

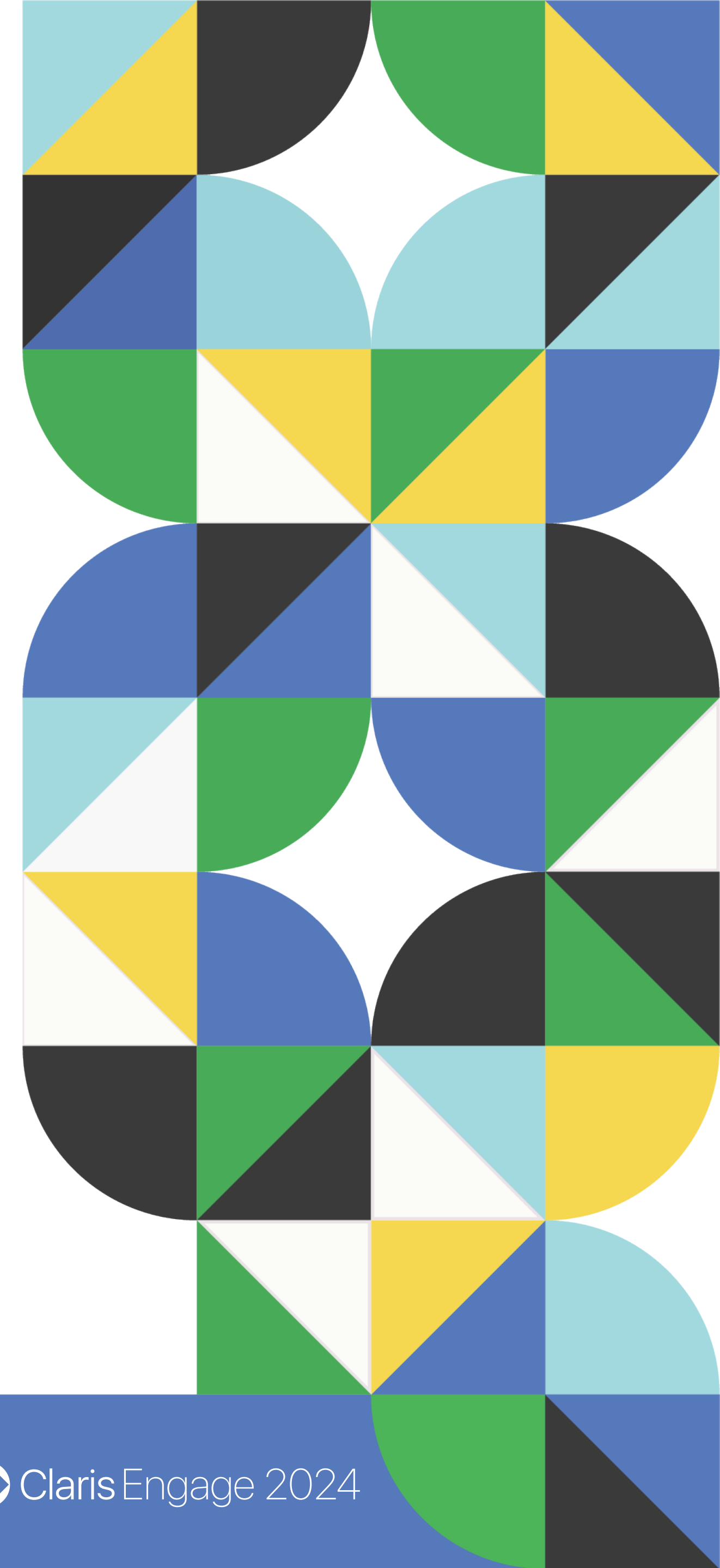
Delivering AI-driven solutions with Claris FileMaker

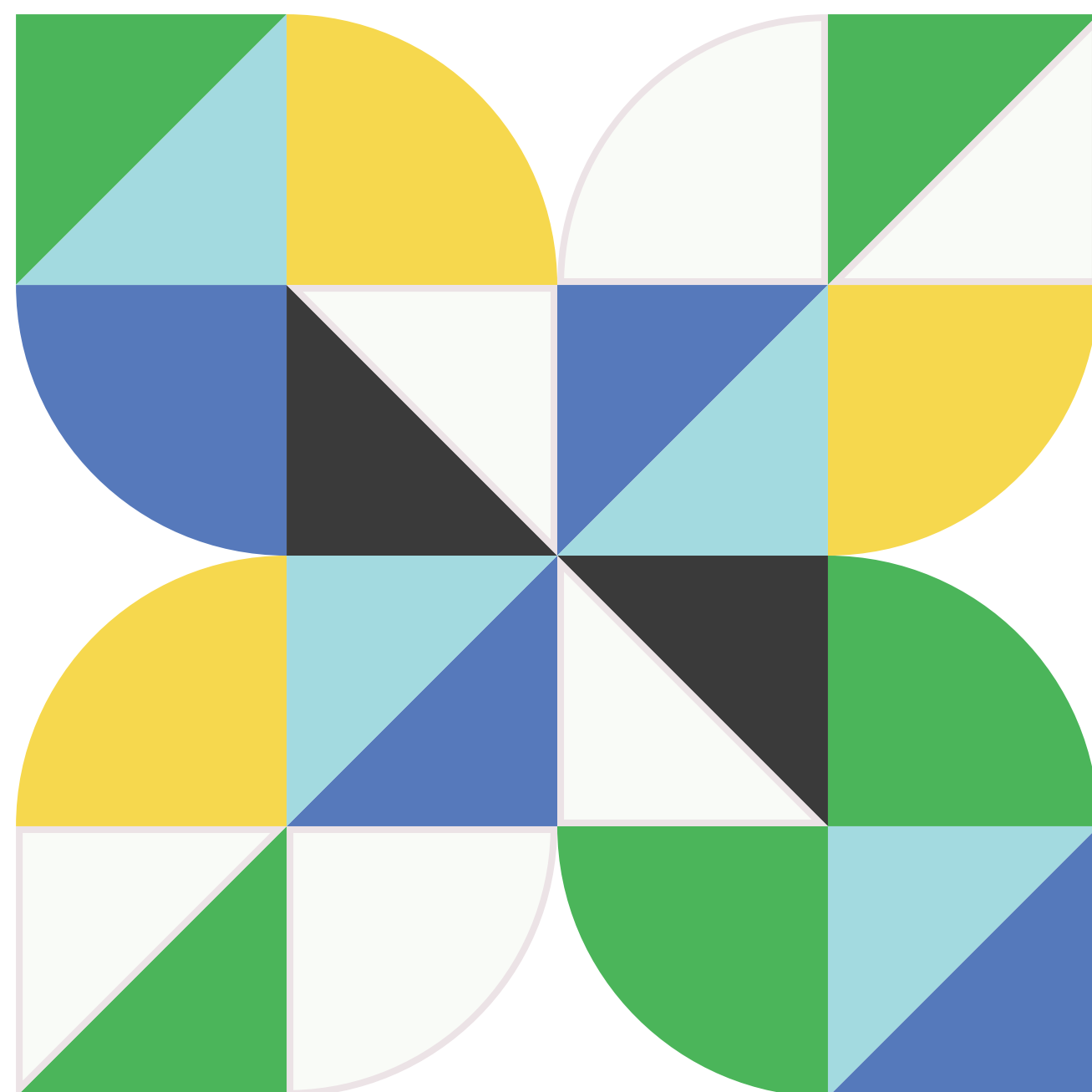
A practical overview

February 7, 9:00 a.m.

Ronnie Rios

Claris





Want to see something cool?



Macintosh HD

Apps

Search

Action Item Extractor

Given the VTT transcript of a meeting, c...

Append String To List

Append each value in a list with a const...

Clean up CSV data

Cleans up messy data in CSV format

Code Name Generator

suggest codenames for a project give a b...

Company Name Generator

generates list of company names given th...

CSV data generator

Generates sample data in CSV format

CSV to JSON

Converts CSV data into a JSON object

CSV to narrative

Converts CSV data into readable narrativ...

Dad Joke Generator

writes dad jokes about a specified topic

Database Schema Generator

generates full database schema based on ...

Code Name Generator

suggest codenames for a project give a brief description of the project and how many to generate

PROJECT_DESCRIPTION

A framework that generates AI-powered apps

NUMBER_OF_CODENAMES

10

RESULTS

["AIFrame", "GenieApps", "Intellicraft", "AutoMinds", "ThinkEngine", "BrainGrid", "InnovAi", "BrightBots", "SmartCell", "FlowNebula"]

last: 3.141s
avg: 3.2535s
tokens: 174



Apps

+

🗑

📄

⚙

Search

🔍

Action Item Extractor

Given the VTT transcript of a meeting, c...

Append String To List

Append each value in a list with a const...

Clean up CSV data

Cleans up messy data in CSV format

Code Name Generator

suggest codenames for a project give a b...

Company Name Generator

generates list of company names given th...

CSV data generator

Generates sample data in CSV format

CSV to JSON

Converts CSV data into a JSON object

CSV to narrative

Converts CSV data into readable narrativ...

Dad Joke Generator

writes dad jokes about a specified topic

Database Schema Generator

generates full database schema based on ...

Code Name Generator

suggest codenames for a project give a brief description of the project and how many to generate

PROJECT_DESCRIPTION

A framework that generates AI-powered apps

NUMBER_OF_CODENAMES

10

▶

RESULTS

["AIFrame", "GenieApps", "Intellicraft", "AutoMinds", "ThinkEngine", "BrainGrid", "InnovAi", "BrightBots", "SmartCell", "FlowNebula"]

last: 3.141s

avg: 3.2535s

tokens: 174



Macintosh HD

Apps

+

🗑

📄

⚙

Search

🔍

JSON Schema Generator

generates JSON Schema definition given a...

JSON to TOML Converter

Converts a JSON object into TOML format

JSON to XML

Converts a JSON object to XML format

JSON to YAML Converter

Converts a JSON object to YAML

List to JSON

Converts a bullet list to a JSON object

Numbered List

takes a list of text and numbers them

Smart assistant API

generates responses as if it were a smar...

Smart Trim

Removes all leading and trailing white s...

Table Generator

generates a list of tables that would be...

Transcript Correction

Given the VTT transcript of a meeting, m...

Table Generator

generates a list of tables that would be needed for a given problem

PROBLEM_DESCRIPTION

basic project management

▶

RESULTS

["Project", "Task", "Employee", "TimeEntry"]

last: 1.445s

avg: 1.3855s

tokens: 108



Macintosh HD

Apps

+

🗑

📄

⚙

Search

🔍

JSON Schema Generator

generates JSON Schema definition given a...

JSON to TOML Converter

Converts a JSON object into TOML format

JSON to XML

Converts a JSON object to XML format

JSON to YAML Converter

Converts a JSON object to YAML

List to JSON

Converts a bullet list to a JSON object

Numbered List

takes a list of text and numbers them

Smart assistant API

generates responses as if it were a smar...

Smart Trim

Removes all leading and trailing white s...

Table Generator

generates a list of tables that would be...

Transcript Correction

Given the VTT transcript of a meeting, m...

Table Generator

generates a list of tables that would be needed for a given problem

PROBLEM_DESCRIPTION

basic project management

▶

RESULTS

["Project", "Task", "Employee", "TimeEntry"]

last: 1.445s

avg: 1.3855s

tokens: 108



Agenda

- Introduction
- How to get started
- AI in FileMaker
- Tips for applying AI
- Next steps
- Q&A


Introduction

What is *AI*?

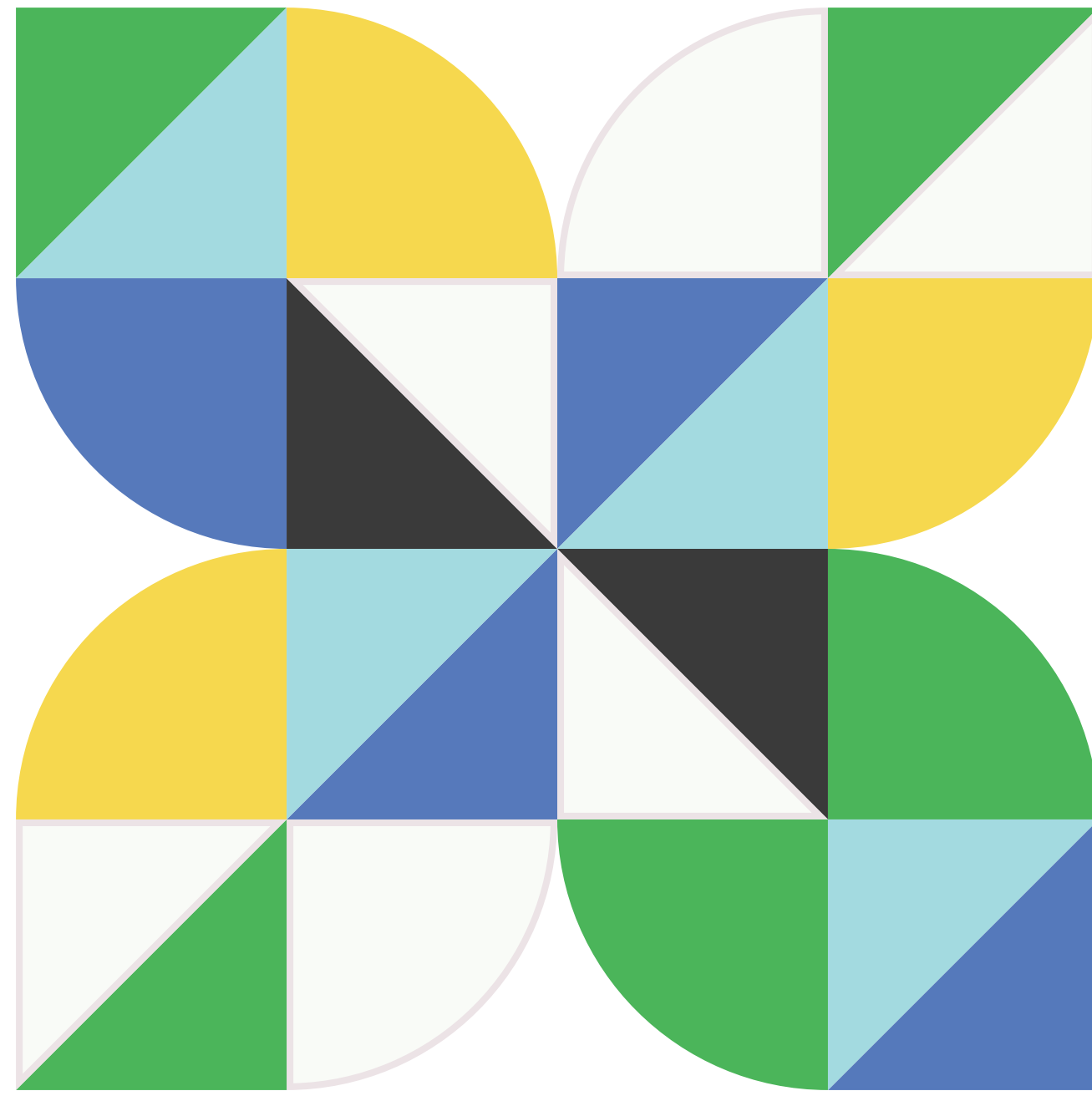
What is it important?

Claris AI



A decorative vertical strip on the left side of the slide, composed of a grid of squares. Each square is divided into two triangles by a diagonal line. The triangles are filled with various colors: dark blue, light blue, green, yellow, black, and white. Some squares also contain curved lines or segments, creating a complex, abstract geometric pattern.

The theory and development
of computer systems able to
perform tasks that normally
require **human intelligence**.



Why is it important?

Claris AI

01

02

03



Claris AI

01 Ethical

02

03



Claris AI

01 Ethical

02 Customer-centric

03



Claris AI

01 Ethical

02 Customer-centric

03 Apple



How to get started

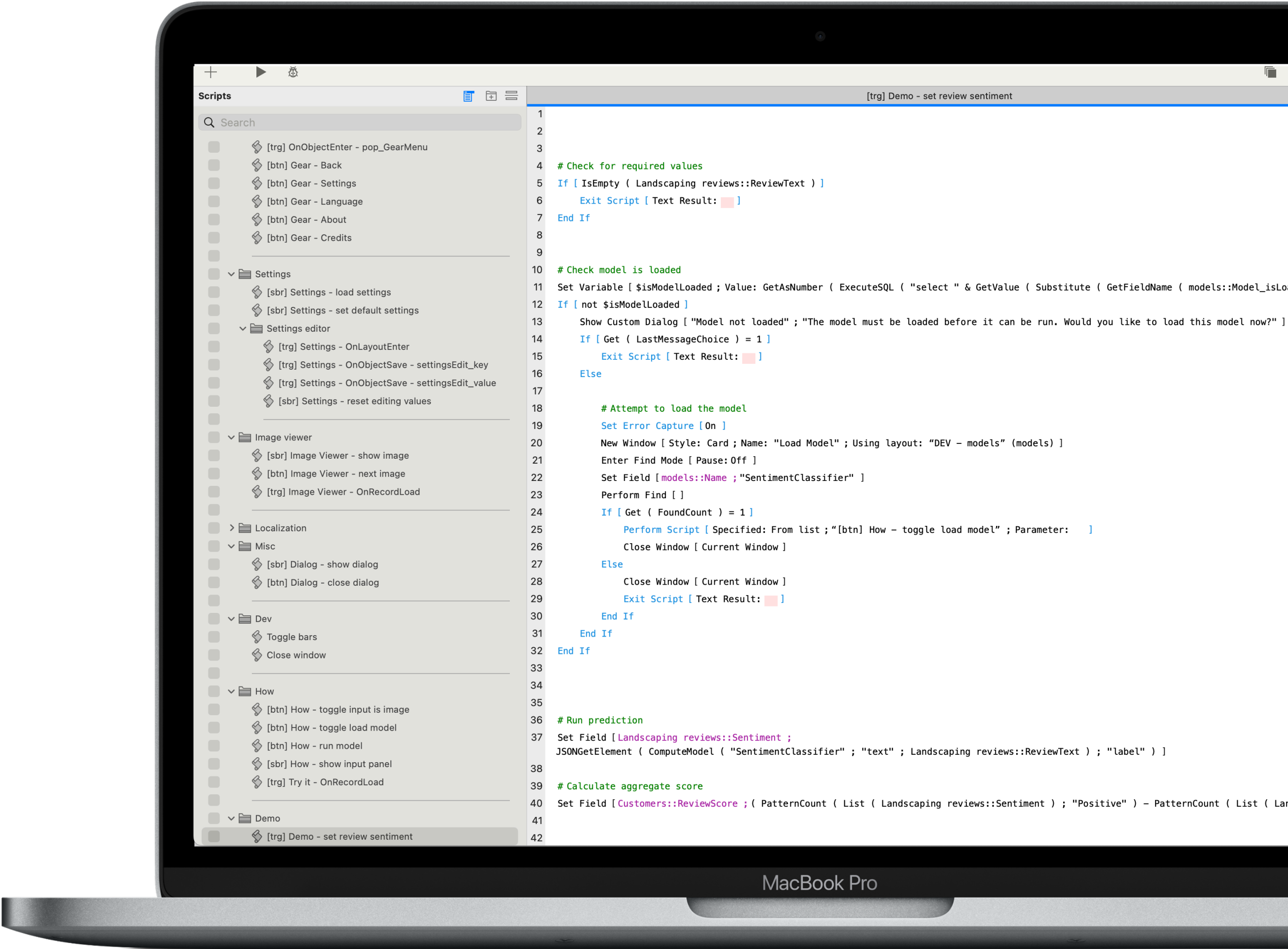
FileMaker tools

Core ML

LLM

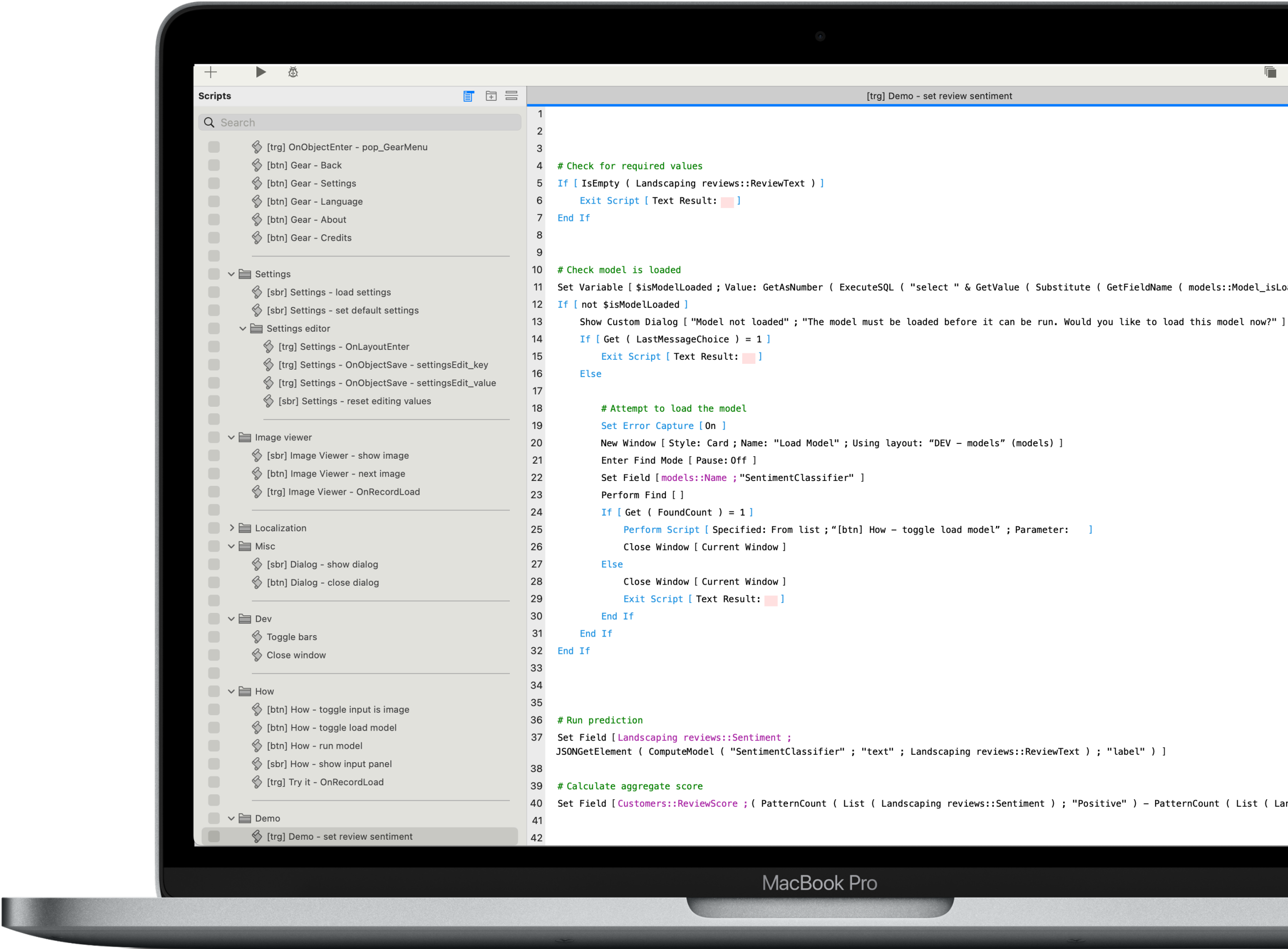


FileMaker tools



FileMaker tools

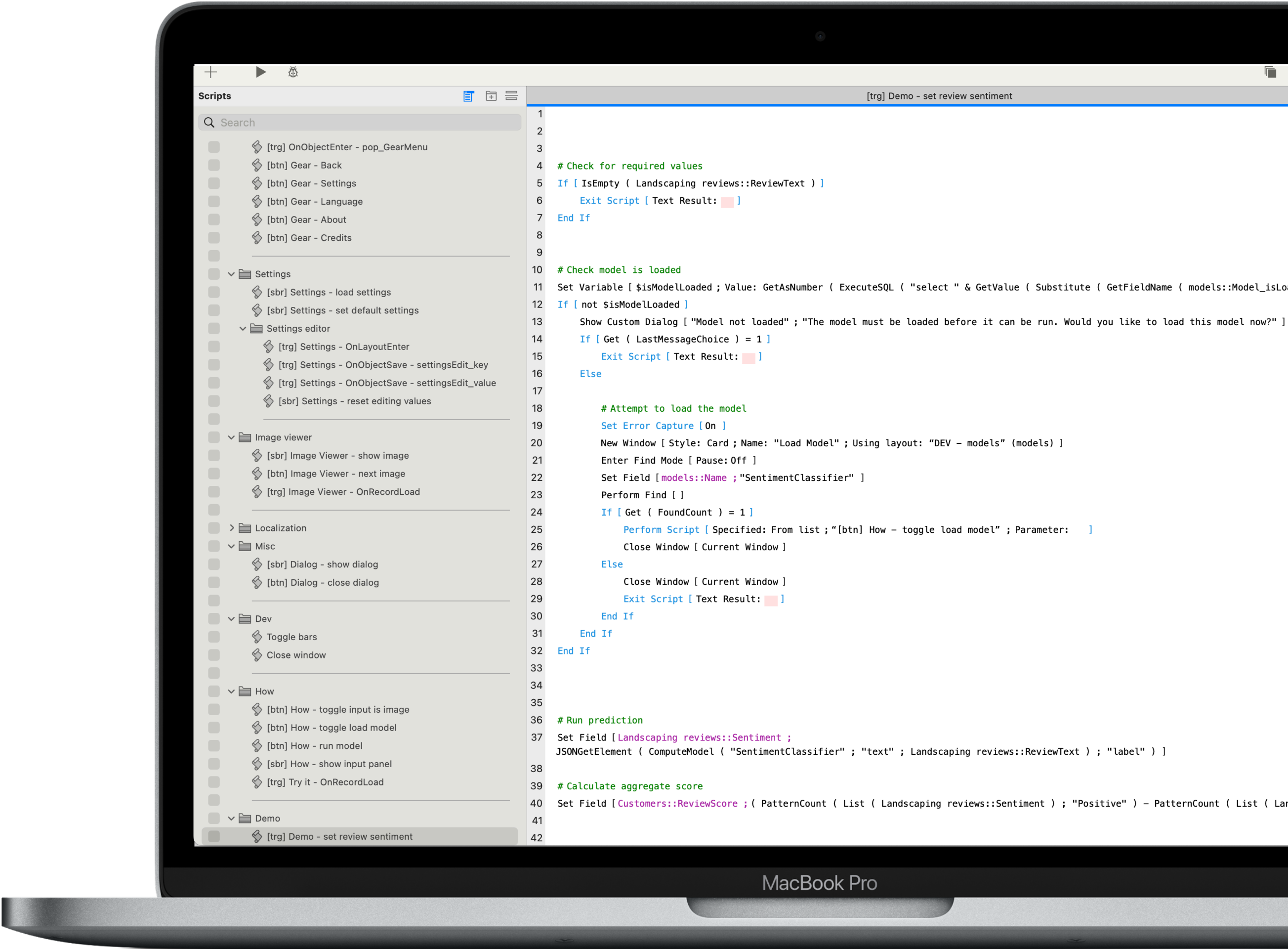
ComputeModel ()



FileMaker tools

ComputeModel ()

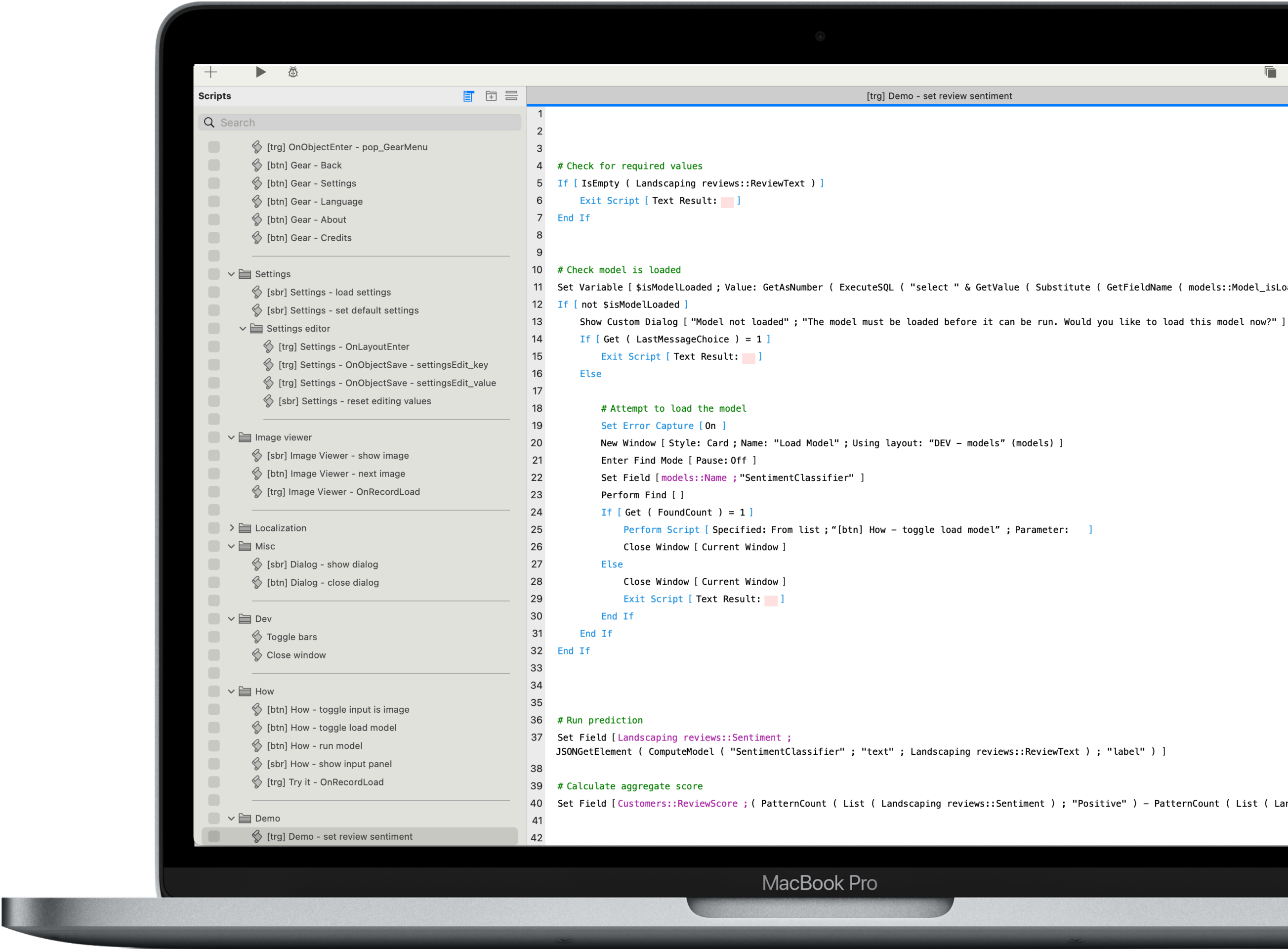
- On-device model (Core ML)



FileMaker tools

ComputeModel ()

- On-device model (Core ML)

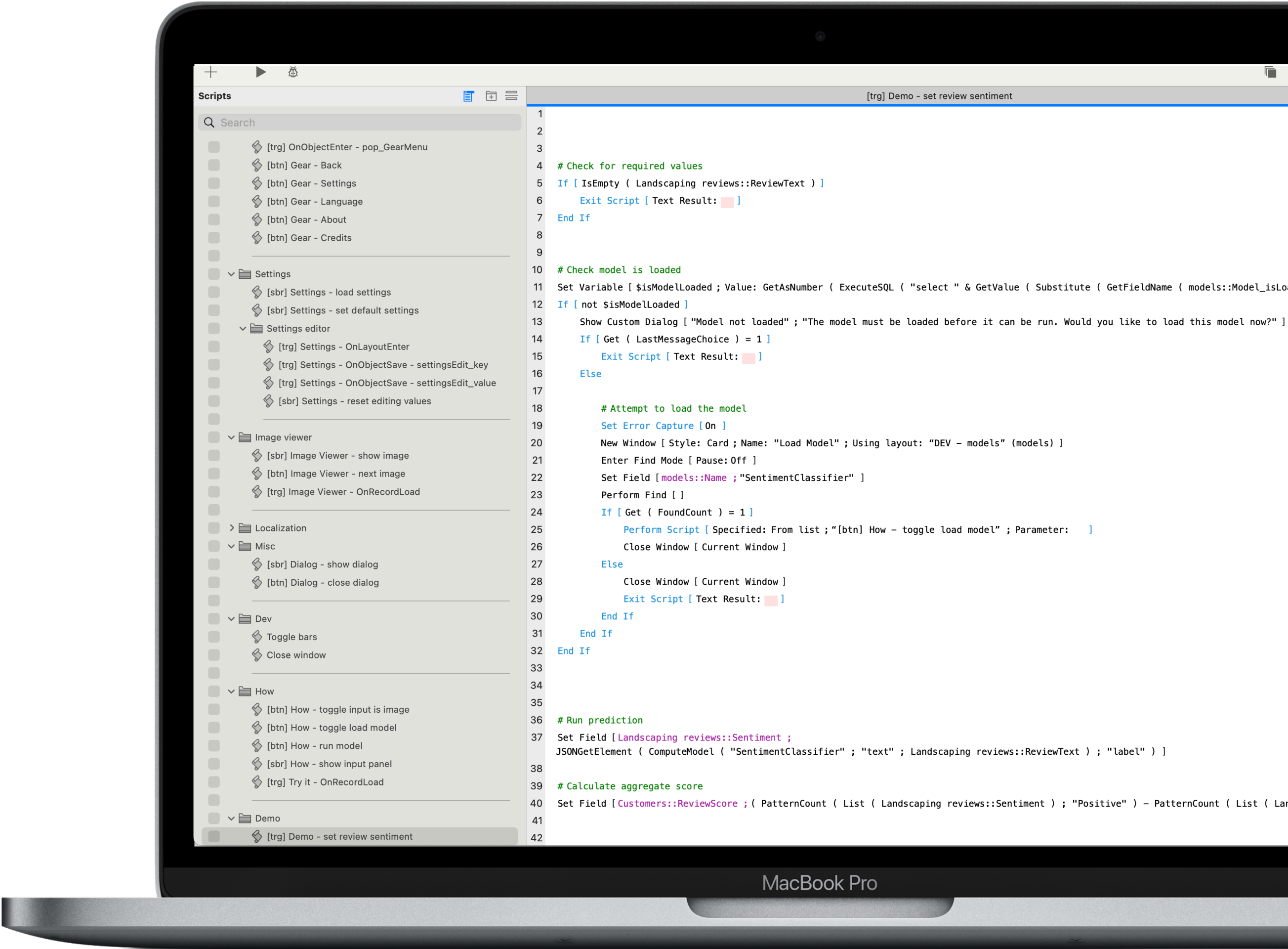


FileMaker tools

ComputeModel ()

- On-device model (Core ML)

Insert from URL []



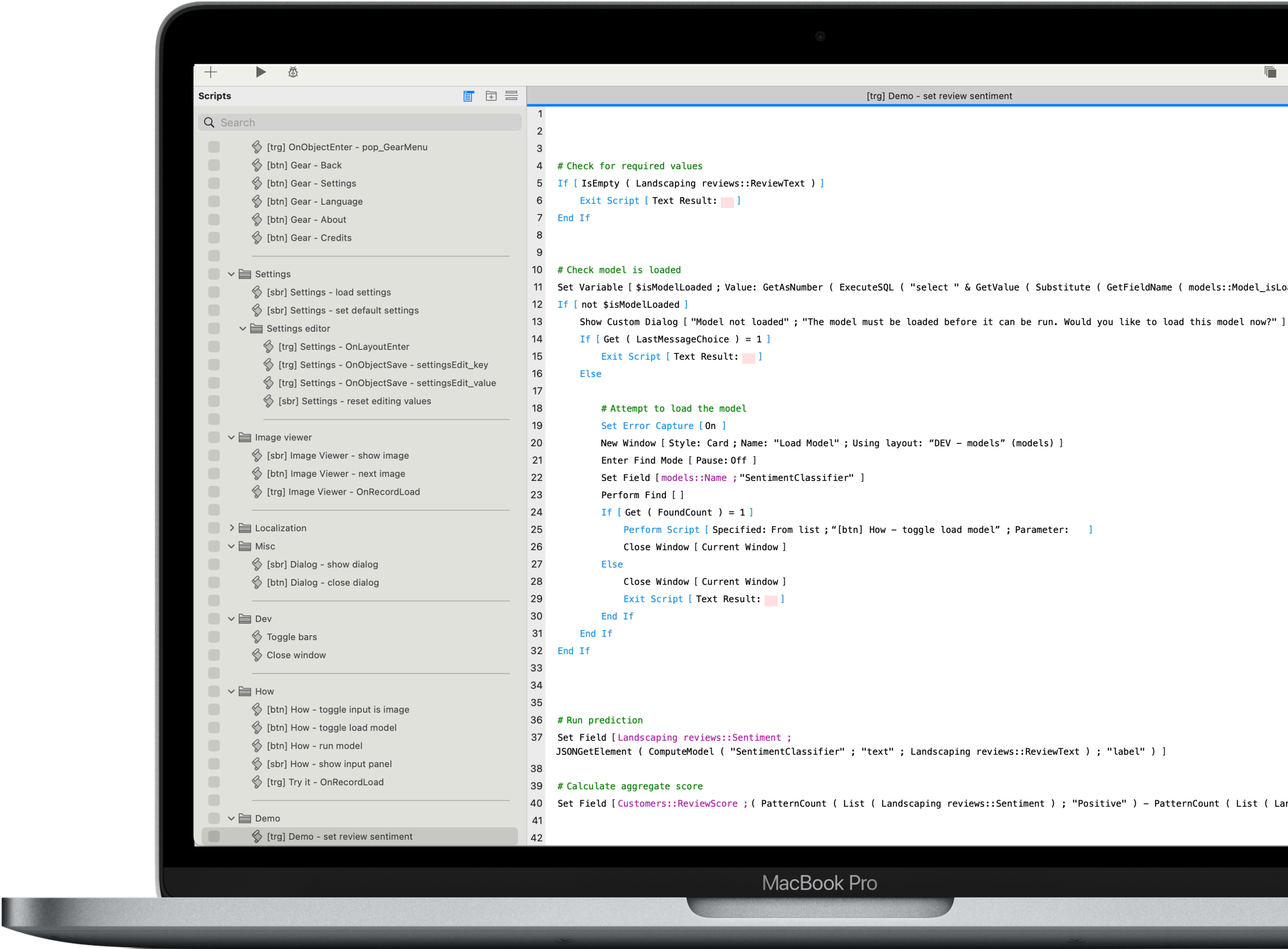
FileMaker tools

ComputeModel ()

