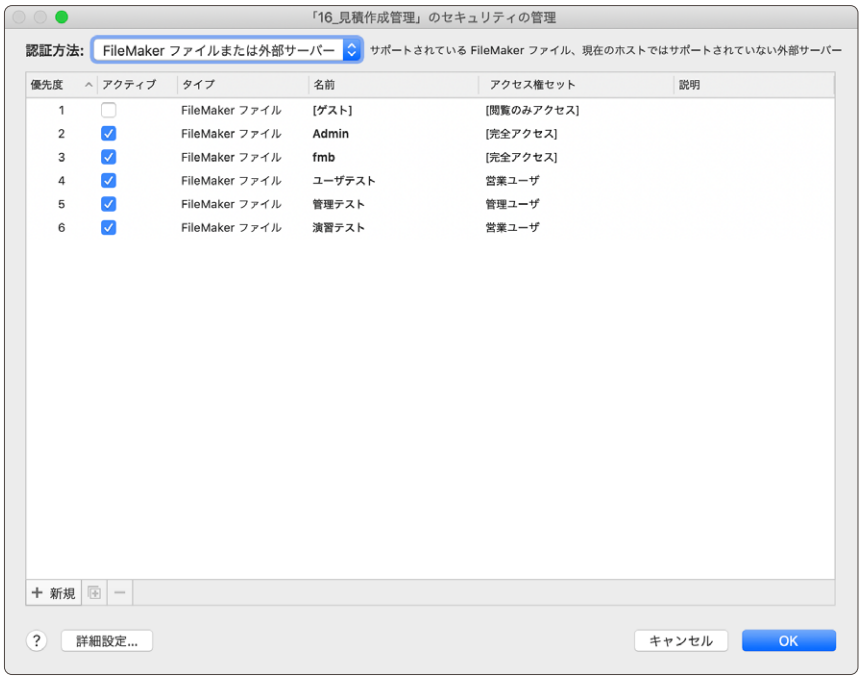


図 16-23 登録を確認

(6)　「ホーム」レイアウトに戻ります。

(7)　「再ログイン」ボタンをクリックしてログインします。
・アカウント名　「演習テスト」
・パスワードは「1111」です

(8)　［パスワード変更］ダイアログが表示されますので「2222」などわかりやすいものに変更します。

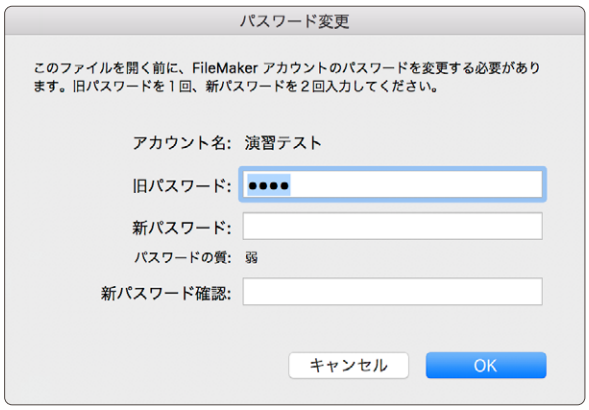


図 16-24［パスワード変更］ダイアログ

(9)　ステータスツールバー、カスタムメニューの表示を確認してください。
（「起動処理 _ 演習」スクリプトでは［完全アクセス］アクセス権セットのアカウント以外の場合のカスタムメニューを「ユーザ用メニュー（演習）」に変更しておきました）

演習ここまで

やってみよう

同じように「アカウント停止」ボタンをクリックしてアカウントが非アクティブになっているか確認してみましょう。

演習ファイルを閉じて、次のファイルを開きます（以下のアカウントでログインします）。

・アカウント名　　Admin
・パスワード　　1111

なお、スクリプトが演習ファイルに作成されています。

・「レコード削除」スクリプト

このスクリプトが割り当てられたボタンも各レイアウトに用意しました。

（ここから、本書に付属している演習ファイルの「演習 16_5」フォルダの中の「16_ 見積作成管理 .fmp12」ファイルをベースに進めます）

16.5 レコード削除の注意

カスタム App ではレコードが作成されて蓄積されるだけではありません。削除もできます。

削除について注意したい点があります。「**削除ルール**」です。［リレーションシップ編集］ダイアログの次の設定です。

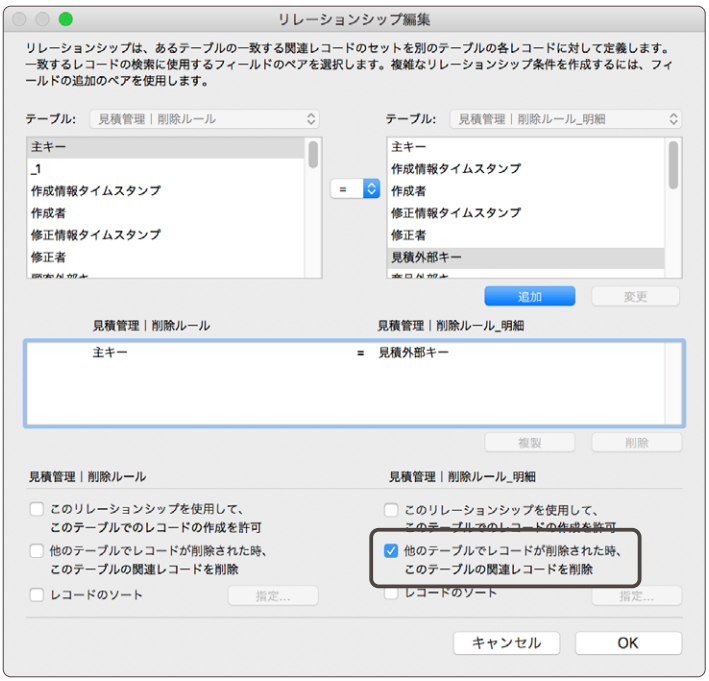


図 16-25［リレーションシップ編集］ダイアログ

［他のテーブルでレコードが削除された時、このテーブルの関連レコードの削除を許可］
テーブルのレコードが削除されたとき、その参照先のテーブルの関連レコードが削除されます。

