

MCP WORKSHOP

ISHITA GUPTA

WHY DO WE NEED MCP?

You are a busy student who tells people you want to get lunch, but never actually follows up



WHY DO WE NEED MCP?

You want to build a simple agent that helps with this problem



WHY DO WE NEED MCP?

You want to build a simple agent that

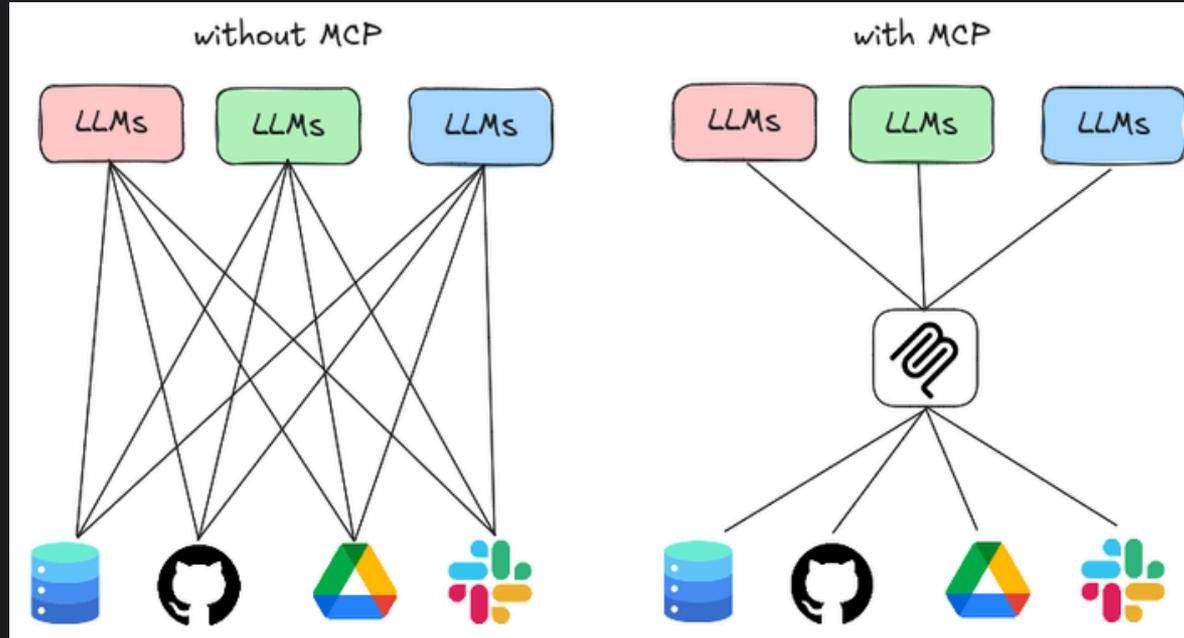
1. Checks your calendar
2. Texts your friend some times you are free to get lunch



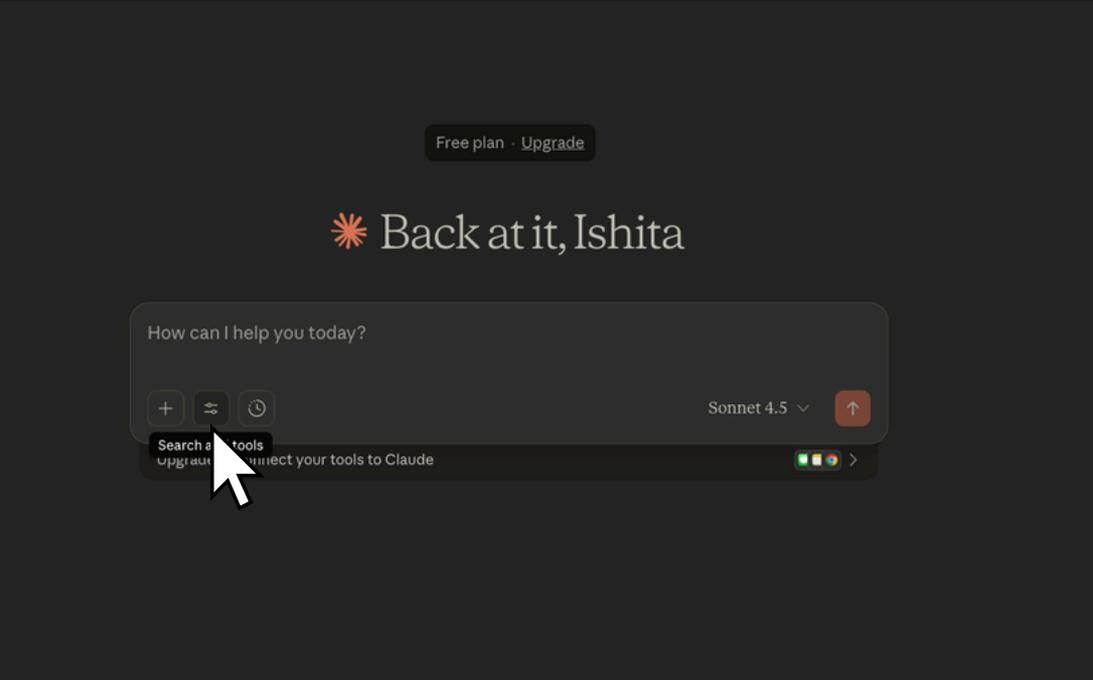
WHY DO WE NEED MCP?



WHY DO WE NEED MCP?



LETS MAKE AGENT!



LETS MAKE AGENT!

✨ Back at it, Ishita

How can I help you today?



Sonnet 4.5



Search menu

Use style

Extended thinking

Research

Set up

Web search

Drive search

Stu

Gmail search

Lea

Calendar search

sub

Notion

Control Chrome

Read and Send iMessages

+ Add connectors

Life stuff

From Calendar

or your

Research project

Analyze and organize

research

LETS MAKE AGENT!

Connectors

Unlock more with Claude when you connect with remote and local tools. Choose from Anthropic-reviewed tools, add a custom one — or [learn more about connectors](#).