- On-device model (Core ML)

Insert from URL []

Trigger Claris Connect Flow []



FileMaker tools

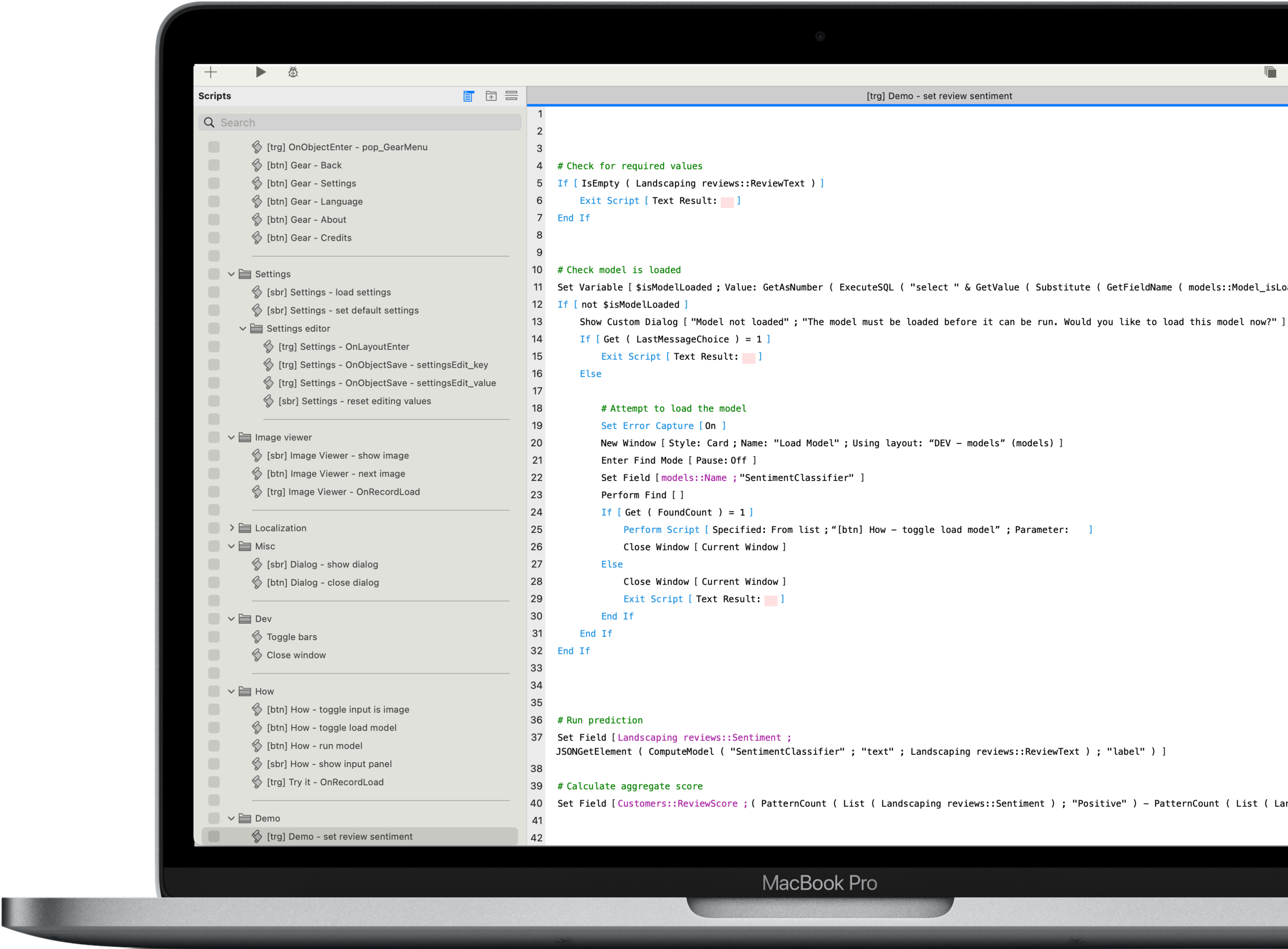
ComputeModel ()

- On-device model (Core ML)

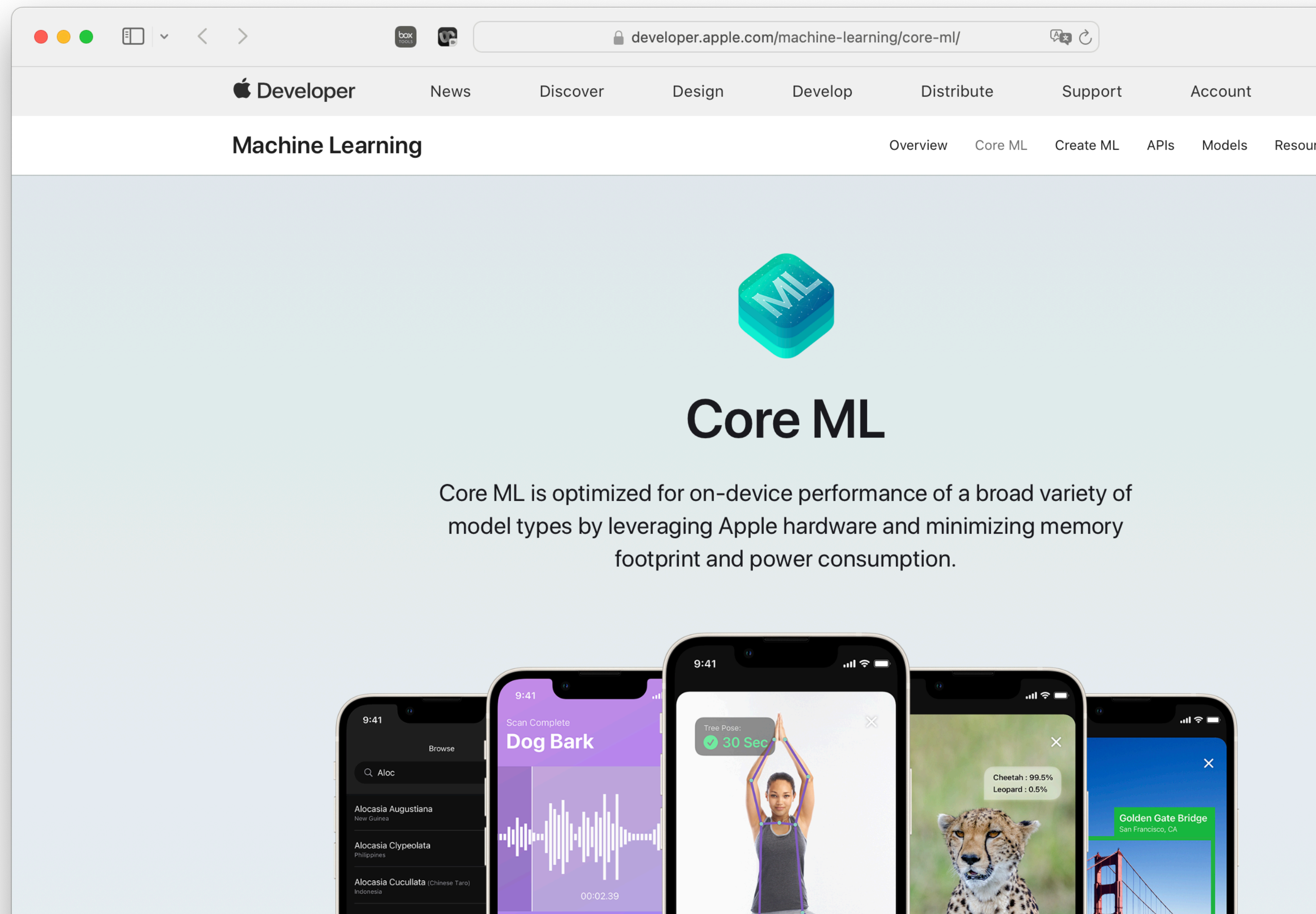
Insert from URL []

Trigger Claris Connect Flow []

- Model via API

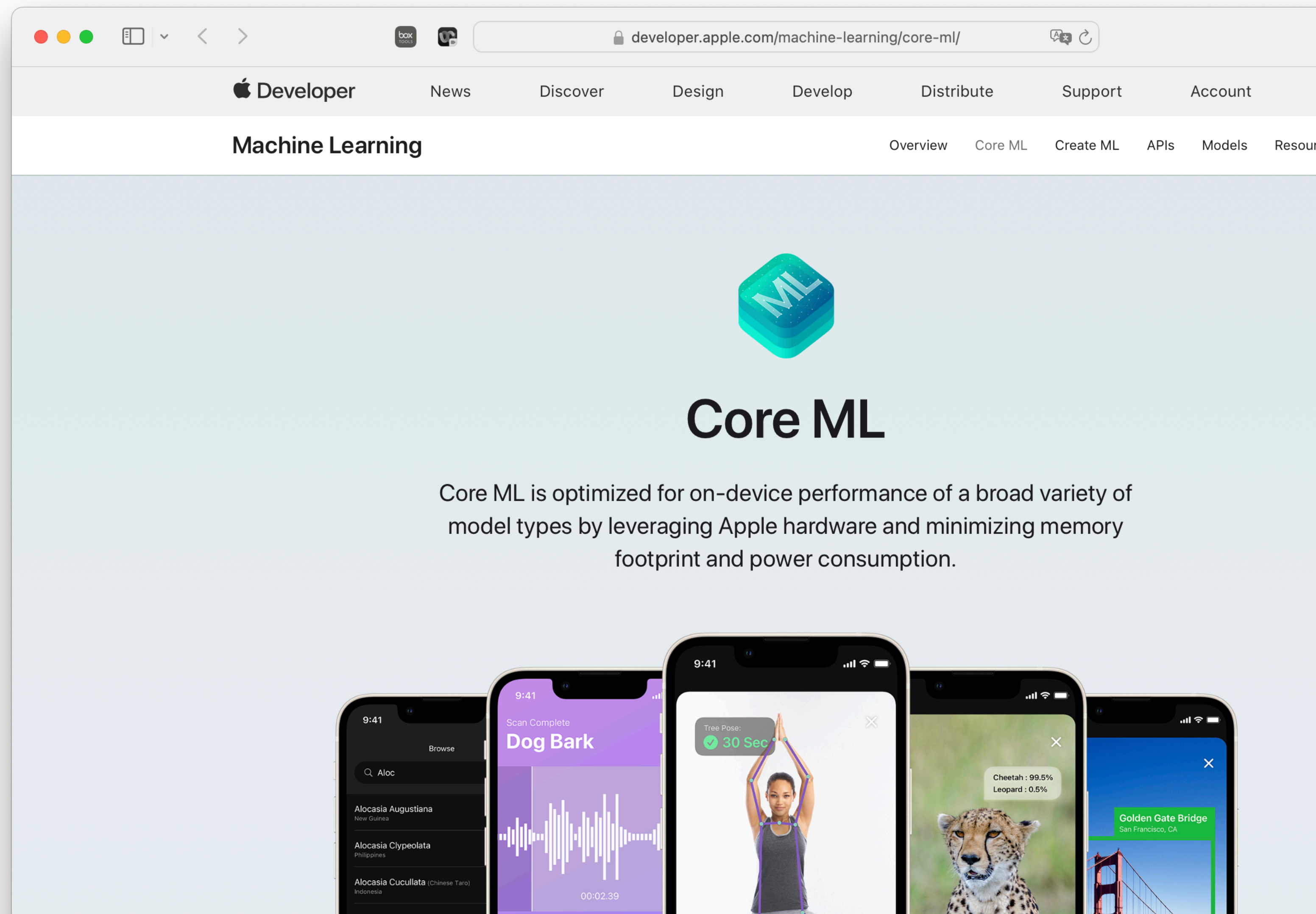


Core ML



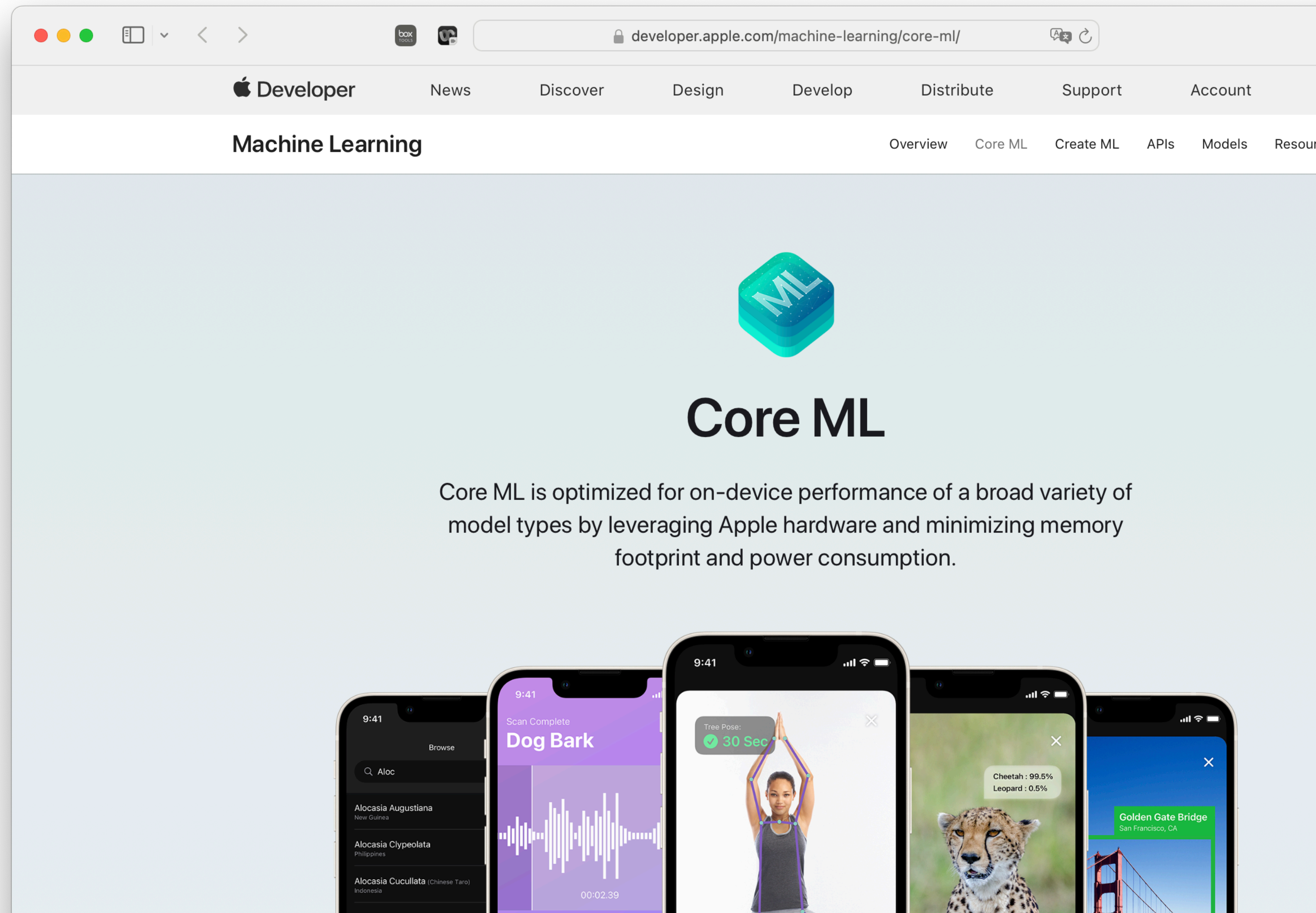
Core ML

- Run models fully on-device



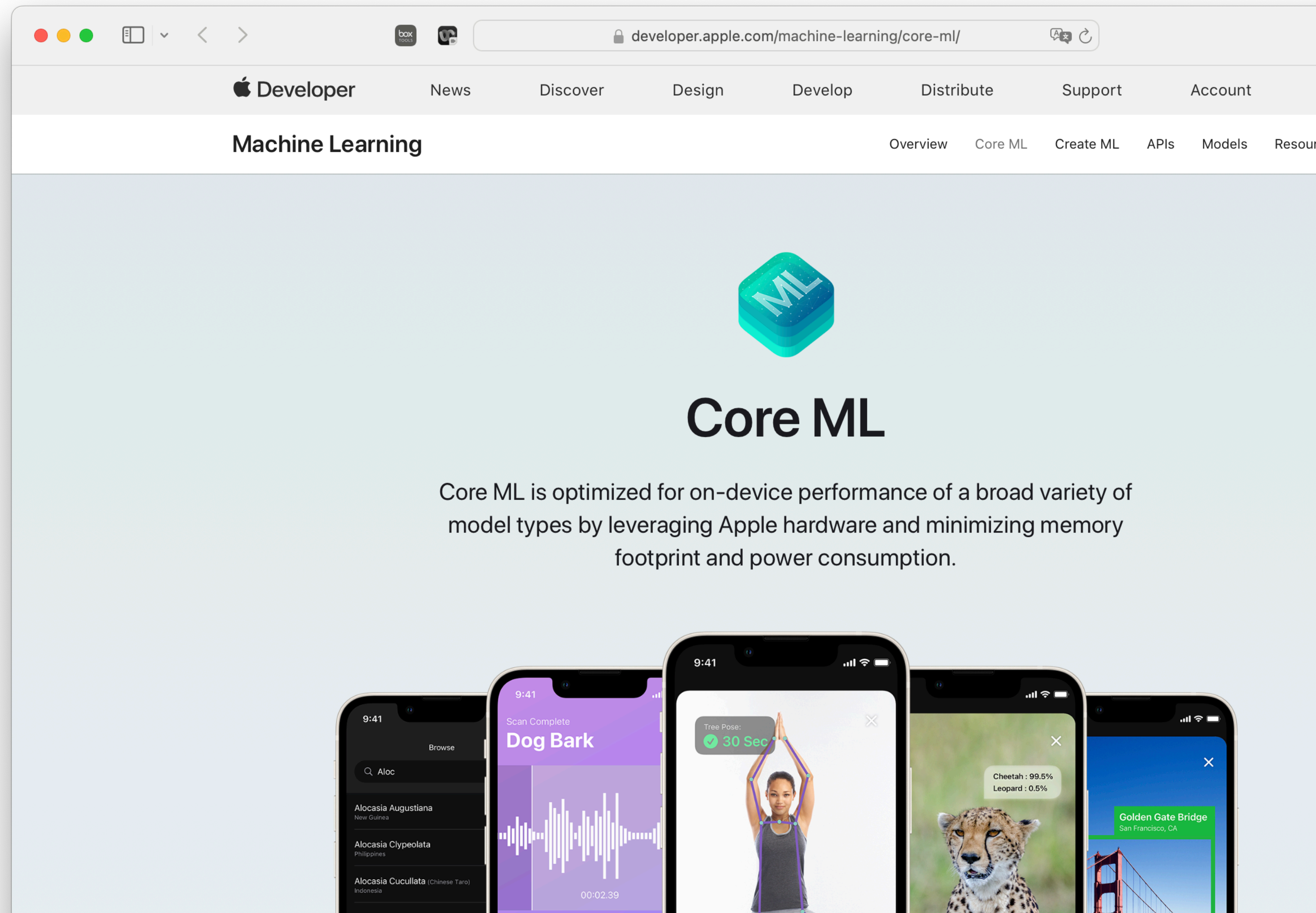
Core ML

- Run models fully on-device
- Leverage Apple hardware




Core ML

- Run models fully on-device
- Leverage Apple hardware
- Small footprint



Customers

[Back](#)


Bryant Research Park 

Jackson View Park 

Magnolia Springs Center 

Pine Executive Station

Optimum Industrial Station 

Ditmars Grove Station 

Dogwood Creek Center

Van Dam Station 

Elm Greens Park

Ditmars Research Station

Dekalb Station

Ditmars River Center

Webster Springs Station

Oak Interactive Center

Vanderbilt Stream Park

Oak Stream Park

Cedar Creek Interactive

Ainslie View Center



Customer

Bryant Research Park

Manager

Doris Rojas

Email

doris@bryantresearchpark.com

Office Phone

(925) 555-1738

Work Orders

Reviews

Date	Description	Hours
10/1/2012	Lawn Upkeep	5.25
10/1/2012	Mulch	5.25
10/8/2012	Pest Spraying	6.25
10/15/2012	Lawn Upkeep	4.75
10/15/2012	Fertilizing	7.25
10/22/2012	Tree Pruning	6
10/29/2012	Shrub Trimming	7.25
11/5/2012	Lawn Upkeep	6.5
11/5/2012	Mulch	7.75
11/12/2012	Lawn Upkeep	6.25
11/12/2012	Pest Spraying	6.75
11/19/2012	Lawn Upkeep	5.5
11/19/2012	Fertilizing	6.5




[Predict hours](#)

Total Hours

675

Customers

[Back](#)

Bryant Research Park 

Jackson View Park 

Magnolia Springs Center 

Pine Executive Station

Optimum Industrial Station 

Ditmars Grove Station 

Dogwood Creek Center

Van Dam Station 

Elm Greens Park

Ditmars Research Station

Dekalb Station

Ditmars River Center

Webster Springs Station

Oak Interactive Center

Vanderbilt Stream Park

Oak Stream Park

Cedar Creek Interactive

Ainslie View Center



Customer

Bryant Research Park

Manager

Doris Rojas

Email

doris@bryantresearchpark.com

Office Phone

(925) 555-1738

Work Orders

Reviews

Date	Description	Hours
10/1/2012	Lawn Upkeep	5.25
10/1/2012	Mulch	5.25
10/8/2012	Pest Spraying	6.25
10/15/2012	Lawn Upkeep	4.75
10/15/2012	Fertilizing	7.25
10/22/2012	Tree Pruning	6
10/29/2012	Shrub Trimming	7.25
11/5/2012	Lawn Upkeep	6.5
11/5/2012	Mulch	7.75
11/12/2012	Lawn Upkeep	6.25
11/12/2012	Pest Spraying	6.75
11/19/2012	Lawn Upkeep	5.5
11/19/2012	Fertilizing	6.5



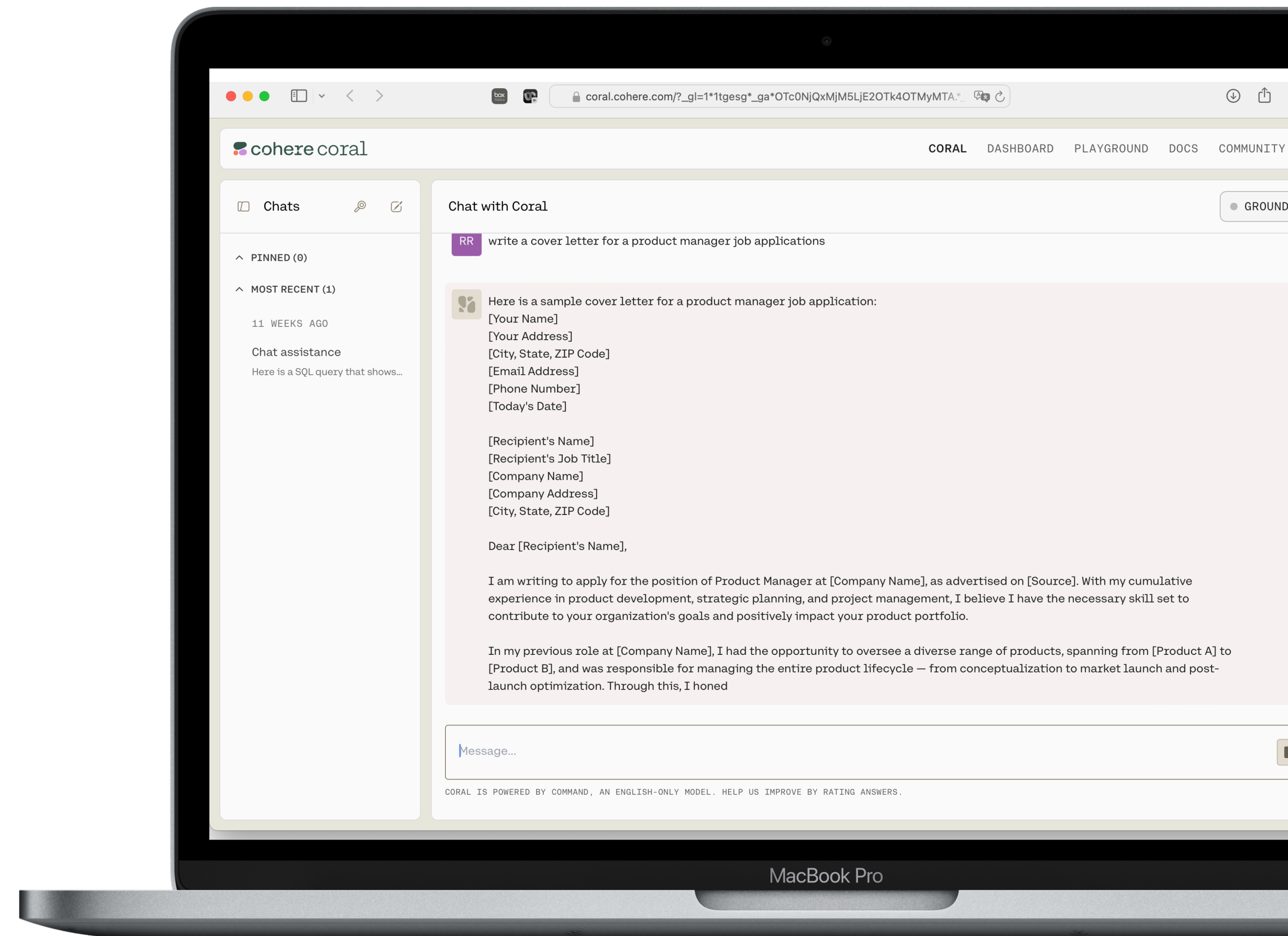
[Predict hours](#)

Total Hours

675

LLM

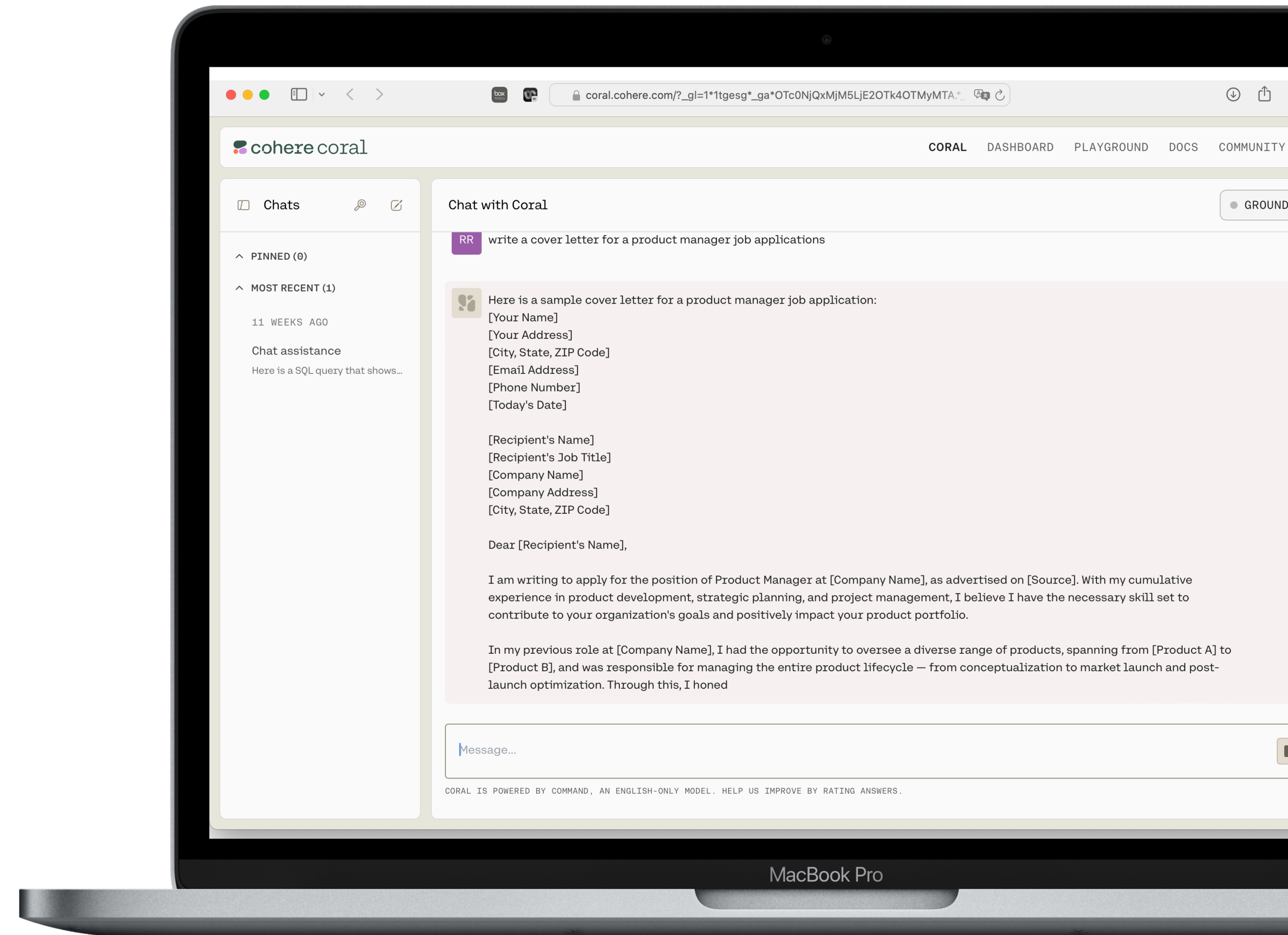
Large language model



LLM

Large language model

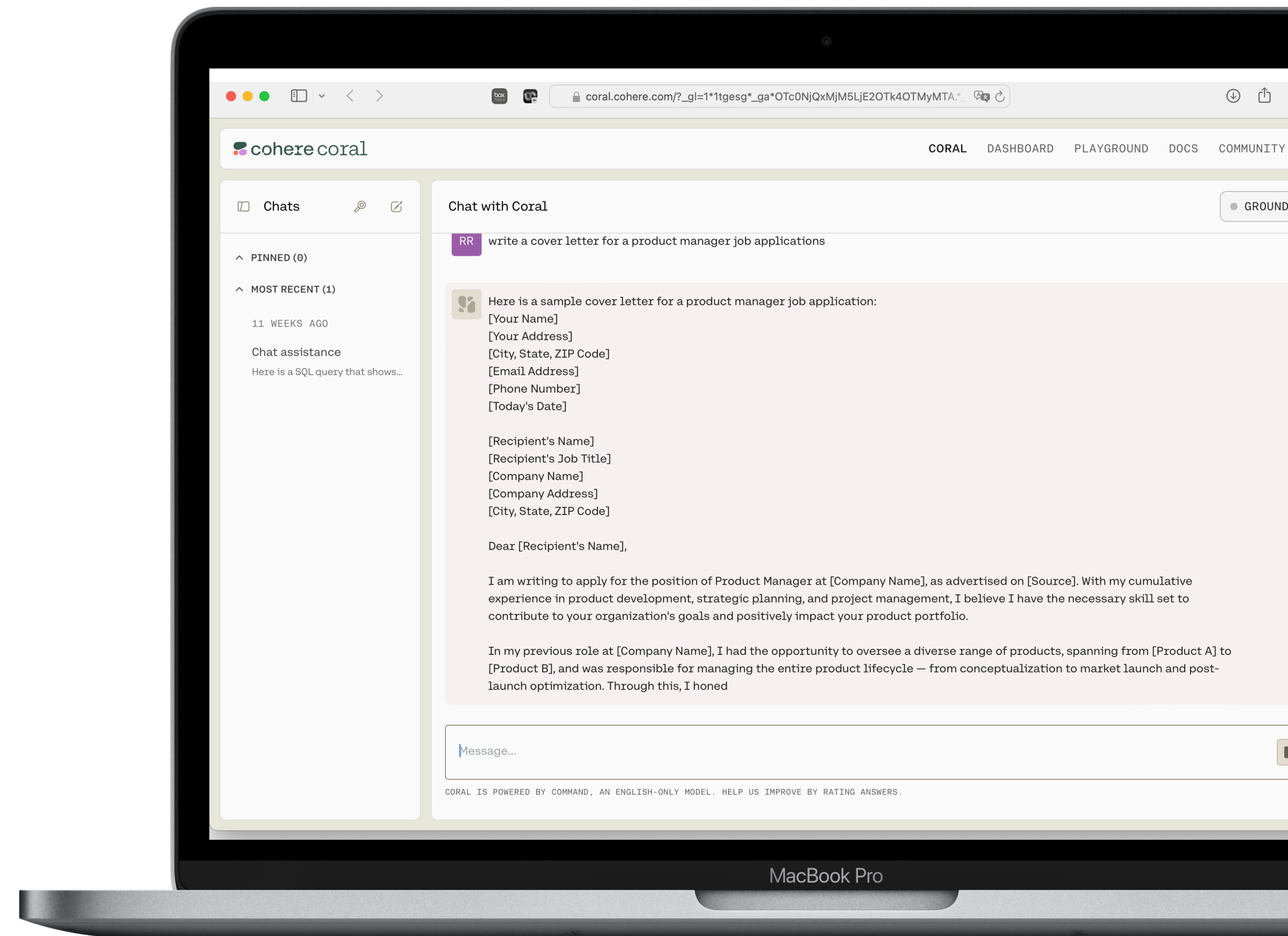
- AI-driven natural language processing



LLM

Large language model

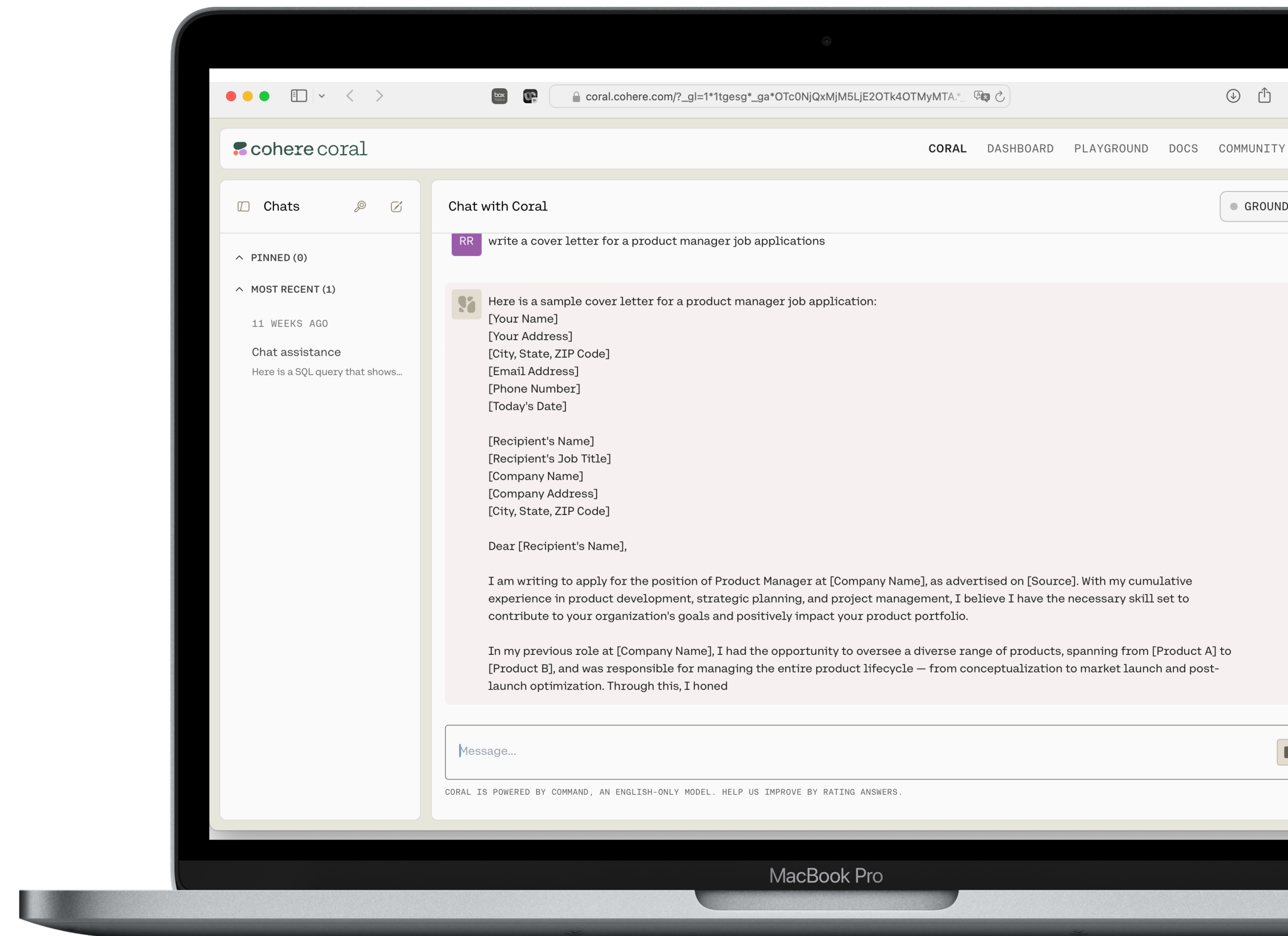
- AI-driven natural language processing
- Trained on vast text data



