

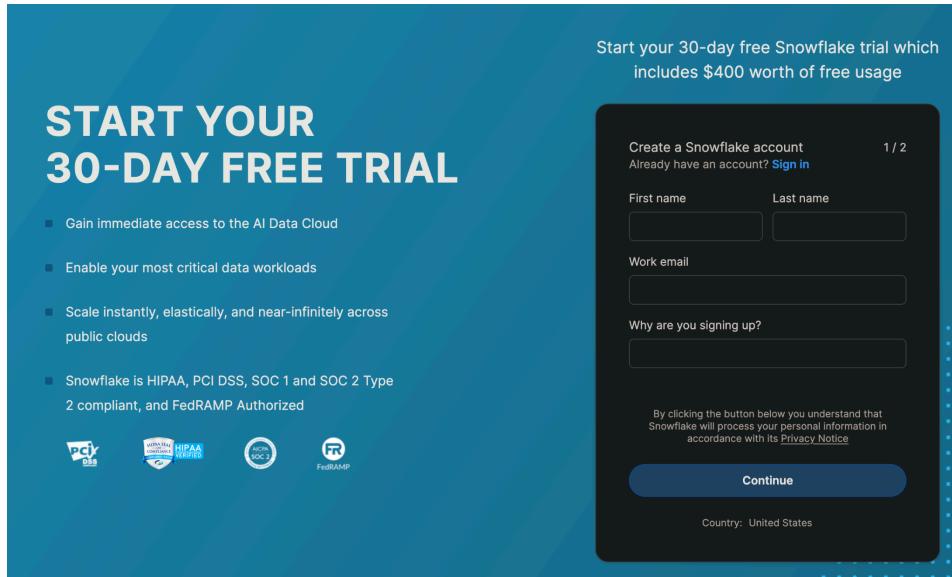
# Table of Contents

<b>Module 1: Setting Up.....</b>	<b>2</b>
Signing-up for Snowflake Trail Account.....	2
Loading in Data.....	5
Showcase the Datasets.....	5
Optional: Ingest Data via Local File.....	6
Showcase setup.sql.....	9
Optional: Ingest Data via S3 Buckets.....	11
Setting Up and Executing a SQL File in Snowflake Workspace.....	14
<b>Module 2: Cortex Search.....</b>	<b>17</b>
Setting Up Cortex Search.....	17
Using Cortex Search.....	21
<b>Module 3: Cortex Analyst.....</b>	<b>24</b>
Setting Up Cortex Analyst.....	24
Optional: Showcase and Explain Semantic Views & Models.....	26
Using Cortex Analyst.....	27
<b>Module 4: User Defined Functions.....</b>	<b>28</b>
Setting Up UDF/UDTF via SQL File.....	28
Executing and Using UDF.....	29
Verifying the Results of Running the UDF/UDTF.....	32
<b>Module 5: Snowflake Intelligence.....</b>	<b>33</b>
Setting Up Custom Snowflake Intelligence.....	33
Creating an Agent.....	33
Setting up the Tools.....	38
Using Custom Snowflake Intelligence Agent.....	46
<b>Module 6: Snowflake Badge.....</b>	<b>48</b>
Setup Autograder.....	48
Run Autograding Scripts.....	48

# Module 1: Setting Up

## Signing-up for Snowflake Trail Account

1. Visit [mlh.link/snowflake-ai-trial](https://mlh.link/snowflake-ai-trial)



Start your 30-day free Snowflake trial which includes \$400 worth of free usage

**CREATE A SNOWFLAKE ACCOUNT**

1 / 2

Already have an account? [Sign in](#)

First name  Last name

Work email

Why are you signing up?

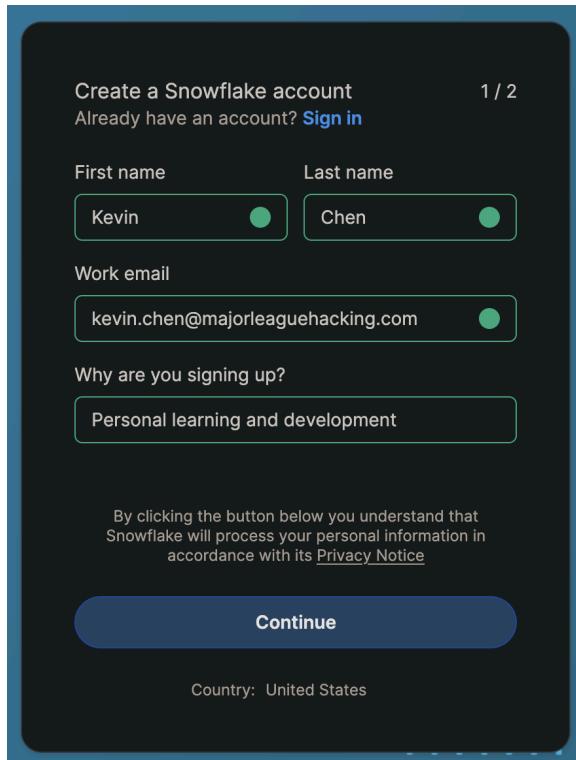
Personal learning and development

By clicking the button below you understand that Snowflake will process your personal information in accordance with its [Privacy Notice](#)

**CONTINUE**

Country: United States

2. Fill out all the information and select "Personal learning and development"



Create a Snowflake account

1 / 2

Already have an account? [Sign in](#)

First name  Last name

Work email

Why are you signing up?

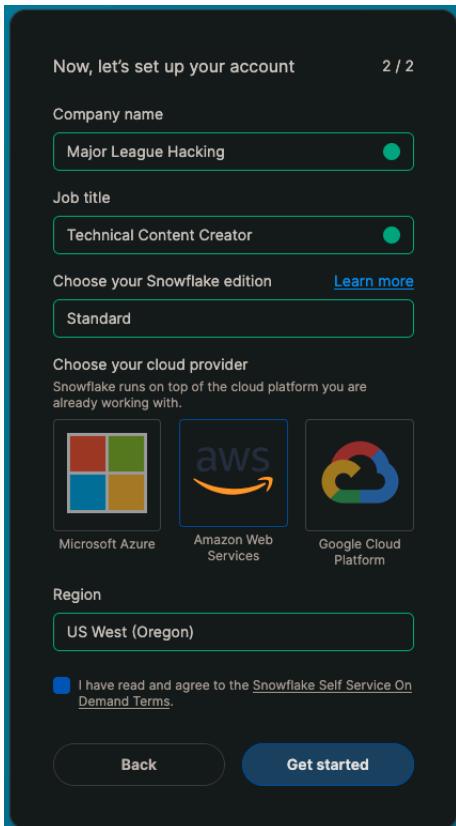
Personal learning and development

By clicking the button below you understand that Snowflake will process your personal information in accordance with its [Privacy Notice](#)

**CONTINUE**

Country: United States

- On the next page, enter your personal details (company name and role can be fictional). Choose the Standard edition, as it includes all AI/ML features at \$2/credit, so your trial credits will stretch further, but any edition will work for this workshop. You should see your cloud provider and region pre-selected for you, with the value of AWS and US West (Oregon). If for any reason you do not see that, select AWS and US West (Oregon) manually.