Web Desktop extensions Manage connectors

 Filesystem 244K Let Claude access your filesystem to read and write files.	 Read and Send IM... 198K Send, read, and manage messages through Apple's Messages app.
 Control Chrome 189K Control Google Chrome browser tabs, windows, and navigation.	 Figma Dev Mode 64K Provides important design information and context when generating code from Figm...
 Windows-MCP 68K Lightweight MCP Server that enables Claude to interact with Windows OS.	 Read and Write Apple Notes 58K Read, write, and manage notes in Apple Notes.
 PDF Tools - Analyze, Extra... 51K Let Claude work with PDFs on your computer: read, analyze, fill forms, extrac...	 Control your Mac 42K Execute AppleScript to automate tasks on macOS.
 Context7 48K Up-to-date Code Docs For Any Prompt.	 Spotify (AppleScript) 16K Control Spotify via AppleScript.
 Kapture Browser Automat... 15K Control web browsers through the Model Context Protocol using Chrome Dev Tools.	 Postman MCP Server (Min... 13K Connect your AI to your APIs on Postman.
 ClickUp 9.8K Search, create, and retrieve tasks and documents, add comments, and track tim...	 Airtable MCP Server 9.3K Read and write access to Airtable databases via the Model Context Protocol.
 Desktop Commander 8.4K	 Brave (AppleScript) 7.5K

AGENT ACTIVITY 😊



MCP ISN'T MAGIC:

- When you asked the agent to send a message, what do you think it did first — before touching any tool?
- How do you think it decided which tool could help it complete that request?
- Once it picked a tool, what information do you think it had to pass along for the tool to actually work?

HOW IT WORKS

Tool Selection — choosing which external capability to use.

Tool Invocation — the actual call and context that goes out.



HOW IT WORKS

Tool Description — how an agent knows what each tool does.

Tool Calling — what parameters to pass into the tool



WHERE ITS STORED

```
~/Library/Application Support/Claude/claude_desktop_config.json
```



IMESSAGE TOOLS

iMessage Tools

1. send_imessage

- **Description:** Send an iMessage using Messages app
- **Parameters:**
 - `message` (required): Message content to send
 - `recipient` (required): Phone number or email of the recipient

2. search_contacts

- **Description:** Search contacts by name, phone, or email
- **Parameters:**
 - `query` (required): Search query

3. read_imessages

- **Description:** Read iMessages from a specific contact
- **Parameters:**
 - `phone_number` (required): Phone number of the contact to read messages from
 - `limit` (optional): Maximum number of messages to retrieve (default: 10)

4. get_unread_imessages

- **Description:** Get all unread iMessages
- **Parameters:**
 - `limit` (optional): Maximum number of unread messages to retrieve (default: 10)

IMESSAGE TOOLS

```
json
{
  "name": "Read and Send iMessages:send_message",
  "description": "Send an iMessage using Messages app",
  "parameters": {
    "type": "object",
    "properties": {
      "message": {
        "type": "string",
        "description": "Message content to send"
      },
      "recipient": {
        "type": "string",
        "description": "Phone number or email of the recipient"
      }
    },
    "required": ["message", "recipient"]
  }
}
```

```
json
{
  "name": "Read and Send iMessages:search_contacts",
  "description": "Search contacts by name, phone, or email",
  "parameters": {
    "type": "object",
    "properties": {
      "query": {
        "type": "string",
        "description": "Search query"
      }
    },
    "required": ["query"]
  }
}
```

GOOGLE CALENDAR TOOLS

Google Calendar Tools

1. list_gcal_calendars

- **Description:** List all available calendars in Google Calendar
- **Parameters:**
 - `page_token` (optional): Token for pagination

2. fetch_gcal_event

- **Description:** Retrieve a specific event from a Google calendar
- **Parameters:**
 - `calendar_id` (required): The ID of the calendar containing the event
 - `event_id` (required): The ID of the event to retrieve

3. list_gcal_events

- **Description:** List or search events from a specific Google Calendar. Supports free text search and time filtering.
- **Parameters:**
 - `calendar_id` (optional): Calendar ID (default: 'primary')
 - `query` (optional): Free text search terms to find events
 - `time_min` (optional): Lower bound for event's end time (RFC3339 format)
 - `time_max` (optional): Upper bound for event's start time (RFC3339 format)
 - `time_zone` (optional): Time zone for the response (IANA format)
 - `max_results` (optional): Maximum number of events per calendar (default: 25)
 - `page_token` (optional): Token for pagination

4. find_free_time

- **Description:** Find free time periods across a list of calendars
- **Parameters:**
 - `calendar_ids` (required): List of calendar IDs to analyze
 - `time_min` (required): Lower bound for free time search (RFC3339 format)
 - `time_max` (required): Upper bound for free time search (RFC3339 format)
 - `time_zone` (optional): Time zone for the response (IANA format)

These tools allow me to help you manage iMessages and Google Calendar events!

GOOGLE CALENDAR TOOLS

```
json
{
  "name": "list_gcal_calendars",
  "description": "List all available calendars in Google Calendar.",
  "parameters": {
    "type": "object",
    "title": "ListCalendarsInput",
    "properties": {
      "page_token": {
        "type": ["string", "null"],
        "default": null,
        "description": "Token for pagination",
        "title": "Page Token"
      }
    }
  }
}
```

```
json
{
  "name": "fetch_gcal_event",
  "description": "Retrieve a specific event from a Google calendar.",
  "parameters": {
    "type": "object",
    "title": "GetEventInput",
    "properties": {
      "calendar_id": {
        "type": "string",
        "description": "The ID of the calendar containing the event",
        "title": "Calendar Id"
      },
      "event_id": {
        "type": "string",
        "description": "The ID of the event to retrieve",
        "title": "Event Id"
      }
    },
    "required": ["calendar_id", "event_id"]
  }
}
```

WHAT DID I DO THIS SUMMER?

MCP Evals!

Built a Big Agent System

Tried to See if it did ok?

WHAT DID I DO THIS SUMMER?

WHAT DID I DO THIS SUMMER?

MCP

Evals!

WHAT DID I DO THIS SUMMER?

MCP
Evals!



WHAT DID I DO THIS SUMMER?

MCP
Evals!



WHAT DID I DO THIS SUMMER?