この設定は取り扱い注意です。削除を実行するレイアウトに設定した関連レコードが表示されていなくても削除されます。

演習ファイルに設定してありますので、［データベースの管理］ダイアログを開いて確認します。リレーションシップグラフに専用の TOG を作成しました（次図を参照）。

・「見積管理｜削除ルール」TO と「見積管理｜削除ルール _ 明細」TO

注意が必要なのは、この削除ルールは設定した TO が指定されているレイアウトだけではなく他のレイアウト（同じ実体テーブルを元にした TO が指定されているもの）でも同様の削除となる点です。



図 16-26 削除ルール用の TOG

なお、複数の関連で設定を行うと、あるテーブルのレコードを削除したときに他の関連テーブルの関連レコードも削除されます。例えば、次のような設定がされているとします。

・「見積管理」TO と「見積管理 _ 明細」TO の関連
（見積レコードを削除したら関連の明細レコードも削除）
・「見積管理 _ 明細」TO と「見積管理 _ 明細 _ 商品」TO
（見積明細レコードを削除したら関連の商品レコードも削除）

このとき、見積レコードを削除すると、関連の明細レコード、関連の商品レコードも削除されます。**「連鎖削除」**と呼びます。

削除の設定は、専用の TOG を作成し、事前に十分に検討することをお勧めします。

演習で削除を試してみましょう。

「演習 16_5」フォルダ

演習 16-5 削除

- **使用ファイル：16_ 見積作成管理 .fmp12**

(1) 「ホーム」レイアウトで「新規見積登録」ボタンをクリックします（カスタムダイアログでは［OK］をクリックします）。

(2) 新しい見積レコードが作成されます。レコード数を覚えてください。201 件のはずです。

(3) 箱アイコンのボタンをクリックして商品を何か選択して明細行を 2 行入力します。

(4)　「z_ 見積明細」レイアウトに切り替えて、レコード数を覚えてください。
明細行が 2 行であれば、演習ファイルでは 977 件のはずです。

(5)　「見積詳細」レイアウトに戻ってゴミ箱アイコンボタンをクリックして、先ほど作成した見積レコードを削除します。

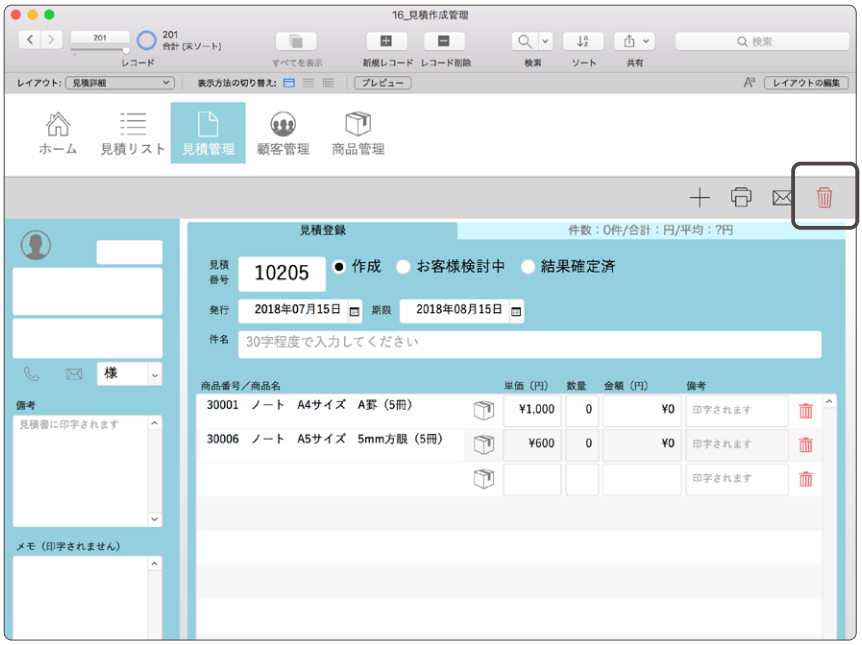


図 16-27 ゴミ箱アイコンボタン

(6)　レコード数を確認します。

- 見積レコードは 200 件
- 「z_ 見積明細」レイアウトでは 975 件

(7)　リレーションシップグラフの TOG で設定した削除ルールで次のことが起こりました。

「見積情報」レコードを削除すると、関連の「見積明細」レコードも削除される

(8)　もう一度「見積詳細」レイアウトで右上の「+」ボタンをクリックして見積レコードを作成し、明細行を 2 行入力します。同じように件数を確認してください。

a. 「z_ 見積明細」レイアウトに切り替えてます。
b. 一番最後のレコードをアクティブにします。
c. ステータスツールバーの「削除」ボタンをクリックしてレコードを削除します。
d. 「見積詳細」レイアウトに切り替えます
e. 見積のレコードは削除されていません。
f. しかし明細行が 1 レコード削除されています。

演習ここまで

これが削除ルールです。設定後は思わぬレコード削除に注意してください。

16.6 排他

1 つのカスタム App を複数のユーザが使用すると、ごくまれに 1 つのレコードに 2 人以上のユーザがアクセスして次図のようなメッセージが表示されることがあります。