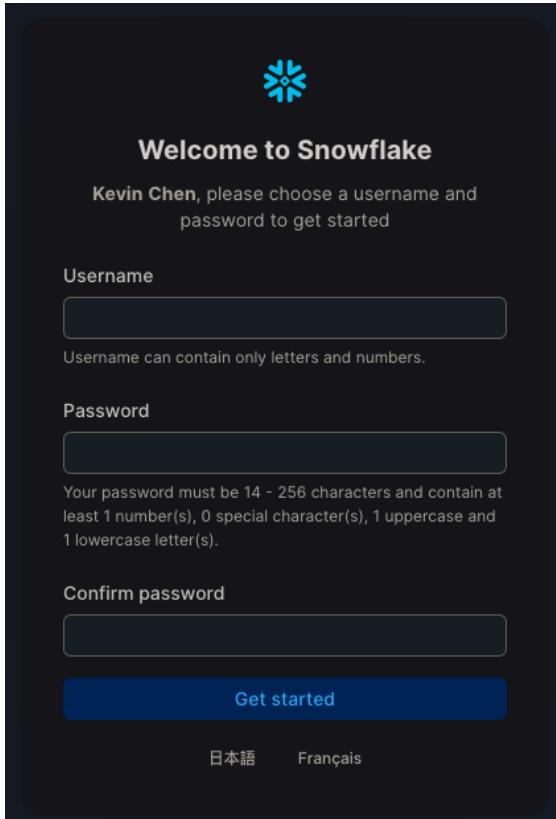
- Answer the next two optional pages if desired
- Check for an email and activate your account!

The left screenshot shows a message: "Check your inbox! An email to activate your account has been sent to kevin.chen@majorleaguehacking.com. It may take a few minutes to arrive. Meanwhile, here are a few resources to check out:

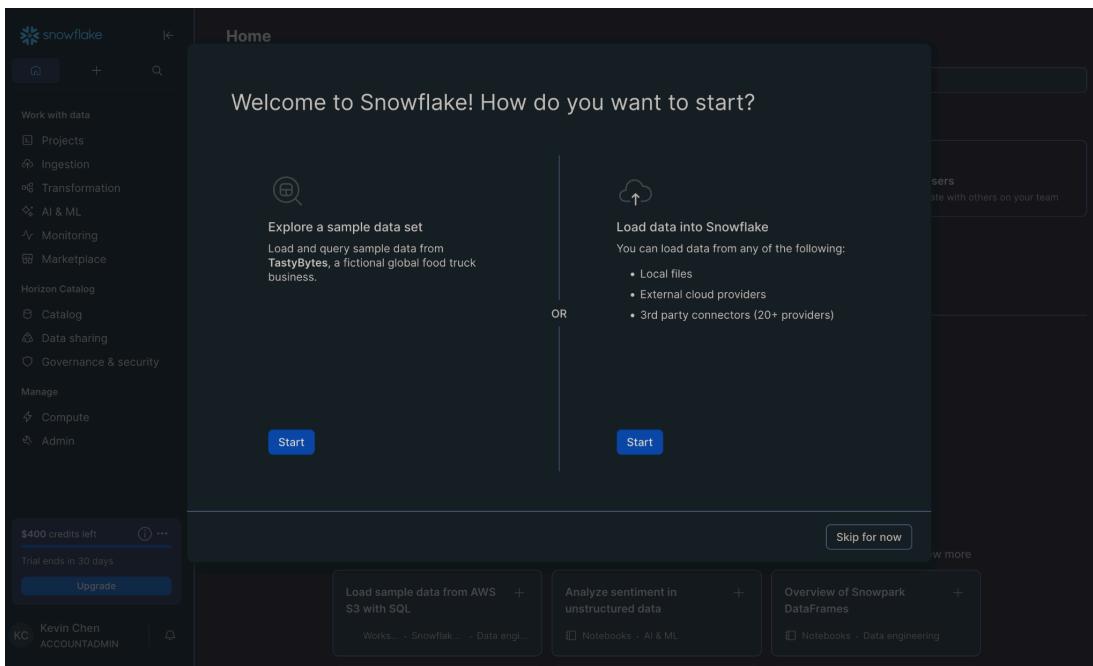
- Get started with [Snowflake documentation](#)
- Sign up for a free, instructor-led [Virtual Hands-On Lab](#)
- Explore industry-specific user cases and walkthroughs in our [Solutions Center](#)

The right screenshot shows an email from "Snowflake Computing <no-reply@snowflake.net>" to "kevin.chen". The subject is "Activate your Snowflake account". The body of the email says: "Congratulations on getting started with Snowflake! Click the button below to activate your account." Below the button is the text: "This activation link is temporary and will expire in 72 hours." and "Save this for later Once you activate your account, you can access it at <https://apchrjp-vac81038.snowflakecomputing.com/console/login>".

6. Set up your user account for your new Snowflake account.



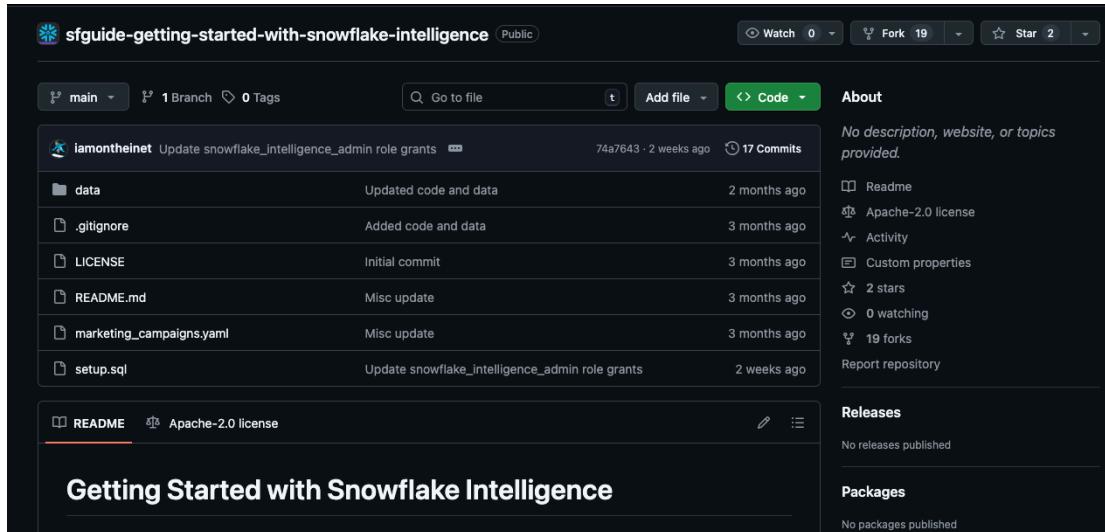
7. Perfect, leave that page up, and you are all setup to continue back to the slides!