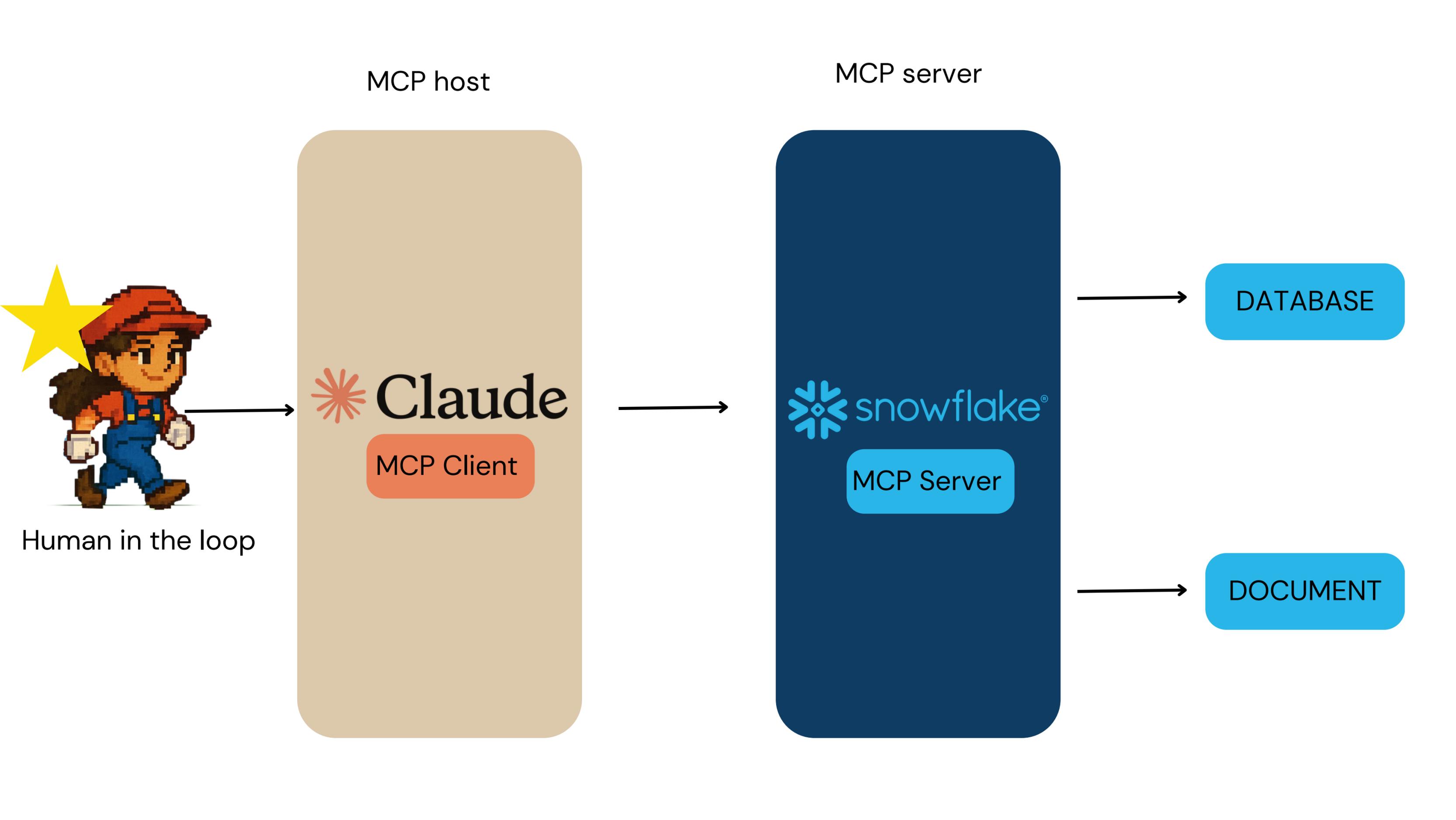
MCP
Evals!



WHAT DID I DO THIS SUMMER?

MCP
Evals!





MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liabilities each have. Then reason about why this could be the case



Claude

MCP Client

```
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:
Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned by these operators (although they could still be returned by the overall query with the use of @not).
TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is interpreted in UTC.
These operators can be combined into a single filter object.
Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):
{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
] }
###
```

DATABASE

```
SELECT
  1. chunk
  2. relative_path
  3. file_url
  4. language
FROM cortex_search_tutorial_db.public.doc_chunks
```

snowflake

MCP Server

Human in the loop

Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
M sec_filings_analyst
query: 'What are the top 10 most profitable companies by net income or profits in the most recent year available?'
}
```

Response

```
[{"type": "text", "text": "\u201cI apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability"}]
```

Let me try a more specific query for net income data:

```
M sec_filings_analyst
Request
{
  query: 'Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10'
}
```

Response

```
[{"type": "text", "text": "\u201cThis is our interpretation of your"}]
```

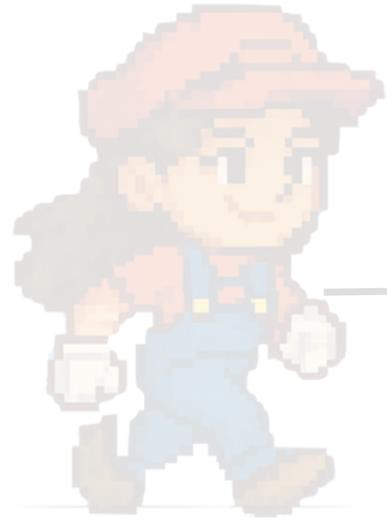
```
Users > igueta > Library > Application Support > Claude > / tools_config.yaml
tools_config.yaml
- service_name: 'airbnb_svc'
- descriptions:
  - 'cooking/baking' - amenities: 'Kitchen'
  - 'kitchen/entertainment' - amenities: 'TV', 'Table top'
  - 'parking/car' - amenities: 'Free parking on premises', 'Paid parking on premises'
  - 'families/kids' - amenities: 'Family/kid friendly', 'room_type: Entire home/apt'
  - 'pets/pets' - amenities: 'Pets allowed'
  - 'laundry/washing' - amenities: 'Washer', 'Dryer'
  - 'water/cold' - amenities: 'Heating', 'Indoor fireplace'
  - 'swimming' - amenities: 'Pool'
  - 'outdoor space' - amenities: 'Patio or balcony', 'Garden or backyard'
  - 'romantic/cozy' - amenities: 'Hot tub', 'Indoor fireplace', 'Private entrance'
- EXAMPLE FILTERS:
  Family trip: (@and: ((@eq: ('market': 'Sydney')), (@gte: ('bedrooms': 2))), (@contains: ('amenities': 'Family/kid friendly')))
  Budget cooking: (@and: ((@lte: ('price': 300)), (@contains: ('amenities': 'Kitchen'))))
  Pet-friendly: (@and: ((@contains: ('amenities': 'Pets allowed')), (@contains: ('amenities': 'Garden or backyard'))))
- database_name: 'CORTEX_SEARCH_TUTORIAL_DB'
- schema_name: 'PUBLIC'
- columns: []
- limit: 10
- service_name: 'sec_filings_search'
- descriptions:
  - 'SEC filings and corporate financial documents search service.'
- FILTERABLE ATTRIBUTES:
  - language (TEXT): document language
  - Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures.'
- database_name: 'CORTEX_SEARCH_TUTORIAL_DB'
- schema_name: 'PUBLIC'
- columns: ['chunk', 'relative_path', 'file_url', 'language']
- limit: 10
- analyst_services:
  - service_name: 'revenue_analytics'
  - semantic_model: 'CORTEX_ANALYST_REVENUE_TIMESERIES_RAW_DATA/revenue_timeseries.yaml'
  - descriptions:
    - Analyst service that provides insights into revenue trends, time series analysis, and financial performance metrics for business intelligence and forecasting.
  - service_name: 'sec_filings_analyst'
  - semantic_model: 'CORTEX_SEARCH_TUTORIAL_DB_SEC_DATA/SEMANTIC_MODELS/new_sec_semantic_model.yaml'
  - descriptions:
    - This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companies. When querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time. For example, if the user asks "What are the revenue trends across telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and then compare them and summarize the trends.
```

DOCUMENT

MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liabilities each have. Then reason about why this could be the case



Human in the loop