図 16-28 誰かが編集中のレコードを編集しようとしたときのエラー

これは「**排他制御**」により、一度に 2 人以上のユーザが同一のレコードを編集することができないように「**ロック**」されるためです。全置換や照合インポートでも同じように機能します。

レコードは、別のユーザが編集中でも表示は可能です。しかし、そのレコードを他のユーザが編集しようとするとエラーになります。先に編集しているユーザによって開かれてロックされているからです。つまり、普段の作業やスクリプト作成時に、全置換やインポートを行う処理が含まれる場合に注意する必要があります。

ユーザによるロックは、フィールドにカーソルを入れて編集を開始したタイミングで起こります。スクリプト中でレコードをロックするためには、以下のスクリプトステップが用意されています。

レコード／検索条件を開く

エラー処理を行い、エラーコードが 200 であれば、そのレコードはロックされています。なお、誰かが既にロックしている場合、強制解除はできません。

処理後に、レコードのロックを解除するスクリプトステップは以下です。

レコード／検索条件確定

記録する
スクリプトで複数レコードを変更する処理中に、ロックされていたレコードの「主キー」などを記録する仕組み（どこかに保存するなど）を加えて、処理後に確認できるようにすれば、処理漏れとなったレコードの検索や再処理に役立ちます。

全置換やインポートの場合、どのレコードがロックされていたかはわかりません。複数のレコードを変更する場合に 1 件ずつ確実に処理する方法として Loop 処理でレコードがロックされていないか確認するなどの工夫が可能です。

また、照合インポートを行いたい場合にもレコードのロックについての注意が必要です。時間を調整して確実な実行を考えます（なるべく誰も使っていない時間帯など）。

実行する時間の調整

大量のレコードを一度に処理する場面は、あまり多くありません。例えば、

- データのメンテナンス（新しいフィールドを作成したために既存のレコードに初期値を格納したい、など）
- 定期的に必要な処理（決まった期日までになにも起きなかったレコードに一律に同じ値を入力したい、など）

このような場合、

- 一時的にカスタム App を運用停止にして作業する
- FileMaker Cloud や FileMaker Server でスクリプトの実行をスケジュールする

など、排他を気にせずに確実な処理を行うように調整します。

16.7 起動と終了の処理

最後にカスタム App の「起動」と「終了」について考えます。

16.7.1 起動：作業開始準備

ユーザがカスタム App で作業を開始できるように、準備をするのが「**起動処理**」です。

ログイン
登録のあるアカウントだけがカスタム App を使えるようにします。

最初に表示されるレイアウト
作業を開始するレイアウトが表示されるようにします。

「見積作成管理 .fmp12」では、これまでに、クライアントデバイスで条件分岐するように「起動処理 _ 演習」スクリプトで作業を行いました。また［完全アクセス］以外に次の 2 種類のアクセス権セットが作成されています。これらについて起動処理で行うべきことがあります。

・管理ユーザアクセス権セット
・営業ユーザアクセス権セット

ステータスツールバー
ステータスツールバーの表示有無を決定します。やはり「起動処理 _ 演習」スクリプトで作業を行いました。

メニューセット
カスタムメニューの使用有無を決定します。同じく「起動処理 _ 演習」スクリプトで作業を行いました。

これらは、アクセス権セットの名前で条件分岐します。

・管理ユーザアクセス権セット　→　ステータスツールバー非表示／「ユーザ用」カスタムメニューセット
・営業ユーザアクセス権セット　→　ステータスツールバー非表示／「ユーザ用」カスタムメニューセット
・[完全アクセス］　→　ステータスツールバー表示／標準のカスタムメニューセット

演習ファイルには「開発者かどうか」カスタム関数が作成されていますので、そちらを利用しています。

・[完全アクセス］アクセス権セットのアカウントかどうか

という条件は、使い勝手が良い条件です。

また、別の観点があります。複数のユーザが共有するカスタム App の場合、ホストへアップロードする前に行っておきたいことがあります。

グローバル格納フィールドの初期化
グローバル格納フィールドは、様々に利用されています。起動時に初期化していますが終了時には行っていません。このままだと、開発者がローカル環境で手入力あるいはスクリプトで入力していた値がグローバル格納されたまま、FileMaker Cloud や FileMaker Server で展開されることになります。

もしも起動時に初期化されないエラーが起きた場合に、開発者のための値をユーザが利用できてしまうかもしれません。

「グローバル格納値クリア」スクリプトを作成しておきました。「ユーティリティ」テーブルに作成されているグローバル格納フィールドを初期化するものです。こうして 1 つのスクリプトでまとめておこなうと他のスクリプトからサブスクリプトとして利用できるので便利です。

他にもありますが、これらのことは起動処理で行っておきましょう。

```
OnFirstWindowOpen
```

［ファイルオプション］の「スクリプトトリガ」タブで「OnFirstWindowOpen」トリガに設定します。

16.7.2 終了

ユーザがカスタム App で作業を終了するときは、ウインドウを閉じてファイルが終了できれば良いはずです。しかし、いくつか行っておきたいことがあります。

ただし、ローカル環境でファイルを閉じたときだけです。ローカル環境では、ウインドウの位置や表示していたレイアウト、対象レコード、グローバル格納の値を覚えています。これらを、あるべき内容にしておかなければなりません。