## Loading in Data

### Showcase the Datasets

1. Visit [mlh.link/snowflake-data](https://mlh.link/snowflake-data)



**About**  
*No description, website, or topics provided.*

- Readme
- Apache-2.0 license
- Activity
- Custom properties
- 2 stars
- 0 watching
- 19 forks

Report repository

**Releases**  
 No releases published

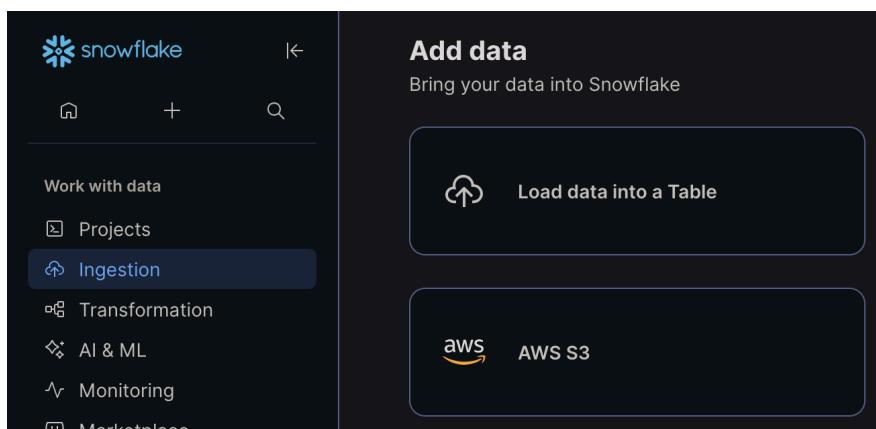
**Packages**  
 No packages published

**Getting Started with Snowflake Intelligence**

2. Look through the data folder and explore some files. Explain that you could directly ingest these files via the Snowflake UI.

Name	Last commit message	Last commit date
...		
marketing_campaign_metrics.csv	Added code and data	3 months ago
products.csv	Added code and data	3 months ago
sales.csv	Added code and data	3 months ago
social_media_mentions.csv	Added code and data	3 months ago
support_cases.csv	Updated code and data	2 months ago

3. Show them “Ingestion > Load data into Table” being the location where they would do so



**Add data**  
 Bring your data into Snowflake

**Load data into a Table**

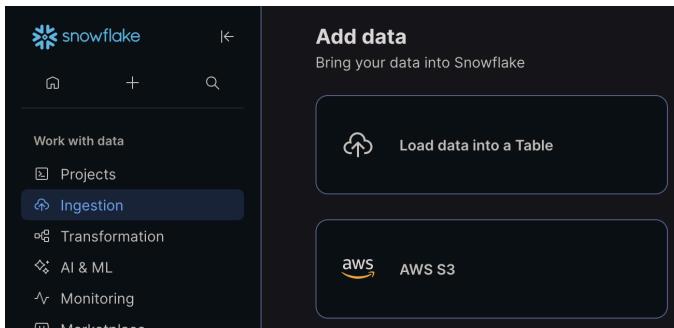
**AWS S3**

### Optional: Ingest Data via Local File

1. Download any one of the .csv files

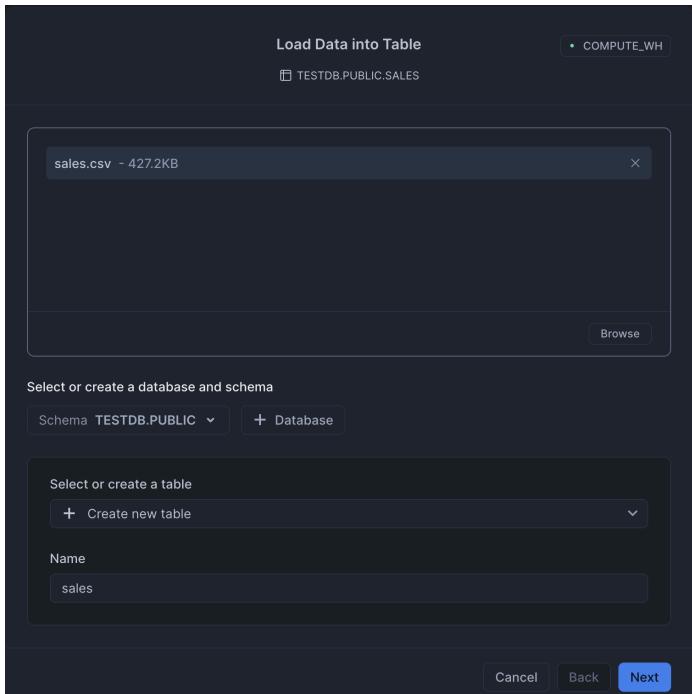
Name	Last commit message	Last commit date
...		
marketing_campaign_metrics.csv	Added code and data	3 months ago
products.csv	Added code and data	3 months ago
sales.csv	Added code and data	3 months ago
social_media_mentions.csv	Added code and data	3 months ago
support_cases.csv	Updated code and data	2 months ago

2. Go to "Ingestion > Load data into Table"



3. Click Browse and upload the file you downloaded. Click + Database, name it TestDB, and confirm. Finally, enter your desired name for the new table before loading.

This screenshot shows the 'Load Data into Table' wizard. Step 1: Select or create a database and schema. It displays a 'Browse' button for uploading files, a note about supported formats (CSV/TSV, json, orc, avro, parquet), and a note about the file size limit (250MB). Step 2: Select or create a table. It shows a dropdown menu with the message '+ Please select database first'. At the bottom, there are 'Cancel', 'Back', and 'Next' buttons.