LLM

Large language model

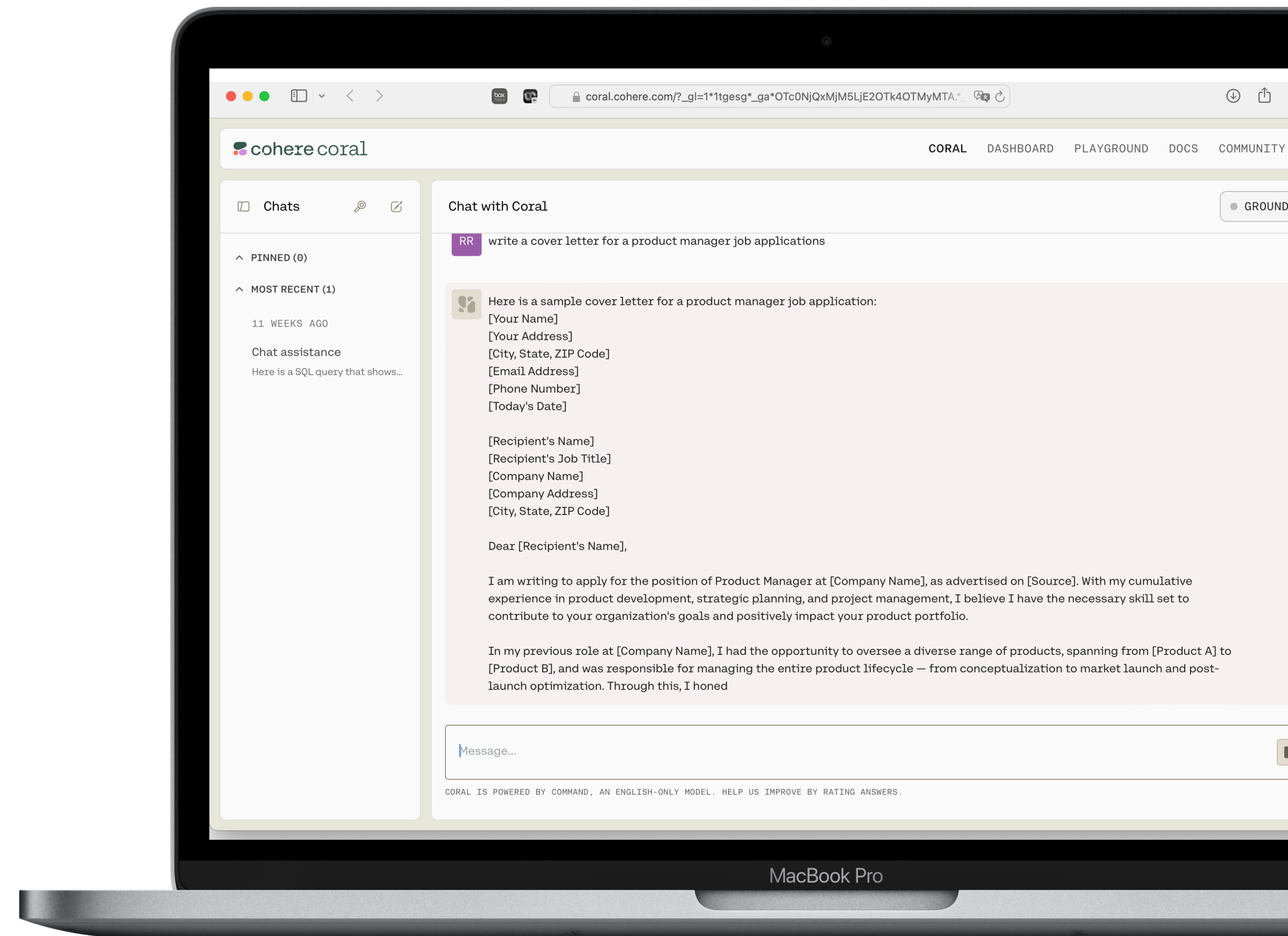
- AI-driven natural language processing
- Trained on vast text data
- Mimics human-like conversation



LLM

Large language model

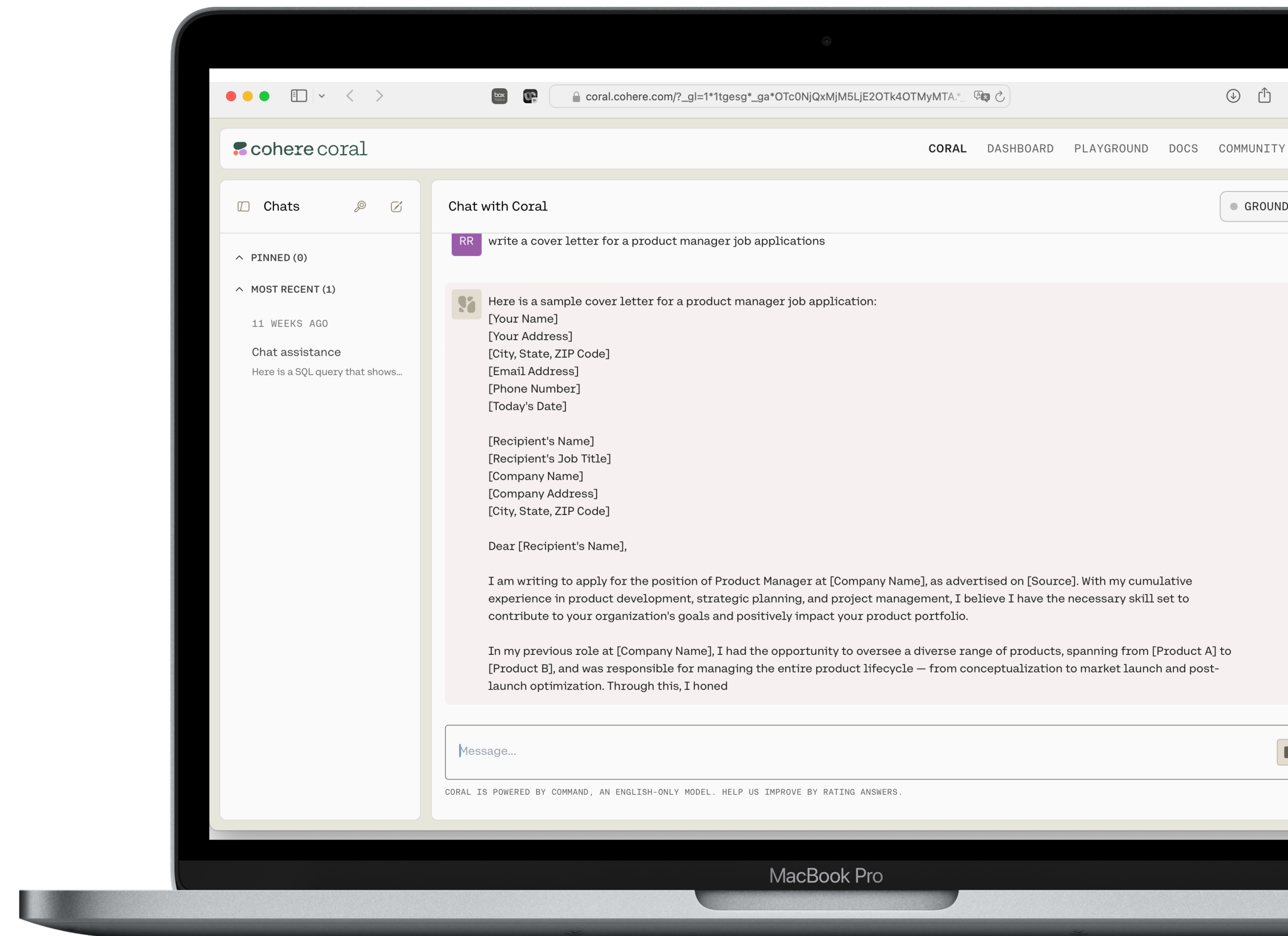
- AI-driven natural language processing
- Trained on vast text data
- Mimics human-like conversation
- Performs diverse language tasks



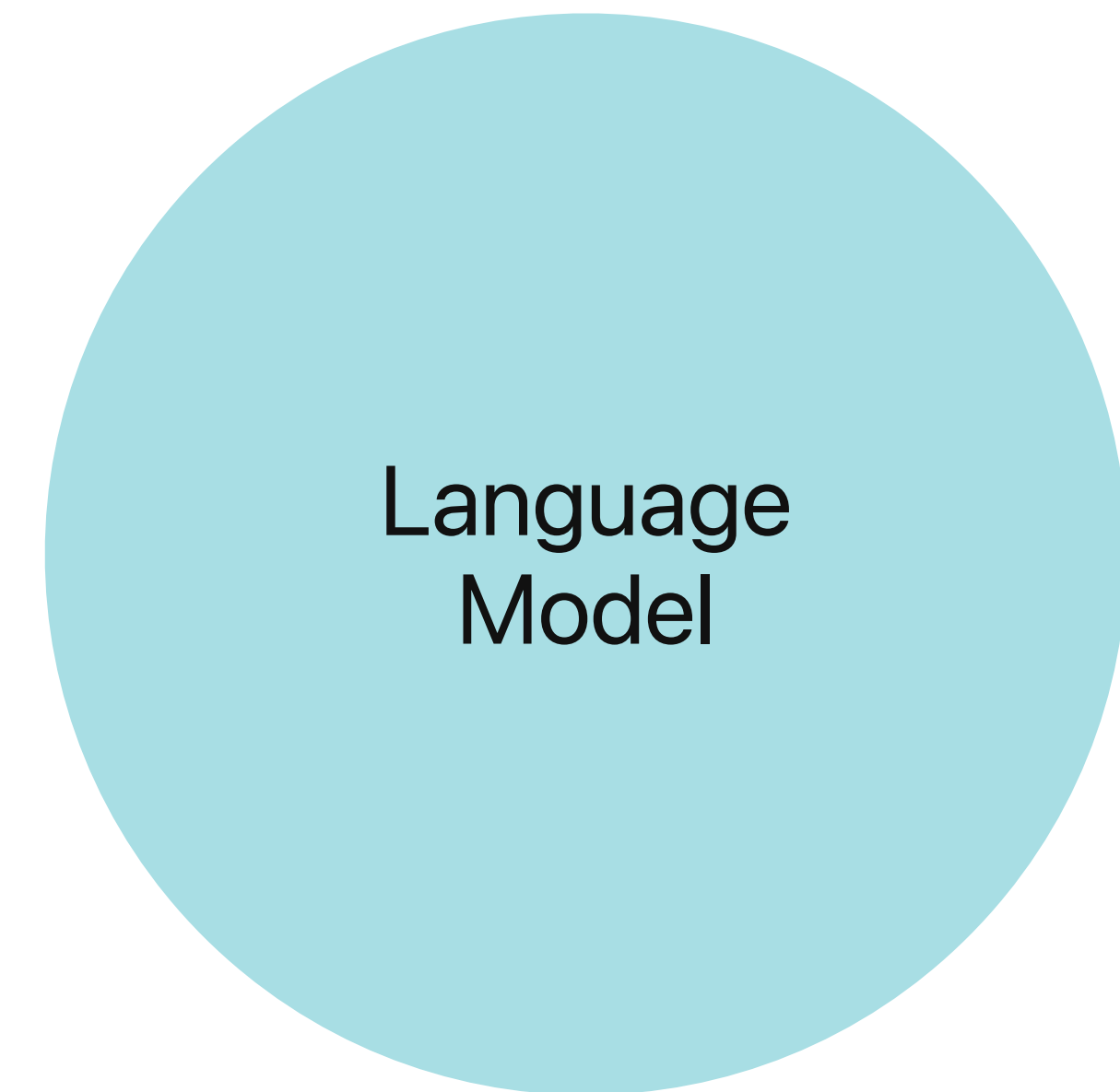
LLM

Large language model

- AI-driven natural language processing
- Trained on vast text data
- Mimics human-like conversation
- Performs diverse language tasks
- Ruled by probabilities

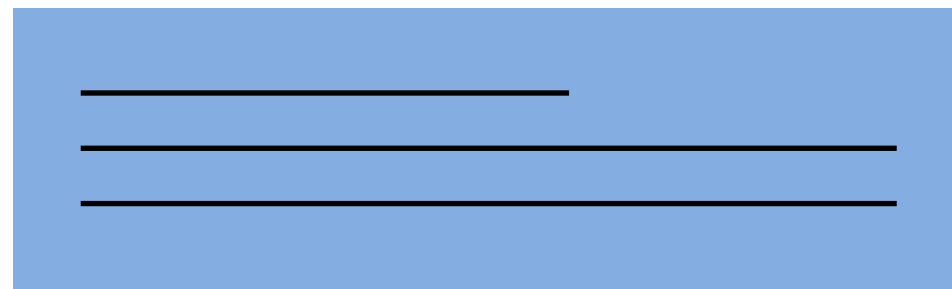


Completion



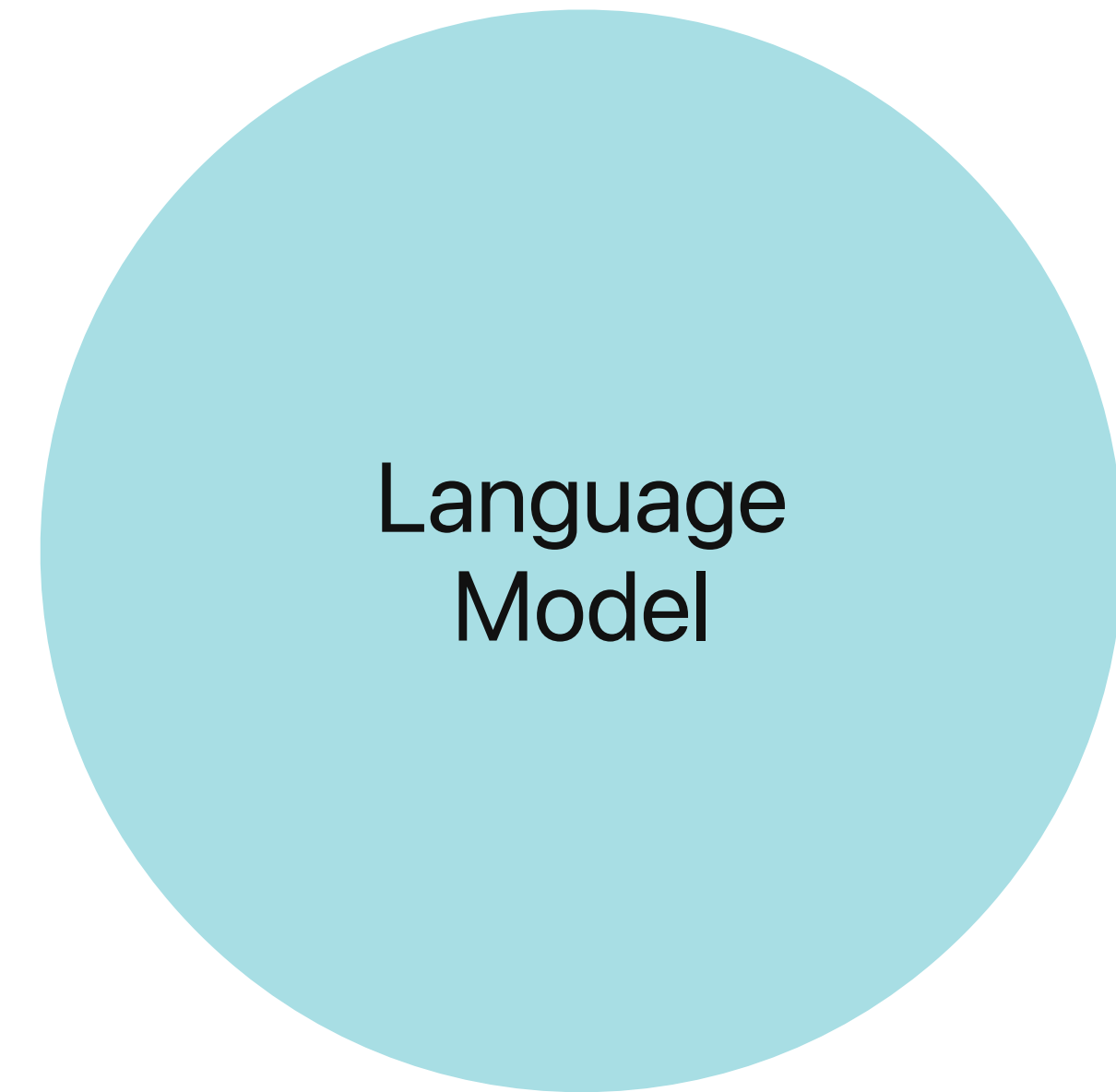
Completion

Prompt



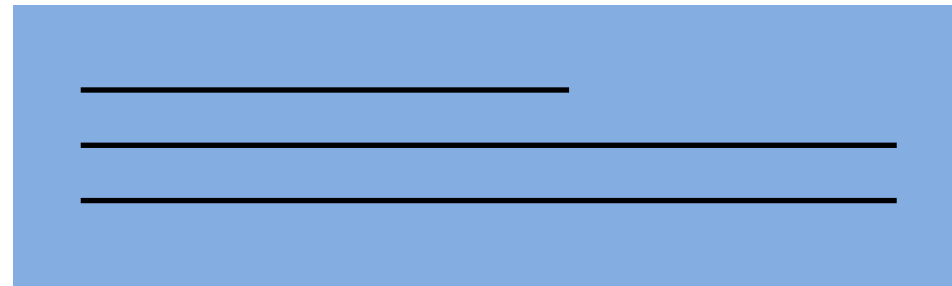
A blue rectangular box representing a prompt input field. It contains three horizontal lines for text entry.

Language
Model



Completion

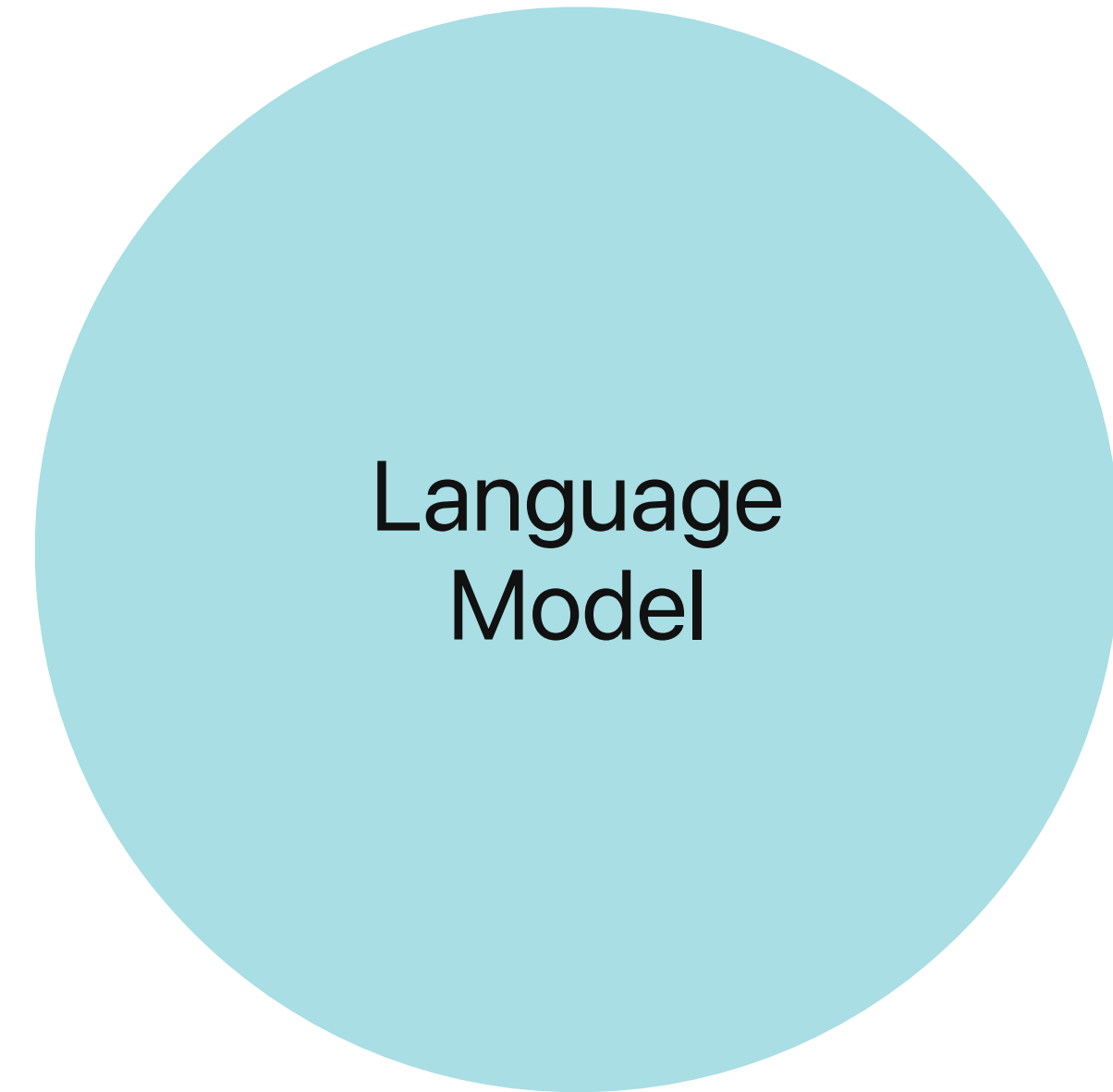
Prompt



Input

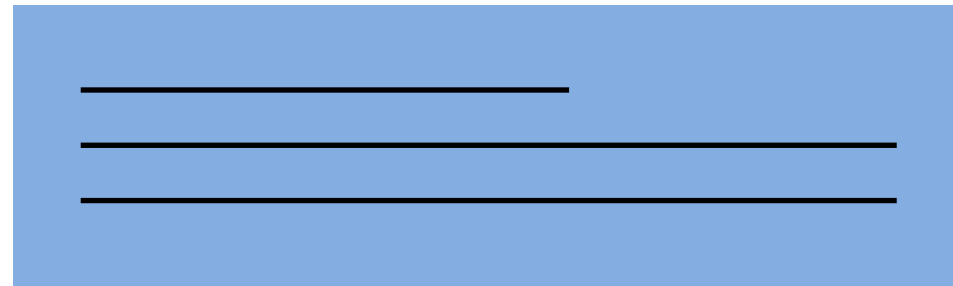


Language
Model



Completion

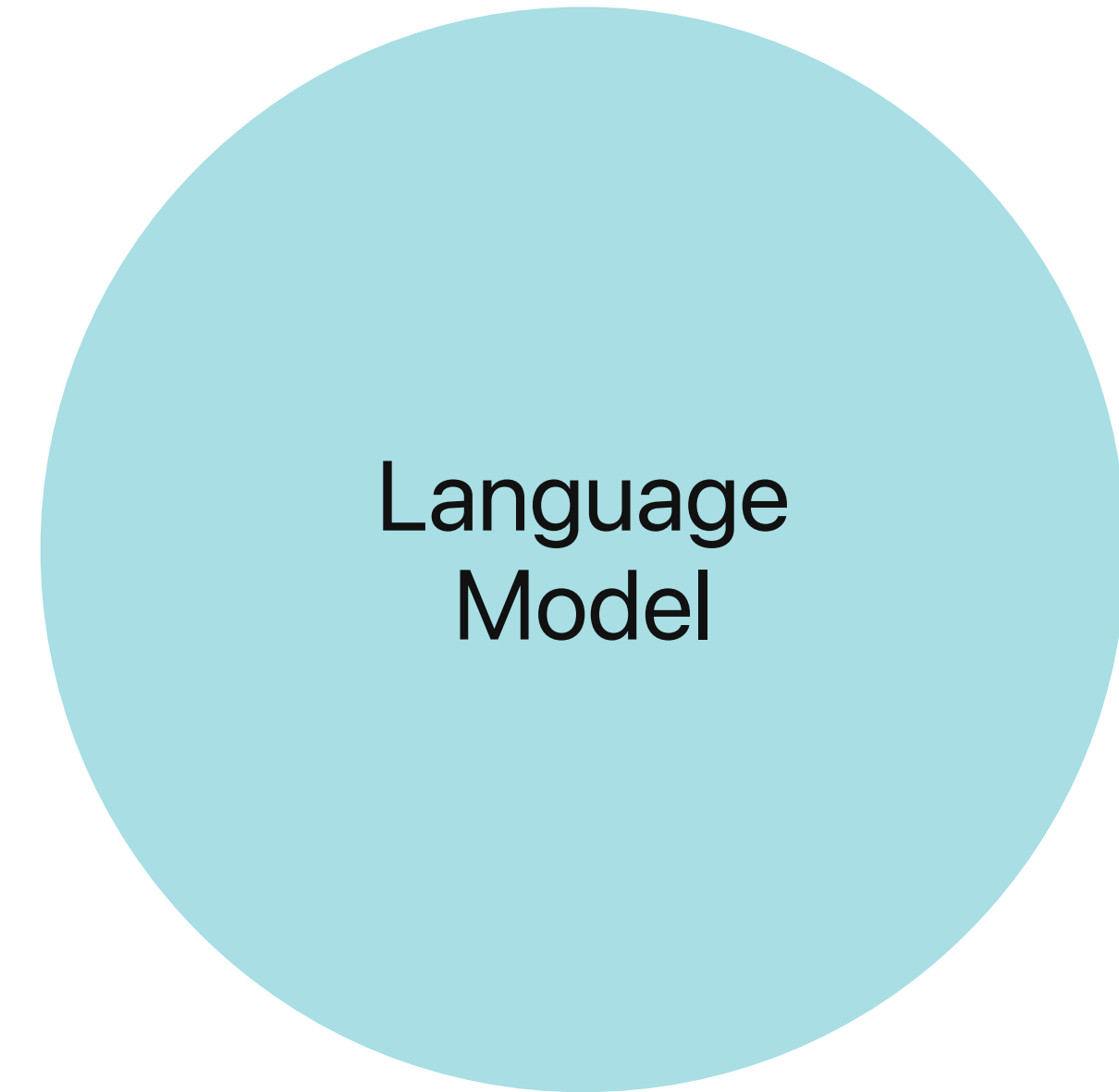
Prompt



Input



Language
Model

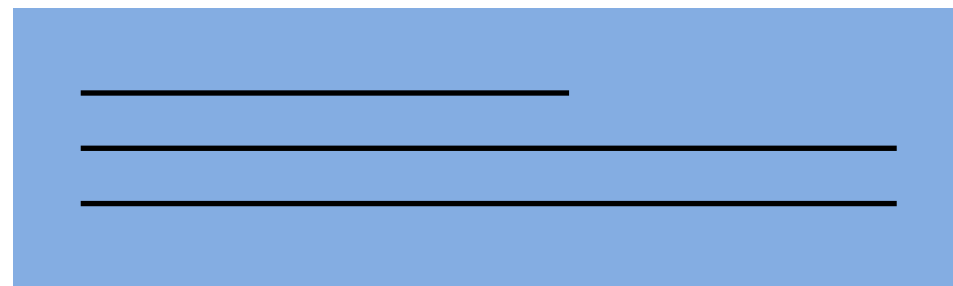


Output



Completion

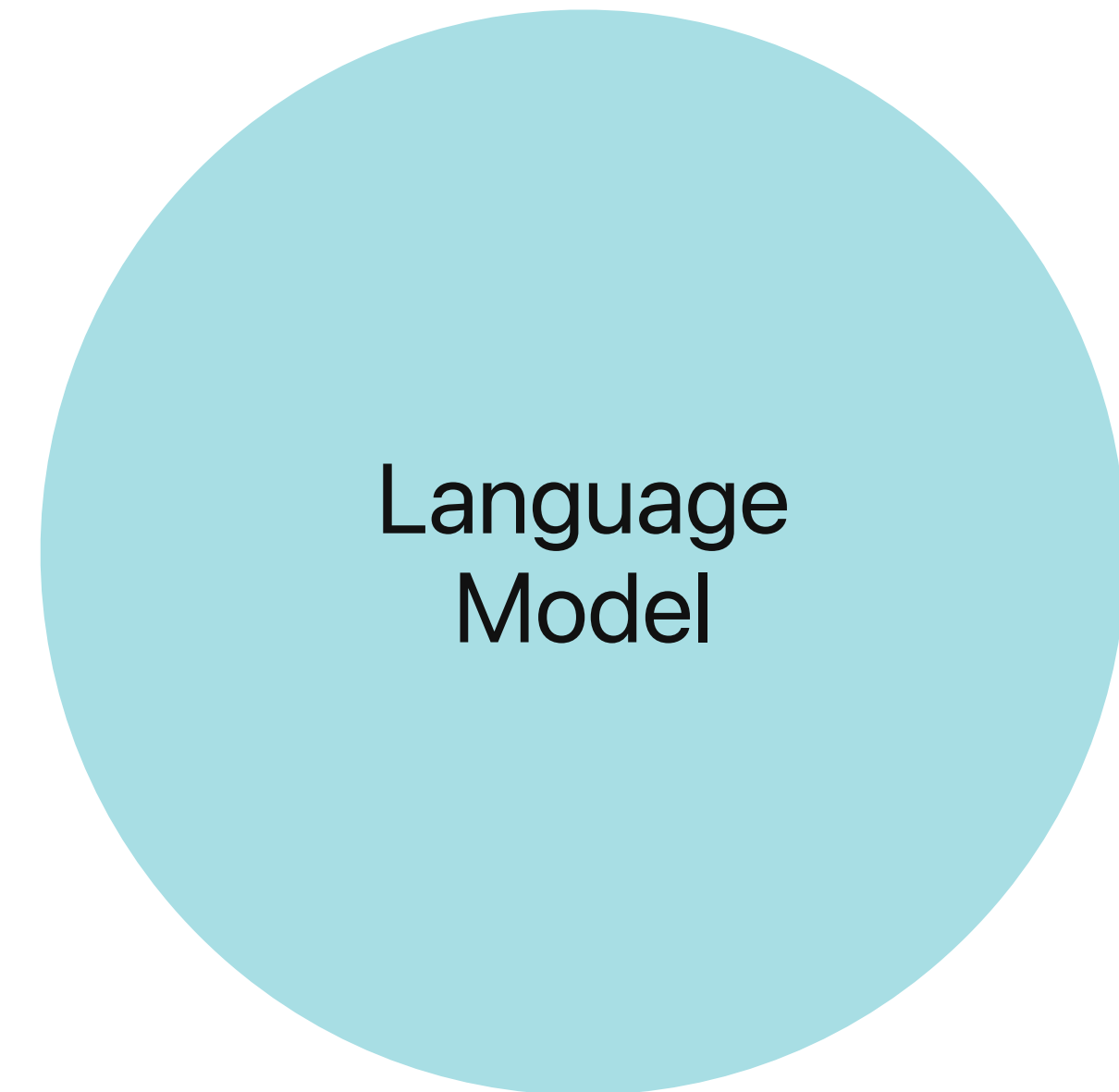
Prompt



Input



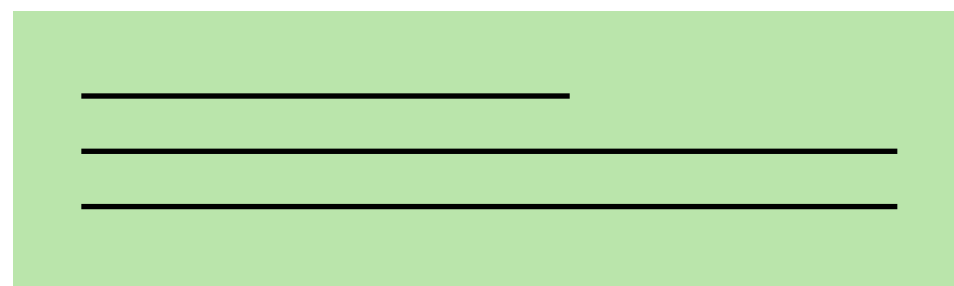
Language
Model



Output



Completion



Completion

Prompt

Completion

Completion

Prompt

"In simple terms, photosynthesis is..."

Completion

Completion

Prompt

"In simple terms, photosynthesis is..."

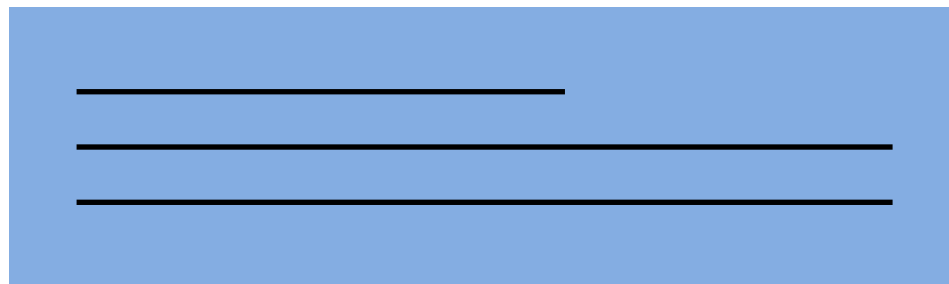
Completion

"...how plants make their food using sunlight."

Chat completion API

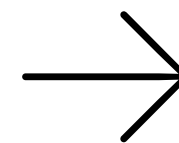
Chat completion API

Prompt



Chat completion API

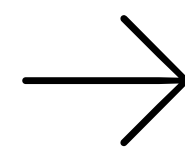
Prompt



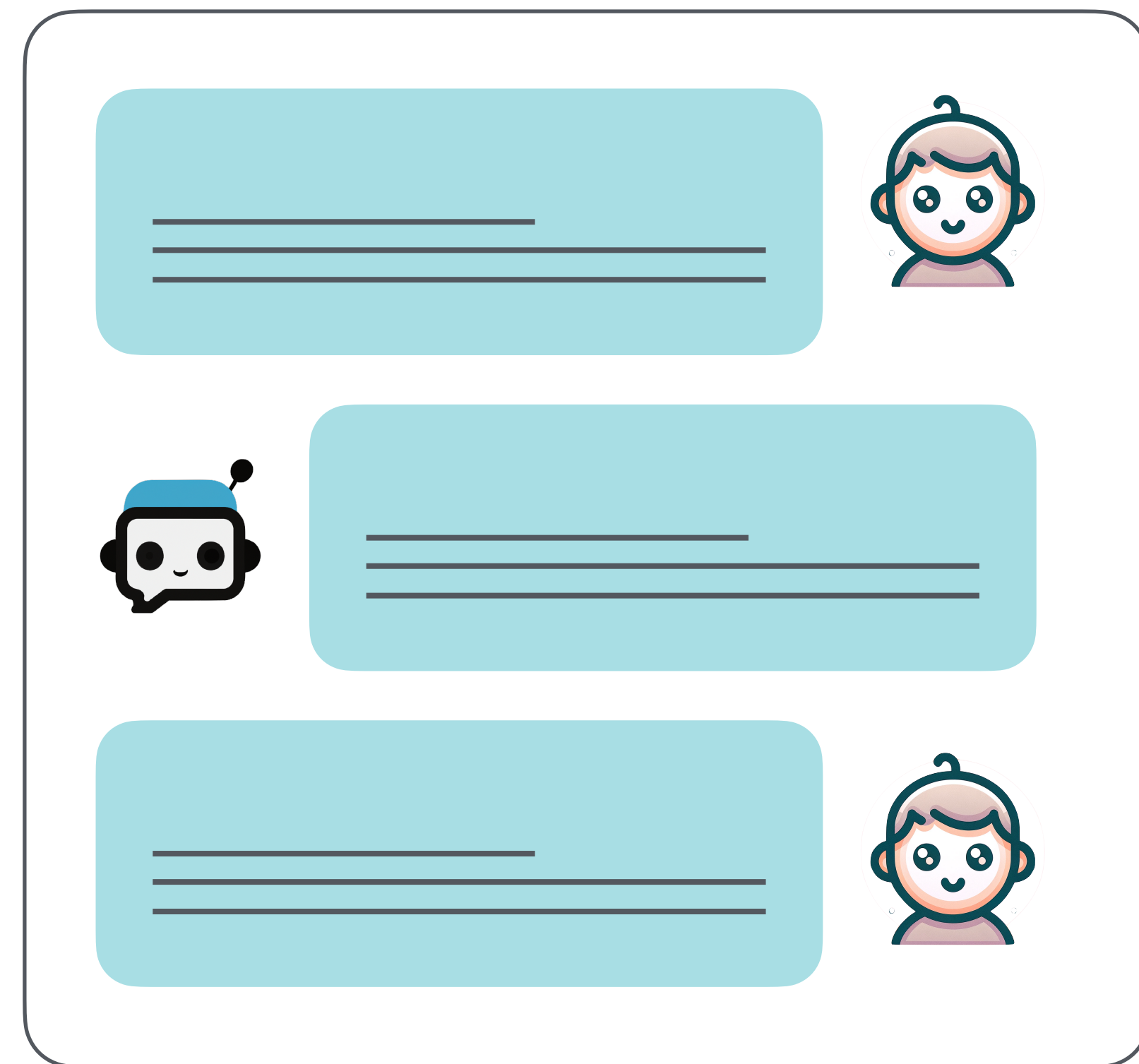
Messages



Prompt

A blue rectangular box containing three horizontal lines, representing a text input field for a prompt.

Messages

A large rounded rectangle containing three chat messages. The first and third messages are from a user (represented by a person icon) and are on the right side. The second message is from a bot (represented by a robot icon) and is on the left side. Each message is in a light blue box with three horizontal lines for text.