全レイアウトで全レコードを表示する
対象レコードを覚えたままにしないために行います。

グローバル格納値のクリア
グローバル格納フィールドのデータをクリアします。

ウインドウの位置を整える
ウインドウの位置も覚えたままにせず揃えるためです。

ローカル環境で開いているか見分ける
次の計算式で条件分岐しています。Get (ホスト IP アドレス) は、ファイルがホストされていない場合は、「空欄」を返します。それを IsEmpty 関数で確認しています。

```
IsEmpty ( Get ( ホストIPアドレス )
```

```
OnLastWindowClose
```

［ファイルオプション］の「スクリプトトリガ」タブで「OnLastWindowClose」トリガに設定します。

「終了処理」スクリプトは、次の内容です。

```
If [ IsEmpty ( Get ( ホストIPアドレス ) ) ]

    スクリプト実行 [ 「全レイアウトで全レコード」 ]

    レイアウト切り替え [ 「起動画面」 ( z_ユーティリティ ) ]
    レコード / 検索条件確定 [ 入力値の制限を無視 ; ダイアログなし ]
    ウインドウの調整 [ 収まるようにサイズ変更 ]

    スクリプト実行 [ 「グローバル格納値クリア」 ]

    ウインドウの移動 / サイズ変更 [ 現在のウインドウ ; 上 : 5 ; 左 : 5 ]
End If
```

16.7.3 その他

レイアウトモードの定規や書式設定バーの表示なども FileMaker Cloud や FileMaker Server にアップロードする前に確認しておくことをお勧めします。

詳細はプロダクトドキュメンテーションセンター（https://www.claris.com/ja/resources/documentation/）などを参照してください。

追補 1

ここでは、FileMaker Cloud を使ったカスタム App の共有方法について紹介します。詳しくは、クラリス・ジャパン株式会社のサイト（https://www.claris.com/ja/）のドキュメントを参照してください。

FileMaker Cloud チームでのカスタム App の共有

FileMaker Cloud は、1 つの「ホスト」に 1 つ対応する「チーム」によってカスタム App を共有するユーザを管理します。1 つのチームの中には、1 つ以上の「グループ」を作って、さらに細かく、カスタム App へのアクセスを制御することができます。

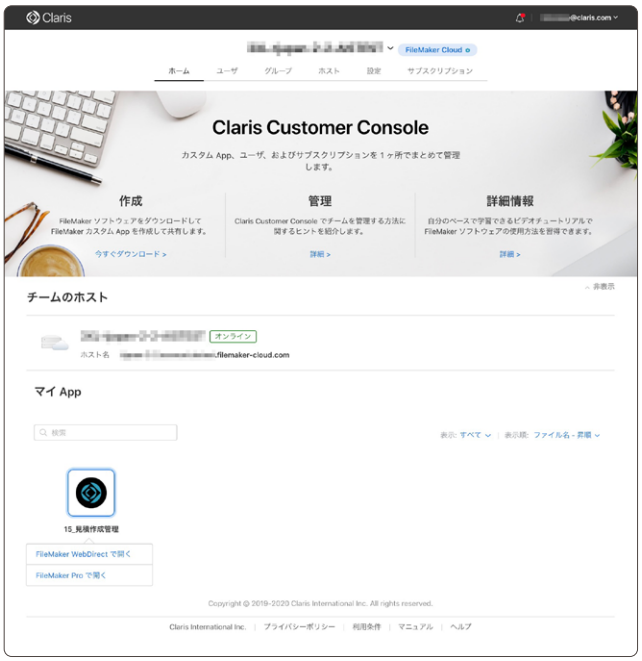
チームは、「チームマネージャ」によって管理されます。チームマネージャは、1 つのチームに 2 人以上設定することが推奨されています。

FileMaker Cloud でのカスタム App の共有は、「チームマネージャ」が次の手順で実施します。

1. ユーザをチームに招待する
2. チーム内にグループを作成する
3. 共有するカスタム App にユーザおよび／またはグループを追加する
4. カスタム App を FileMaker Cloud にアップロードする

Claris Customer Console (CCC)

チームやユーザの管理やカスタム App の管理は、チームマネージャが Claris Customer Console (CCC) を使って実施します。CCC は FileMaker Cloud と Claris Connect のリソースを管理する Web アプリケーションです。CCC では、Claris ID アカウント、チームとホストの設定、サブスクリプションの管理（アップグレード）、グループの管理などが可能です。詳しくは、CCC のヘルプを参照してください。



追補図 1　Claris Customer Console の「ホーム」画面（チームマネージャ用）

追補 2

FileMaker Pro 19 から Web ビューアで JavaScript を使用したスクリプトを作成できるようになりました。また、JavaScript と FileMaker のスクリプトがお互いに呼び出して情報を渡すこともできます。

ここでは、FileMaker Pro 19 ヘルプの「Web ビューアで JavaScript を使用したスクリプト作成」に記載された例をご紹介します。詳しい手順や設定、スクリプトの内容は、ヘルプを参照してください。