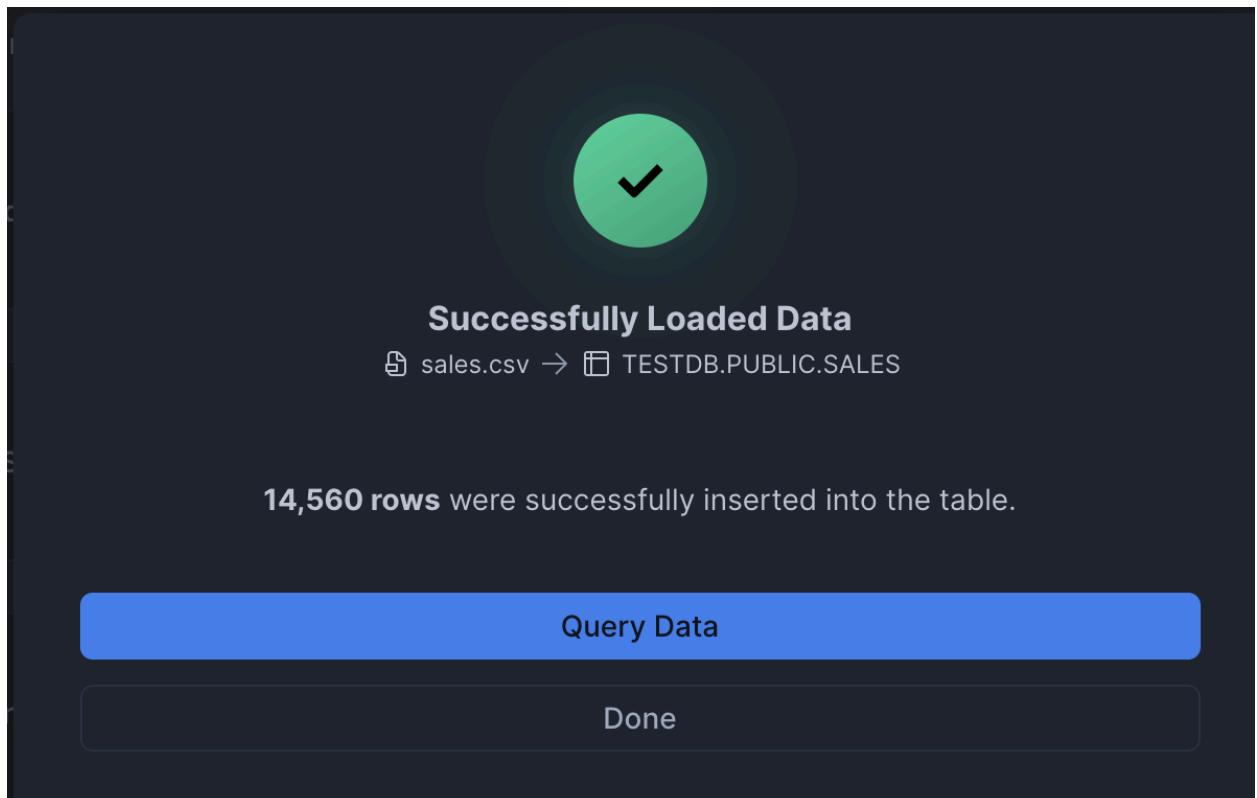
- Click "Next" to open the schema configuration screen for your new table. Verify the column definitions and data types are correct, then click "Load".

The screenshot shows the 'Edit Schema' section of the 'Load Data into Table' interface. It lists 5 Columns:

DATA TYPE	COLUMN NAME	COLUMN DATA
DATE	date	2025-05-16, 2025-05-16, 2025-05-16,...
VARCHAR	region	North, North, North, North, North
NUMBER	product_id	1, 2, 3, 4, 5
NUMBER	units_sold	28, 32, 32, 25, 27
NUMBER	sales_amount	2199.67, 1039.35, 692.7, 866.81, 1473.0

On the left, there's a 'File format' section for 'Delimited Files (CSV or TSV)' and an 'Edit Schema' section for '5 Columns'. Below that is a 'What should happen if an error is encountered while loading a file?' dropdown set to 'Do not load any data (default)'. At the bottom are 'Show SQL', 'Cancel', 'Back', and 'Load' buttons.

- After loading completes, and you see "Successfully Loaded Data," click "Done". Then go to "Catalog > TestDB/Public/Tables/<TABLENAME>" to view the imported records.

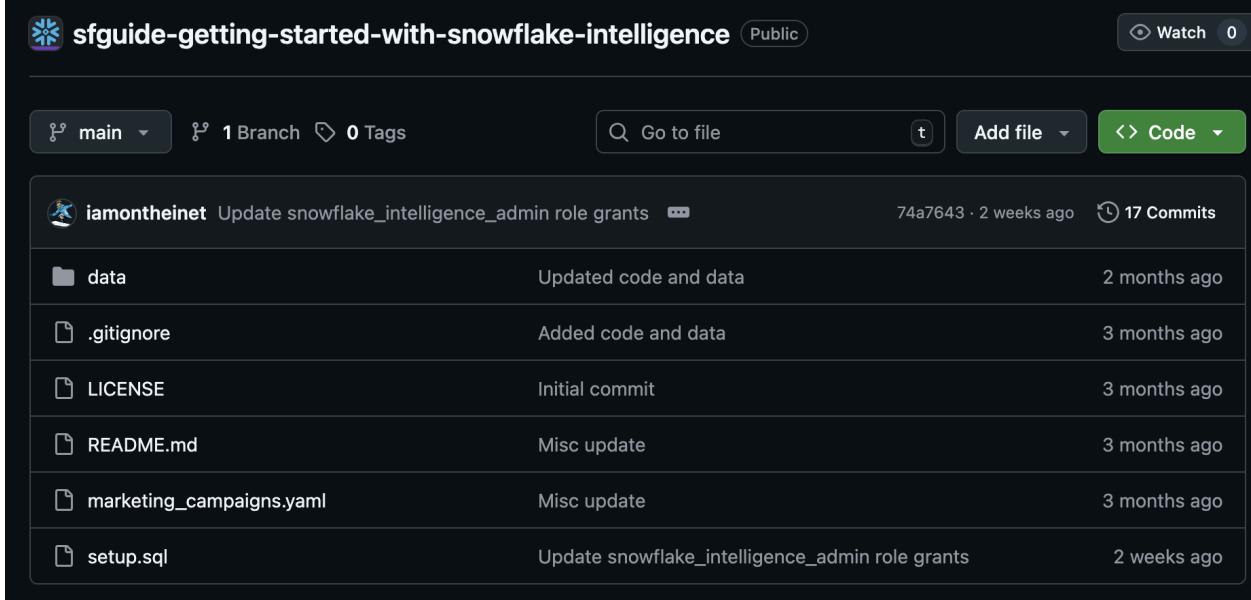


The screenshot shows the Snowflake Database Explorer interface. On the left sidebar, under the "Catalog" section, the "SALES" table is selected. The main panel displays the "TESTDB / PUBLIC / SALES" table. At the top of the table view, there are buttons for "Describe Table", "Load Data", and other options. Below the table header, a message states "100 Rows • Updated 1 minute ago". The table has columns: DATE, REGION, PRODUCT\_ID, UNITS SOLD, and SALES\_AMOUNT. The data preview shows 100 rows of sales data, with the last row being dated 2025-05-16 and having a value of 29231.00.