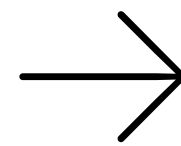
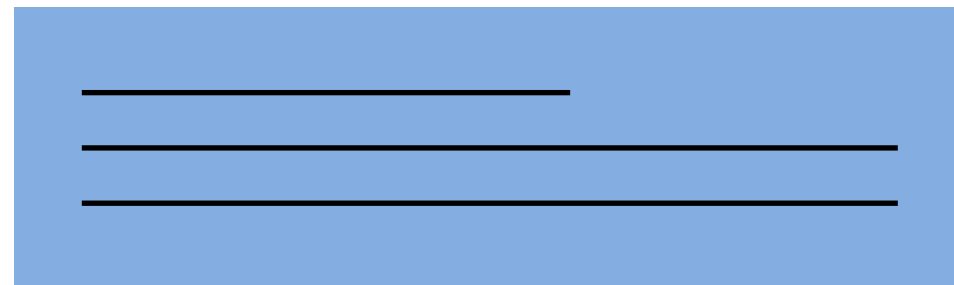
Messages

User: [Message 1]

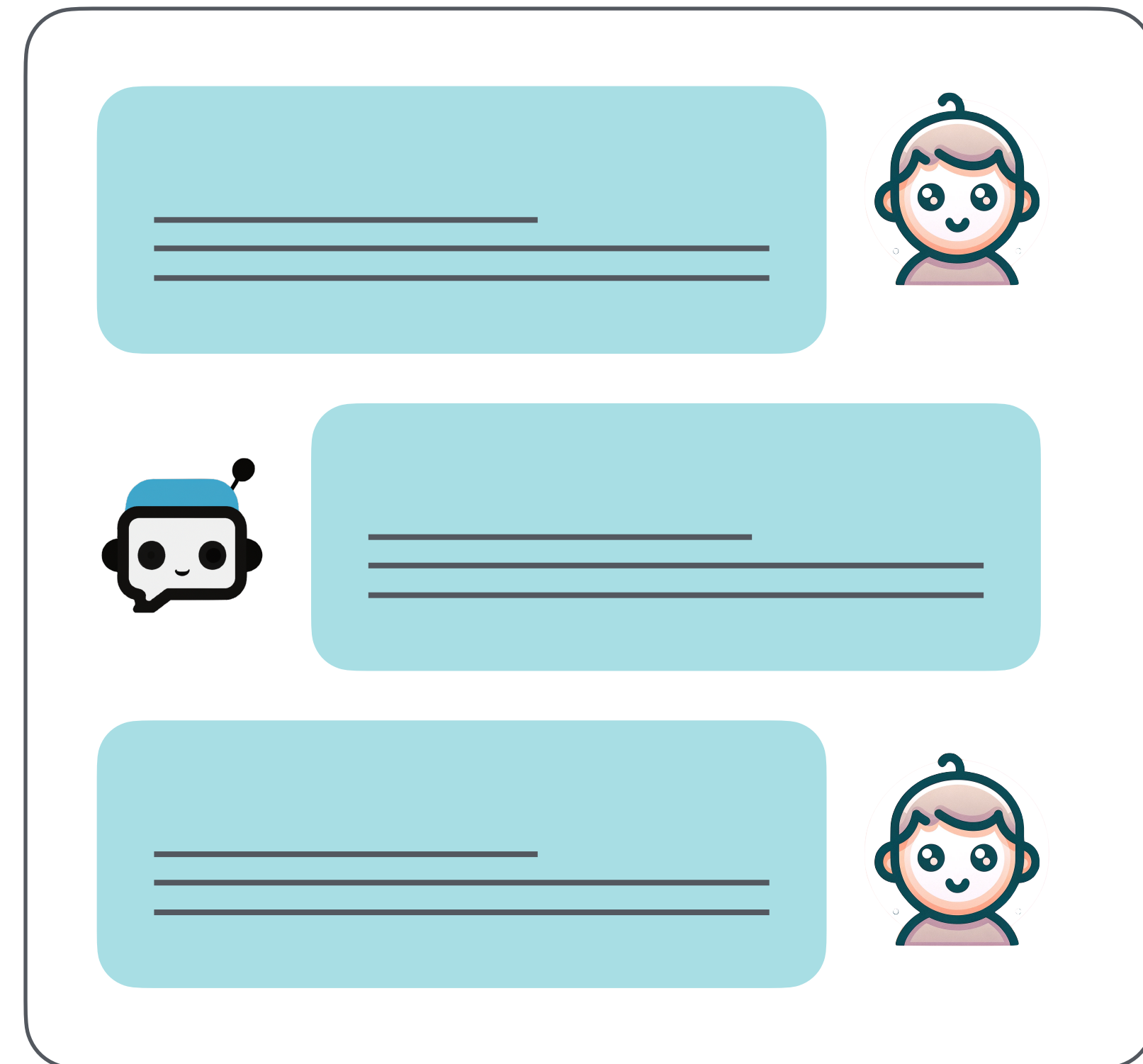
Bot: [Message 2]

User: [Message 3]

Prompt



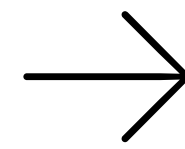
Messages



The 'Messages' container shows a sequence of three messages:

- Message 1: A light blue bubble on the right containing three horizontal lines, with a person icon to its right.
- Message 2: A light blue bubble on the left containing three horizontal lines, with a robot icon to its left.
- Message 3: A light blue bubble on the right containing three horizontal lines, with a person icon to its right.

Prompt

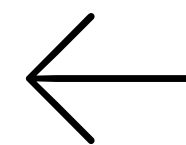


Messages

A cartoon icon of a person with brown hair and a purple shirt.

A cartoon icon of a robot with a blue head and a white body.

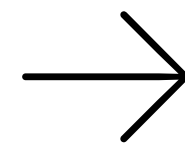
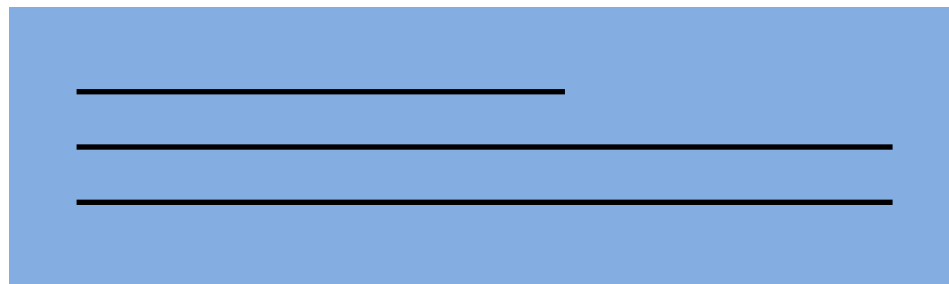
Completion



A cartoon icon of a robot with a blue head and a white body.

Chat completion API

Prompt



Messages

```
[  
  { "role" : "system", "content" : "Act  
like a 5th grade science teacher" },  
  { "role" : "user", "content" : "What is  
photosynthesis?" },  
  { "role" : "assistant", "content" :  
    "how plants make their food using  
sunlight" }  
]
```

Chat completion API

Messages

```
[  
  { "role" : "system", "content" : "Act like a 5th grade science teacher" },  
  { "role" : "user", "content" : "What is photosynthesis?" },  
  { "role" : "assistant", "content" : "how plants make their food using sunlight" }  
]
```

Chat completion API

Roles

Chat completion API

Roles

User

Request from human.

Chat completion API

Roles

User

Request from human.

Assistant

Response from the model.

Chat completion API

Roles

User

Request from human.

Assistant

Response from the model.

System

High-level instructions to define context or behavior.

AI in FileMaker

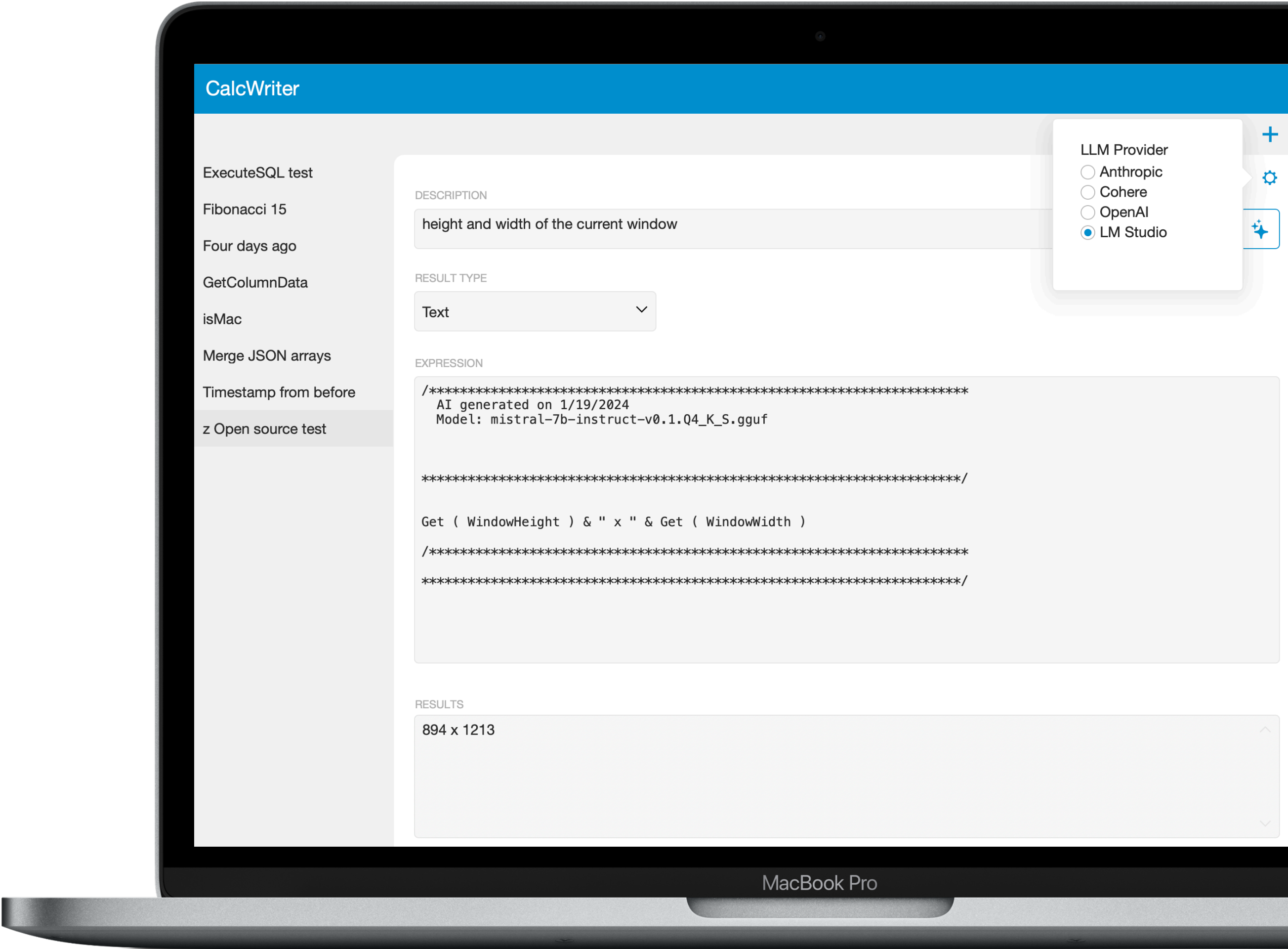
Generate calculations

Find by natural language

Generating charts

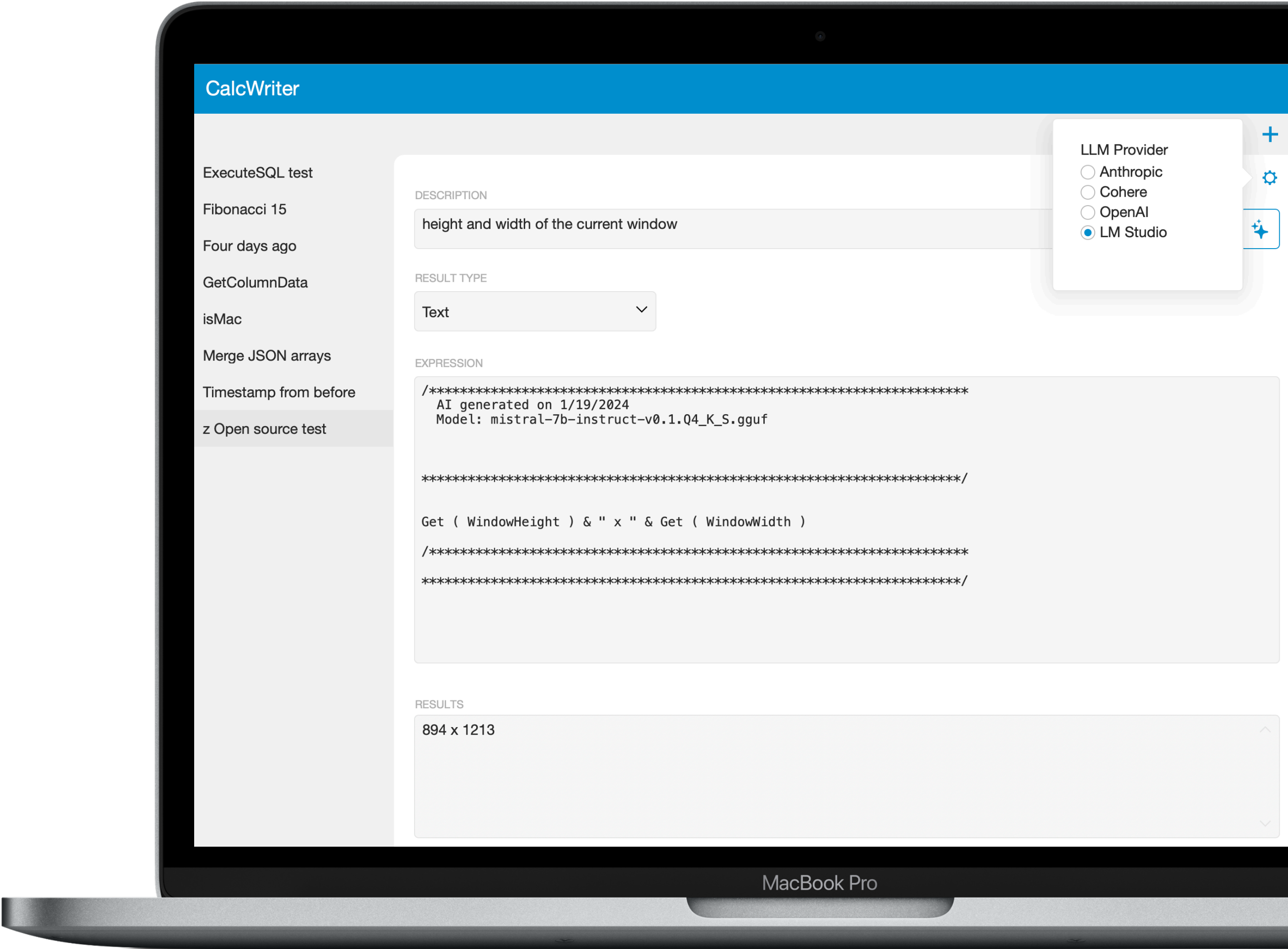


CalcWriter



CalcWriter

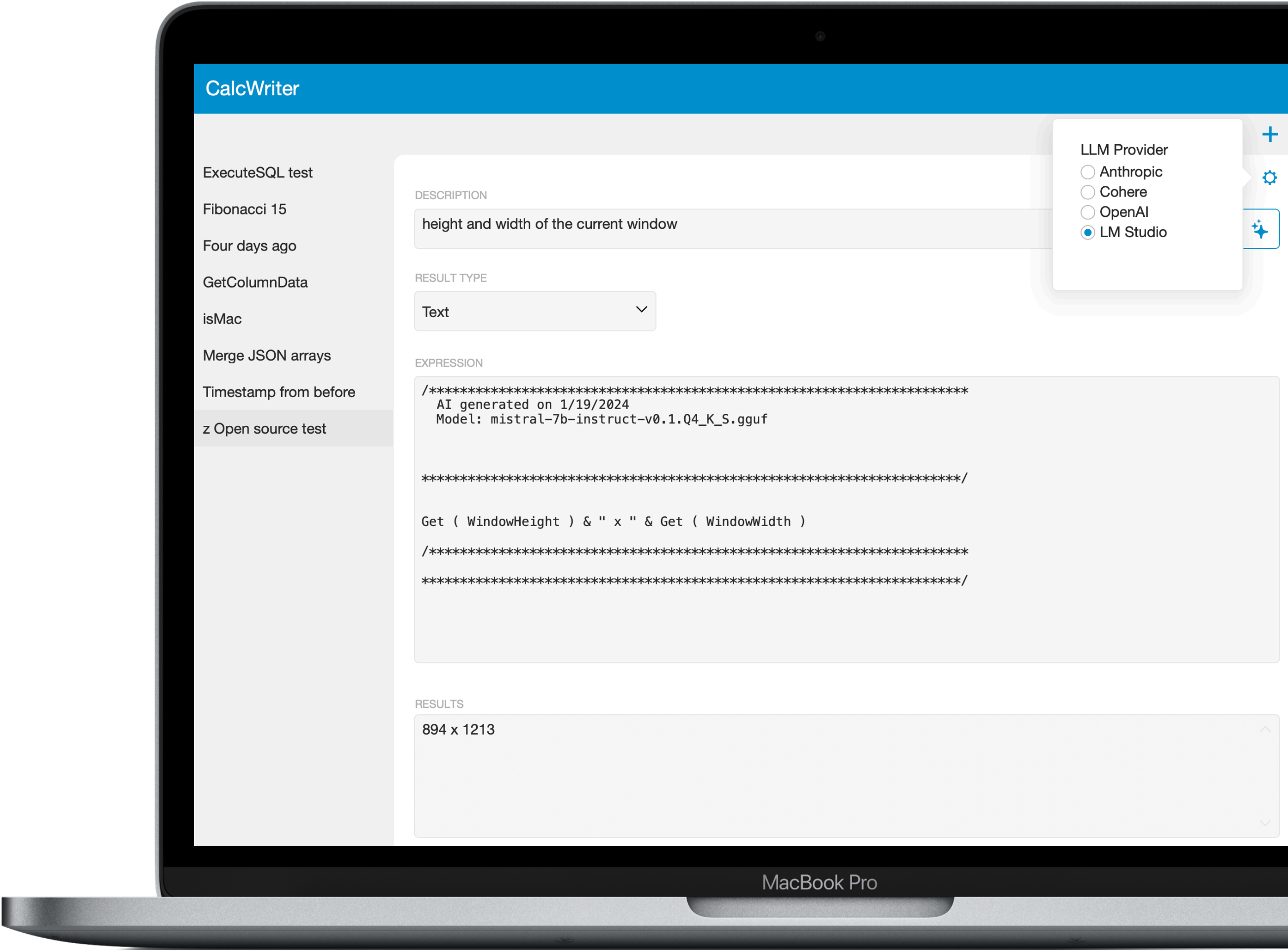
Generate calculations



CalcWriter

Generate calculations

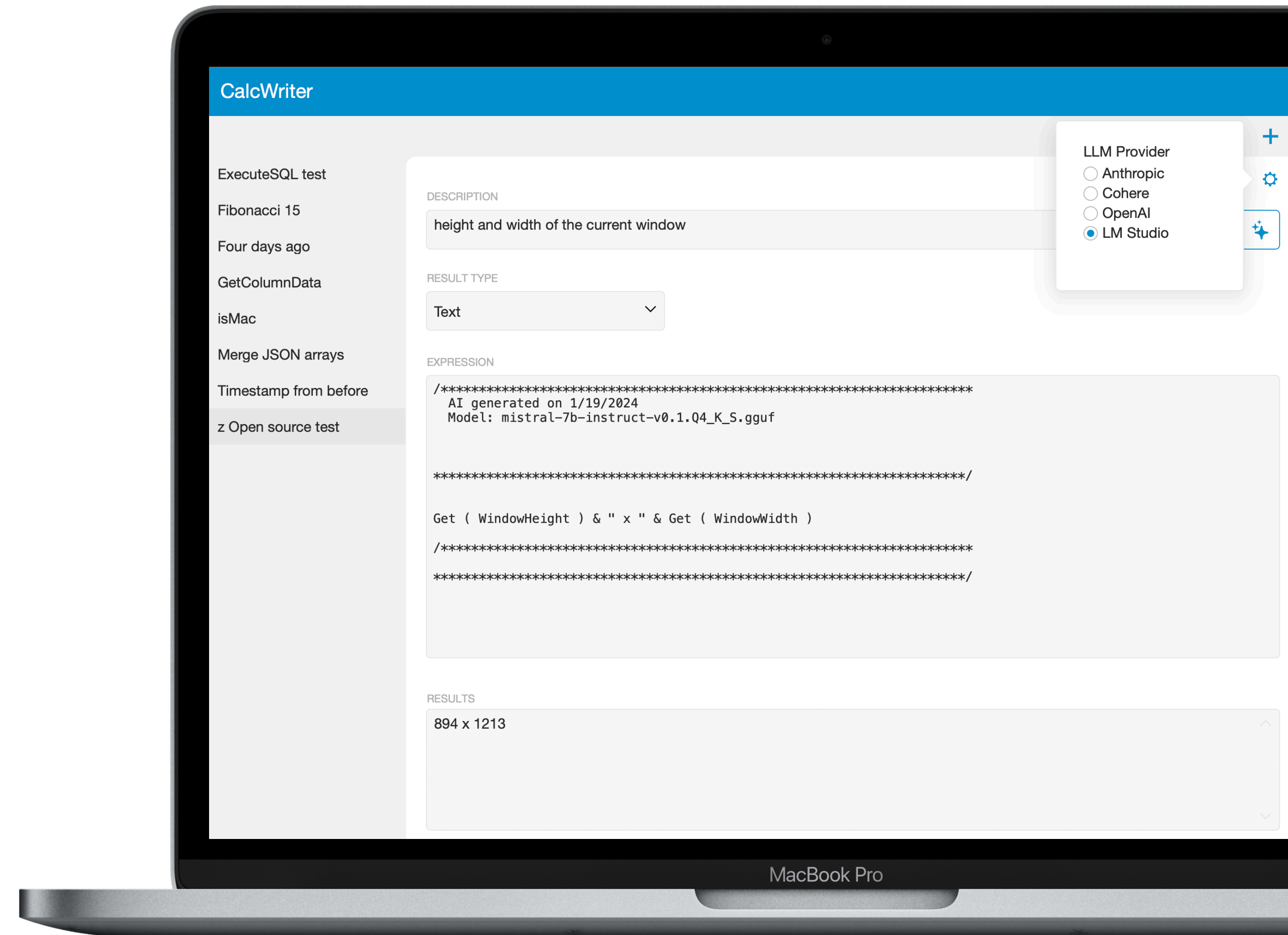
- Focus on answer



CalcWriter

Generate calculations

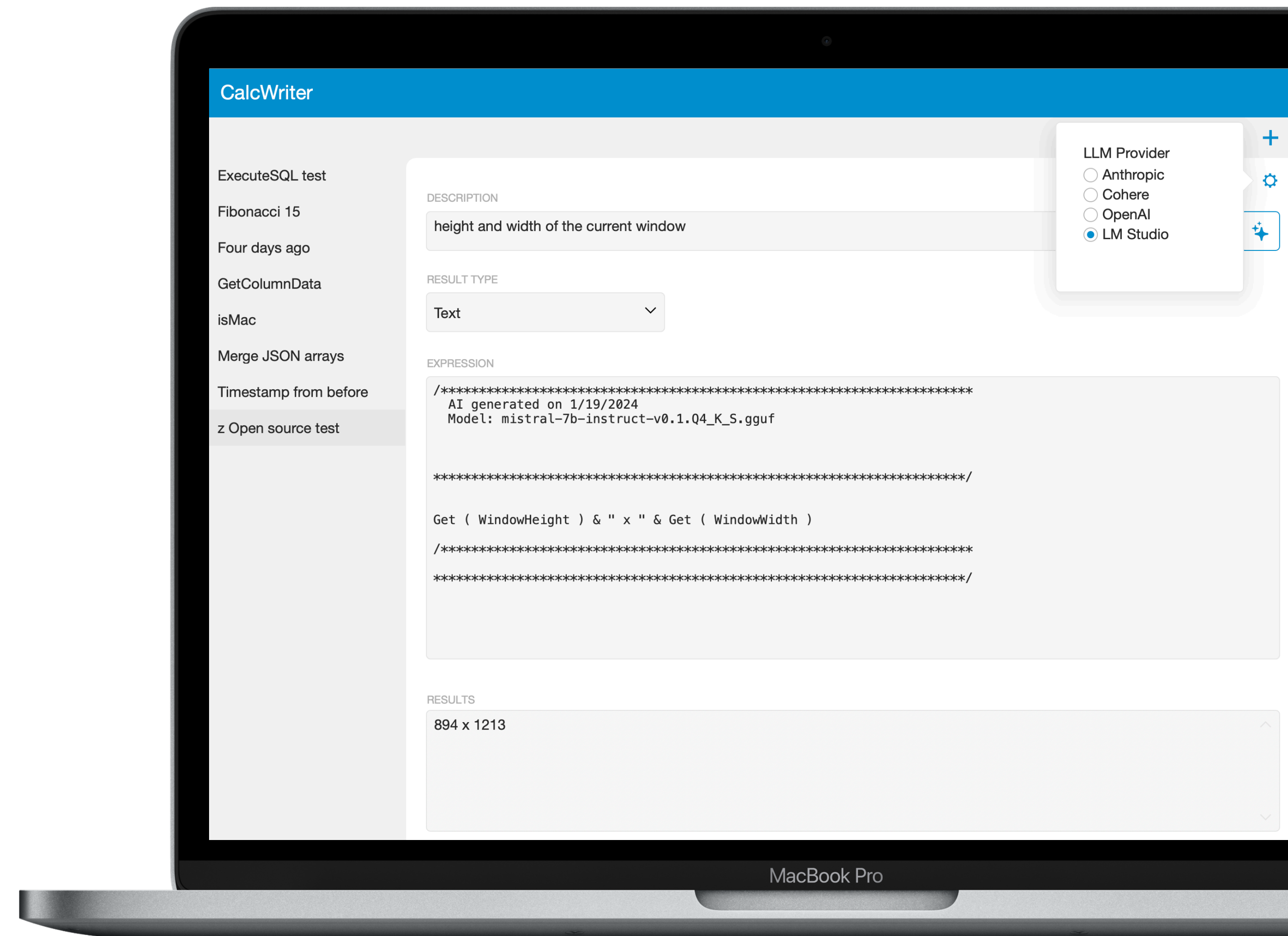
- Focus on answer
- No recursion



CalcWriter

Generate calculations

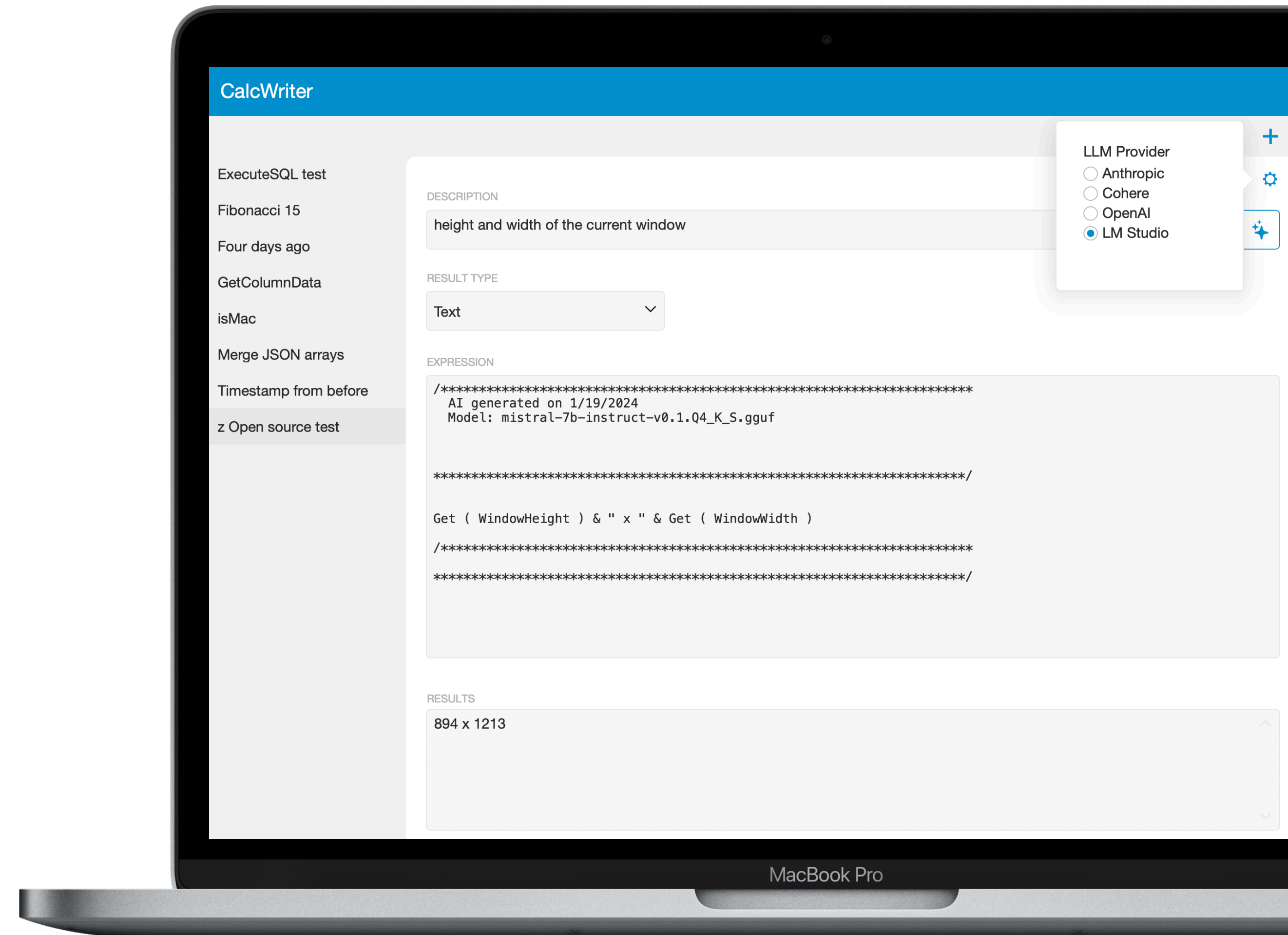
- Focus on answer
- No recursion
- Self-contained



CalcWriter

Generate calculations

- Focus on answer
- No recursion
- Self-contained
- Use in-line comments





CalcWriter

Timestamp from before

Timestamp of 7 hours ago

Timestamp

```

/*****
AI generated on 10/15/2023

```

*****/

```
Get ( CurrentTimestamp ) - 7 * 3600
```

/*****

*****/



CalcWriter

Timestamp from before

Timestamp of 7 hours ago

Timestamp

```

/*****
AI generated on 10/15/2023

```

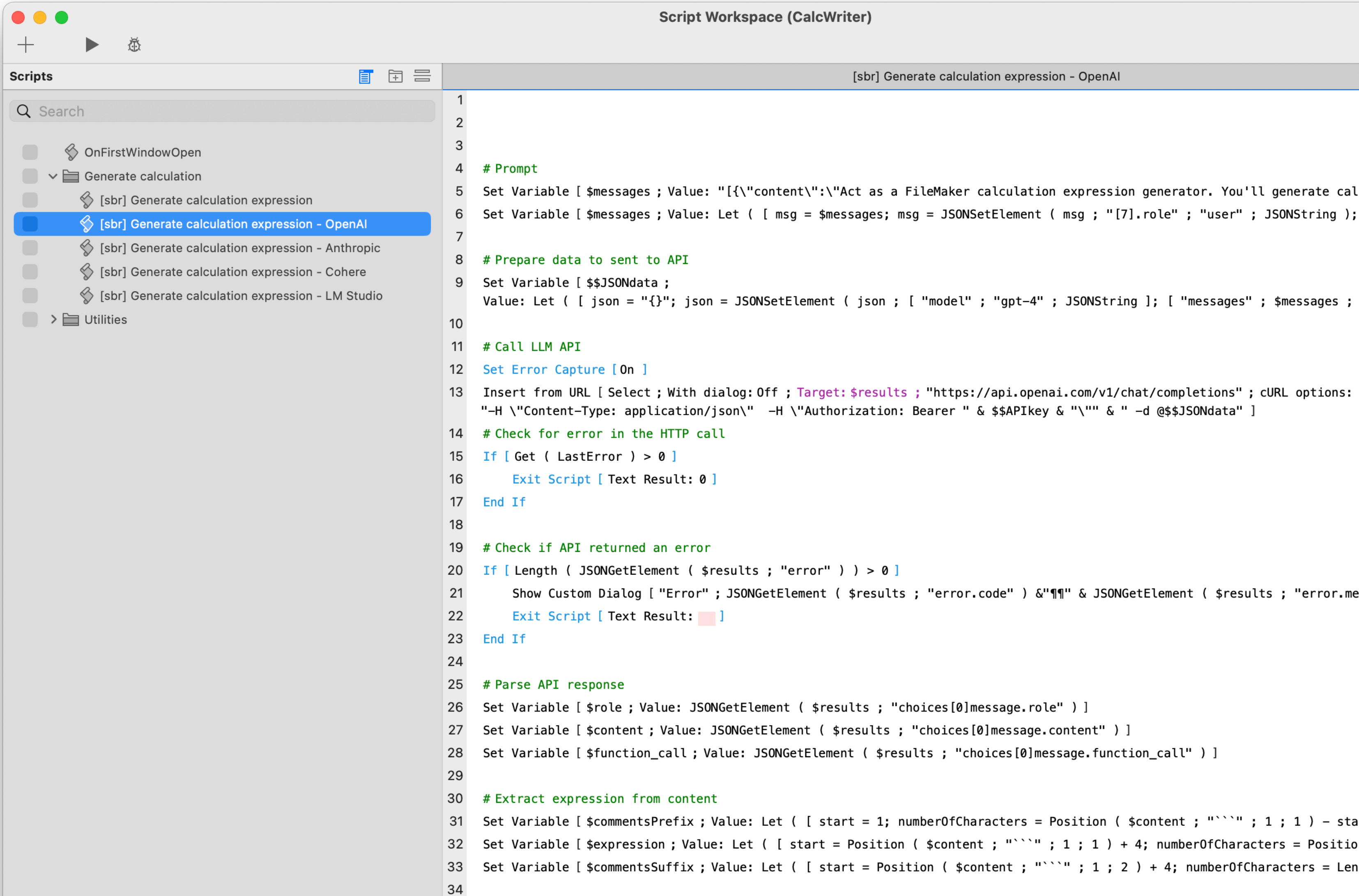
*****/

```
Get ( CurrentTimestamp ) - 7 * 3600
```