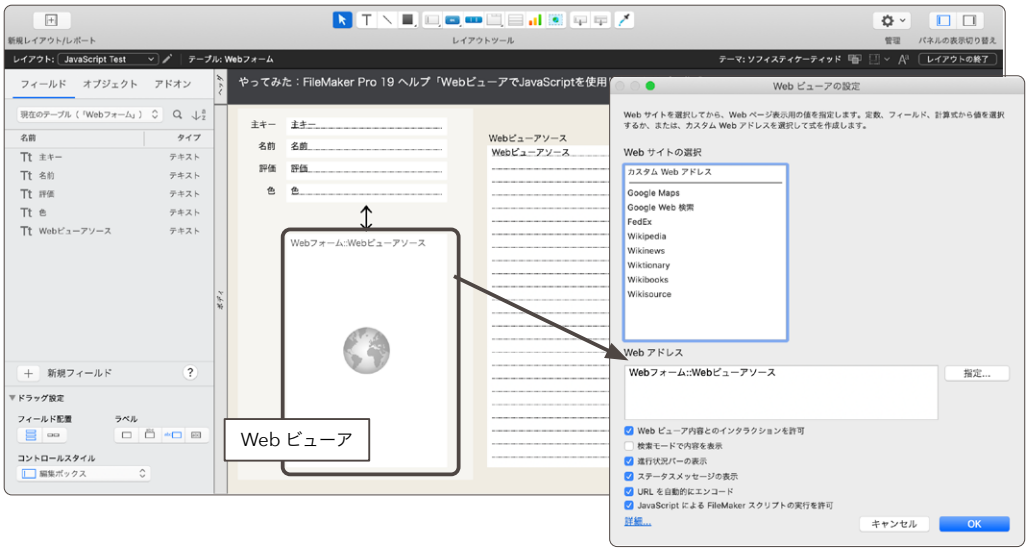
例：Web ビューアに作成したユーザインタフェースとテーブル間でデータをやり取りする

この例では、レイアウト上の Web ビューアに作成したユーザインタフェースとテーブル間で、次の機能を実現します。

・ユーザインタフェースで設定した値を FileMaker テーブルのレコードに設定する
・変更したレコードの値をユーザインタフェースに反映する

手順

1. テーブル「Web フォーム」を作成します。
2. 「Web フォーム」に対するレイアウト上に、Web ビューアを追加します。
3. 同じレイアウト上に、テーブル「Web フォーム」の各フィールドオブジェクトを追加します。



追補図 2　レイアウトおよび Web ビューアの設定

4. ブラウズモードでレコードを追加します。
5. 「Web フォーム ::Web ビューアソース」フィールド（グローバル格納）に、ユーザインタフェースを実現するデータ URL を入力します。（次ページを参照してください）
6. [スクリプトワークスペース] で「Web ビューアでユーザデータを設定」というスクリプトを作成します。このスクリプトは、レコードの値を Web ビューアに反映します。
7. 「Web フォームからのデータを格納」というスクリプトを作成します。このスクリプトは、Web ビューアでユーザが [送信] ボタンをクリックすると、Web フォームに設定された値が「Web フォーム」テーブルのフィールドに格納されます。
8. フィールドデータが変更された場合に Web ビューアを更新するため、レイアウトモードで、「Web ビューアでユーザデータを設定」スクリプトを実行する OnRecordLoad および OnRecordCommit スクリプトトリガを追加します。

「Web フォーム ::Web ビューアソース」フィールドに入力するデータ URL（手順 5）

```
data:text/html,
<html>
  <head>
    <style>
      div { padding-bottom: 0.5em; }
    </style>
  </head>

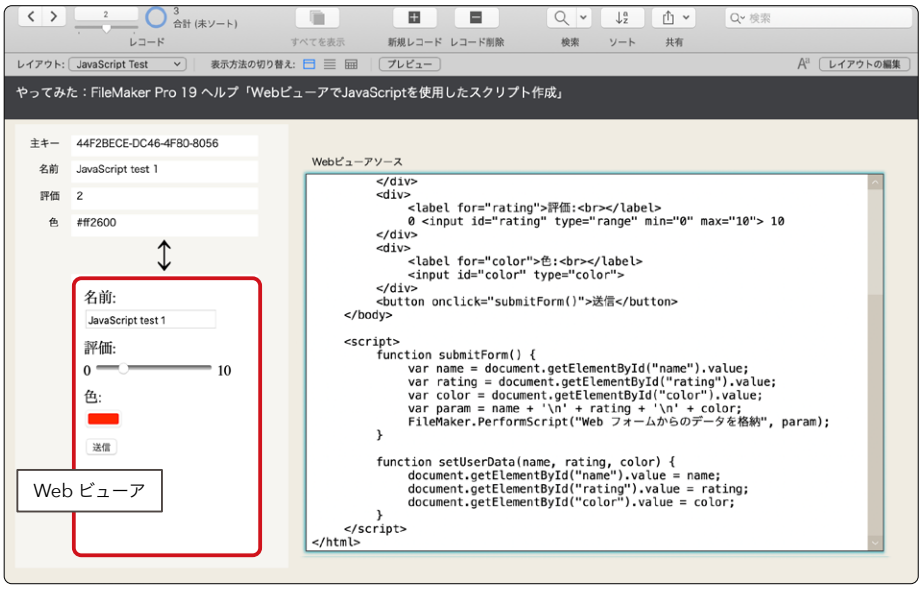
  <body>
    <div>
      <label for="name">名前:<br></label>
      <input id="name" type="text" value="default">
    </div>
    <div>
      <label for="rating">評価:<br></label>
      0 <input id="rating" type="range" min="0" max="10"> 10
    </div>
    <div>
      <label for="color">色:<br></label>
      <input id="color" type="color">
    </div>
    <button onclick="submitForm()">送信</button>
  </body>

  <script>
    function submitForm() {
      var name = document.getElementById("name").value;
      var rating = document.getElementById("rating").value;
      var color = document.getElementById("color").value;
      var param = name + '\n' + rating + '\n' + color;
      FileMaker.PerformScript("Web フォームからのデータを格納", param);
    }

    function setUserData(name, rating, color) {
      document.getElementById("name").value = name;
      document.getElementById("rating").value = rating;
      document.getElementById("color").value = color;
    }
  </script>
</html>
```

スクリプト：Web ビューアでユーザデータを設定（手順 6）

```
スクリプト一時停止 / 続行 [ 間隔 ( 秒 ): .001]
Web ビューアで JavaScript を実行 [ オブジェクト名: "Web ビューア";
  関数名: "setUserData";
  引数: Web フォーム::名前, Web フォーム::評価, Web フォーム::色 ]
```