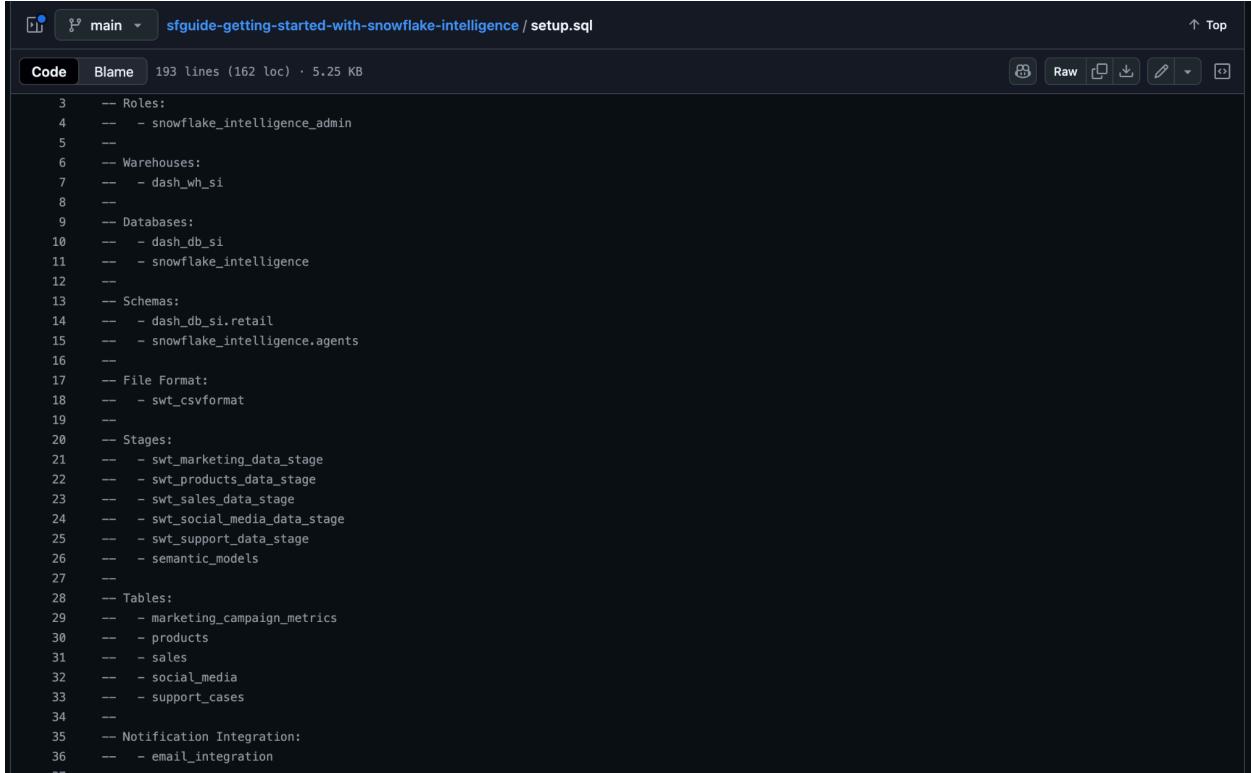
	DATE	REGION	PRODUCT_ID	UNITS SOLD	SALES_AMOUNT
1	2025-05-16	North	1	28	2199.67
2	2025-05-16	North	2	32	1039.35
3	2025-05-16	North	3	32	692.70
4	2025-05-16	North	4	25	866.81
5	2025-05-16	North	5	27	1473.00
6	2025-05-16	North	6	26	810.15
7	2025-05-16	North	7	26	1468.63
8	2025-05-16	North	8	35	2139.86
9	2025-05-16	North	9	31	2126.71
10	2025-05-16	North	10	24	2301.86
11	2025-05-16	North	11	46	3437.98
12	2025-05-16	North	12	49	29231.00

## Showcase setup.sql

1. Show the attendees the setup.sql script



The screenshot shows a GitHub repository named "sfguide-getting-started-with-snowflake-intelligence". The repository is public and has 1 branch and 0 tags. The main branch contains several files: "data", ".gitignore", "LICENSE", "README.md", "marketing\_campaigns.yaml", and "setup.sql". The "setup.sql" file was updated 2 weeks ago to "Update snowflake\_intelligence\_admin role grants". Other commits include "Initial commit" (3 months ago), "Added code and data" (3 months ago), and "Updated code and data" (2 months ago).

The screenshot shows the content of the "setup.sql" file. The file contains 193 lines of SQL code, totaling 162 loc and 5.25 KB. The code is organized into sections: Roles, Warehouses, Databases, Schemas, File Format, Stages, Tables, and Notification Integration. Key roles defined include "snowflake\_intelligence\_admin" and "snowflake\_intelligence". Warehouses include "dash\_wh\_si". Databases include "dash\_db\_si" and "snowflake\_intelligence". Schemas include "dash\_db\_si.retail" and "snowflake\_intelligence.agents". File formats include "svt\_csvformat". Stages include "svt\_marketing\_data\_stage", "svt\_products\_data\_stage", "svt\_sales\_data\_stage", "svt\_social\_media\_data\_stage", "svt\_support\_data\_stage", and "semantic\_models". Tables include "marketing\_campaign\_metrics", "products", "sales", "social\_media", and "support\_cases". Notification integration includes "email\_integration".

```

3 -- Roles:
4 --   - snowflake_intelligence_admin
5 --
6 -- Warehouses:
7 --   - dash_wh_si
8 --
9 -- Databases:
10 --   - dash_db_si
11 --   - snowflake_intelligence
12 --
13 -- Schemas:
14 --   - dash_db_si.retail
15 --   - snowflake_intelligence.agents
16 --
17 -- File Format:
18 --   - svt_csvformat
19 --
20 -- Stages:
21 --   - svt_marketing_data_stage
22 --   - svt_products_data_stage
23 --   - svt_sales_data_stage
24 --   - svt_social_media_data_stage
25 --   - svt_support_data_stage
26 --   - semantic_models
27 --
28 -- Tables:
29 --   - marketing_campaign_metrics
30 --   - products
31 --   - sales
32 --   - social_media
33 --   - support_cases
34 --
35 -- Notification Integration:
36 --   - email_integration
37 --