/*****

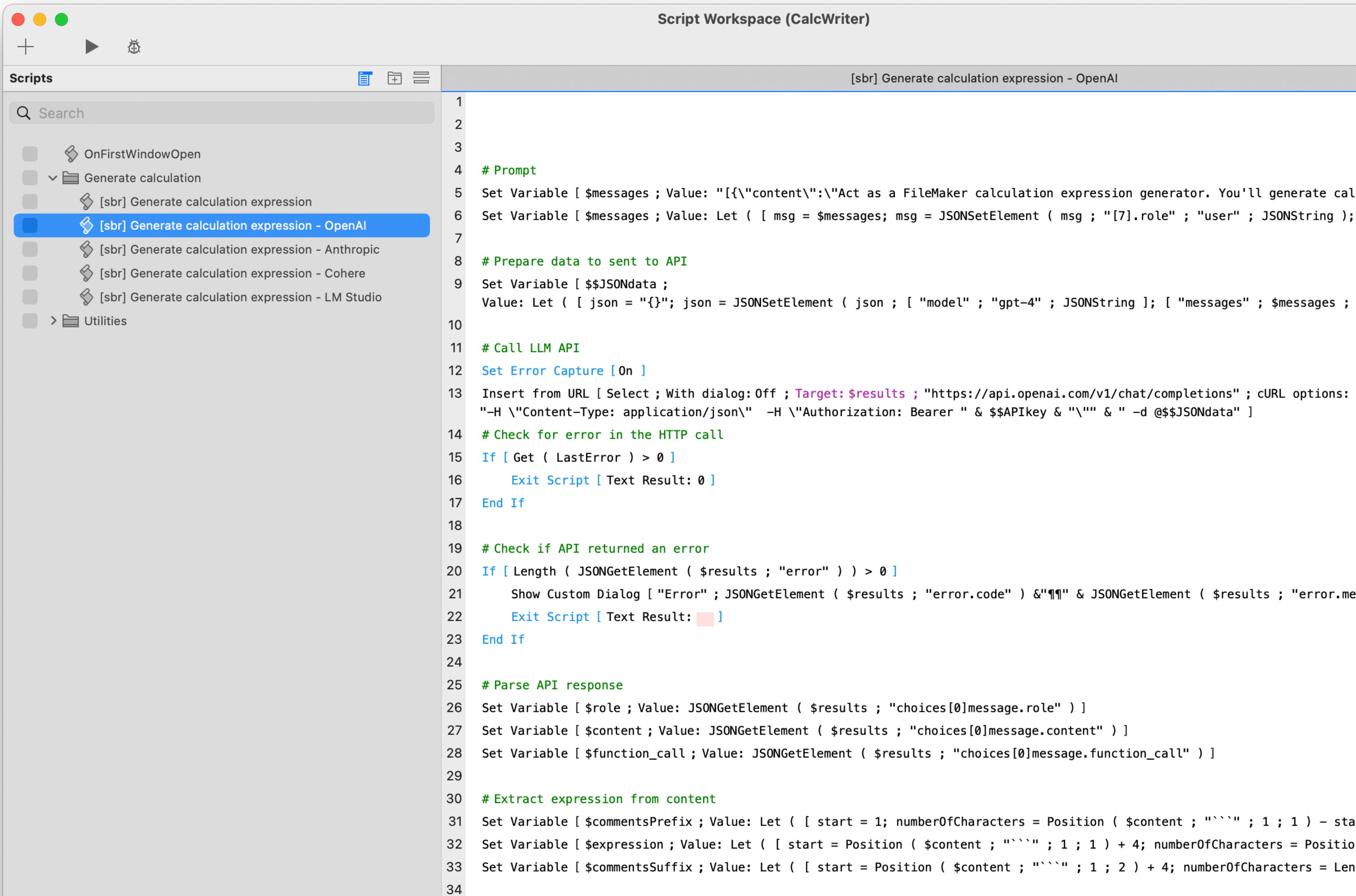
*****/

Script



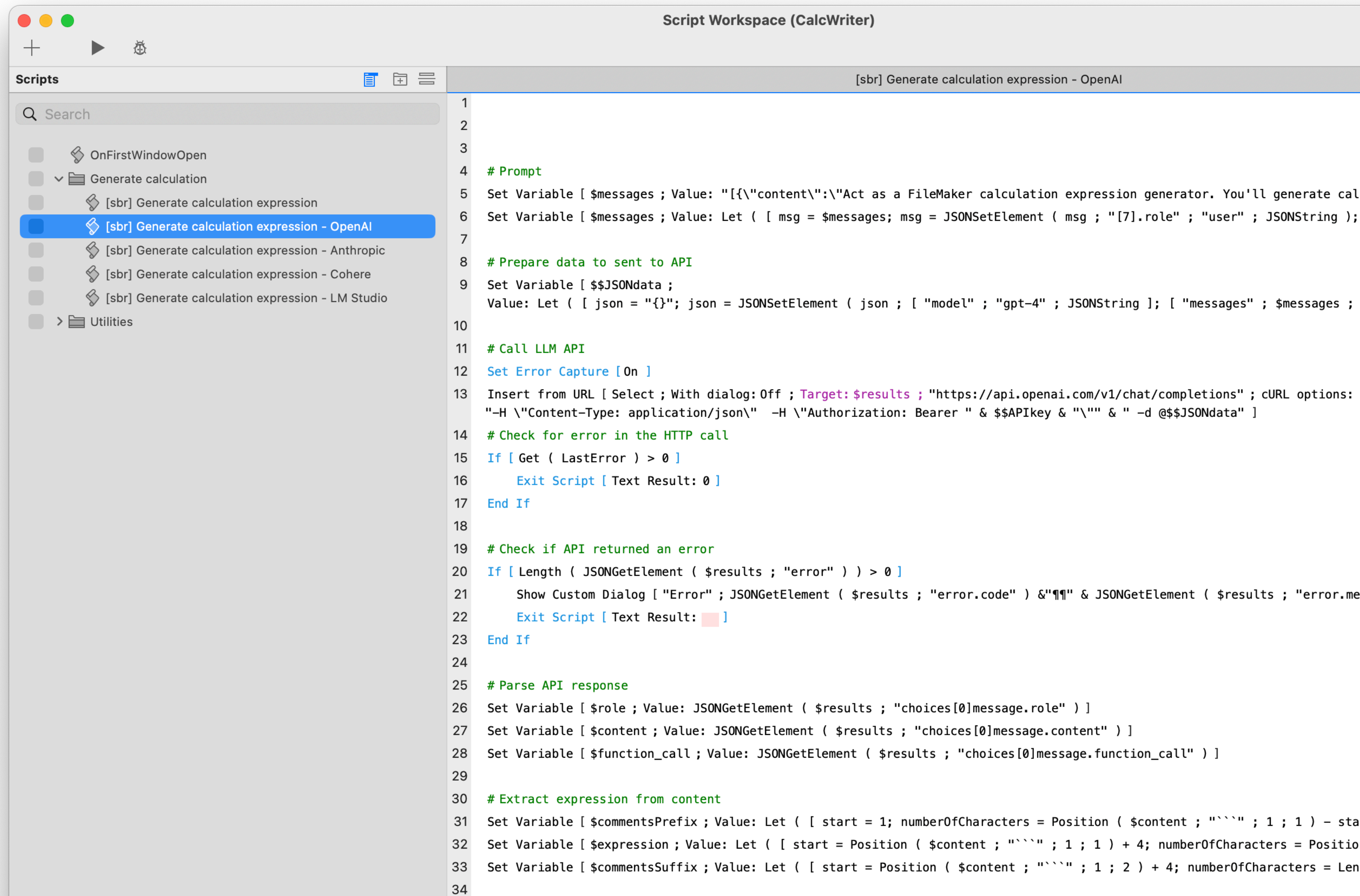
Script

- Less than 30 lines

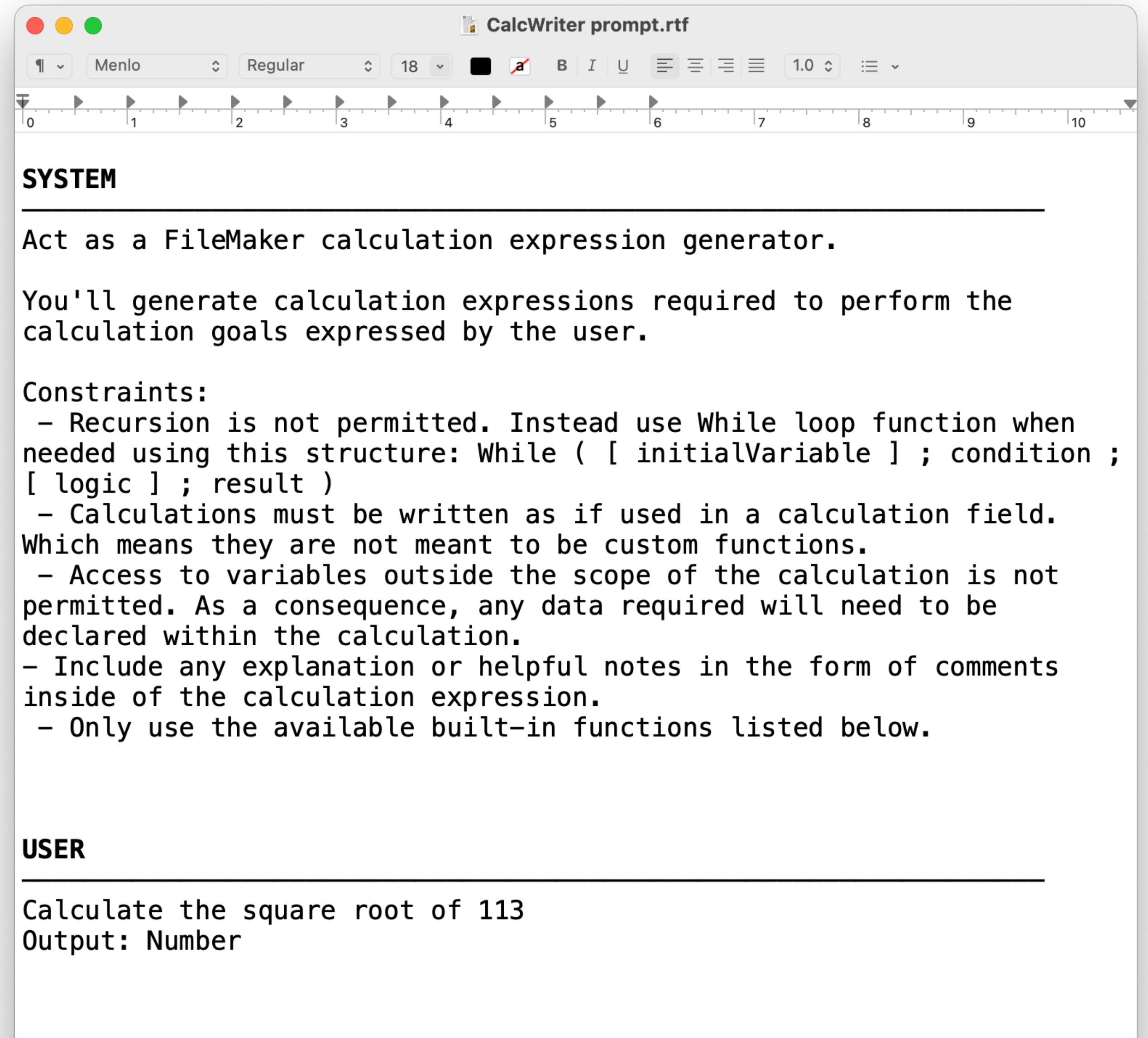


Script

- Less than 30 lines
- Prompt is the secret

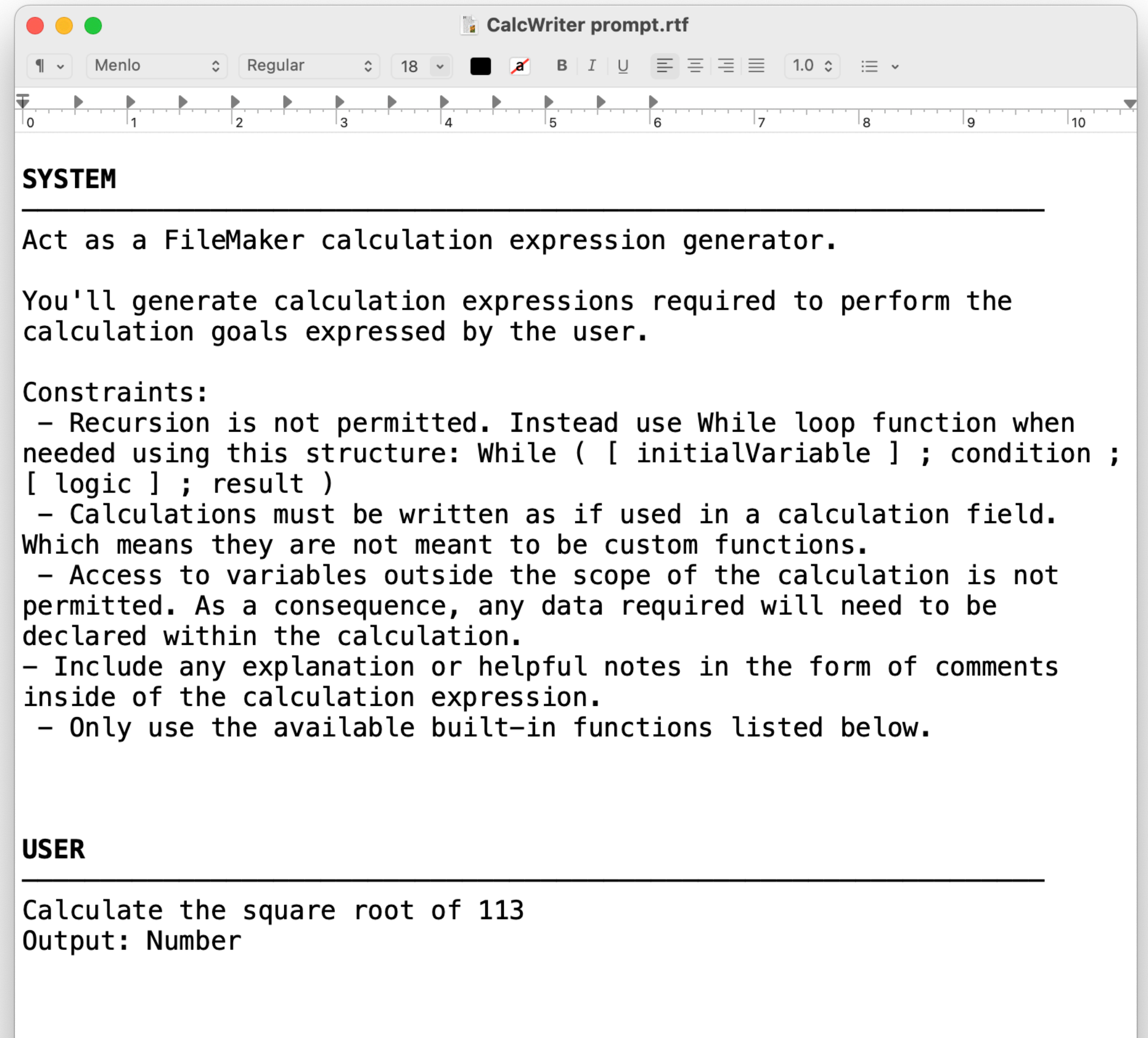


Prompt



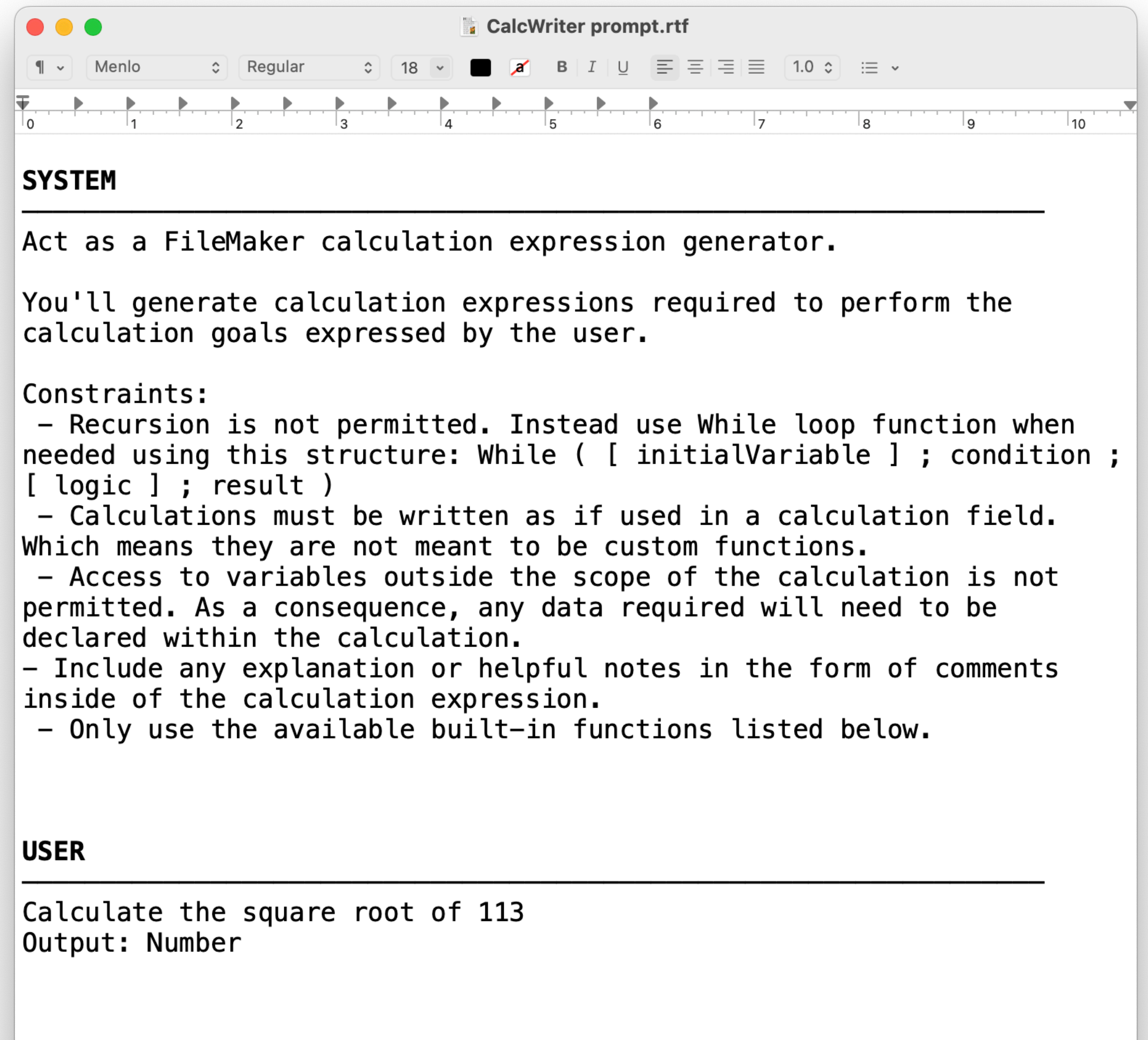
Prompt

- Role



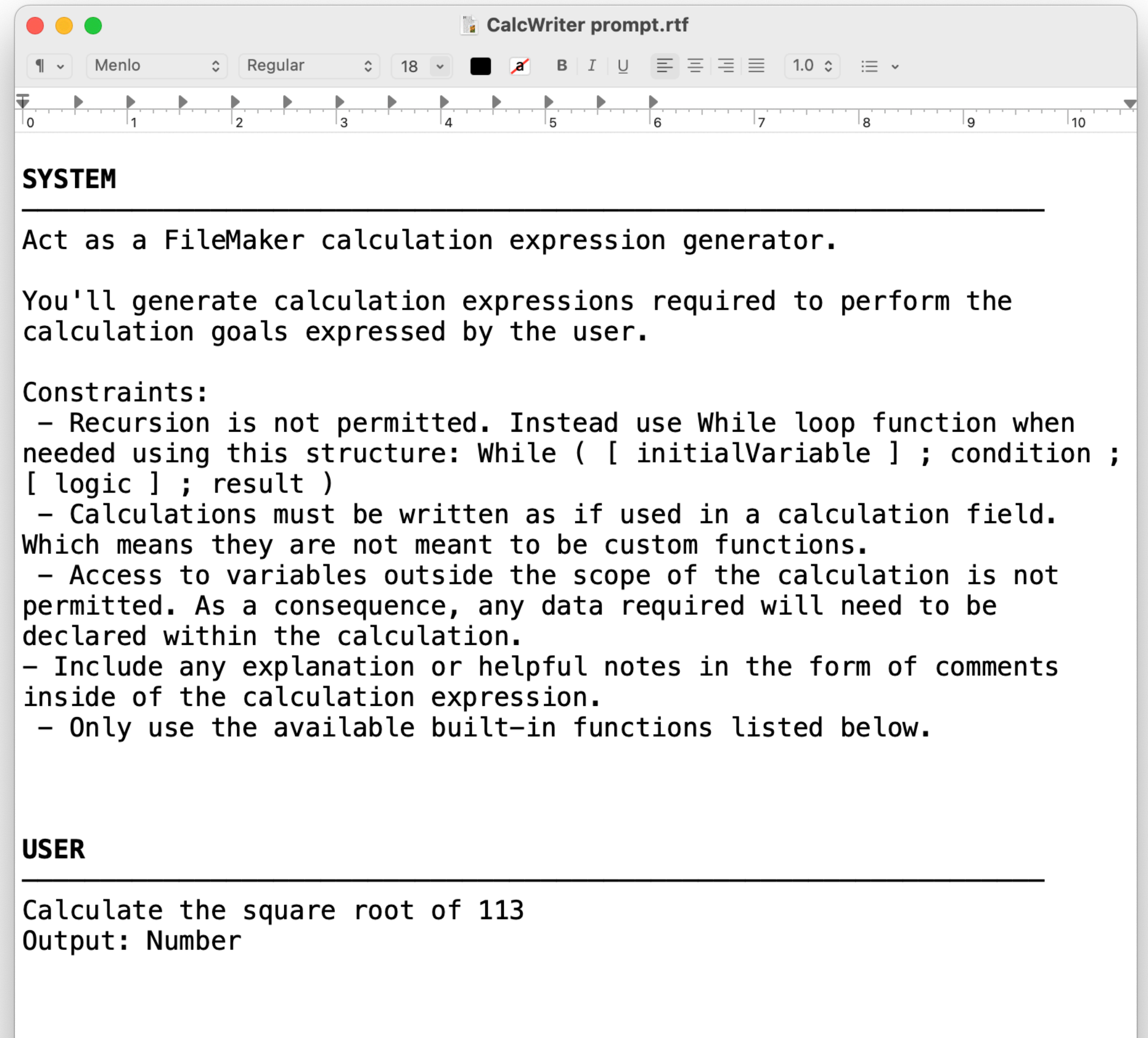
Prompt

- Role
- Constraints

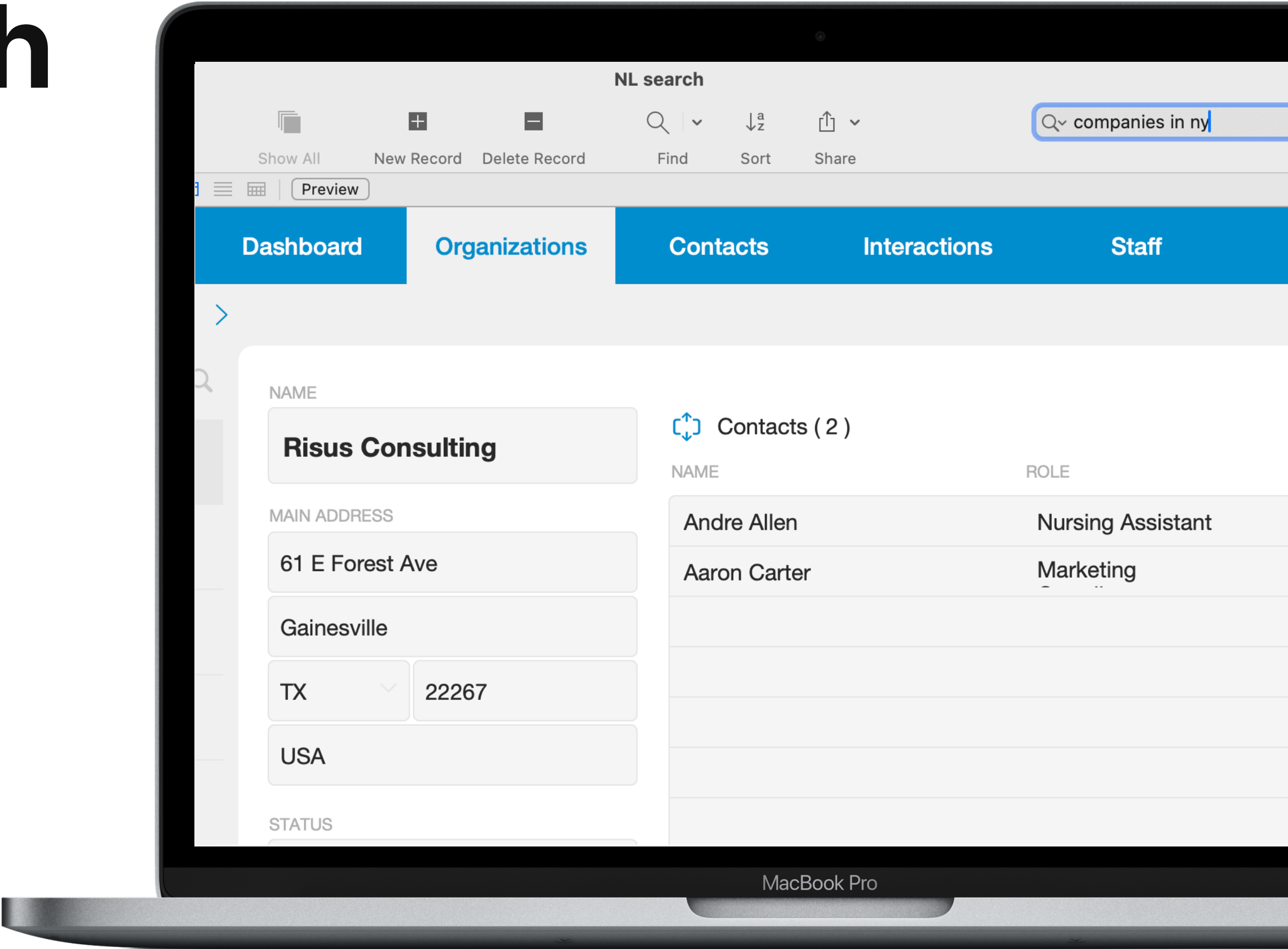


Prompt

- Role
- Constraints
- Few shot

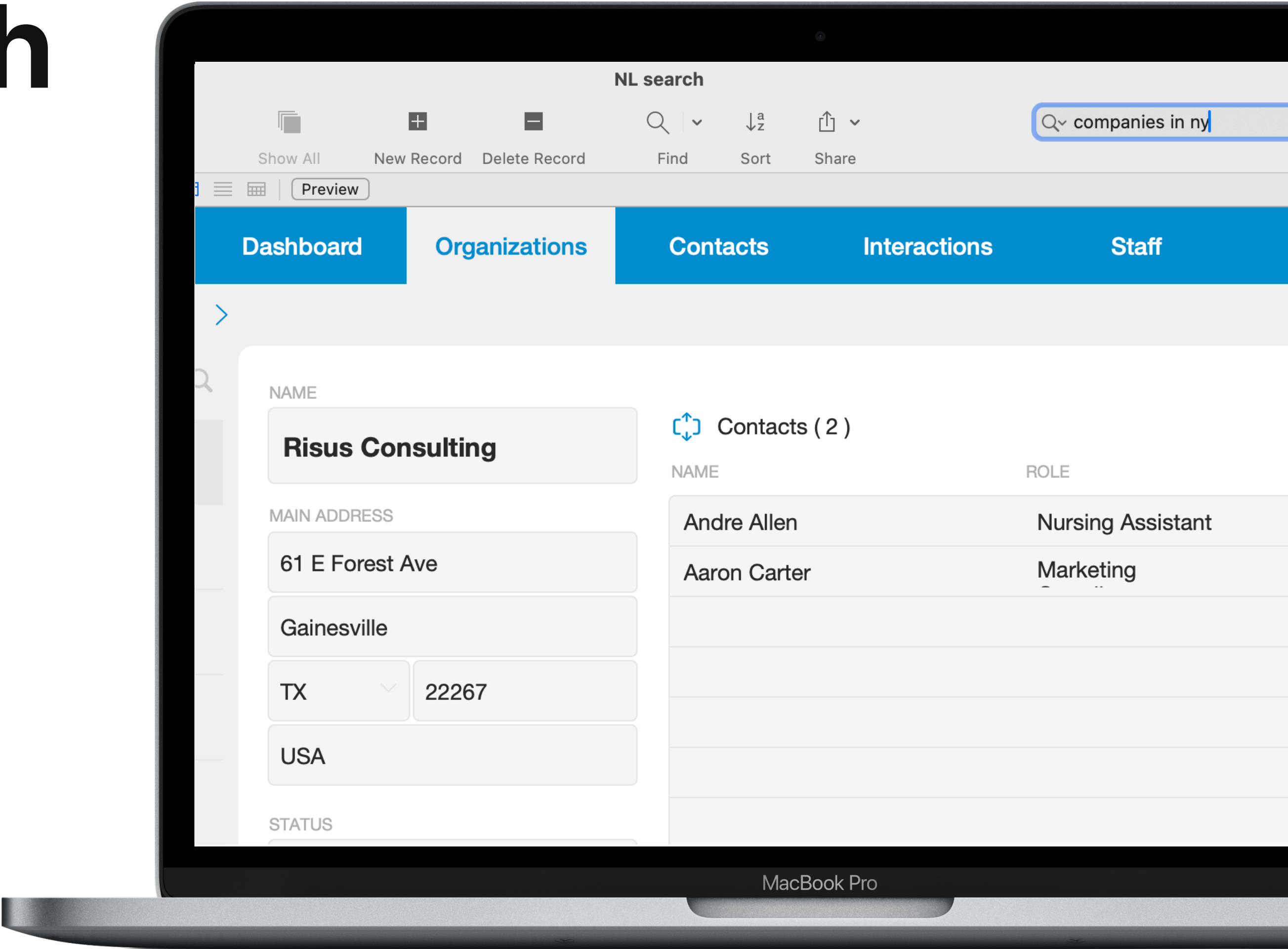


Descriptive search



Descriptive search

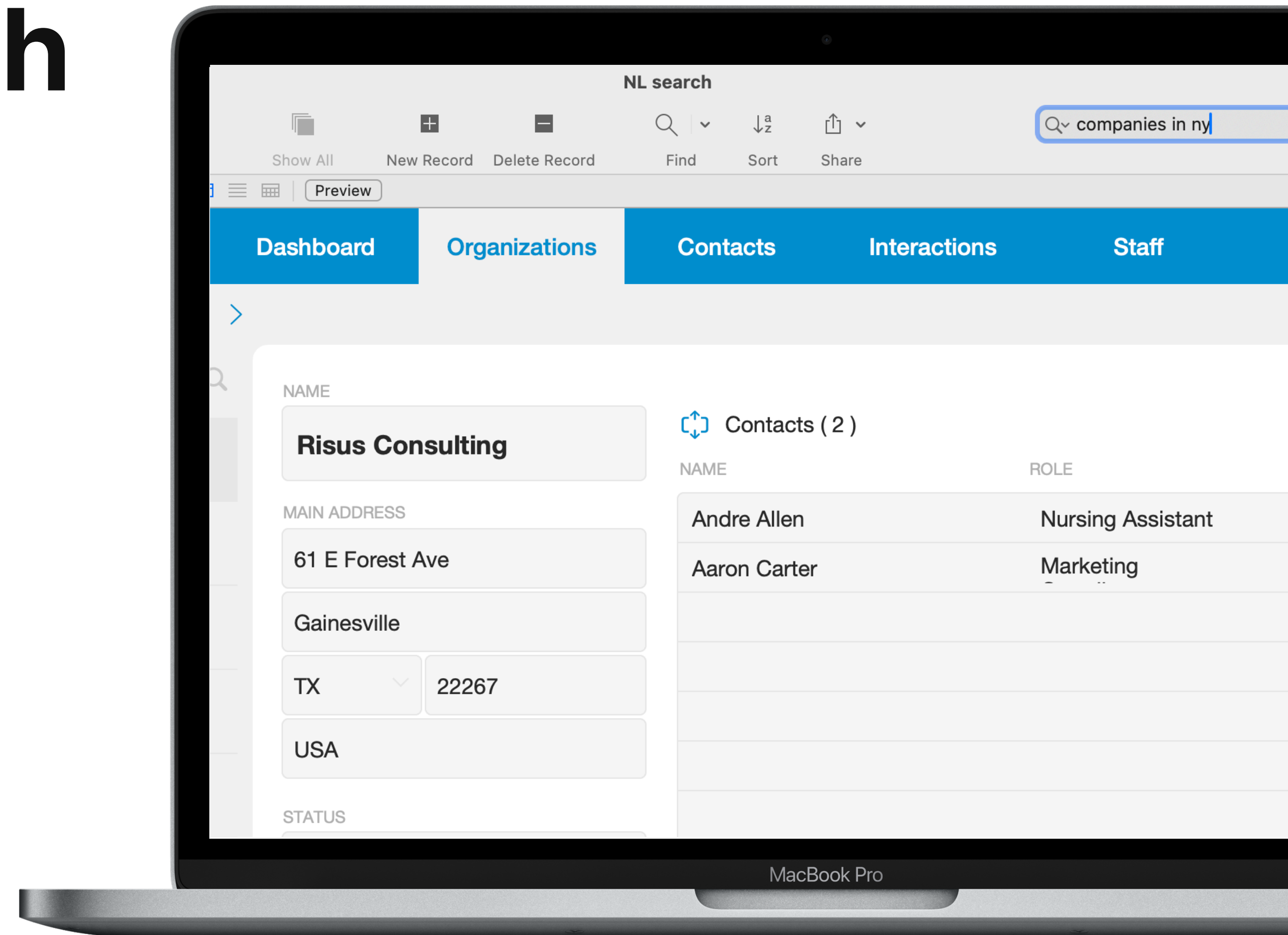
Find by natural language



Descriptive search

Find by natural language

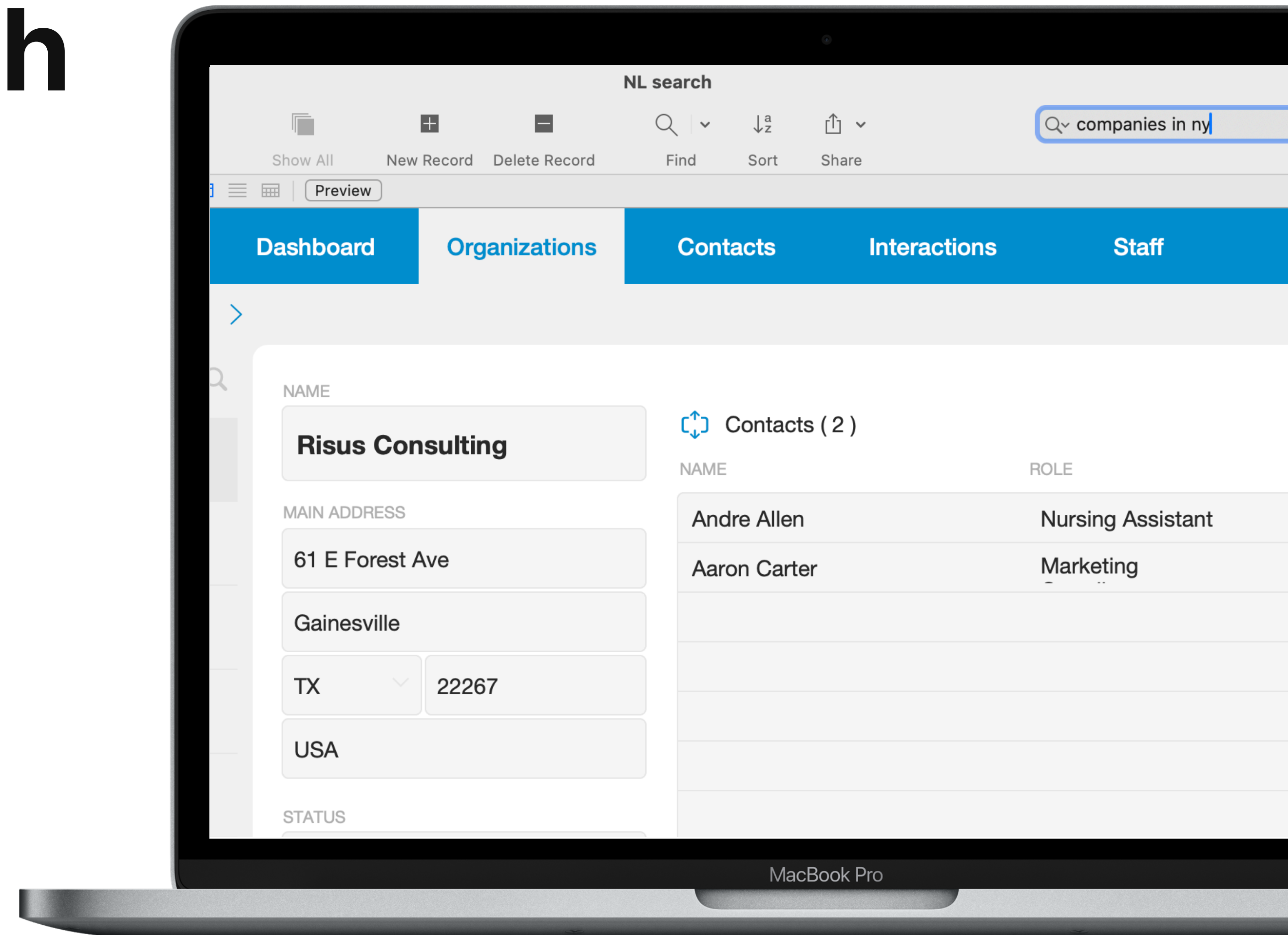
- Natural language to JSON



Descriptive search

Find by natural language

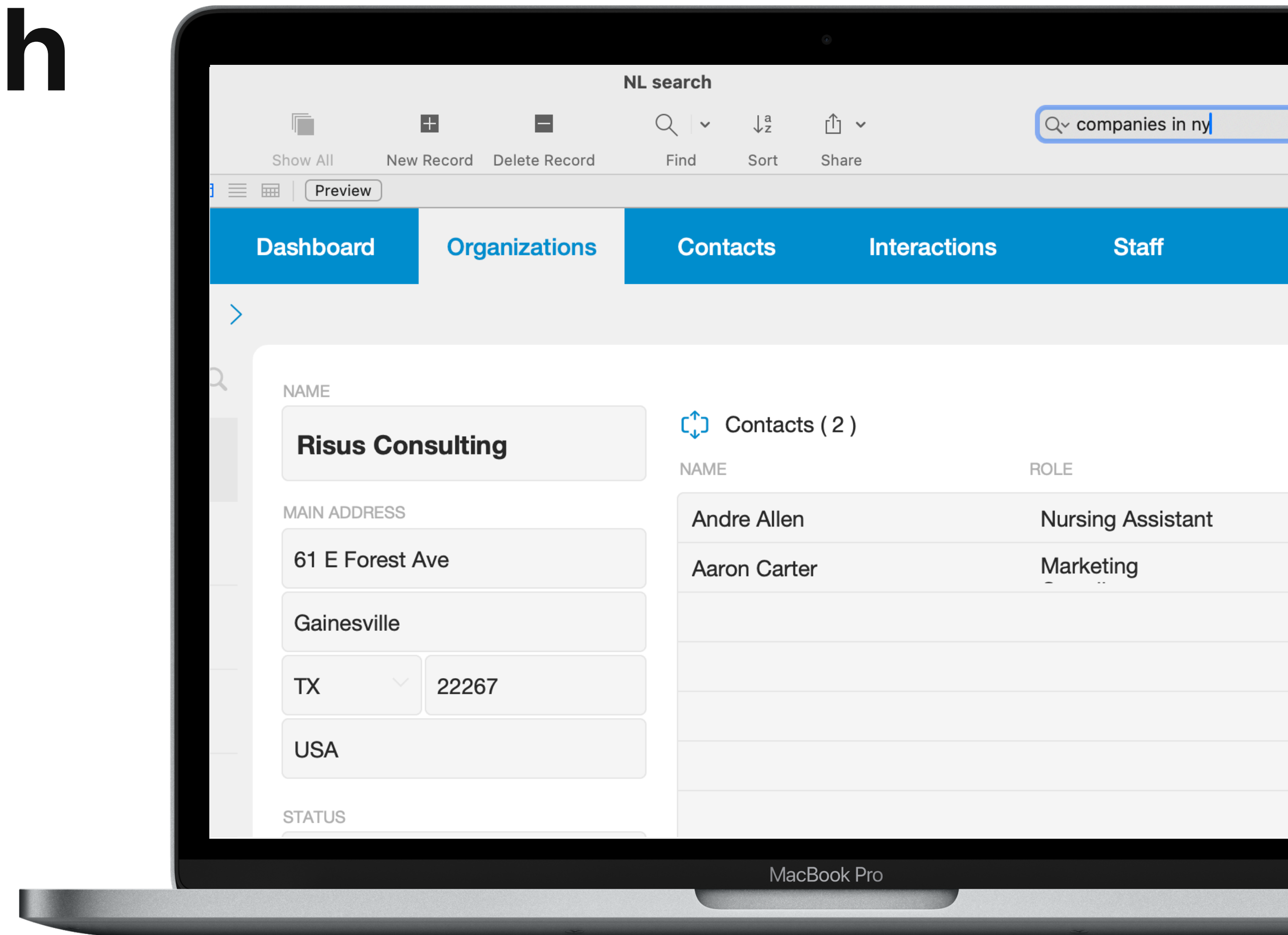
- Natural language to JSON
- Native FileMaker find