```
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

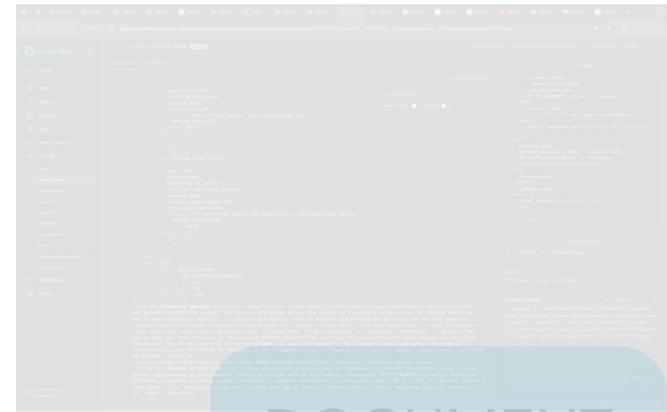
The following usage notes apply:
Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned by these operators (although they could still be returned by the overall query with the use of @not).
TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is interpreted in UTC.
These operators can be combined into a single filter object.
Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):
{
  "@and": [
    { "@gte": { "numeric_col": 10.5 } },
    { "@lte": { "numeric_col": 12.5 } }
  ]
}
```



Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
M sec_filings_analyst
{
  "query": "Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10"
}
```

```
1. sec_filings_analyst
2. sec_filings_analyst
3. sec_filings_analyst
4. sec_filings_analyst
5. sec_filings_analyst
6. sec_filings_analyst
7. sec_filings_analyst
8. sec_filings_analyst
9. sec_filings_analyst
10. sec_filings_analyst
11. sec_filings_analyst
12. sec_filings_analyst
13. sec_filings_analyst
14. sec_filings_analyst
15. sec_filings_analyst
16. sec_filings_analyst
17. sec_filings_analyst
18. sec_filings_analyst
19. sec_filings_analyst
20. sec_filings_analyst
21. sec_filings_analyst
22. sec_filings_analyst
23. sec_filings_analyst
24. sec_filings_analyst
25. sec_filings_analyst
26. sec_filings_analyst
27. sec_filings_analyst
28. sec_filings_analyst
29. sec_filings_analyst
30. sec_filings_analyst
31. sec_filings_analyst
32. sec_filings_analyst
33. sec_filings_analyst
34. sec_filings_analyst
35. sec_filings_analyst
36. sec_filings_analyst
37. sec_filings_analyst
38. sec_filings_analyst
39. sec_filings_analyst
40. sec_filings_analyst
41. sec_filings_analyst
42. sec_filings_analyst
43. sec_filings_analyst
44. sec_filings_analyst
45. sec_filings_analyst
46. sec_filings_analyst
47. sec_filings_analyst
48. sec_filings_analyst
49. sec_filings_analyst
50. sec_filings_analyst
51. sec_filings_analyst
52. sec_filings_analyst
53. sec_filings_analyst
54. sec_filings_analyst
55. sec_filings_analyst
56. sec_filings_analyst
57. sec_filings_analyst
58. sec_filings_analyst
59. sec_filings_analyst
60. sec_filings_analyst
61. sec_filings_analyst
62. sec_filings_analyst
63. sec_filings_analyst
64. sec_filings_analyst
65. sec_filings_analyst
66. sec_filings_analyst
67. sec_filings_analyst
68. sec_filings_analyst
69. sec_filings_analyst
70. sec_filings_analyst
71. sec_filings_analyst
72. sec_filings_analyst
73. sec_filings_analyst
74. sec_filings_analyst
75. sec_filings_analyst
76. sec_filings_analyst
77. sec_filings_analyst
78. sec_filings_analyst
79. sec_filings_analyst
80. sec_filings_analyst
81. sec_filings_analyst
82. sec_filings_analyst
83. sec_filings_analyst
84. sec_filings_analyst
85. sec_filings_analyst
86. sec_filings_analyst
87. sec_filings_analyst
88. sec_filings_analyst
89. sec_filings_analyst
90. sec_filings_analyst
91. sec_filings_analyst
92. sec_filings_analyst
93. sec_filings_analyst
94. sec_filings_analyst
95. sec_filings_analyst
96. sec_filings_analyst
97. sec_filings_analyst
98. sec_filings_analyst
99. sec_filings_analyst
100. sec_filings_analyst
```



MCP host

MCP server

Let me start by querying the SEC filings analyst service to get the most profitable companies.

M sec_filings_analyst

```
query: what are the top 10 most profitable companies by net income or profit
in the most recent year available?`
}
```

Response

```
[{"type": "text", "text": "{\"text\": \"I apologize, but I need more clarity on how
to measure profitability. While we have financial data in the semantic model, we
need to specify which measure (tag/variable) should be used to determine net income
or profit. The model contains various financial metrics but we need to know the
specific measure to use for profitability\""}]
```

Let me try a more specific query for net income data:

M sec_filings_analyst

Request

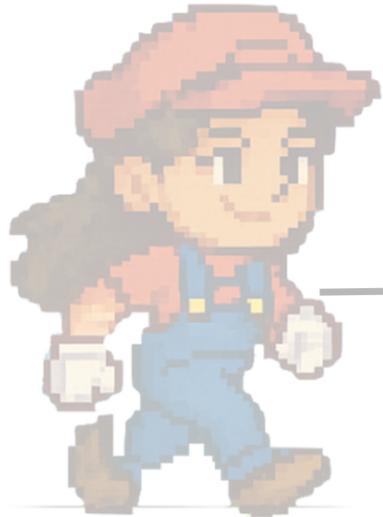
```
{
  `query`: `Show me the companies with the highest net income values, including the
net income amount and company name, sorted by net income descending, limit to top
10`
}
```