```

2. Scroll through and explain what the script is doing. Note: We won't be using lines 151 - 189 in this Workshop.
  - a. 42: Switch to the ACCOUNTADMIN role.
  - b. 44-47: Create the snowflake\_intelligence\_admin role and grant it account-level privileges (warehouse, database, integration creation).
  - c. 49-52: Assign snowflake\_intelligence\_admin to the current user and set it as the default role and warehouse.
  - d. 54: Activate the snowflake\_intelligence\_admin role.
  - e. 55-62: Provision core environments
    - i. dash\_db\_si database with retail schema and dash\_wh\_si warehouse
    - ii. snowflake\_intelligence database with agents schema and grant agent creation
  - f. 64-66: Set session context to dash\_db\_si.retail and dash\_wh\_si.
  - g. 68-71: Define the swt\_csvformat file format for CSV ingestion.
  - h. 74-149: For each dataset (marketing, products, sales, social\_media, support):
    - i. Create an external stage pointing to the corresponding S3 path
    - ii. Create the target table with appropriate columns
    - iii. Load data into the table via COPY INTO
  - i. 151: Create the semantic\_models stage with server-side encryption.
  - j. 153-189: Configure email\_integration and define the send\_email Python stored procedure to wrap SYSTEM\$SEND\_EMAIL.
  - k. 191: Enable cross-region Cortex features via the CORTEX\_ENABLED\_CROSS\_REGION account parameter.
  - l. 193: Display a setup completion confirmation message.

## Optional: Ingest Data via S3 Buckets

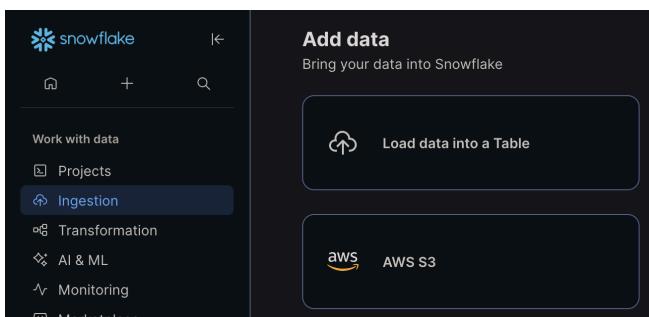
1. Grab one of the S3:// endpoints from the `setup.sql` script

```

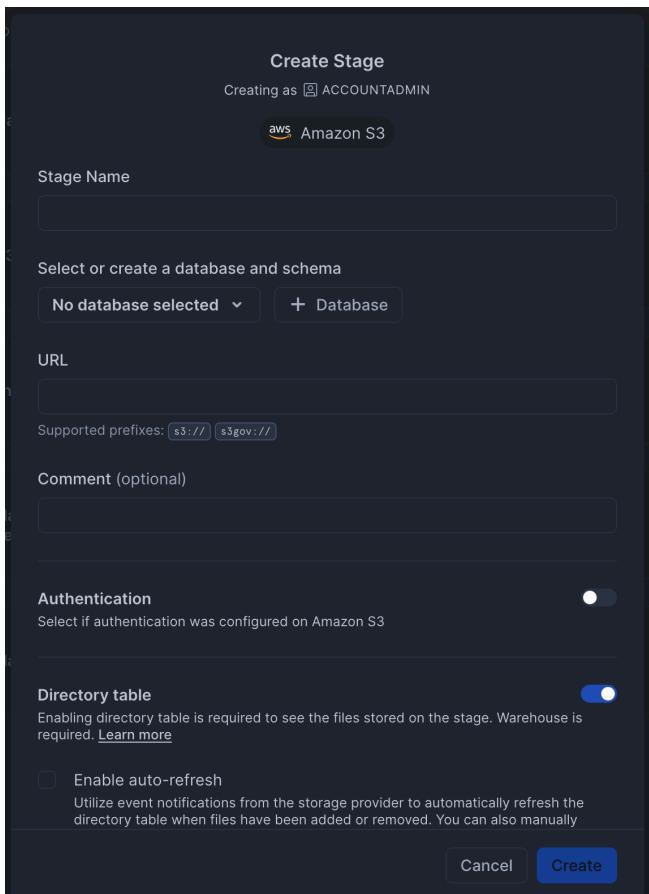
89      -- create table products and load data from s3 bucket
90      create or replace stage swt_products_data_stage
91          file_format = swt_csvformat
92          url = 's3://sfquickstarts/sfguide_getting_started_with_snowflake_intelligence/product/';

```

2. Go to "Ingestion > AWS S3"



3. Enter a name for your stage and paste in the S3 URL from the `setup.sql` script, then click "Create".

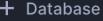


The screenshot shows the 'Create Stage' dialog. At the top, it says 'Creating as ACCOUNTADMIN'. Below that is a section for 'Amazon S3' with a 'Stage Name' input field containing 'sfprod'. Under 'Select or create a database and schema', there's a dropdown menu showing 'No database selected' and a '+ Database' button. The 'URL' input field contains the S3 URL from the setup script: 's3://sfquickstarts/sfguide\_getting\_started\_with\_snowflake\_intelligence/product/'. Below the URL, there's a note about supported prefixes: 'Supported prefixes: s3:// s3gov://'. The 'Comment (optional)' field is empty. In the 'Authentication' section, there's a note: 'Select if authentication was configured on Amazon S3' with a toggle switch set to off. In the 'Directory table' section, there's a note: 'Enabling directory table is required to see the files stored on the stage. Warehouse is required.' with a toggle switch set to on. At the bottom, there are 'Cancel' and 'Create' buttons.

**Create Stage**  
 Creating as  ACCOUNTADMIN

 Amazon S3

**Stage Name**

**Select or create a database and schema**  
 Schema TESTDB.PUBLIC  

**URL**  
  
 Supported prefixes: `s3://` `s3gov://`

**Comment (optional)**

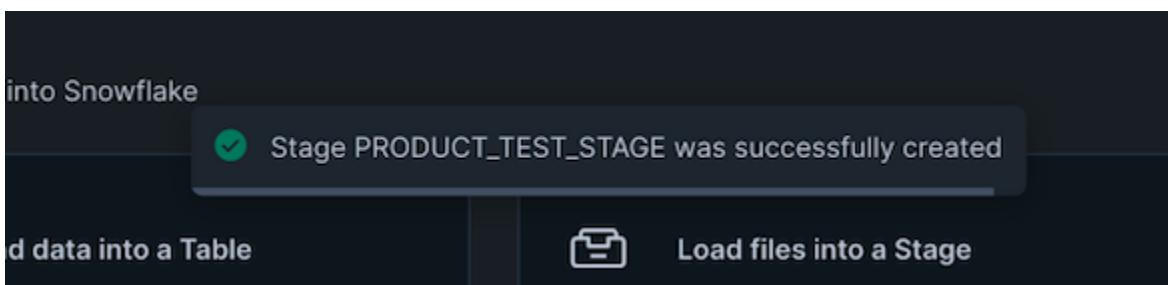
**Authentication**  
 Select if authentication was configured on Amazon S3 

**Directory table**  
 Enabling directory table is required to see the files stored on the stage. Warehouse is required. [Learn more](#) 

**Enable auto-refresh**  
 Utilize event notifications from the storage provider to automatically refresh the directory table when files have been added or removed. You can also manually

**Create**

- Once created, go to “Catalog > TestDB/Public/Stages/<STATGENAME>” to view the files available for loading.