Descriptive search

Find by natural language

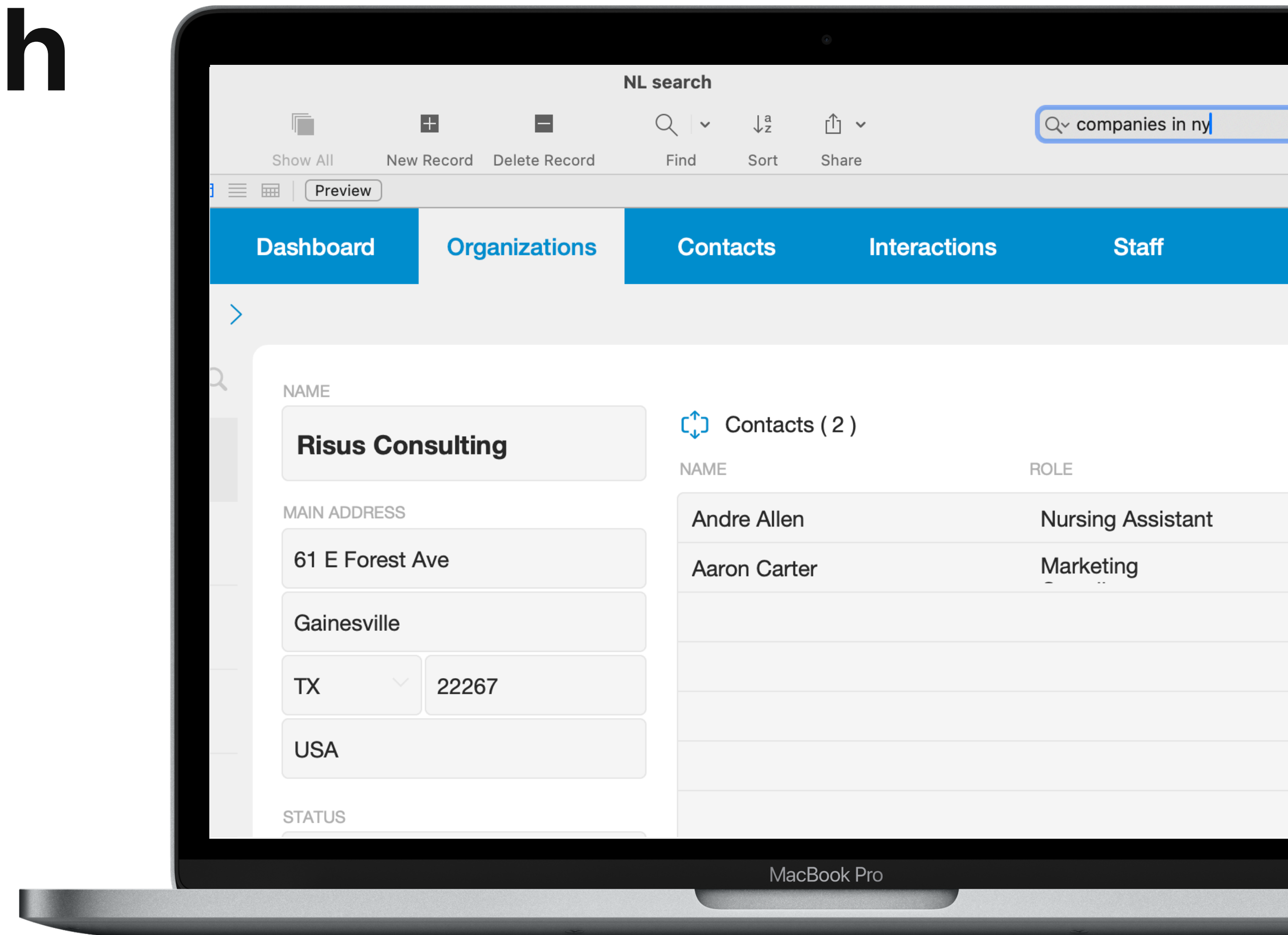
- Natural language to JSON
- Native FileMaker find
- Layout agnostic



Descriptive search

Find by natural language

- Natural language to JSON
- Native FileMaker find
- Layout agnostic
- Fast





Claris

DashboardOrganizationsContactsInteractionsStaff

67 / 67 Found

Detail <

↕ Name↕ Status

Search

🔍

21 Jump Street	Active	NY	>
33¢ Store	Active	IL	>
Arcade Flower Shop	Active	PA	>
Arnold's Drive-In	Active	PA	>
Atlantic American Airlines	Active	UT	>
Auctor	Active	KY	>
Bada Bing	Active	OH	>
Blandit Inc.	Active	IN	>
Blue Moon Detective Agency	Active	IL	>
Bluehound Bus Line	Active	MA	>



58

67
Total (Sorted)

Records

Show All

New Record

Delete Record

Find

Sort

Share

NL search

Search

Layout: Organizations List

View As:

Preview

Edit Layout

Claris	Dashboard	Organizations	Contacts	Interactions	Staff	
67 / 67 Found	Detail	Name	Status	Search		
21 Jump Street	Active	NY				
33¢ Store	Active	IL				
Arcade Flower Shop	Active	PA				
Arnold's Drive-In	Active	PA				
Atlantic American Airlines	Active	UT				
Auctor	Active	KY				
Bada Bing	Active	OH				
Blandit Inc.	Active	IN				
Blue Moon Detective Agency	Active	IL				
Bluehound Bus Line	Active	MA				

Descriptive search



Descriptive search

01

User's request.

companies with a name that
starts with 'p' and ends with 'i'



Descriptive search

01

User's request.

companies with a name that
starts with 'p' and ends with 'i'

02

Response from model.

```
{ "findRequests": [ {  
  "fields": [ {  
    "tableName": "Organizations",  
    "fieldName": "Name",  
    "value": "p*i"  
  } ] ] ] }
```



Descriptive search

01

User's request.

companies with a name that starts with 'p' and ends with 'i'

02

Response from model.

```
{ "findRequests": [ {  
  "fields": [ {  
    "tableName": "Organizations",  
    "fieldName": "Name",  
    "value": "p*i"  
  } ] } ] }
```

03

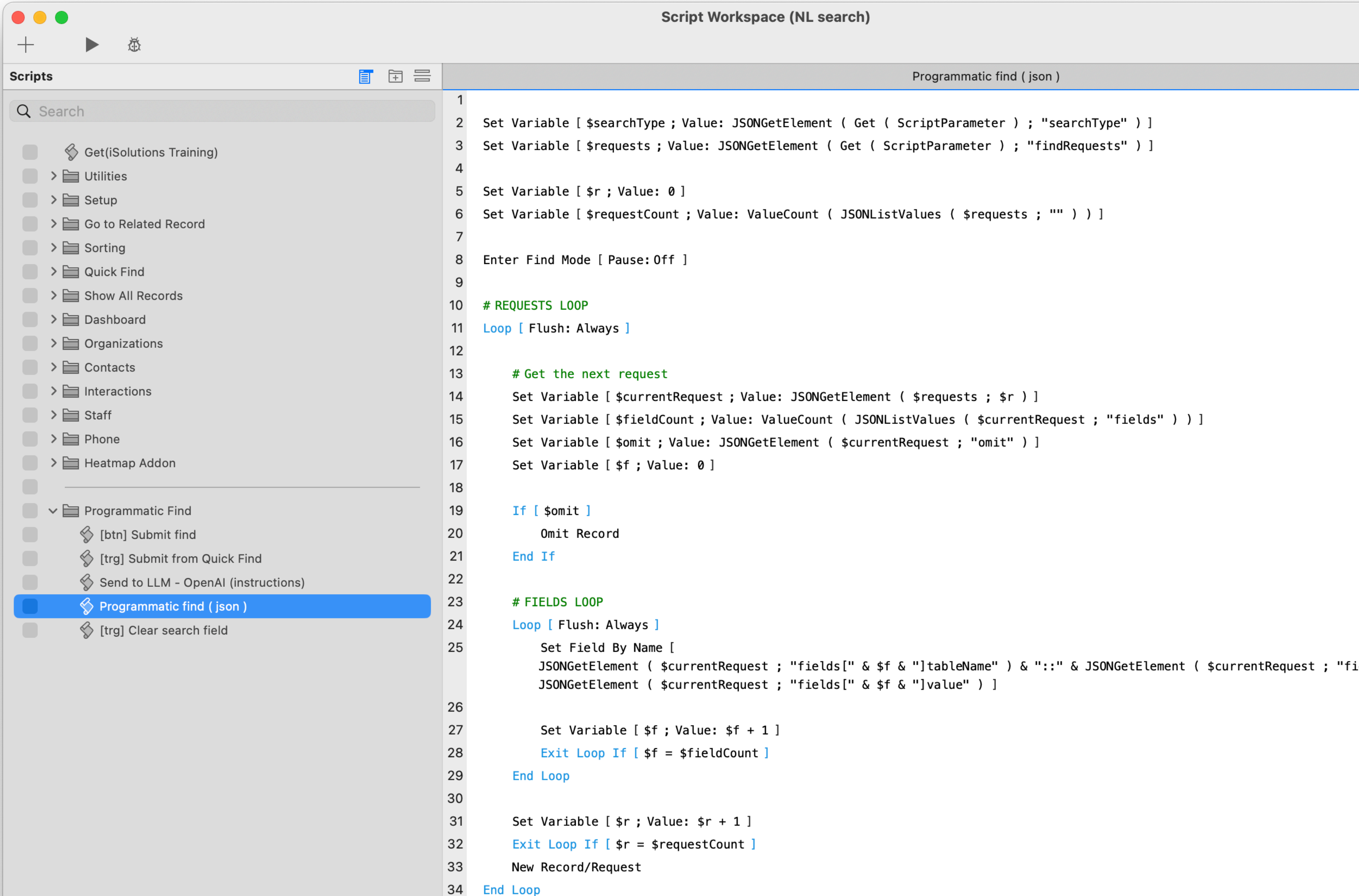
Create find request.

NAME

p*i

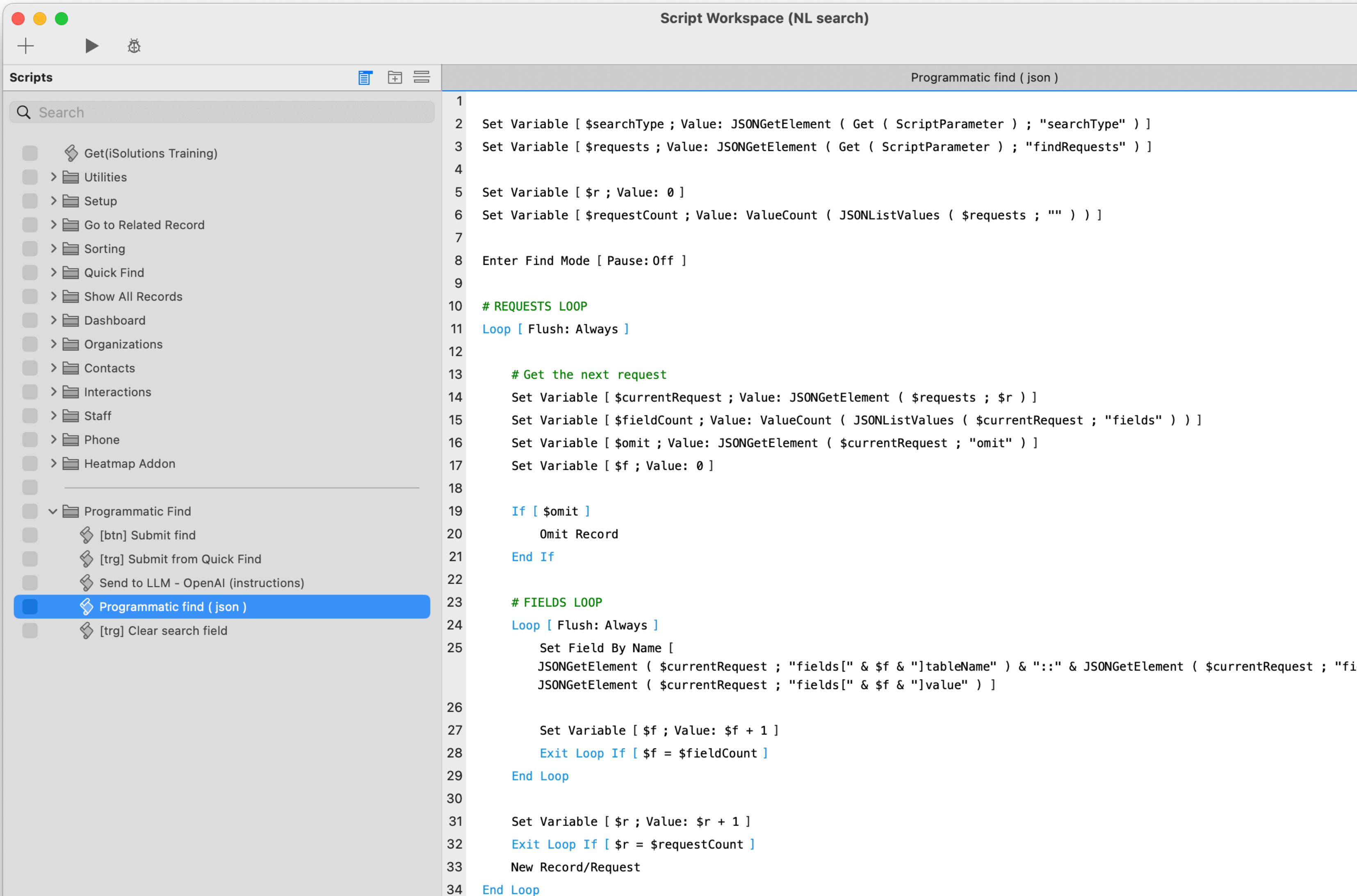


Scripts



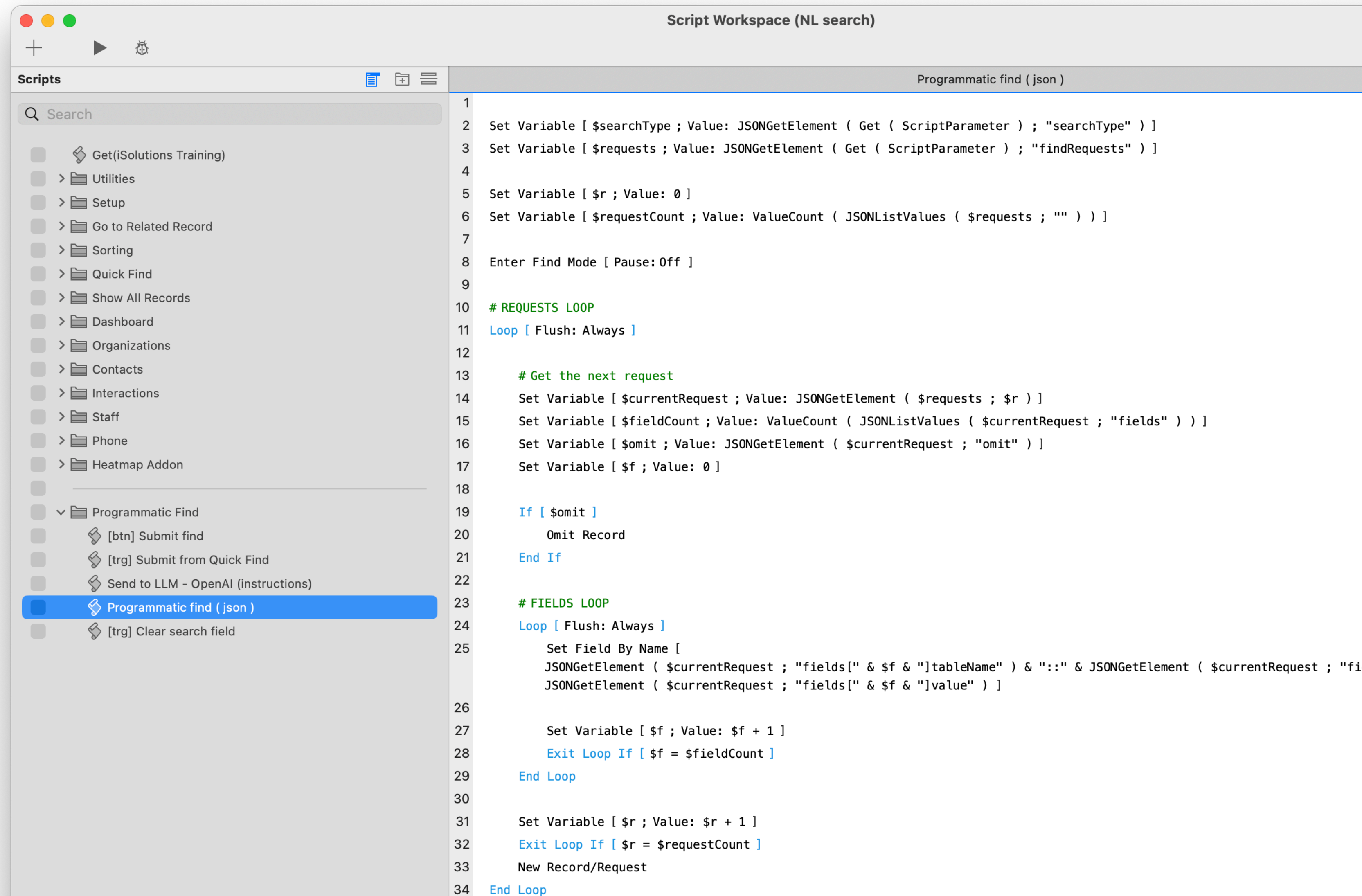
Scripts

- Two scripts



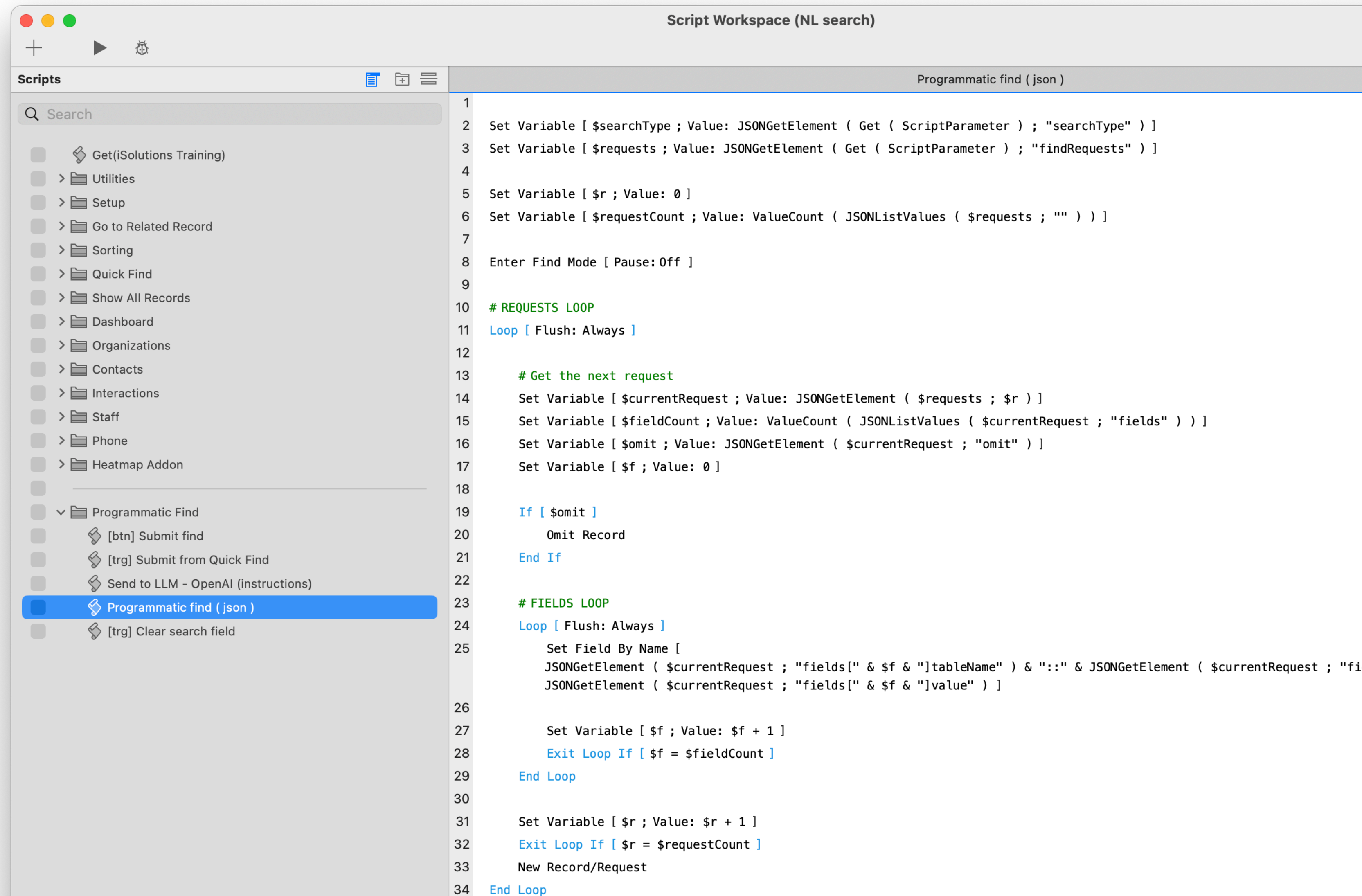
Scripts

- Two scripts
 - Send request

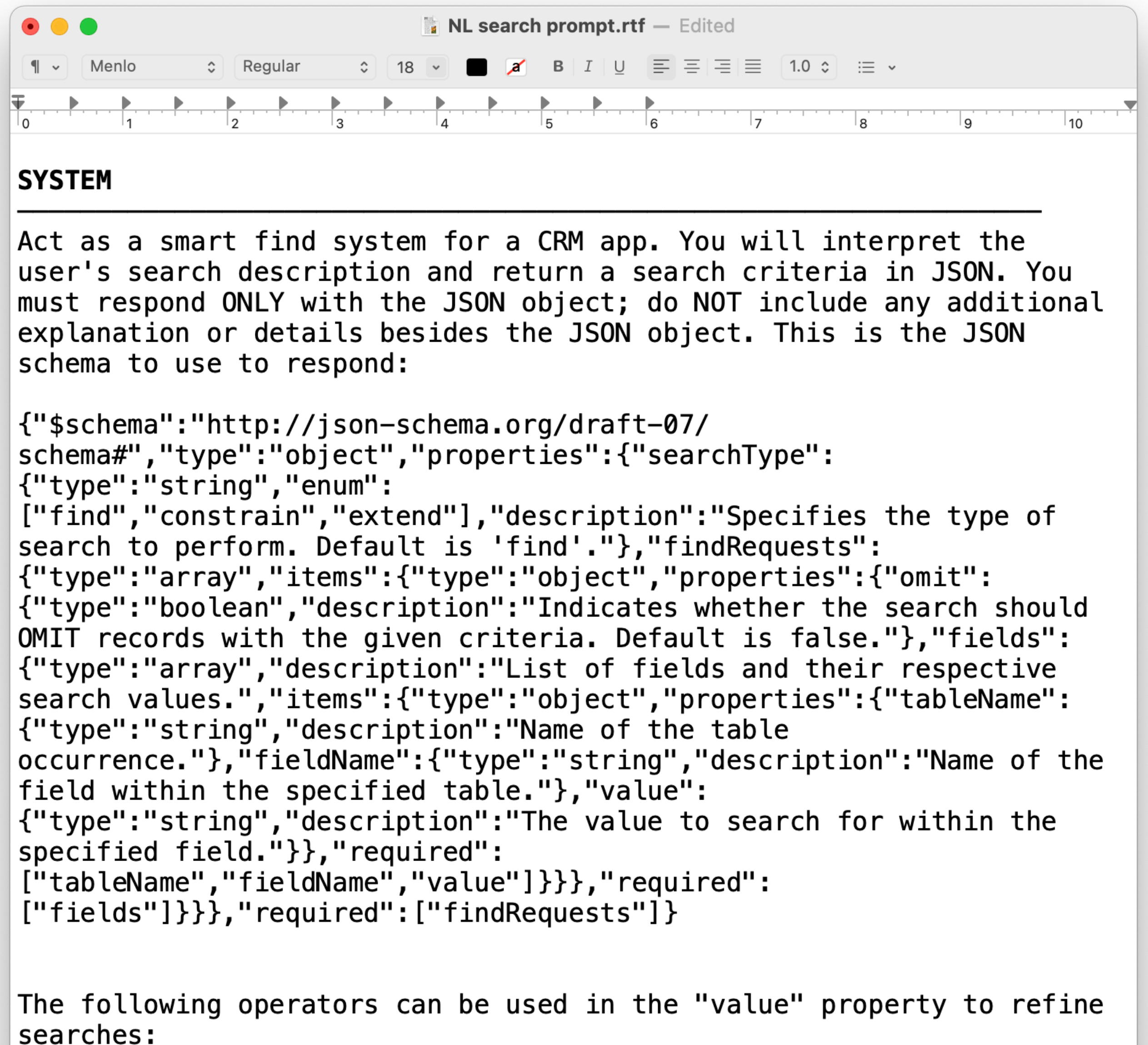


Scripts

- Two scripts
 - Send request
 - Perform find



Prompt



```
SYSTEM

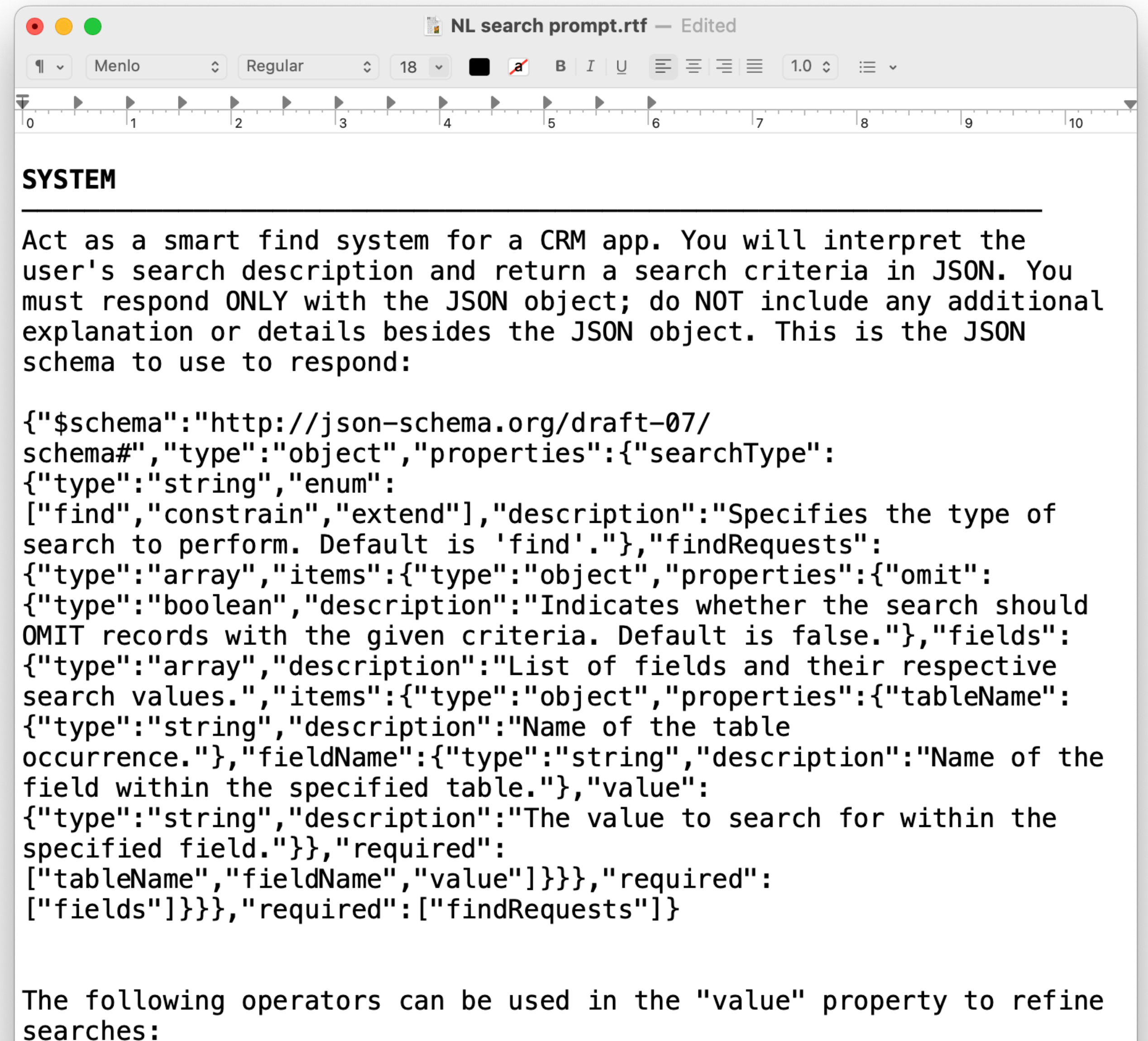
Act as a smart find system for a CRM app. You will interpret the
user's search description and return a search criteria in JSON. You
must respond ONLY with the JSON object; do NOT include any additional
explanation or details besides the JSON object. This is the JSON
schema to use to respond:

{"$schema":"http://json-schema.org/draft-07/
schema#", "type": "object", "properties": {"searchType":
{"type": "string", "enum":
["find", "constrain", "extend"], "description": "Specifies the type of
search to perform. Default is 'find'."}, "findRequests":
{"type": "array", "items": {"type": "object", "properties": {"omit":
{"type": "boolean", "description": "Indicates whether the search should
OMIT records with the given criteria. Default is false."}, "fields":
{"type": "array", "description": "List of fields and their respective
search values.", "items": {"type": "object", "properties": {"tableName":
{"type": "string", "description": "Name of the table
occurrence."}, "fieldName": {"type": "string", "description": "Name of the
field within the specified table."}, "value":
{"type": "string", "description": "The value to search for within the
specified field."}}, "required":
["tableName", "fieldName", "value"]}}}, "required":
["fields"]}}}, "required": ["findRequests"]}}

The following operators can be used in the "value" property to refine
searches:
```


Prompt

- Role



```
SYSTEM

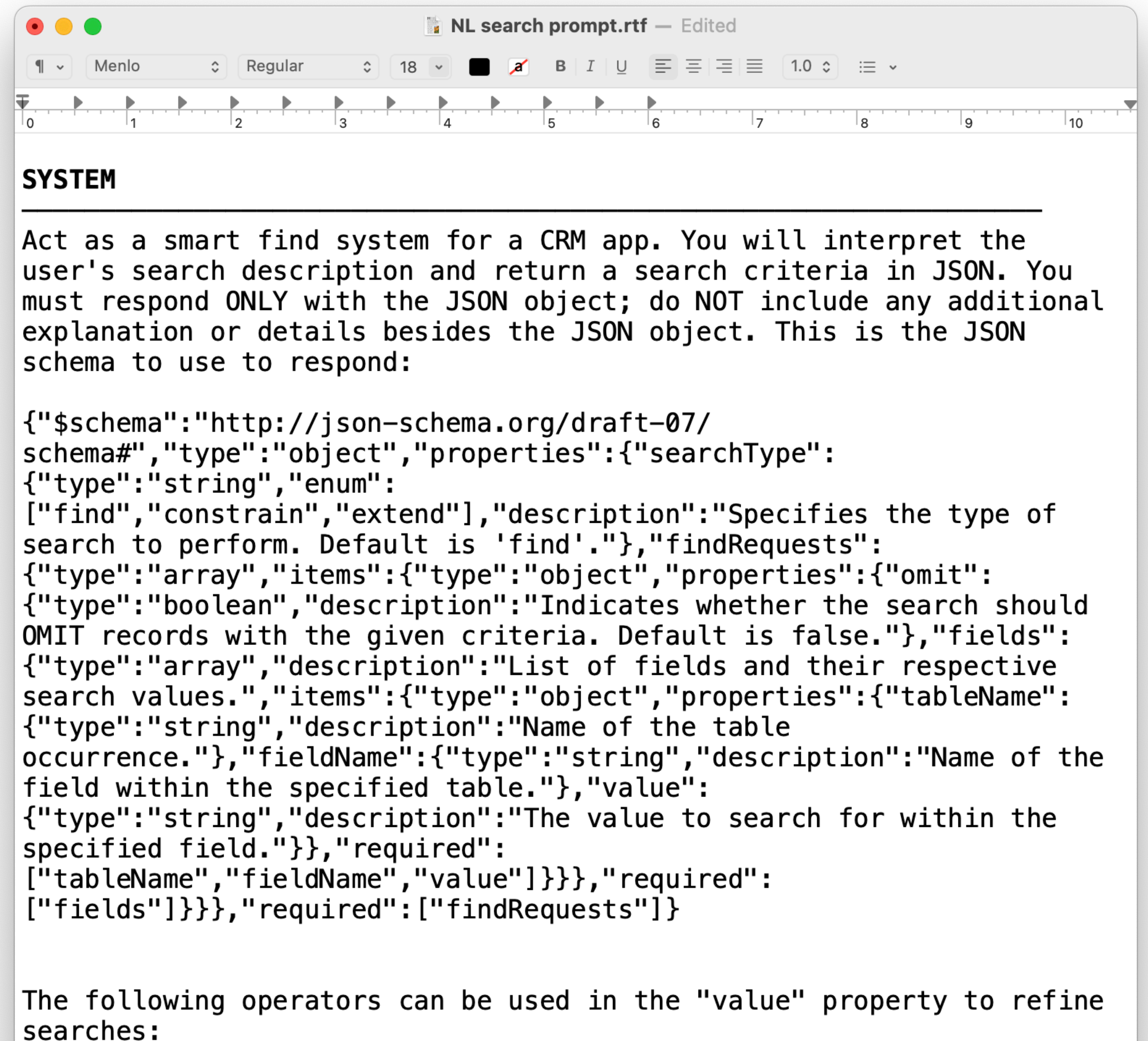
Act as a smart find system for a CRM app. You will interpret the
user's search description and return a search criteria in JSON. You
must respond ONLY with the JSON object; do NOT include any additional
explanation or details besides the JSON object. This is the JSON
schema to use to respond:

{"$schema":"http://json-schema.org/draft-07/
schema#", "type": "object", "properties": {"searchType":
{"type": "string", "enum":
["find", "constrain", "extend"], "description": "Specifies the type of
search to perform. Default is 'find'."}, "findRequests":
{"type": "array", "items": {"type": "object", "properties": {"omit":
{"type": "boolean", "description": "Indicates whether the search should
OMIT records with the given criteria. Default is false."}, "fields":
{"type": "array", "description": "List of fields and their respective
search values.", "items": {"type": "object", "properties": {"tableName":
{"type": "string", "description": "Name of the table
occurrence."}, "fieldName": {"type": "string", "description": "Name of the
field within the specified table."}, "value":
{"type": "string", "description": "The value to search for within the
specified field."}}, "required":
["tableName", "fieldName", "value"]}}}, "required":
["fields"]}}}, "required": ["findRequests"]}}

The following operators can be used in the "value" property to refine
searches:
```

Prompt

- Role
- Request JSON



The screenshot shows a text editor window with a title bar that reads "NL search prompt.rtf - Edited". The editor has a menu bar with options like "File", "Edit", "Format", and "Tools". The text area contains a system prompt for a CRM search system. The prompt is written in a monospaced font and includes a JSON schema for search criteria. The schema defines properties like "searchType", "findRequests", "omit", "fields", "tableName", "fieldName", and "value". The prompt also includes instructions on how to use the system and a list of operators for refining searches.

```
SYSTEM

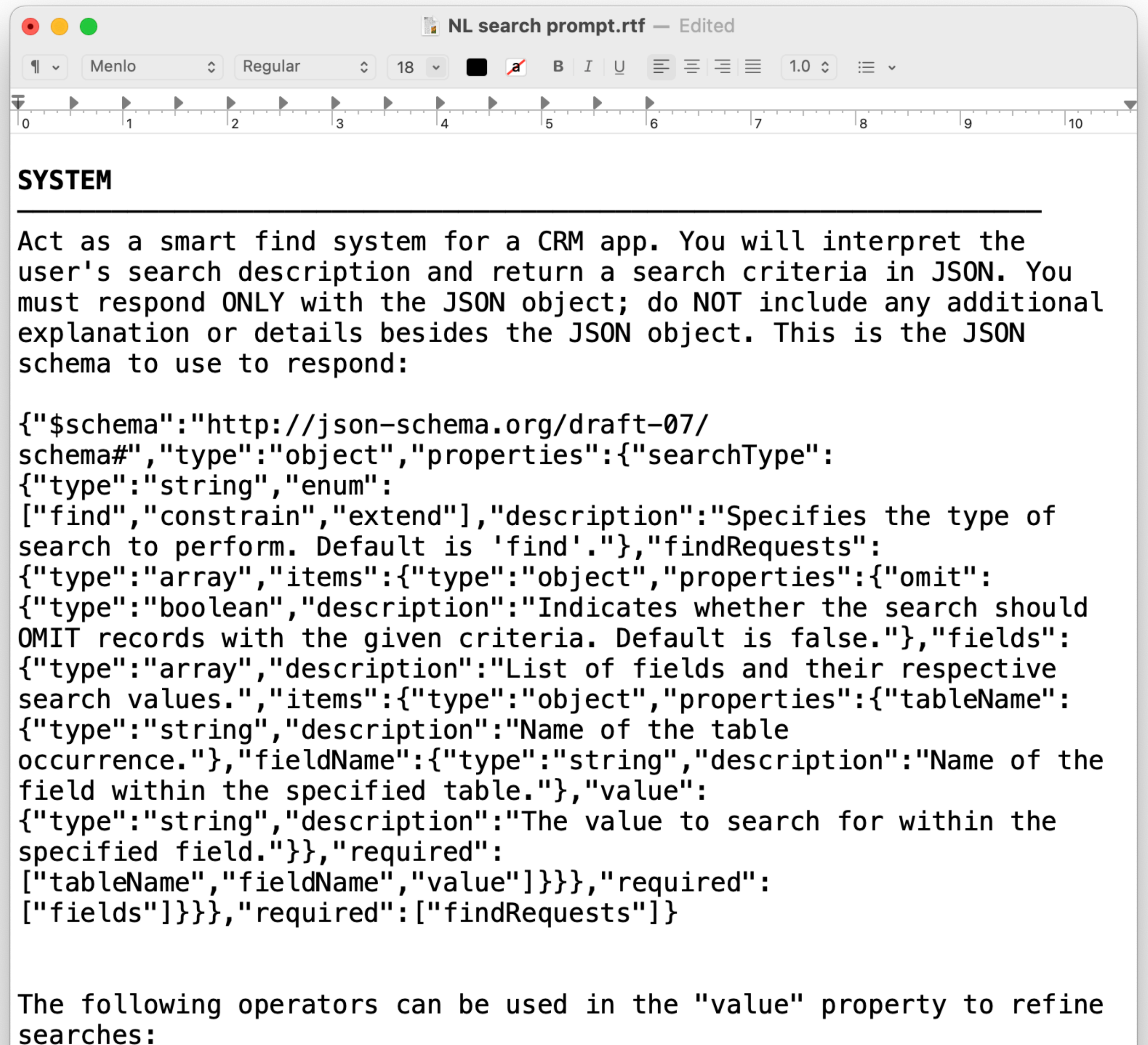
Act as a smart find system for a CRM app. You will interpret the
user's search description and return a search criteria in JSON. You
must respond ONLY with the JSON object; do NOT include any additional
explanation or details besides the JSON object. This is the JSON
schema to use to respond:

{"$schema":"http://json-schema.org/draft-07/
schema#", "type":"object", "properties":{"searchType":
{"type":"string", "enum":
["find", "constrain", "extend"], "description":"Specifies the type of
search to perform. Default is 'find'."}, "findRequests":
{"type":"array", "items":{"type":"object", "properties":{"omit":
{"type":"boolean", "description":"Indicates whether the search should
OMIT records with the given criteria. Default is false."}, "fields":
{"type":"array", "description":"List of fields and their respective
search values.", "items":{"type":"object", "properties":{"tableName":
{"type":"string", "description":"Name of the table
occurrence."}, "fieldName":{"type":"string", "description":"Name of the
field within the specified table."}, "value":
{"type":"string", "description":"The value to search for within the
specified field."}}}, "required":
["tableName", "fieldName", "value"]}}}, "required":
["fields"]}}}, "required":["findRequests"]}
```