追補図 3　JavaScript 連携の実装例

スクリプト：Web フォームからのデータを格納（手順 7）

```
変数を設定 [$jsParameters ; 値: Get (スクリプト引数)]
フィールド設定 [Web フォーム::名前 ; GetValue ($jsParameters ; 1)]
フィールド設定 [Web フォーム::評価 ; GetValue ($jsParameters ; 2)]
フィールド設定 [Web フォーム::色 ; GetValue ($jsParameters ; 3)]
```

動作の確認

レイアウト上に配置した Web ビューアには、「名前」（テキスト）、「評価」（スライダー）、「色」（カラーピッカー）を入力できるユーザインタフェースが表示されます。

Web ビューア内のユーザインタフェースからそれぞれの値を設定して「送信」ボタンをクリックすると、設定された値が「Web フォーム」テーブルのフィールドが自動的に挿入されます。

また、「Web フォーム」テーブルの各フィールドの値を更新してレコードを確定すると、Web ビューア内の値も更新されます。

■

索引

記号

$$ 変数、グローバル変数　394, 492
$ 変数、ローカル変数　394, 492

数字

1 対 1　43
1 対多　41, 74

アルファベット

A

Abs　345, 688
Average　412
AVPlayer オプション設定　620
AVPlayer 再生　620, 622
AVPlayer 再生状態設定　620

B

Base64Decode　411
Base64Encode　411

C

Case　129, 151, 339, 390, 397, 399, 401, 402, 404, 424, 427
Ceiling　344
CF 再帰ドット　425
CF 再帰左から文字 2　426
CF 消費税　424
CF 年齢　424
Char　448
Choose　391
Code　448
Count　412, 415

D

Date　128, 151, 330, 341, 384, 424, 492
Day　128, 330, 384, 424, 492
DayName　384
DayOfWeek　384
DayOfYear　384

E

Else　484
Else If　484, 487
End If　484
End Loop　548
Entitiy Relationship　38
ER 図　38
Evaluate　392
EvaluationError　331
Exact　352, 357, 358, 367
ExecuteSQL　408
EXIF　176
Exit Loop IF　548

F

FieldNames　255
FileMaker Go　608
[FileMaker WebDirect の設定] ダイアログ　675
FileMaker ファイル認証　661
Filter　347, 349, 394
FilterValues　347
Floor　344
fmreauthenticate10　660
foreign key　48
Furigana　348

G

GetAsBoolean　391, 392
GetAsDate　365, 385, 424
GetAsNumber　366
GetAsTime　385
GetAsTimestamp　385
GetAVPlayerAttribute　620
GetContainerAttribute　195, 196, 397, 399, 411
GetFieldName　393, 456, 541, 545
GetHeight　411
GetSensor　623
GetThumbnail　411
Get(UUID)　56, 343
Get(UUID 番号)　56, 343
GetValue　347, 498, 499, 500
GetWidth　411
Get(アカウントアクセス権セット名)　647
Get(アカウント拡張アクセス権)　660
Get(アカウントタイプ)　663
Get(アクティブフィールド名)　456
Get(アクティブポータル行番号)　456
Get(アクティブレイアウトオブジェクト名)　456
Get(アクティブレコード番号)　325, 456
Get(アプリケーションバージョン)　615, 616, 648
Get(ウインドウ名)　456
Get(ウインドウモード)　297, 363, 365
Get(オープンデータファイル情報)　558
Get(現在のアクセス権セット名)　422, 427
Get(現在の時刻 UTC ミリ秒)　386
Get(最終エラー)　517, 527, 532, 535, 551
Get(最終メッセージ選択)　510, 512, 516, 519, 523, 534, 695
Get(システムプラットフォーム)　345, 397, 399, 401, 402, 404, 688, 689
Get(スクリプトの結果)　495
Get(スクリプト引数)　460, 461, 487,

492, 498, 500
Get(対象レコード数)　456
Get(タイムスタンプ)　331, 385
Get(デスクトップパス)　397, 399, 401, 402, 404, 525
Get(デバイス)　615, 616, 648
Get(テンポラリパス)　525
Get(ドキュメントパス)　525
Get(トリガ外部イベント)　450, 620
Get(トリガキー修飾キー)　448
Get(トリガキー入力)　448
Get(日付)　119, 151, 331, 341, 385, 424, 492
Get(ホスト IP アドレス)　706
Get(ホストのタイムスタンプ)　385
Get(レイアウトテーブル名)　456
Get(レイアウト番号)　233, 456
Get(レイアウト名)　456
Get(レコード番号)　325, 456

H
Hiragana　348
Hour　385

I
If　119, 373, 378, 388, 390, 425, 428, 484, 487, 493, 497, 501, 507, 516, 519, 523, 527, 532, 536, 547, 551, 553, 559, 615, 620, 688
Int　344
IsEmpty　119, 378, 391, 392, 424, 481, 487, 559, 695, 706
IsValid　391, 392

J
JSONFormatElements　417
JSONGetElement　498, 506, 507
JSONSetElement　498, 506, 508
JSONString　506
JSON 関数　417
JSON フォーマットデータ　506

K
KanaHankaku　348
KanaZenkaku　348, 350
KanjiNumeral　348
Katakana　348

L
Left　346, 425
LeftValues　347
LeftWords　346
Length　345
Let　393, 394, 413, 415, 424, 492
List　412
Loop　548, 549, 552, 556, 695