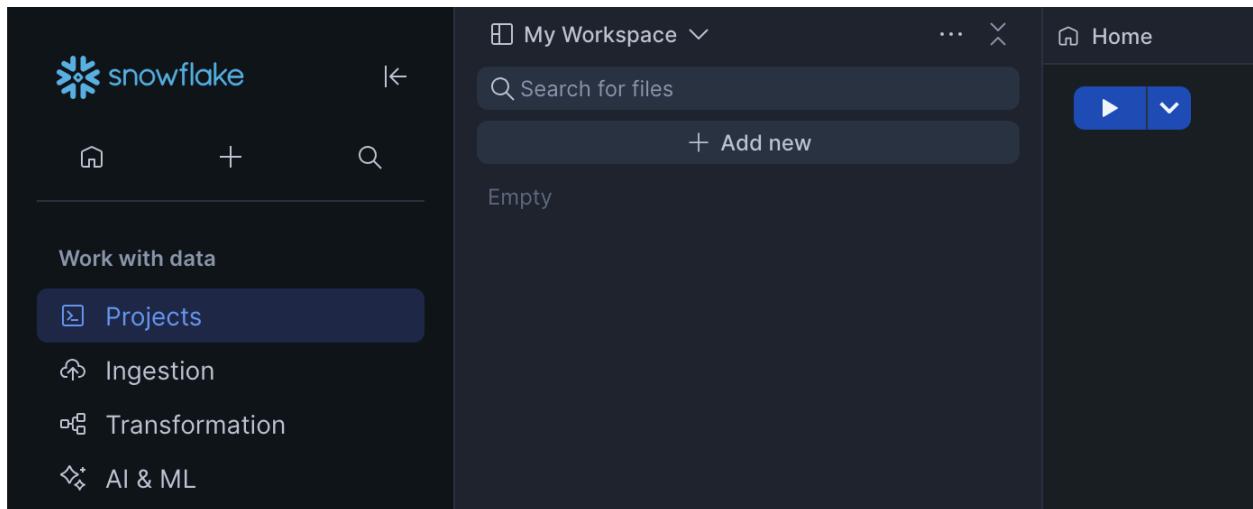


- Click the three dots next to the CSV file and choose "Load into table".

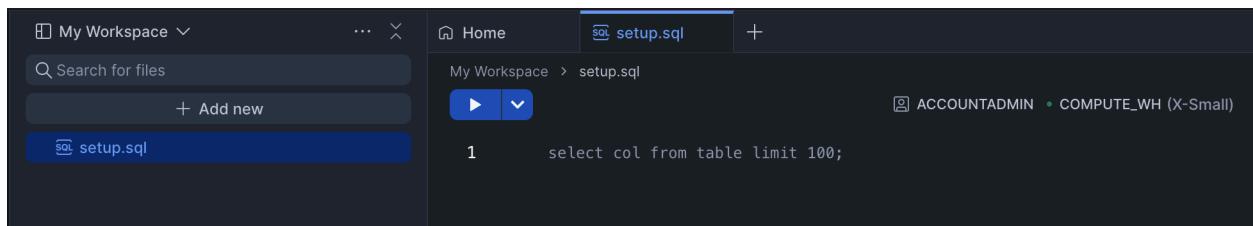
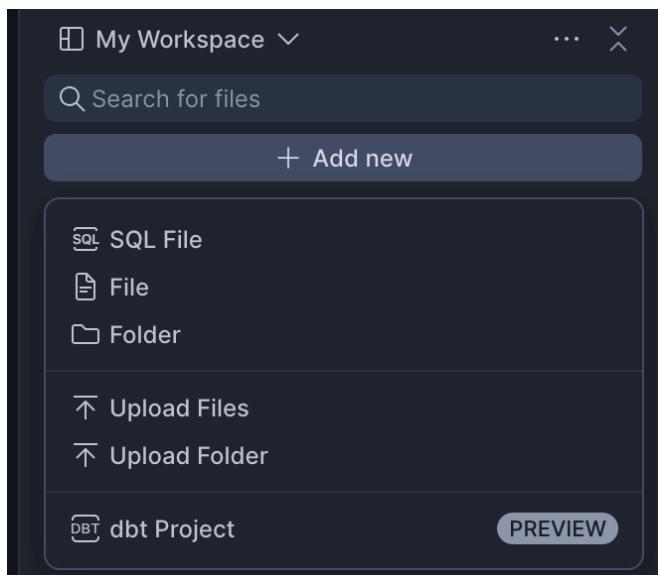
- Follow the same prompts as in Step 3 for Ingest Data via Local File, creating a new table in TESTDB and defining the table schema.

## Setting Up and Executing a SQL File in Snowflake Workspace

1. Go to “Projects” which should take you to “My Workspace”



2. Click on “+ Add new” and select “SQL File” and name it `setup.sql`



- Navigate back to the GitHub repo, open `setup.sql`, copy its contents, and paste them into the SQL file you created in your workspace.

```

1  -- Summary of objects created in this script:
2  --
3  -- Roles:
4  --   - snowflake_intelligence_admin
5  --
6  -- Warehouses:
7  --   - dash_wh_si
8  --
9  -- Databases:
10 --   - dash_db_si
11 --   - snowflake_intelligence
12 --
13 -- Schemas:
14 --   - dash_db_si.retail
15 --   - snowflake_intelligence.agents
16 --
17 -- File Format:
18 --   - swt_csvformat
19 --
20 -- Stages:
21 --   - swt_marketing_data_stage
22 --   - swt_products_data_stage
23 --   - swt_sales_data_stage
24 --   - swt_social_media_data_stage
25 --   - swt_support_data_stage
26 --   - semantic_models

```

- If you haven't yet, briefly explain the purpose of `setup.sql`: it automates role, warehouse, database, schema, stage, and table creation, then loads all datasets from S3. Click "Run All" to execute the entire script. You should see:

"Congratulations! Snowflake Intelligence setup has completed successfully!"