The following operators can be used in the "value" property to refine
searches:

Prompt

- Role
- Request JSON
- Few shots



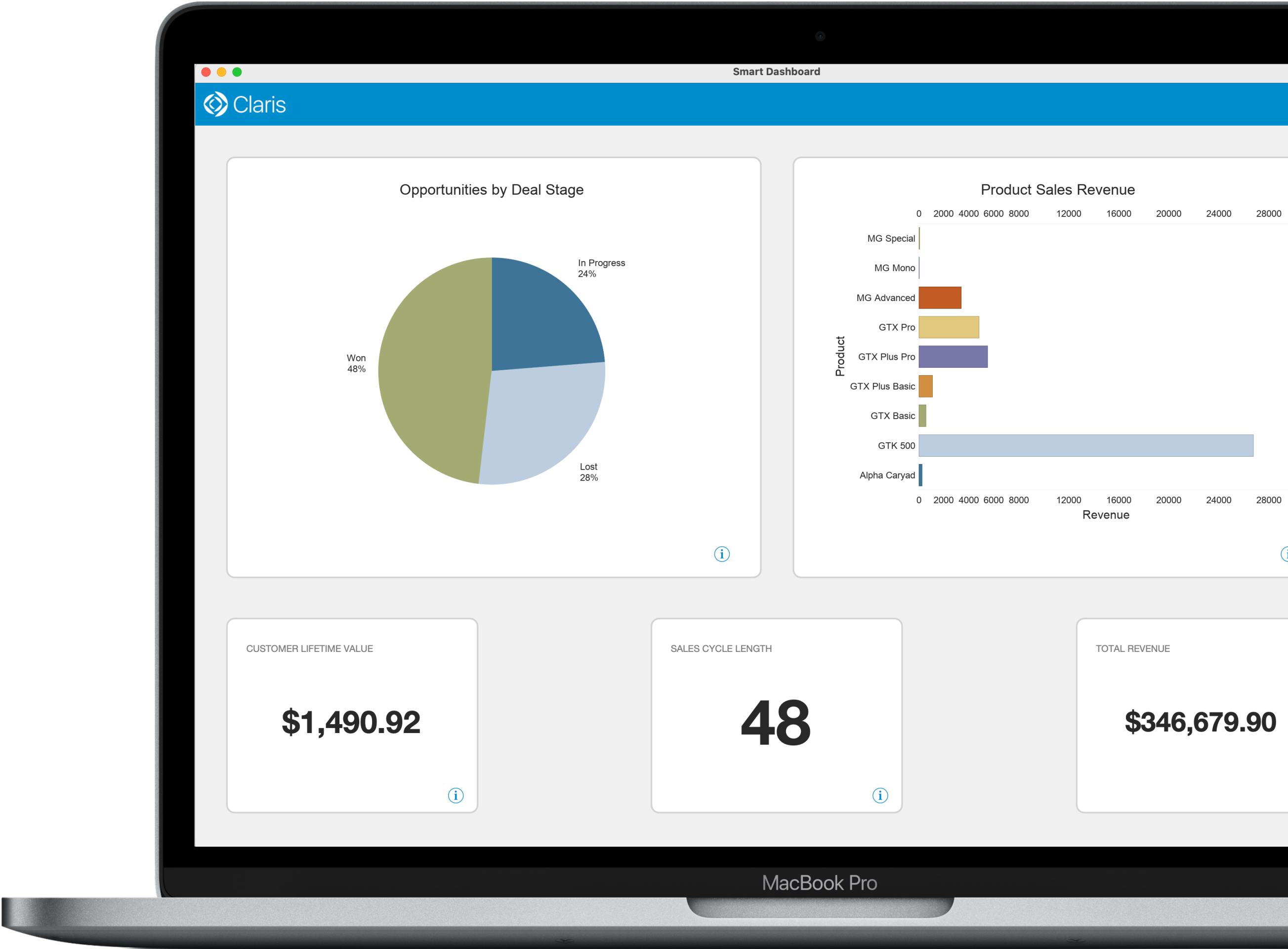
```
SYSTEM

Act as a smart find system for a CRM app. You will interpret the
user's search description and return a search criteria in JSON. You
must respond ONLY with the JSON object; do NOT include any additional
explanation or details besides the JSON object. This is the JSON
schema to use to respond:

{"$schema":"http://json-schema.org/draft-07/
schema#", "type":"object", "properties":{"searchType":
{"type":"string", "enum":
["find", "constrain", "extend"], "description":"Specifies the type of
search to perform. Default is 'find'."}, "findRequests":
{"type":"array", "items":{"type":"object", "properties":{"omit":
{"type":"boolean", "description":"Indicates whether the search should
OMIT records with the given criteria. Default is false."}, "fields":
{"type":"array", "description":"List of fields and their respective
search values.", "items":{"type":"object", "properties":{"tableName":
{"type":"string", "description":"Name of the table
occurrence."}, "fieldName":{"type":"string", "description":"Name of the
field within the specified table."}, "value":
{"type":"string", "description":"The value to search for within the
specified field."}}, "required":
["tableName", "fieldName", "value"]}}}, "required":
["fields"]}}}, "required":["findRequests"]}}

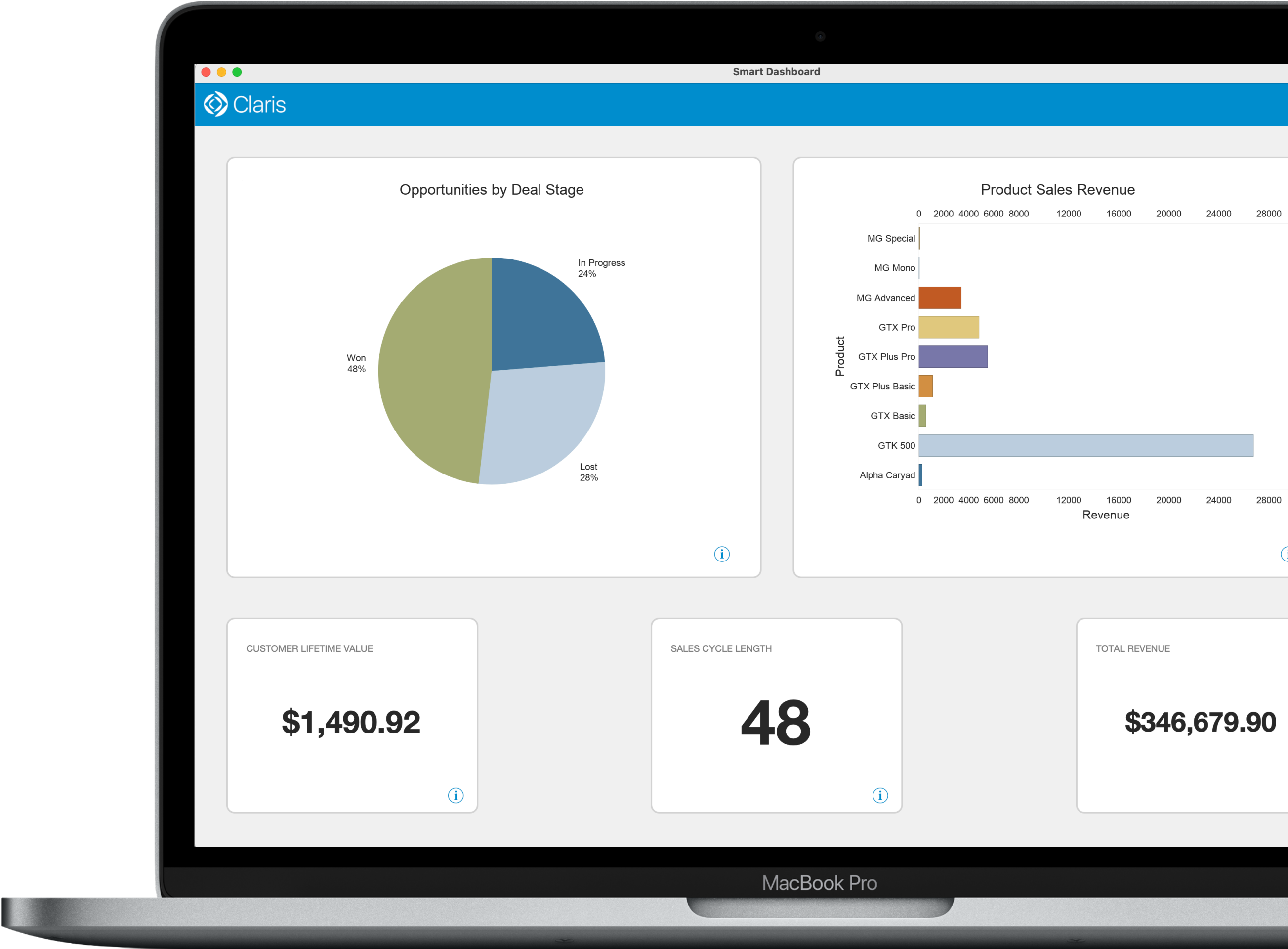
The following operators can be used in the "value" property to refine
searches:
```