M
Max　412
Middle　346
MiddleValues　347
MiddleWords　346
Min　412
Minute　385
Mod　344
Month　128, 330, 384, 389, 424, 492, 592
MonthName　384

N
not　352, 353, 391

O
OAuth　662
OnExternalcommandReceived　450, 620
OnFileAVPlayerChange　453, 620
OnFirstWindowOpen　452, 616, 706
OnGestureTap　449
OnLastWindowClose　453, 707
OnLayoutEnter　448, 545
OnLayoutExit　448
OnLayoutKeystroke　448
OnLayoutSizeChange　449
OnModeEnter　449
OnModeExit　449
OnObjectAVPlayerChange　452, 620
OnObjectEnter　451
OnObjectExit　451
OnObjectKeyStroke　451
OnObjectModity　451
OnObjectSave　451
OnObjectValidate　451
OnPanelSwitch　451
OnRecordCommit　448
OnRecordLoad　447
OnRecordRevert　448
OnTimer スクリプトをインストール　454
OnTimer トリガ　454
OnViewChange　449
OnWindowClose　453
OnWindowOpen　453

P
PatternCount　615, 648
PDCA　6
primary key　48

Q
Quote　366

R
Random　344
Right　346

RightValues　347
RightWords　346
RomanHankaku　146, 348, 349, 394
Round　339, 344, 413, 415

S
Seconds　385
Self　119, 146, 349, 350, 392, 394
SetRecursion　418, 426
Sign　345
SSL　677
Substitute　366, 394
Sum　338, 339, 412, 413, 415

T
Time　385
Timestamp　385
TO　60
TOG　61
Trim　346, 349
TrimAll　346
Truncate　344

U
URL を開く　614
UUID　56, 343, 629

V
ValueCount　346
VerifyContainer　411

W
Web ビューア　309
WeekOfYear　384, 592
While　418, 420, 426
WordCount　345

X
X の関連　327

Y
Year　128, 330, 341, 384, 389, 424, 492, 592
YearName　384

かな

あ
アカウント　641
アカウントの有効化　697
アカウントを追加　695
アクセス権セット　644, 649
値一覧　277
値一覧（関連利用）　288
アンカー / ブイ　67
暗号化　672

い
依存　55
一致するレコードを検索　528
イベントエンティティ　40
印刷　516, 519, 523, 527
印刷設定　516, 519, 523, 527
インスペクタ　259
インターフェースファイル　638

う
ウインドウ　472
ウインドウタイトルの設定　541
ウインドウ内容の再表示　549, 559
ウインドウの移動／サイズ変更　541
ウインドウのオプション　542
ウインドウの固定　549
ウインドウの調整　463, 501, 507, 541, 615
ウインドウを選択　457, 541, 546, 547
ウインドウを閉じる　541, 546
運用管理者　626

え
エラー処理　432, 481, 517, 519, 520, 523, 527, 532, 535, 536, 547, 551, 553, 613, 619, 695, 703
演算子　62, 329
エンティティ　38

お
オープン格納　178
オブジェクトへ移動　476
オブジェクト関数　411
オブジェクトタイプフィールド　174
オブジェクトタブ　263
［オブジェクトの管理］ダイアログ　180
オブジェクトへ移動　259, 457, 487, 493

か
カード　542
カードウインドウ　469
開発環境　10
開発者　5
開発者かどうか　422
開発担当者　626
外部格納のパス　178
外部キー　48
外部データソース　664
外部のサーバー認証　662
拡張アクセス権　659
カスタム App　2
カスタムアプリケーション　2
カスタム関数　421
カスタムダイアログを表示　399, 402, 483, 487, 493, 510, 516, 519, 523, 527, 532, 535, 536, 551, 553, 556, 559,

619, 695
カスタムメニュー　680, 682
カスタムメニューセット　682
画面一覧　204
画面遷移　215
関数　329
関数の種類　342
関数パネル　336
関連　57
関連テーブル　81
関連（範囲）　83
関連（複数条件）　82
関連レコード　81
関連レコードへ移動　457, 464, 468, 470, 472, 474, 476, 478, 483, 487, 493, 541

き
キーチェーン　643
企画　11
起動処理　705
機能　26, 429
規模感　20
競合　629
業務　2
業務の流れ　24
切り取り　456

く
クイック検索　528
クイック検索対象　298
クライアント　628
クライアント／サーバー　627
グラフ　568, 593
[グラフ設定] ダイアログ　595
グローバル格納　163
グローバル変数　394

け
計算式　329
[計算式の指定] ダイアログ　332
計算式パネル　334
現在のスクリプト終了　559
現在のスクリプトを終了　433
検索実行　528, 532, 535, 536, 551
検索条件を変更　528
検索モードに切り替え　528, 532, 535, 536, 551, 553

こ
合意　13, 32
交差 / 中間　53
構造設計　37
高度なツール　9
構文　332
高度なツール　9
コピー　456
コメント　427
コンテキスト　65
コントロールスタイル　276

さ
サーバー上のスクリプト実行　613
再帰　418
再計算　331
再評価　331
再ログイン　648
作業　23
索引　81, 153, 331
削除ルール　700
サブスクリプト　495
算術演算子　335
参照先　84
参照元　84

し
資格情報マネージャ　643
識別子　665
システム　2
システム開発プロセス　7
実行　431
自動ログイン　643
集計　568
集計タイプフィールド　572
終了処理　706
主キー　48
取得関数　331, 365
消去　376
小計パート　576
条件付き書式　353
条件分岐　430, 489
照合条件　81
照合フィールド　81
ショートカット　439
新規ウインドウ　541, 544, 547
真偽値　392
新規レイアウト作成ウイザード　234
新規レコード／検索条件　458, 528, 551