Response

```
[{"type": "text", "text": "{\"text\": \"This is our interpretation of your\""}]
```

I notice some unusual values in that result that seem too high. Let me try a different approach

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liabilities each have. Then reason about why this could be the case



Human in the loop

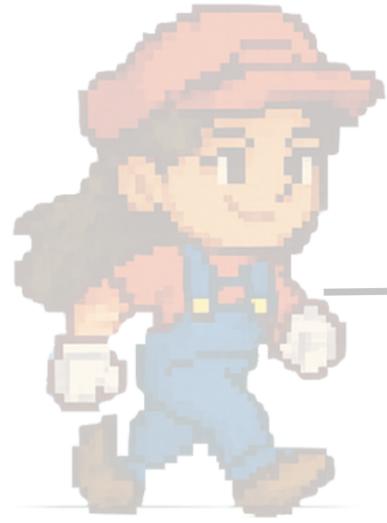
Cortex Search

Cortex Analyst

MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liabilities each have. Then reason about why this could be the case



Human in the loop

Let me start by querying the SEC filings analyst service to get the most profitable companies.

```

M_sec_filings_analyst
query: 'Show me the top 10 most profitable companies by total income for profits in the most recent year available?'
  
```

Response

```

[{"type": "text", "text": "\u005cI apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability"}
  
```

Let me try a more specific query for net income data:

```

M_sec_filings_analyst
Request
{
  query: 'Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10'
}
  
```

Response

```

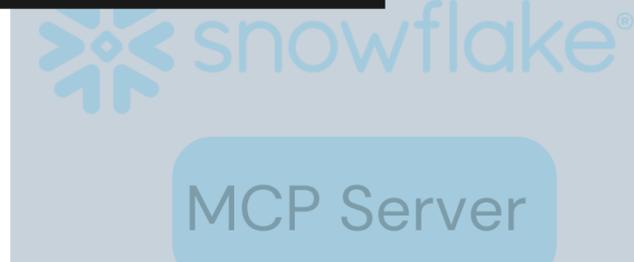
[{"type": "text", "text": "\u005cThis is our interpretation of your"}
  
```

```

Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

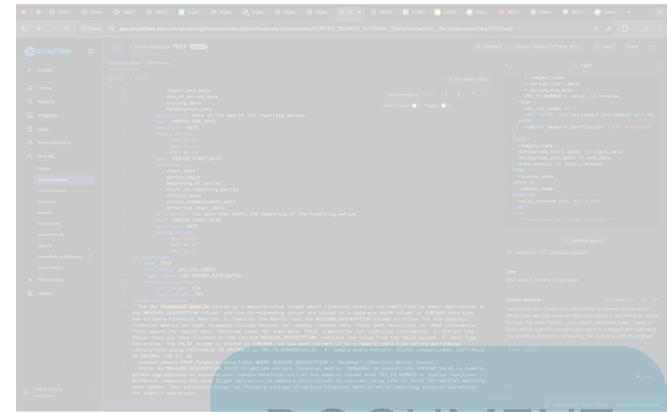
These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:
Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned by these operators (although they could still be returned by the overall query with the use of @not).
TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is interpreted in UTC.
These operators can be combined into a single filter object.
Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):
{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
]}
  
```



```

Users | Google | Library | Application Support | Claude | tools_config.yaml
tools_config.yaml
- name: sec_filings_analyst
  description: 'SEC Filings and Corporate Financial Documents Search Service'
  capabilities:
    - read
  api_url: 'https://api.cortex-search.snowflake.com/v1/public/sec_filings'
  headers:
    Content-Type: application/json
  body:
    query: 'Show me the top 10 most profitable companies by total income for profits in the most recent year available?'
  
```



MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liabilities each have. Then reason about why this could be the case



Claude

MCP Client

```
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned by these operators (although they could still be returned by the overall query with the use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):
{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
] }
```

snowflake

MCP Server

DATA

Human in the loop

Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
M sec_filings_analyst
query: "Who are the top 10 most profitable companies by total income or profit in the most recent year available?"
```

Response

```
[{"type": "text", "text": "\u005c\u005c\u005c\u005cI apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability"}]
```

Let me try a more specific query for net income data:

```
M sec_filings_analyst
Request
{
  query: "Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10"
}
```

Response

```
[{"type": "text", "text": "\u005c\u005c\u005c\u005cThis is our interpretation of your"}]
```

```
Users | Claude | Application Support | Claude | tools_config.yaml
1  = service_name: "airbook_svc"
2  = model: "claude-3.5-sonnet"
3  = provider: "anthropic"
4  = api_base: "https://api.anthropic.com"
5  = api_key: "ANTHROPIC_API_KEY"
6  = headers: {}
7  = metadata: {}
8  = max_tokens: 4096
9  = temperature: 0.0
10 = tool_resources: {}
11 = context_window: 128000
12 = system_prompt: ""
13 = tags: {}
14 = tool_use_behavior: "auto"
15 = tool_use_no_fallback: true
16 = tool_use_warn_on_timeout: true
17 = tool_use_warn_on_no_tool: true
18 = tool_use_warn_on_no_tool_text: ""
19 = tool_use_warn_on_no_tool_timeout: 30
20 = tool_use_warn_on_no_tool_timeout_text: ""
21 = tool_use_warn_on_no_tool_timeout_text_color: "red"
22 = tool_use_warn_on_no_tool_timeout_text_size: 12
23 = tool_use_warn_on_no_tool_timeout_text_weight: "bold"
24 = tool_use_warn_on_no_tool_timeout_text_align: "center"
25 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
26 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
27 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
28 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
29 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
30 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
31 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
32 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
33 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
34 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
35 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
36 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
37 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
38 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
39 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
40 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
41 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
42 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
43 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
44 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
45 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
46 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
47 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
48 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
49 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
50 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
51 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
52 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
53 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
54 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
55 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
56 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
57 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
58 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
59 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
60 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
61 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
62 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
63 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
64 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
65 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
66 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
67 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
68 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
69 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
70 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
71 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
72 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
73 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
74 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
75 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
76 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
77 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
78 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
79 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
80 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
81 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
82 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
83 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
84 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
85 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
86 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
87 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
88 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
89 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
90 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
91 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
92 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
93 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
94 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
95 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
96 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
97 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
98 = tool_use_warn_on_no_tool_timeout_text_font_weight: "bold"
99 = tool_use_warn_on_no_tool_timeout_text_font_color: "red"
100 = tool_use_warn_on_no_tool_timeout_text_font_size: 12
```

DOCUMENT

MCP host

MCP server

I want you to query companies are. Ba
Then reason about