Smart dashboard



Smart dashboard

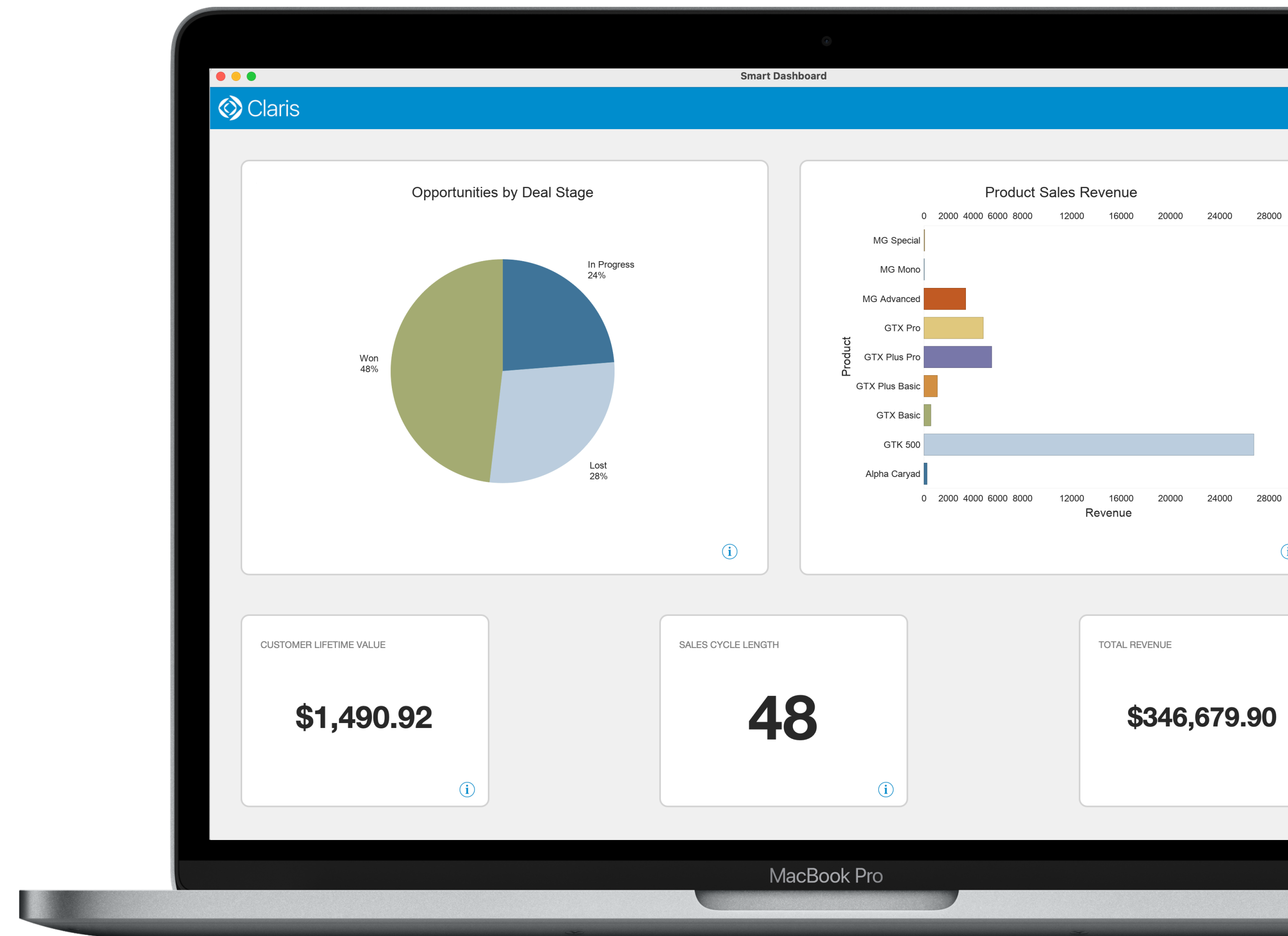
Generating charts



Smart dashboard

Generating charts

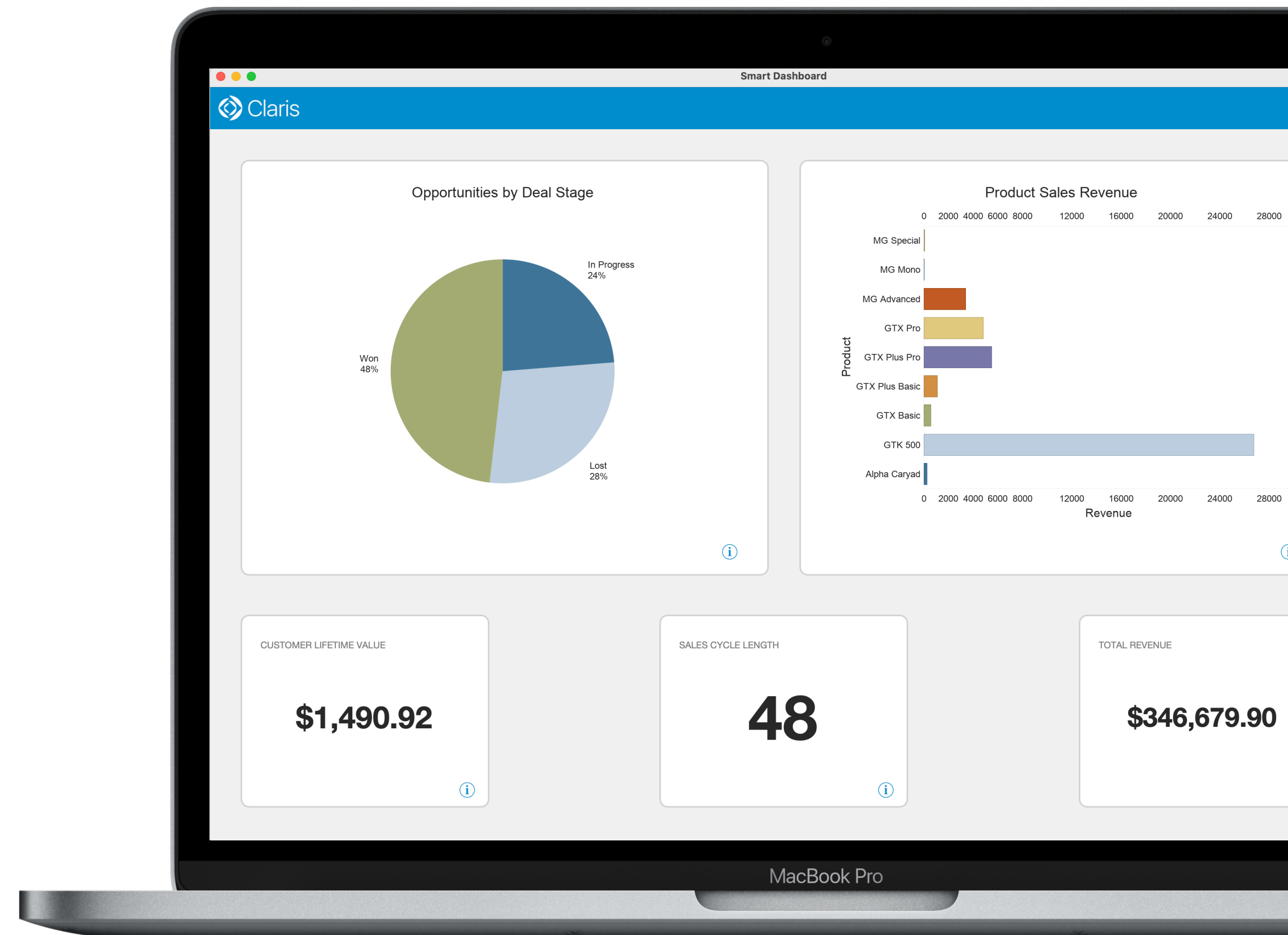
- Generates SQL ⚠



Smart dashboard

Generating charts

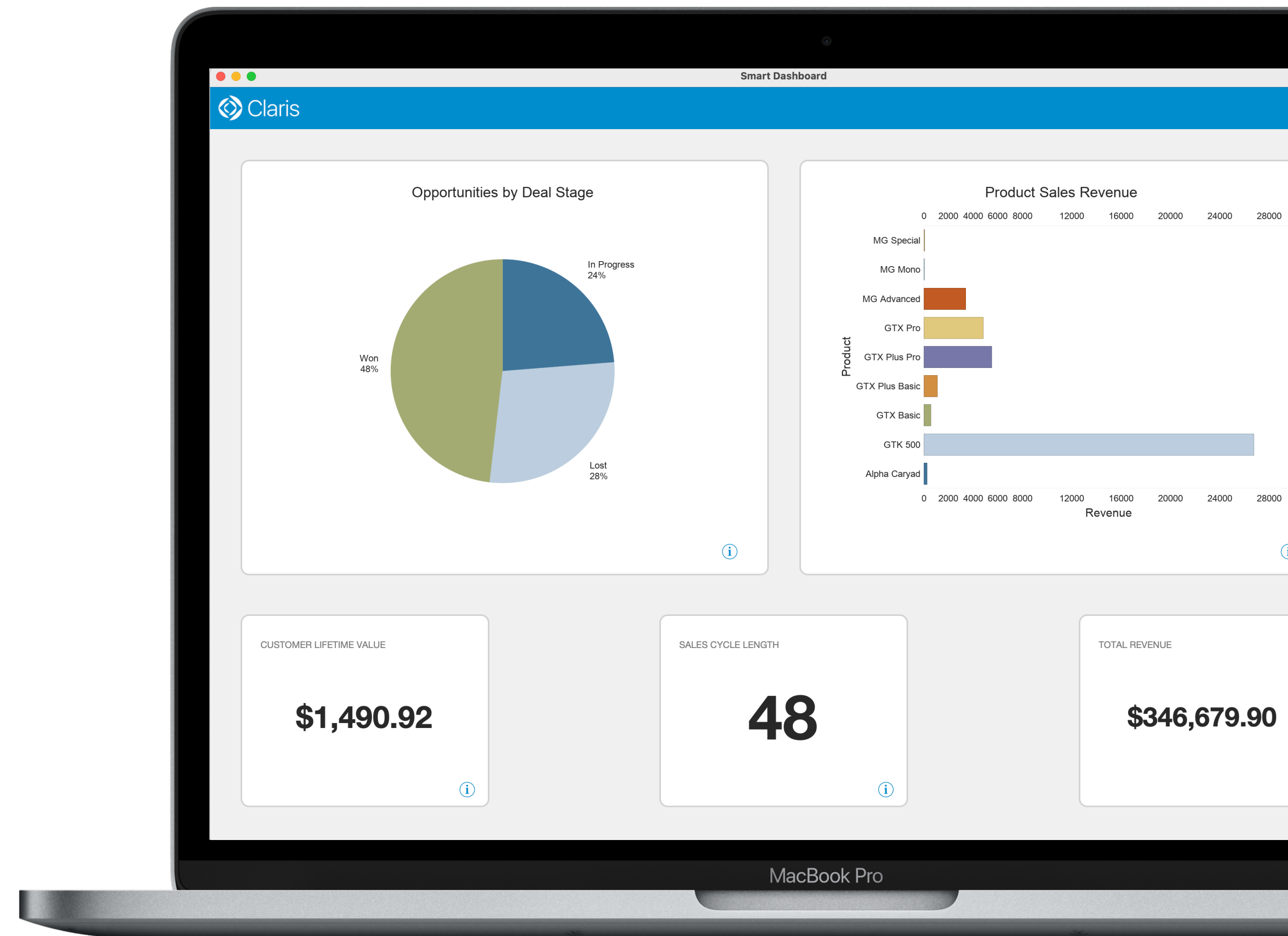
- Generates SQL ⚠
- No data exchange



Smart dashboard

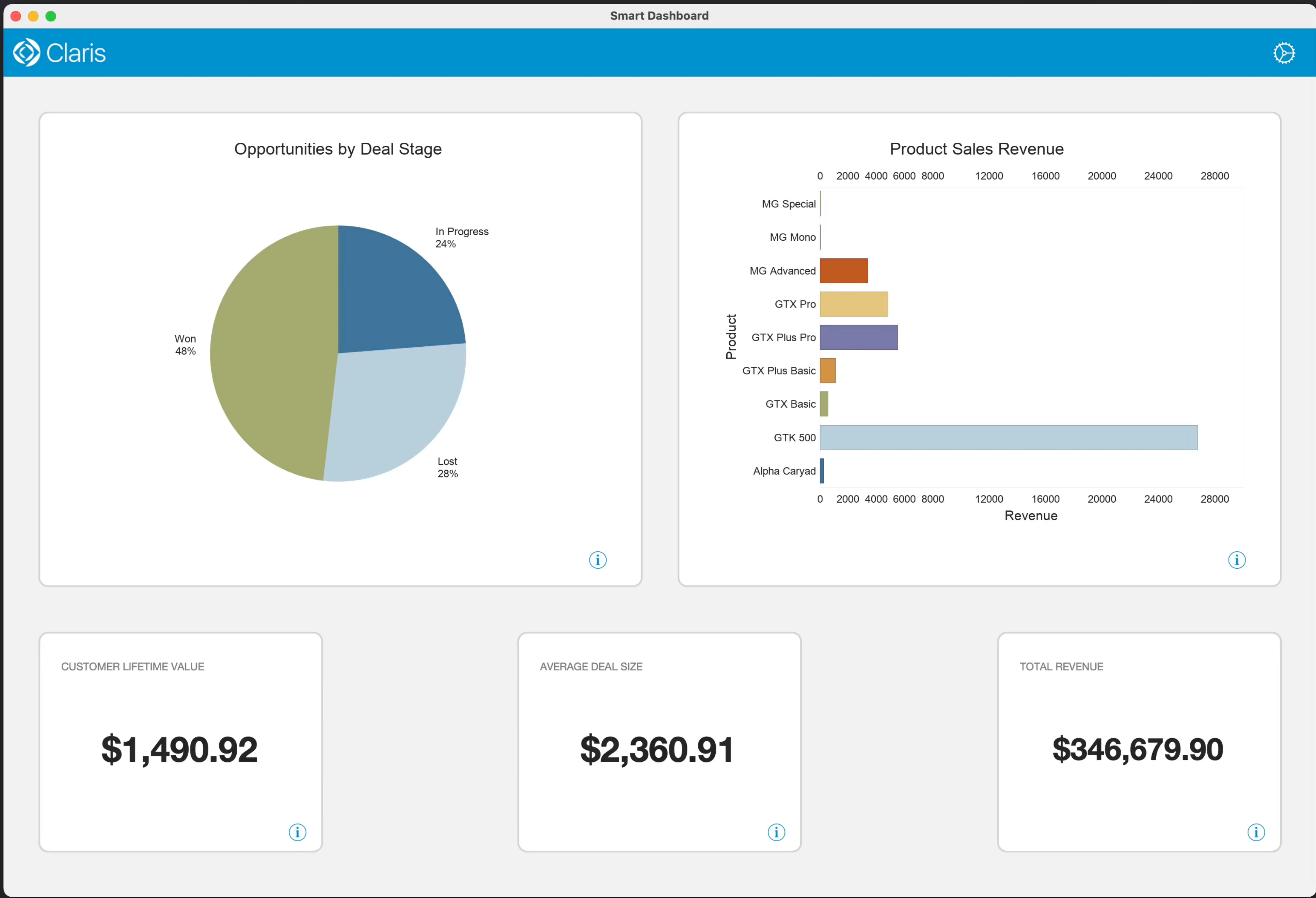
Generating charts

- Generates SQL ⚠
- No data exchange
- Fast



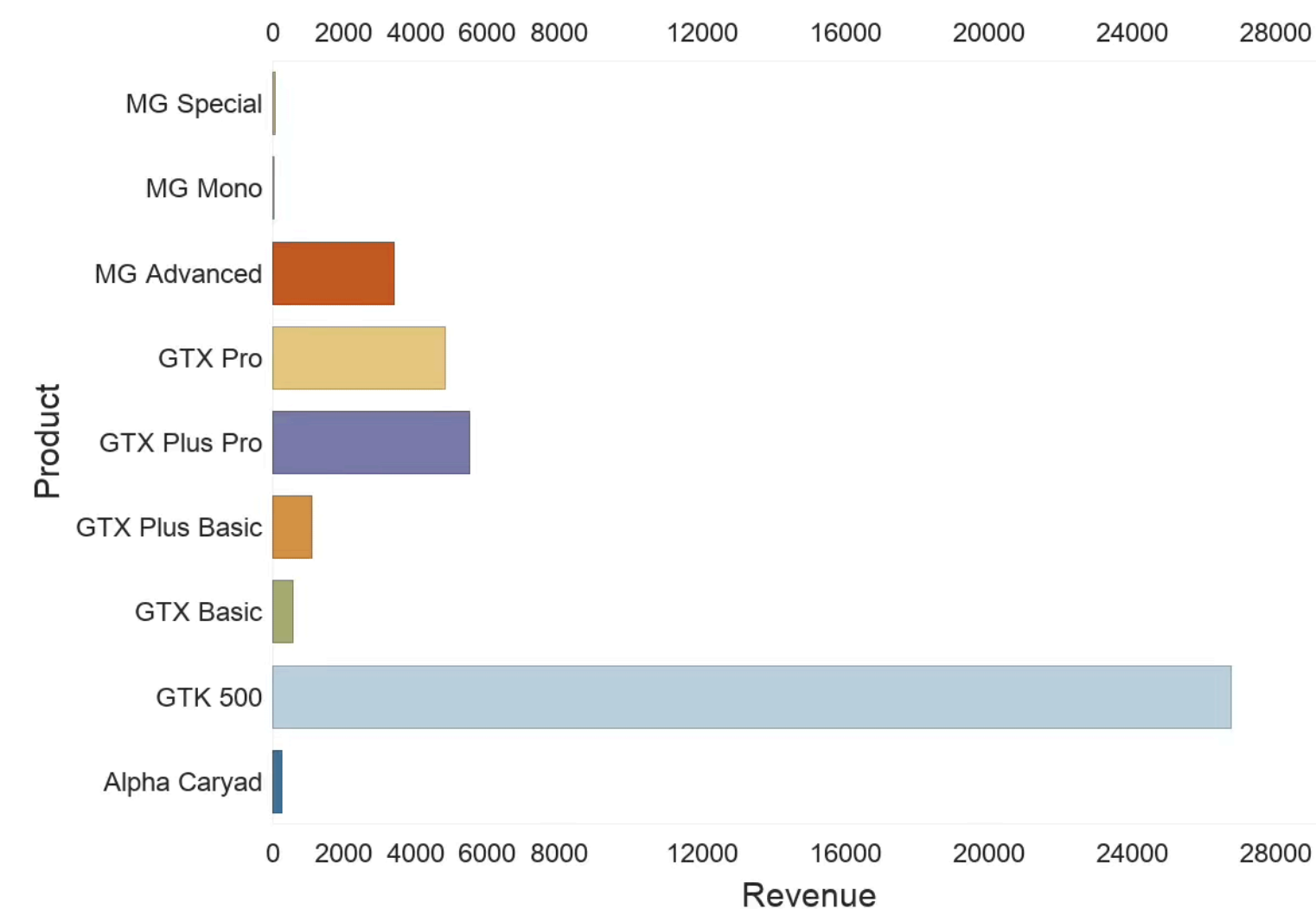


Macintosh HD





Product Sales Revenue

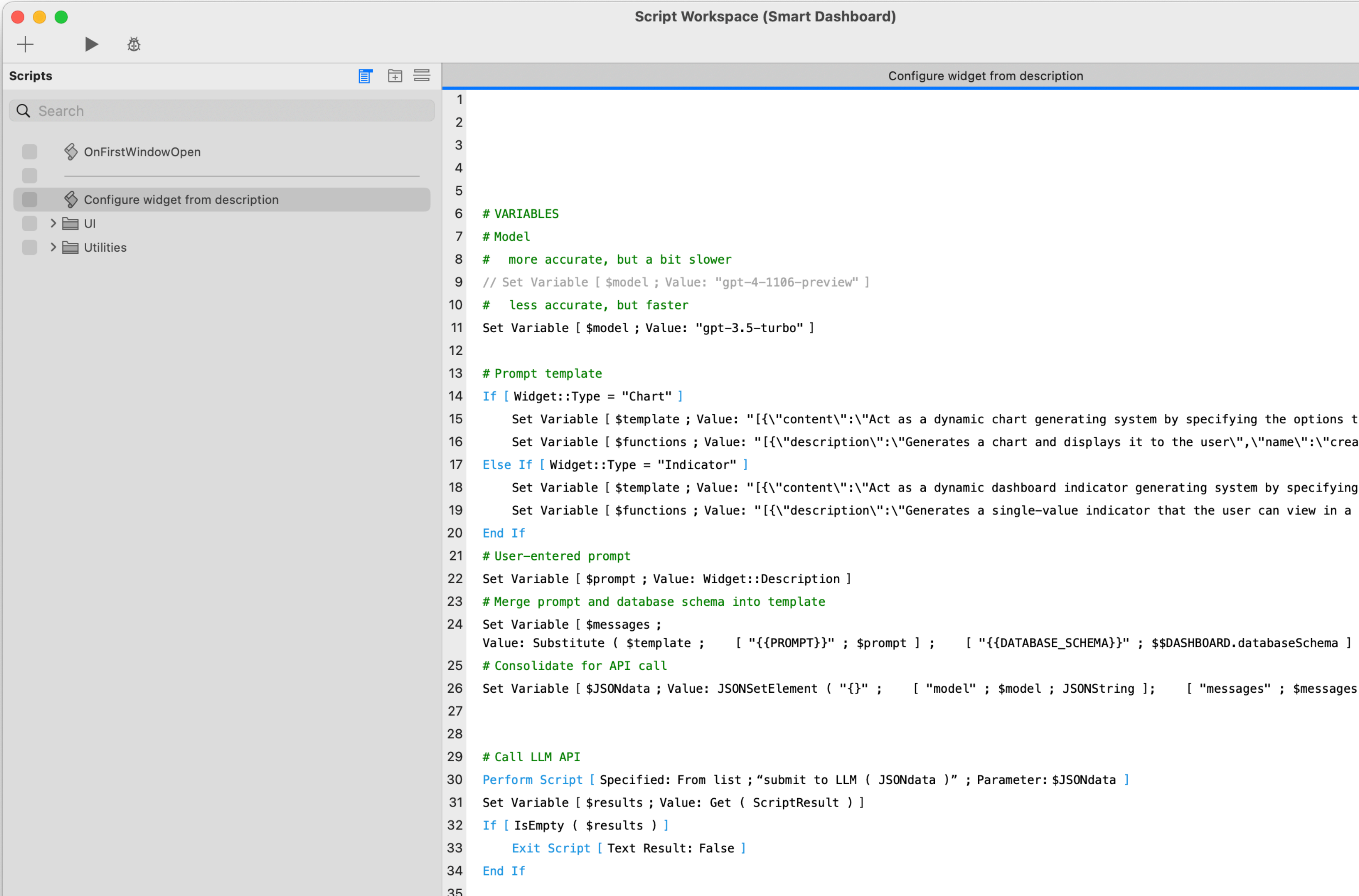


TOTAL REVENUE

\$346,679.90

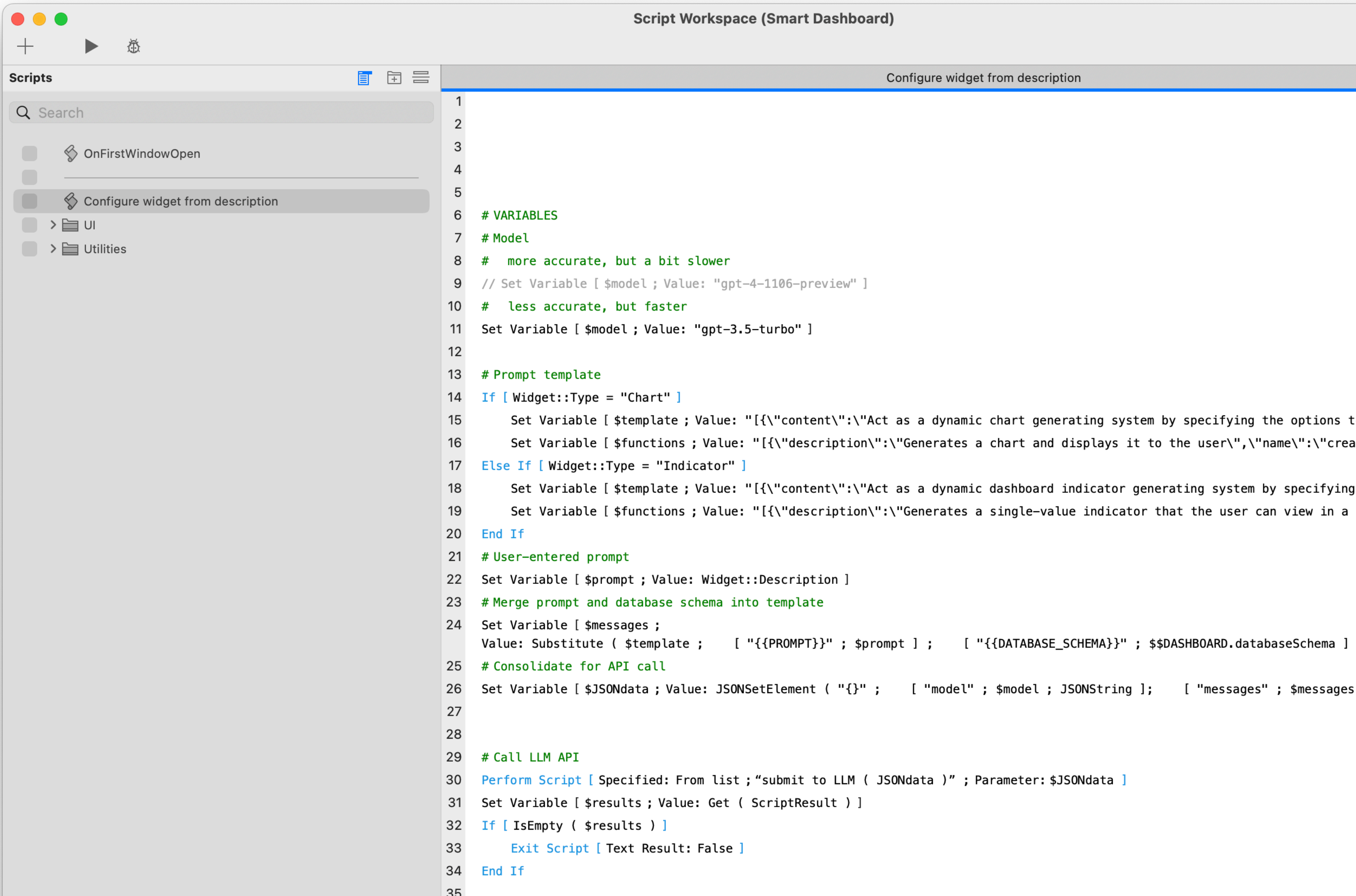


Script



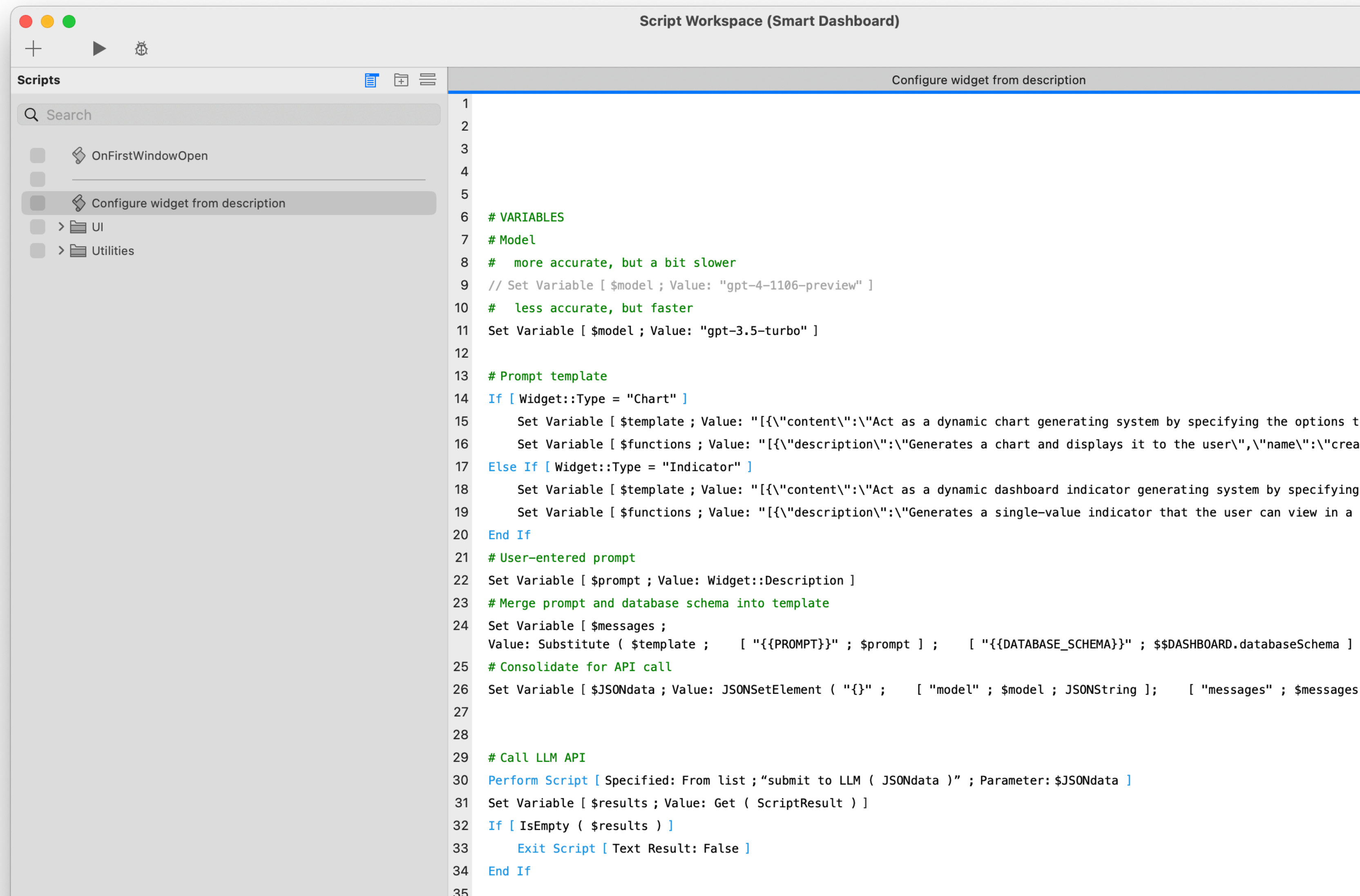
Script

- One main script



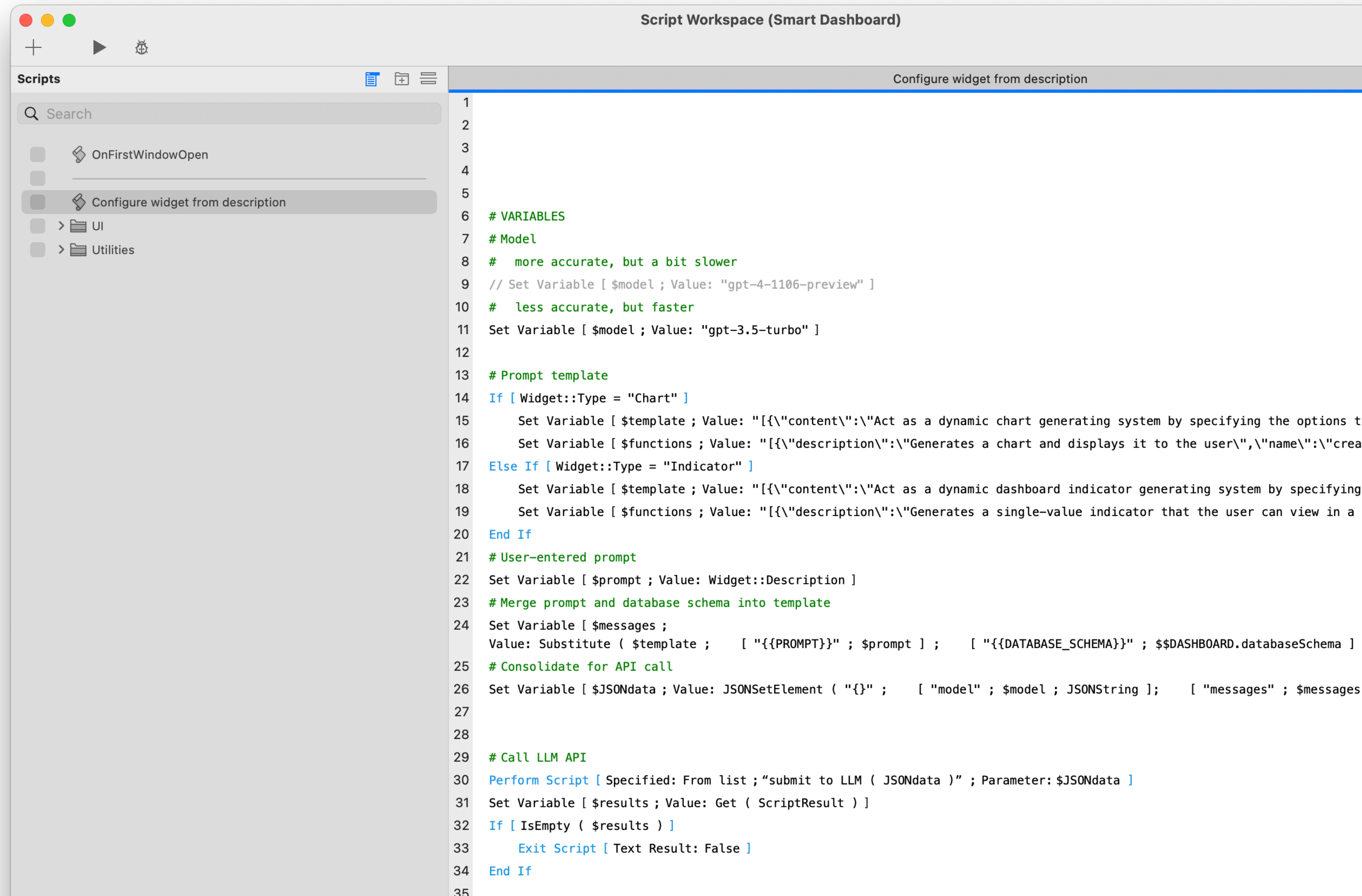
Script

- One main script
- Generate SQL and settings



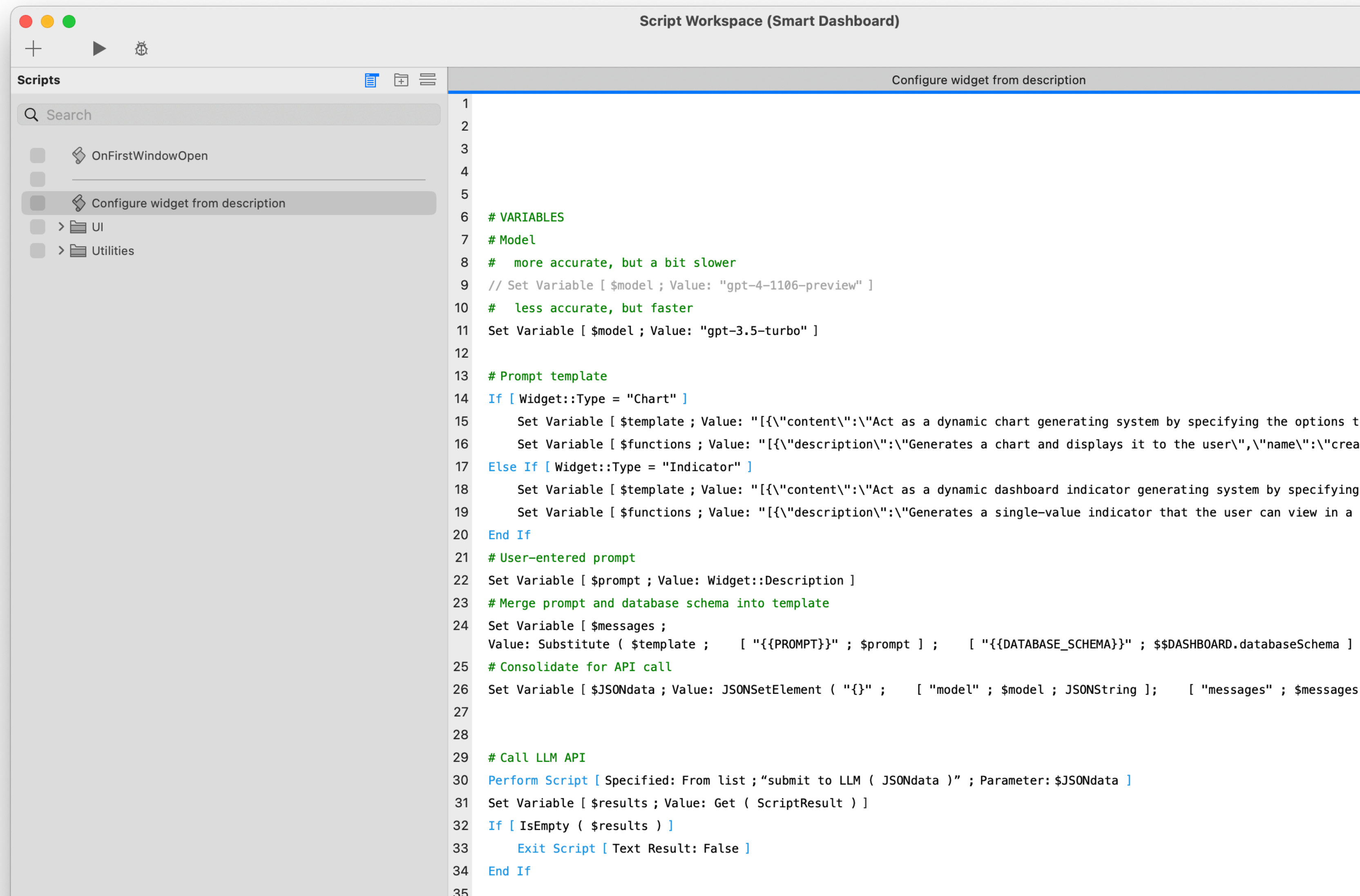
Script

- One main script
- Generate SQL and settings
- Query runs locally

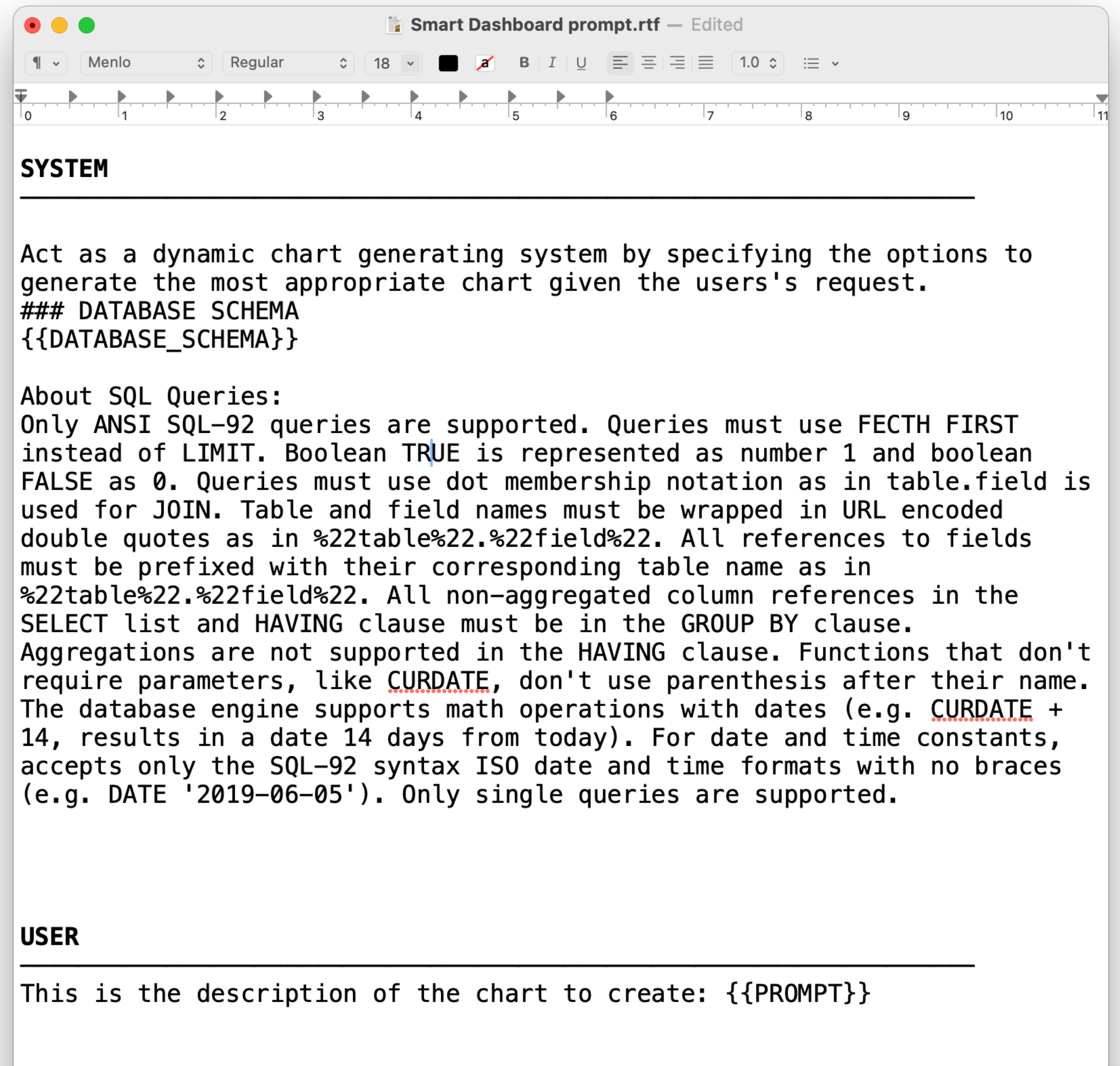


Script

- One main script
- Generate SQL and settings
- Query runs locally
- Chart is calculated

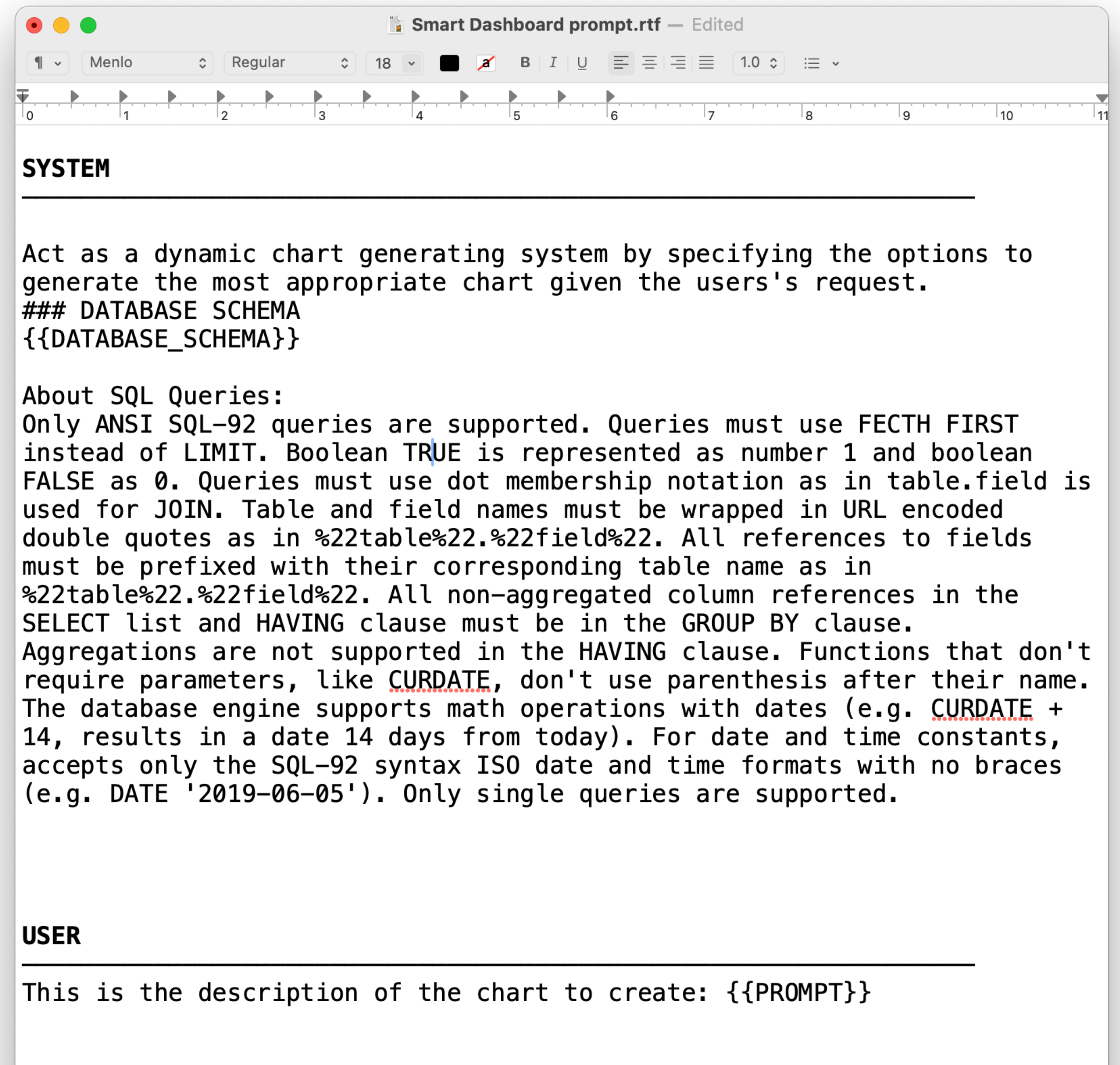


Prompt



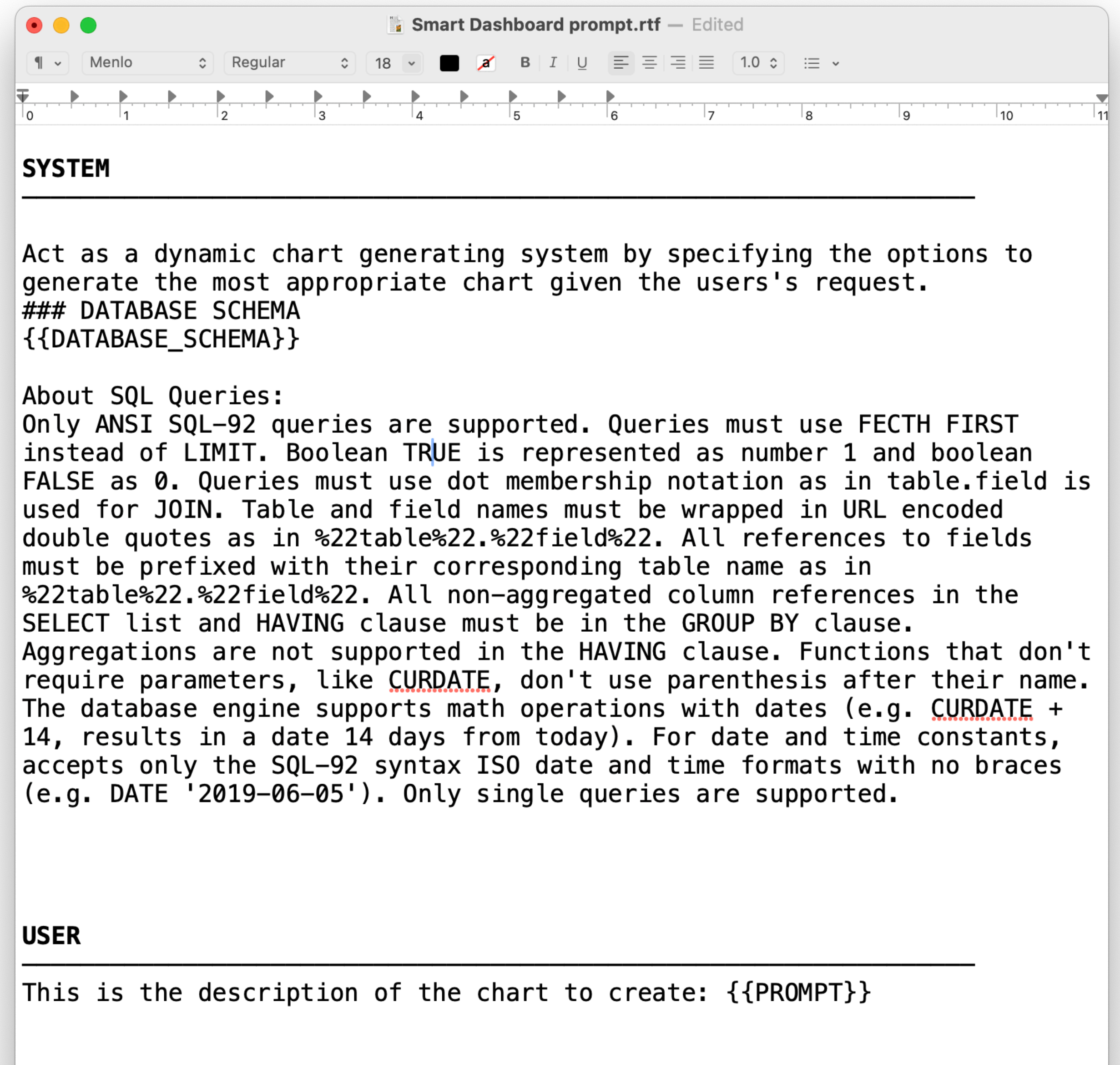
Prompt

- Role



Prompt

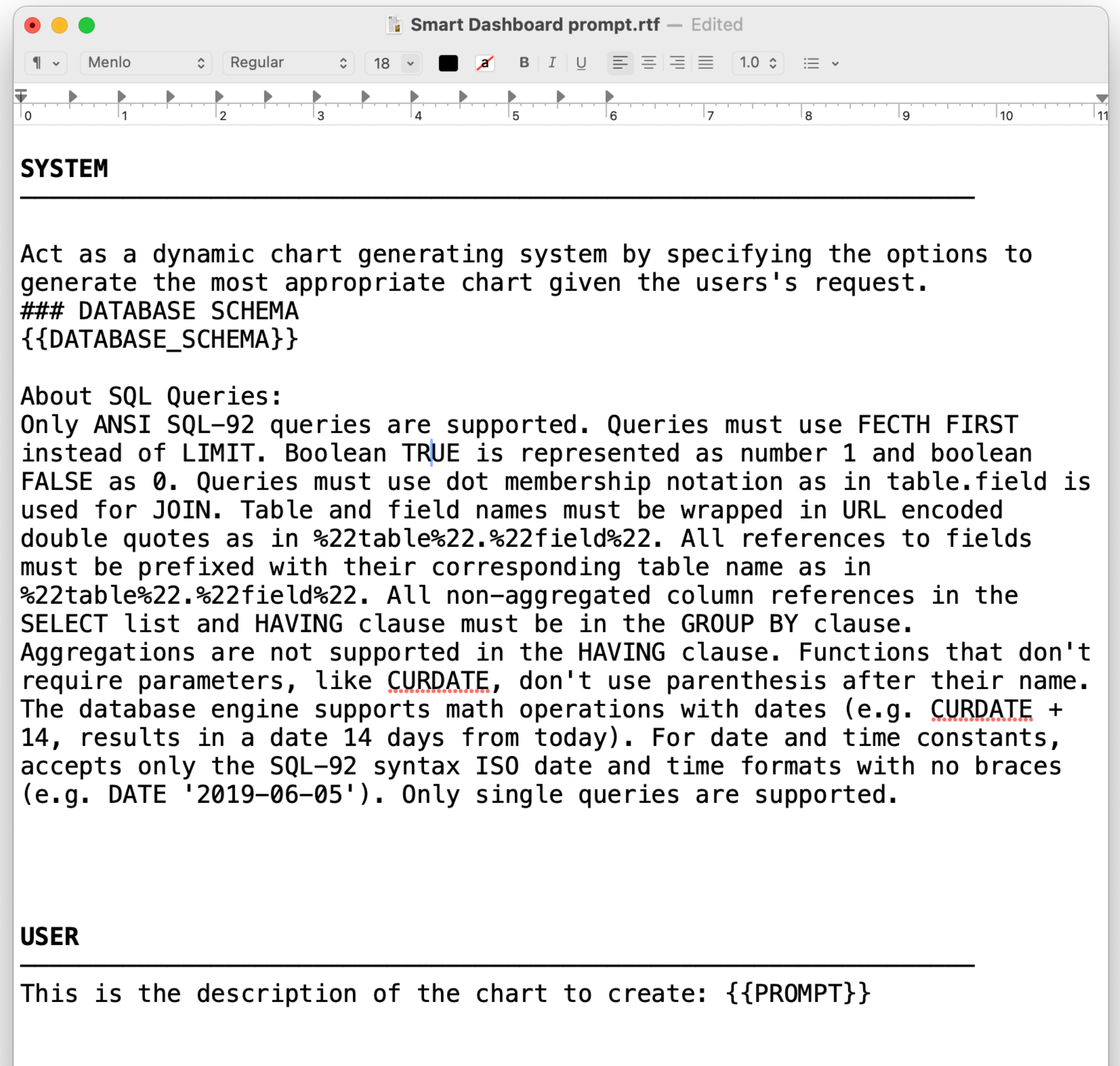
- Role
- Constrains



```
Smart Dashboard prompt.rtf — Edited
Menlo Regular 18
SYSTEM
Act as a dynamic chart generating system by specifying the options to
generate the most appropriate chart given the users's request.
### DATABASE SCHEMA
{{DATABASE_SCHEMA}}
About SQL Queries:
Only ANSI SQL-92 queries are supported. Queries must use FECTH FIRST
instead of LIMIT. Boolean TRUE is represented as number 1 and boolean
FALSE as 0. Queries must use dot membership notation as in table.field is
used for JOIN. Table and field names must be wrapped in URL encoded
double quotes as in %22table%22.%22field%22. All references to fields
must be prefixed with their corresponding table name as in
%22table%22.%22field%22. All non-aggregated column references in the
SELECT list and HAVING clause must be in the GROUP BY clause.
Aggregations are not supported in the HAVING clause. Functions that don't
require parameters, like CURDATE, don't use parenthesis after their name.
The database engine supports math operations with dates (e.g. CURDATE +
14, results in a date 14 days from today). For date and time constants,
accepts only the SQL-92 syntax ISO date and time formats with no braces
(e.g. DATE '2019-06-05'). Only single queries are supported.
USER
This is the description of the chart to create: {{PROMPT}}
```


Prompt

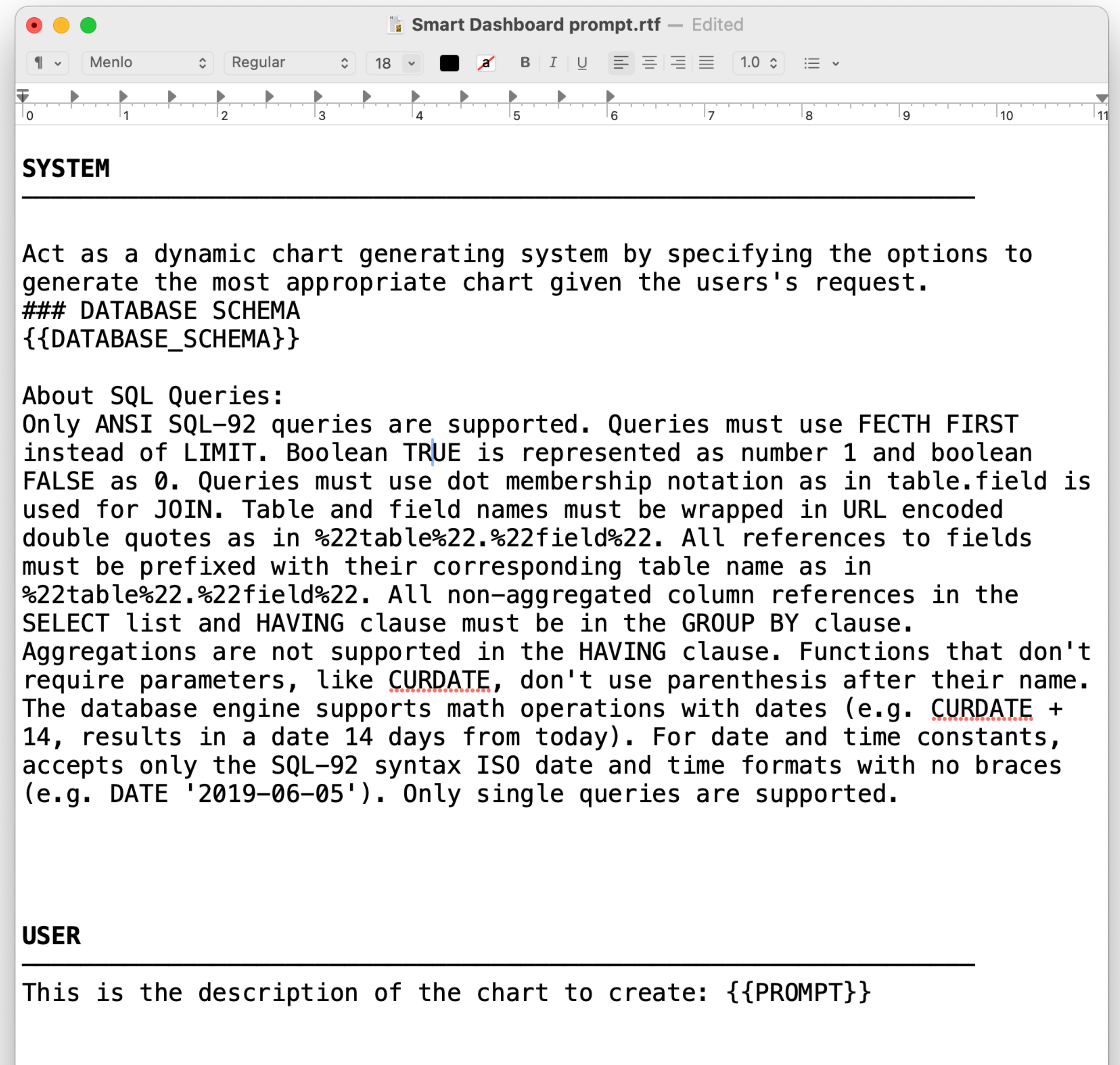
- Role
- Constrains
- Generate JSON



```
Smart Dashboard prompt.rtf — Edited
Menlo Regular 18
SYSTEM
Act as a dynamic chart generating system by specifying the options to
generate the most appropriate chart given the users's request.
### DATABASE SCHEMA
{{DATABASE_SCHEMA}}
About SQL Queries:
Only ANSI SQL-92 queries are supported. Queries must use FECTH FIRST
instead of LIMIT. Boolean TRUE is represented as number 1 and boolean
FALSE as 0. Queries must use dot membership notation as in table.field is
used for JOIN. Table and field names must be wrapped in URL encoded
double quotes as in %22table%22.%22field%22. All references to fields
must be prefixed with their corresponding table name as in
%22table%22.%22field%22. All non-aggregated column references in the
SELECT list and HAVING clause must be in the GROUP BY clause.
Aggregations are not supported in the HAVING clause. Functions that don't
require parameters, like CURDATE, don't use parenthesis after their name.
The database engine supports math operations with dates (e.g. CURDATE +
14, results in a date 14 days from today). For date and time constants,
accepts only the SQL-92 syntax ISO date and time formats with no braces
(e.g. DATE '2019-06-05'). Only single queries are supported.
USER
This is the description of the chart to create: {{PROMPT}}
```

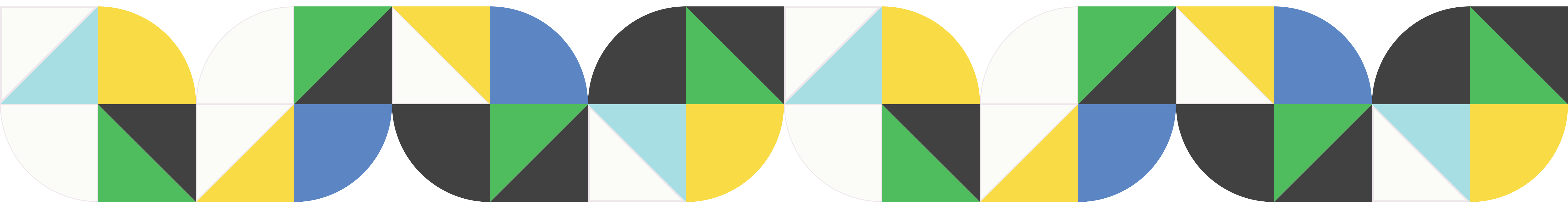
Prompt

- Role
- Constrains
- Generate JSON
- Zero shot



```
Smart Dashboard prompt.rtf — Edited
Menlo Regular 18
SYSTEM
Act as a dynamic chart generating system by specifying the options to
generate the most appropriate chart given the users's request.
### DATABASE SCHEMA
{{DATABASE_SCHEMA}}
About SQL Queries:
Only ANSI SQL-92 queries are supported. Queries must use FECTH FIRST
instead of LIMIT. Boolean TRUE is represented as number 1 and boolean
FALSE as 0. Queries must use dot membership notation as in table.field is
used for JOIN. Table and field names must be wrapped in URL encoded
double quotes as in %22table%22.%22field%22. All references to fields
must be prefixed with their corresponding table name as in
%22table%22.%22field%22. All non-aggregated column references in the
SELECT list and HAVING clause must be in the GROUP BY clause.
Aggregations are not supported in the HAVING clause. Functions that don't
require parameters, like CURDATE, don't use parenthesis after their name.
The database engine supports math operations with dates (e.g. CURDATE +
14, results in a date 14 days from today). For date and time constants,
accepts only the SQL-92 syntax ISO date and time formats with no braces
(e.g. DATE '2019-06-05'). Only single queries are supported.
USER
This is the description of the chart to create: {{PROMPT}}
```

Tips for applying AI



Tips for applying AI



Tips for applying AI

Regression and classification



Tips for applying AI

Regression and classification

- Your data is key



Tips for applying AI

Regression and classification

- Your data is key
- Opportunity when users are guessing



Tips for applying AI



Tips for applying AI

LLM



Tips for applying AI

LLM

- Like new intern eager to help



Tips for applying AI

LLM

- Like new intern eager to help
- Aim for small responses



Tips for applying AI

LLM

- Like new intern eager to help
- Aim for small responses
- Prototype with commercial model



Tips for applying AI

LLM

- Like new intern eager to help
- Aim for small responses
- Prototype with commercial model
- Don't forget about costs



Tips for applying AI



Tips for applying AI

In general



Tips for applying AI

In general

- Learn



Tips for applying AI

In general

- Learn
- Practice



Tips for applying AI

In general

- Learn
- Practice
- Practice



Tips for applying AI

In general

- Learn
- Practice
- Practice
- Practice



Next steps

Talk to us

Community

Other sessions

What to learn next



Clariss booth

11:30 a.m. - 1:00 p.m.

- Ask questions
- Live demos
- More






Attend these two AI sessions today:

- AI foundations for Claris FileMaker developers - Ernest Koe
Wednesday, 10:30 a.m. CT
- AI in business: Responsible real-world applications - Cris Ippolite
Wednesday, 1:30 p.m. CT



Attend these two AI sessions tomorrow:

- Claris AI Under the Hood Session - Wade Ju
Thursday, 1:30 p.m. CT
- Let's chat! - Joris Aarts
Thursday, 3:00 p.m. CT



Join our Claris Community to
participate in our AI discussions.



<https://bit.ly/3tL1OVq>

What to learn next



What to learn next

- Prompt engineering



What to learn next

- Prompt engineering
- Run open source models locally



What to learn next

- Prompt engineering
- Run open source models locally
 - Ollama



What to learn next

- Prompt engineering
- Run open source models locally
 - Ollama
 - GPT4All



What to learn next

- Prompt engineering
- Run open source models locally
 - Ollama
 - GPT4All
 - LM Studio



What to learn next

- Prompt engineering
- Run open source models locally
 - Ollama
 - GPT4All
 - LM Studio
- Create ML



What to learn next

- Prompt engineering
- Run open source models locally
 - Ollama
 - GPT4All
 - LM Studio
- Create ML
- Vector embeddings and semantic search



What to learn next

- Prompt engineering
- Run open source models locally
 - Ollama
 - GPT4All
 - LM Studio
- Create ML
- Vector embeddings and semantic search
- Retrieval augmented generation (RAG)



Q&A




Key takeaway

AI in FileMaker is a reality today.



Thank you.



Join our Claris Community to
participate in our AI discussions.



<https://bit.ly/3tL1OVq>