す
数字関数　344
スキーマ　38
スクリプト実行　495, 497, 501, 507
スクリプトステップ　430
スクリプトステップパネル　438
スクリプトデバッガ　442
スクリプトトリガ　445
スクリプトの一時停止 / 続行　549
スクリプトパネル　434
スクリプト引数　458
スクリプト編集パネル　436
スクリプトワークスペース　433
スタイル　315

スタンドアロン　627
ステークホルダー　13
ステータスツールバー　257, 679
スライドコントロール　306
スワイプ　611

せ
正規化　37, 45
制御コマンド　430
制約　12
セキュア格納　178
セキュリティ　29
センサー情報　623
全スクリプト終了　433
前部総計パート　577

そ
その他の関数　393

た
ダイアログ（モーダル）　542
対象レコードの拡大　528
対象レコードの絞り込み　528
ダイナミック引数　409
対話　510
多対多　42, 76, 87
タッチ　611
タブコントロール　306
タブ順　300

つ
ツールバーの表示切り替え　648
次の場合にオブジェクトを隠す　296, 352

て
定義　57
データ　3, 107
データ移行　30, 634
データ書式（オブジェクトタイプ）　190
データの格納　152, 160
データの同期　628
データビューア　341, 444
データファイル　638
データファイルから読み取る　558, 563
データファイルに書き込む　558, 565
データファイルの位置を取得　558, 563
データファイルの位置を設定　558, 565
データファイルを作成　558
データファイルを閉じる　558, 561
データファイルを開く　558, 561
データ分離モデル　638
[データベースの管理] ダイアログ　58
データモデリング　38
テーブルオカレンス　60
テーブルオカレンスグループ　61
テーマ　315
テキスト関数　345, 348, 365
デバイスから挿入　610, 617
デフォルトアカウント　642
デフォルトのフィールド　110
デフォルトのフィールドの自動作成　110

と
同期　609, 628
統計関数　338, 412
統合　31, 44
ドキュメントウインドウ　541

な
ナビゲーション　224

に
入力値の自動化　116
入力値の制限　130
入力値の制限（その他）　146

ね
ネットワークファイルパス　675

は
[パートの定義] ダイアログ　253
排他　703
バイナリデータ（埋め込み）　177
バイナリデータの保存（外部格納）　178
ハウスキーピングフィールド　110
貼り付け　456
反復　430

ひ
比較演算子　335, 353
非関連　81
引数　340, 489
引数（複数）　498
被計算フィールド　331
日付関数　340
日付と時刻に関する関数　384
評価　331
標準の機能　429

ふ
ファイルアクセス　664
ファイルサイズを取得　565
ファイルのスクリプトトリガ　452
ファイルの存在を取得　558
ファイルの名前変更　558
ファイルを削除　558
ファイルを挿入　402
フィールド移動　299
フィールドオプション　115
フィールド設定　399, 457, 532, 535, 536, 539, 551, 553, 619, 620
フィールドタイプ　109

フィールドタブ　236
フィールド定義　108
フィールド内容のエクスポート　199
フィールド入力　294
フィールドパネル　333
フィールドラベル　93, 371
フィールドを設定　398
フィールドを選択　457
フィールドを名前で設定　457, 540
フォーリンキー　48
フッタ　222
プライマリキー　48
フラグ　392
プレースホルダテキスト　294
フローチャート　17
フローティングドキュメント　542

へ
ヘッダ　222
変数　394, 492, 525
変数を設定　396, 402, 404, 457, 493, 497, 501, 507, 527, 540, 556, 559, 695

ほ
ポータル　308
ポータル（現在のテーブル）　323
ポータル内の行へ移動　457
ポータルの更新　374, 376
ポータルフィルタ　368, 380, 382
ポート番号　632
ホーム画面　218
保守　4
ホスト　628, 629
［ホスト］ダイアログ　674
ポップアップヘルプ　414
ポップオーバーを閉じる　305, 539
ボディ　222

ま
マスタ　40
マスタエンティティ　40
マスタ / 詳細レイアウト　210, 319

め
命名ルール　70, 108
メタデータ　176
メニュー　66, 687
メニューセットのインストール　648
メンテナンス　4

も
モジュール　495
文字列演算子　335
モックアップ　212
モデリング　37

ゆ
ユーザインターフェース　202
ユーザによる強制終了を許可　533

よ
要件　16
要件定義　20

り
リソースエンティティ　40
リレーションシップ　57
リレーションシップグラフ　59
［リレーションシップの編集］ダイアログ　62

る
ループ　430

れ
レイアウトエリア　254
レイアウトオブジェクト　301, 318
レイアウトオブジェクトのスクリプトトリガ　450
レイアウト切り替え　457, 461, 516, 519, 523, 527, 532, 536, 551, 553, 615
［レイアウト設定］ダイアログ　245
［レイアウトの管理］ダイアログ　232
レイアウト切り替え　463
レイアウトのスクリプトトリガ　447
レイアウトバー　257
レイアウトパート　250
レイアウトモード　231
レコード／検索条件確定　532, 535, 539, 540, 551, 559, 703
レコード／検索条件削除　458
レコード／検索条件／ページへ移動　457, 549, 556, 576
レコード／検索条件を開く　703
レコード作成（関連）　85
レコードのソート　576, 591
レコードを PDF として保存　523, 527, 556
レコードを対象外に　528, 551
連鎖削除　701

ろ
ローカル通知の構成　623
ローカル変数　394
ログイン　661
ロック　703
論理演算子　335
論理関数　390, 408
論理式　367
論理値　288

わ
ワイヤーフレーム　208
割り当て　431