```
1 (
2   SELECT
3     chunk,
4     relative_path,
5     file_url,
6     language
7   FROM cortex_search_tutorial_db.public.doc_chunks
8 )
```

Cortex Search supports four matching operators:

1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:

- @and
- @or
- @not

Example:

```
{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
] }
```

```
1 SELECT
2   name,
3   room_type,
4   address,
5   bed_type,
6   bedrooms,
7   beds,
8   cancellation_policy,
9   cleaning_fee,
10  guests_included,
11  price,
12  property_type,
13  description,
14  house_rules,
15  notes,
16  summary,
17  transit,
18  neighborhood_overview,
19  space,
20  parse_json(address)::STRING as country,
21  parse_json(address)::government_area::STRING as government_area,
22  parse_json(address)::market::STRING as market,
23  parse_json(address)::street::STRING as street,
24  parse_json(address)::suburb::STRING as suburb,
25  ARRAY_CONSTRUCT_COMPACT(parse_json(amenities)) as amenities,
26
27  ('Address\n\n' || address || 'Summary\n\n' || summary || '\n\nDescription\n\n' || description || '\n\nNeighborhood
Overview\n\n' || neighborhood_overview || '\n\nSpace\n\n' || space || '\n\nTransit\n\n' || transit || '\n\nAmenities\n\n' ||
amenities) as listing_text
28 FROM
29  cortex_search_tutorial_db.public.airbnb_listings
```

DATABASE



Human in the loop

The screenshot shows the Snowflake Cortex Analyst interface. On the left is a navigation sidebar with options like Home, Search, Projects, Data, AI & ML, Studio, and Monitoring. The main area is split into two panes. The left pane shows a 'Semantic View' for a table named 'TEST', listing columns like 'report_end_date', 'end_of_period_date', and 'closing_date' with their data types and sample values. The right pane shows a SQL query editor with a query that filters for 'financial data is' and aggregates revenue data by company name. A 'Verified query' button is visible below the query. At the bottom, there's a chat interface with a 'User' prompt: 'how does it come to alphabet' and a 'Cortex Analyst' response: 'I apologize, but I need more information to answer this question. While I can see that revenue data is available in the semantic model through the value field in __sec_report_attributes table, I need to know which specific company you want to compare with Alphabet. The question mentions comparing 'the company with the highest'.

DOCUMENT

MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liabilities each have. Then reason about why this could be the case



Claude

MCP Client

```
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:

Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned by these operators (although they could still be returned by the overall query with the use of @not).

TIMESTAMP and DATE filters accept values of the form: YYYY-MM-DD and, for timezone aware dates: YYYY-MM-DD+HH:MM. If the timezone offset is not specified, the date is interpreted in UTC.

These operators can be combined into a single filter object.

Example:
Filtering on rows where NUMERIC column numeric_col is between 10.5 and 12.5 (inclusive):
{ "@and": [
  { "@gte": { "numeric_col": 10.5 } },
  { "@lte": { "numeric_col": 12.5 } }
] }
```

snowflake

MCP Server

DATA

```
SELECT
  company_name,
  report_date,
  relative_path,
  file_url,
  language
FROM cortex_analyst_db.public.doc_chunks
```

Human in the loop

Let me start by querying the SEC filings analyst service to get the most profitable companies.

```
M sec_filings_analyst
query: Who are the top 10 most profitable companies by total income or profit in the most recent year available?
```

Response

```
[{"type": "text", "text": "\u005c\u005c\u005cI apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability"}]
```

Let me try a more specific query for net income data:

```
M sec_filings_analyst
Request
{
  query: 'Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10'
}
```

Response

```
[{"type": "text", "text": "\u005c\u005c\u005cThis is our interpretation of your"}]
```

```
tools_config.yaml
1  # Welcome
2  - name: sec_filings_analyst
3    uri: https://api.sec-analyst.com/v1
4    headers:
5      Authorization: Bearer sk-1234567890
6    schema:
7      type: object
8      properties:
9        query:
10         type: string
11         description: 'SEC Filings and corporate financial documents search service.'
12         required: true
13      required: [query]
14      filterable_attributes:
15        - language (TEXT) document language
16
17  Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures.
18
19  Database name: 'CORTX_SEARCH_TUTORIAL_DB'
20  Schema name: 'PUBLIC'
21  Limit: 10
22  Service name: 'sec_filings_analyst'
23  description: 'SEC Filings and corporate financial documents search service.'
24
25  FILTERABLE ATTRIBUTES:
26  - language (TEXT) document language
27
28  Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures.
29
30  Database name: 'CORTX_SEARCH_TUTORIAL_DB'
31  Schema name: 'PUBLIC'
32  Limit: 10
33  Columns: ['company', 'relative_path', 'file_url', 'language']
34
35  toolset_services:
36  - name: 'revenue_analyst'
37    uri: https://api.revenue-analyst.com/v1
38    headers:
39      Authorization: Bearer sk-1234567890
40    schema:
41      type: object
42      properties:
43        query:
44         type: string
45         description: 'Revenue analysis and forecasting service.'
46         required: true
47      required: [query]
48
49  Analyze revenue that provides insights into revenue trends,
50  low revenue analysis, and financial performance metrics
51  for business intelligence and forecasting.
52
53  Database name: 'CORTX_SEARCH_TUTORIAL_DB'
54  Schema name: 'PUBLIC'
55  Limit: 10
56  Columns: ['company', 'revenue', 'quarter', 'year']
57
58  toolset_services:
59  - name: 'sec_filings_analyst'
60    uri: https://api.sec-analyst.com/v1
61    headers:
62      Authorization: Bearer sk-1234567890
63    schema:
64      type: object
65      properties:
66        query:
67         type: string
68         description: 'SEC Filings and corporate financial documents search service.'
69         required: true
70      required: [query]
71
72  This tool is used to query SEC filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures submitted by public companies.
73  when querying this service, try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time.
74  For example, if the user asks 'What are the revenue trends across telecommunications companies in the last 5 years?', you could first ask 'What is the revenue for AT&T in the last 5 years?' and then
75  then compare that and summarize the trends.
76
77  Database name: 'CORTX_SEARCH_TUTORIAL_DB'
78  Schema name: 'PUBLIC'
79  Limit: 10
80  Columns: ['company', 'revenue', 'quarter', 'year']
```

DOCUMENT

```
report_date: 2023-12-31
relative_path: /reports/2023/10k/10k.html
file_url: https://www.sec.gov/edgar/data/1234567/000123456723000010/12345672300001010k.htm
language: en
```

MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liability they have. Then reason about why this could be the case



Human in the loop

Why am I showing you this?

```

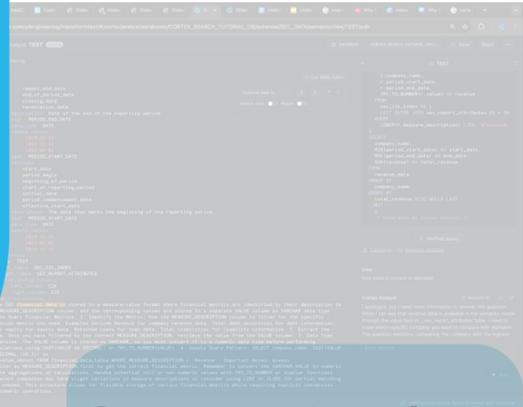
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:
Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned

```

Cortex Search



Cortex Analyst

```

Response
[{"type": "text", "text": "\u0022I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability\u0022"}]

Let me try a more specific query for net income data:

Request
{
  query: 'Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10'
}

Response
[{"type": "text", "text": "\u0022This is our interpretation of your"}]

```

```

18  Twilio (tag: ["brand"], ["name"], ["symbol"], ["type"], ["business"], ["website"], ["family"], ["friendly"]])
19  Robert Mondavi (tag: ["brand"], ["name"], ["symbol"], ["type"], ["business"], ["website"], ["family"], ["friendly"]])
20  Pet-friendly (tag: ["brand"], ["name"], ["symbol"], ["type"], ["business"], ["website"], ["family"], ["friendly"]])
21
22  database_name: "CORTEX_SEARCH_TUTORIAL_DB"
23  schema_name: "PUBLIC"
24  columns: [
25    "id"
26  ]
27  service_name: "sec_filings_search"
28  description: "SEC Filings and corporate financial documents search service."
29
30  FILTERABLE ATTRIBUTES:
31  - language (TEXT): document language
32
33  Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."
34
35  database_name: "CORTEX_SEARCH_TUTORIAL_DB"
36  schema_name: "PUBLIC"
37  columns: ["column", "relative_path", "file_ext", "language"]
38  limit: 10
39  query_services:
40  - service_name: "revenue_analytics"
41    semantic_model: "CORTEX_ANALYST_REPO_REVENUE_ANALYTICS_AWS_DATA/Revenue_Analytics.yaml"
42    description: "Analyze revenue that provides insights into revenue trends, low revenue analysis, and financial performance metrics for business intelligence and forecasting."
43  - service_name: "sec_filings_search"
44    semantic_model: "CORTEX_SEARCH_TUTORIAL_DB_SEC_DATA/SEC_FILINGS_SEARCH/semantic_model.yaml"
45  }
46
47  PROVISIONS:
48  - This tool is used to call services that allow analysts to query SEC filings. The SEC filings include derivative financial reports, disclosures, and other regulatory documents submitted by public companies when carrying this service. Try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time. For example, if the user asks "What are the revenue trends across telecommunications companies in the last 5 years?", you could first ask "What is the revenue for AT&T in the last 5 years?" and then compare that and summarize the trends.
49
50

```


MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liability they have. Then reason about why this could be the case



Human in the loop

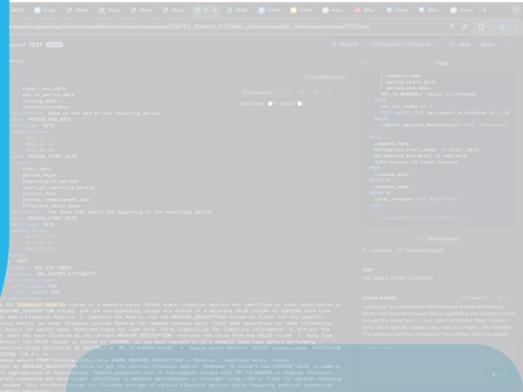
Many Areas where context and instructions are passed in

```
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:
Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Cortex Search



Cortex Analyst

```
Response
[{"type": "text", "text": "\u0022I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability\u0022"}]

Let me try a more specific query for net income data:

Request
{
  query: 'Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10'
}

Response
[{"type": "text", "text": "\u0022This is our interpretation of your"}]
```

```
Twelve (12) [and] [or] [not] [gte] [lte] [contains] [relative_path] [language]
Database name: 'CORTX_SEARCH_TUTORIAL_DB'
Schema name: 'PUBLIC'
Columns: [1]
Limit: 10
Service name: 'sec_filings_search'
Description: 'SEC Filings and corporate financial documents search service.'
Filterable attributes:
- language (TEXT): document language
Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures.'
Database name: 'CORTX_SEARCH_TUTORIAL_DB'
Schema name: 'PUBLIC'
Columns: [column], [relative_path], [language]
Limit: 10
Service name: 'revenue_analytics'
Description: 'Revenue analytics service that provides insights into revenue trends, low revenue analysis, and financial performance metrics for business intelligence and forecasting.'
Service name: 'sec_filings_analyzer'
Description: 'SEC Filings and corporate financial documents search service.'
Filterable attributes:
- This is a text to SQL service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companies when carrying this service. Try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time. For example, if the user asks 'What are the revenue trends across telecommunications companies in the last 5 years?', you could first ask 'What is the revenue for AT&T in the last 5 years?' and then compare that and summarize the trends.
```

MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liabilities they have. Then reason about why this could be the case



Human in the loop

Many different points of failure

```

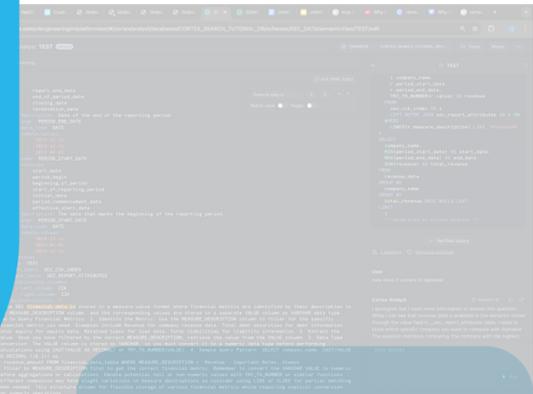
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:
Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned

```

Cortex Search



Cortex Analyst

```

Response
[{"type": "text", "text": "\"I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability\""}]

Let me try a more specific query for net income data:

M sec_filings_analyst

Request
{
  query: 'Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10'
}

Response
[{"type": "text", "text": "\"This is our interpretation of your

```

```

18  Twilio (tag: ["brand"], ["name"], ["symbol"], ["type"], ["business"], ["website"], ["family"], ["friendly"]])
19  Robert Mondavi (tag: ["brand"], ["name"], ["symbol"], ["type"], ["business"], ["website"], ["family"], ["friendly"]])
20  Pet-Friendly (tag: ["brand"], ["name"], ["symbol"], ["type"], ["business"], ["website"], ["family"], ["friendly"]])
21
22  database_name: "CORTX_SEARCH_TUTORIAL_DB"
23  schema_name: "PUBLIC"
24  columns: [
25    "id"
26    "name"
27    "sec_filings_search"
28  ]
29  description: "SEC Filings and corporate financial documents search service."
30
31  FILTERABLE ATTRIBUTES:
32  - language (TEXT): document language
33
34  Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."
35
36  database_name: "CORTX_SEARCH_TUTORIAL_DB"
37  schema_name: "PUBLIC"
38  columns: ["column", "relative_path", "file_size", "language"]
39  limit: 10
40  query_services:
41  - service_name: "revenue_analytics"
42    semantic_model: "CORTX_ANALYST_DATA_REVENUE_ANALYTICS_METADATA"
43    description: "Analyze revenue that provides insights into revenue trends, low revenue analysis, and financial performance metrics for business intelligence and forecasting."
44  - service_name: "sec_filings_analyst"
45    semantic_model: "CORTX_SEARCH_TUTORIAL_DB_SEC_DATA_SEMANTIC_MODEL_METADATA"
46    description: "This is a tool to SEC service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companies when carrying this service. Try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time. For example, if the user asks 'What are the revenue trends across telecommunications companies in the last 5 years?', you could first ask 'What is the revenue for AT&T in the last 5 years?' and then compare that and summarize the trends."
47

```

MCP host

MCP server

I want you to query the cortex analyst service and tell me who the top 10 most profitable companies are. Based on that, I want you to tell me how much debt and liability they have. Then reason about why this could be the case



Human in the loop

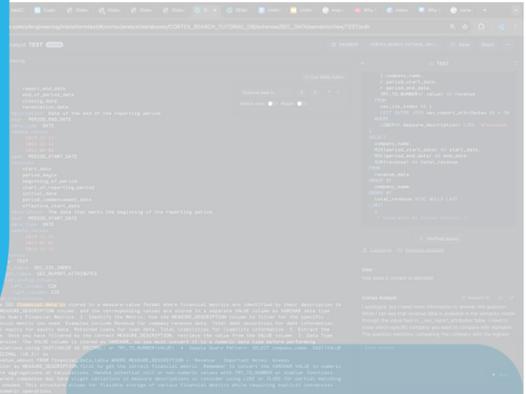
Tool descriptions, calling and instructions are super important!

```
Cortex Search supports four matching operators:
1. TEXT or NUMERIC equality: @eq
2. ARRAY contains: @contains
3. NUMERIC or DATE/TIMESTAMP greater than or equal to: @gte
4. NUMERIC or DATE/TIMESTAMP less than or equal to: @lte

These matching operators can be composed with various logical operators:
- @and
- @or
- @not

The following usage notes apply:
Matching against NaN ('not a number') values in the source query are handled as described in Special values. Fixed-point numeric values with more than 19 digits (not including leading zeroes) do not work with @eq, @gte, or @lte and will not be returned
```

Cortex Search



Cortex Analyst

```
Response
[{"type": "text", "text": "\u0022I apologize, but I need more clarity on how to measure profitability. While we have financial data in the semantic model, we need to specify which measure (tag/variable) should be used to determine net income or profit. The model contains various financial metrics but we need to know the specific measure to use for profitability\u0022"}]

Let me try a more specific query for net income data:

M sec_filings_analyst

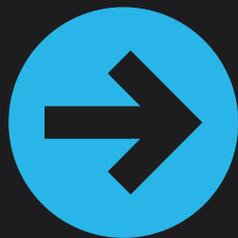
Request
{
  query: 'Show me the companies with the highest net income values, including the net income amount and company name, sorted by net income descending, limit to top 10'
}

Response
[{"type": "text", "text": "\u0022This is our interpretation of your"}]
```

```
18  Twilio (tag: ["brand"], ["name"], ["type"], ["address"], ["website"], ["family"], ["friendly"]])
19  Robert Langdon (tag: ["name"], ["type"], ["address"], ["website"], ["friendly"]])
20  Not-Friendly (tag: ["name"], ["type"], ["address"], ["website"], ["friendly"], ["name or backup"]])
21
22  database_name: "CORTIX_SEARCH_TUTORIAL_DB"
23  schema_name: "PUBLIC"
24  columns: []
25  limit: 10
26  service_name: "sec_filings_analyst"
27  description: "SEC Filings and corporate financial documents search service."
28
29  FILTERABLE ATTRIBUTES:
30  - language (TEXT): document language
31
32  Search through SEC 10-K filings, annual reports, and corporate financial documents including financial statements, risk factors, convertible notes, stock compensation, and regulatory disclosures."
33
34  database_name: "CORTIX_SEARCH_TUTORIAL_DB"
35  schema_name: "PUBLIC"
36  columns: ["column", "relative_path", "file_size", "language"]
37  limit: 10
38  request_services:
39  - service_name: "revenue_analyst"
40  - service_name: "cortex_analyst"
41  description: "Analyze revenue that provides insights into revenue trends, law review analysis, and financial performance metrics for business intelligence and forecasting."
42
43  - service_name: "sec_filings_analyst"
44  - service_name: "cortex_search_tutorial_db_sec_data_semantic_model_sec_data_analyst"
45  description: "This is a tool to SEC service that allows analysts to query SEC filings. The SEC filings include detailed financial reports, disclosures, and other regulatory documents submitted by public companies when utilizing this service. Try to ask simple questions. If the user asks a complicated question, first think about how you could decompose it, and then ask only one part at a time. For example, if the user asks 'What are the revenue trends across telecommunications companies in the last 5 years?', you could first ask 'What is the revenue for AT&T in the last 5 years?' and then compare that and summarize the trends."
46
```


WHAT DID I DO THIS SUMMER?

MCP
Evals!



Dataset
Generation