The screenshot shows the Snowflake UI with the setup.sql script open. A context menu is displayed over the script code, with the 'Run all' option highlighted. Below the editor, the results pane displays a single row of data:

STATUS
Congratulations! Snowflake Intelligence setup has completed successfully!

At the bottom of the screen, the Query History pane shows the execution of the script, listing several recent queries with their execution times and SQL statements.

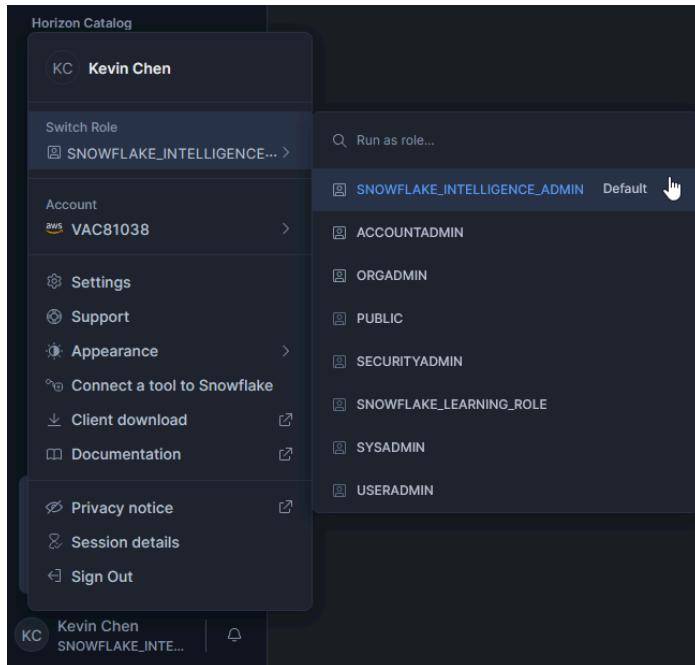
5. Finally, verify the script ran successfully by going to "Catalog" and checking for "DASH\_DB\_SI/RETAIL" DB which should list five tables populated from S3. Also check to make sure "SNOWFLAKE\_INTELLIGENCE" DB should exist and be ready for creating AI agents.

NAME	T	OWNER	RO...	B	C...
MARKETING_CAMPAIGN_METRICS	T...	SNOWFLAKE_INTELLIGENCE...	23	2...	5 ...
PRODUCTS	T...	SNOWFLAKE_INTELLIGENCE...	40	2...	5 ...
SALES	T...	SNOWFLAKE_INTELLIGENCE...	14.6K	8...	5 ...
SOCIAL_MEDIA	T...	SNOWFLAKE_INTELLIGENCE...	273	3...	5 ...
SUPPORT_CASES	T...	SNOWFLAKE_INTELLIGENCE...	30	2...	4 ...

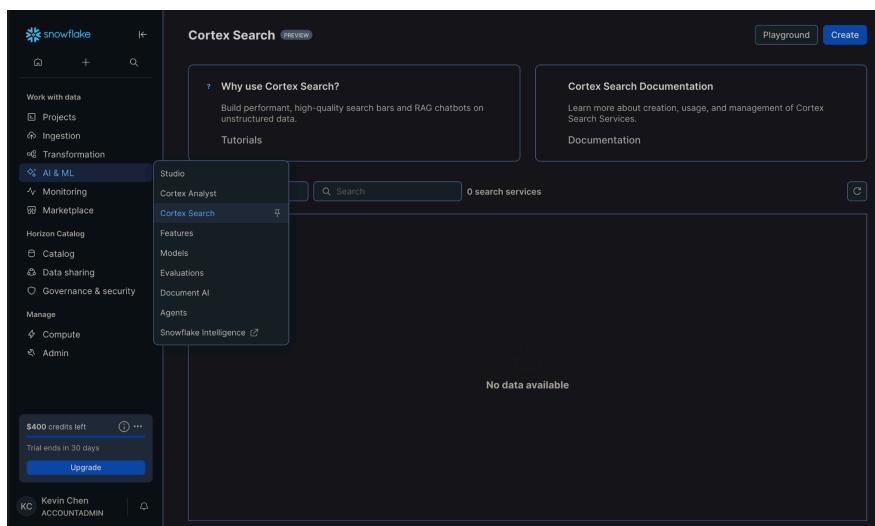
## Module 2: Cortex Search

### Setting Up Cortex Search

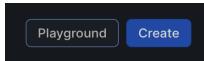
1. In the bottom-left corner, select your account, then switch your role to SNOWFLAKE\_INTELLIGENCE\_ADMIN.



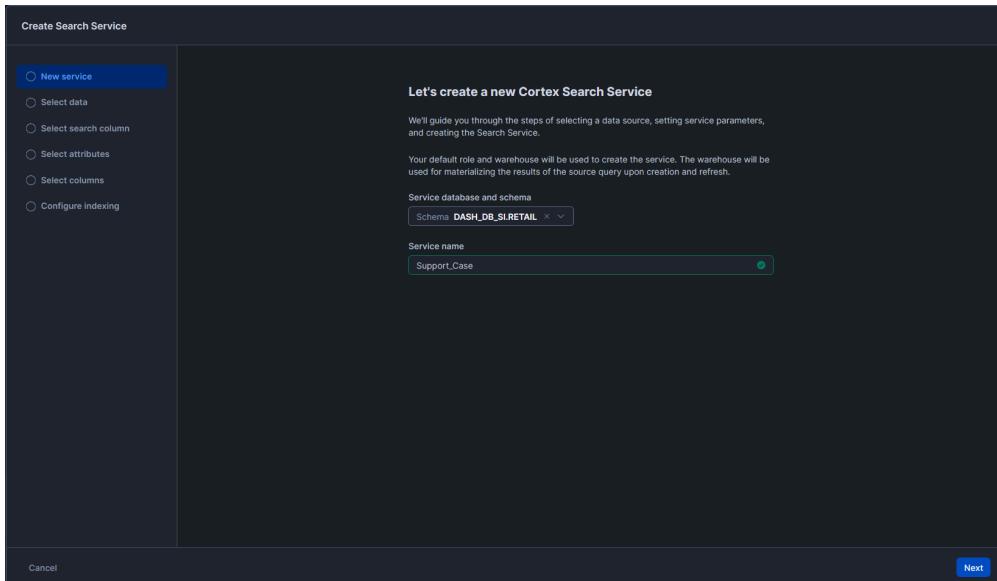
2. Navigate to "AI & ML > Cortex Search"



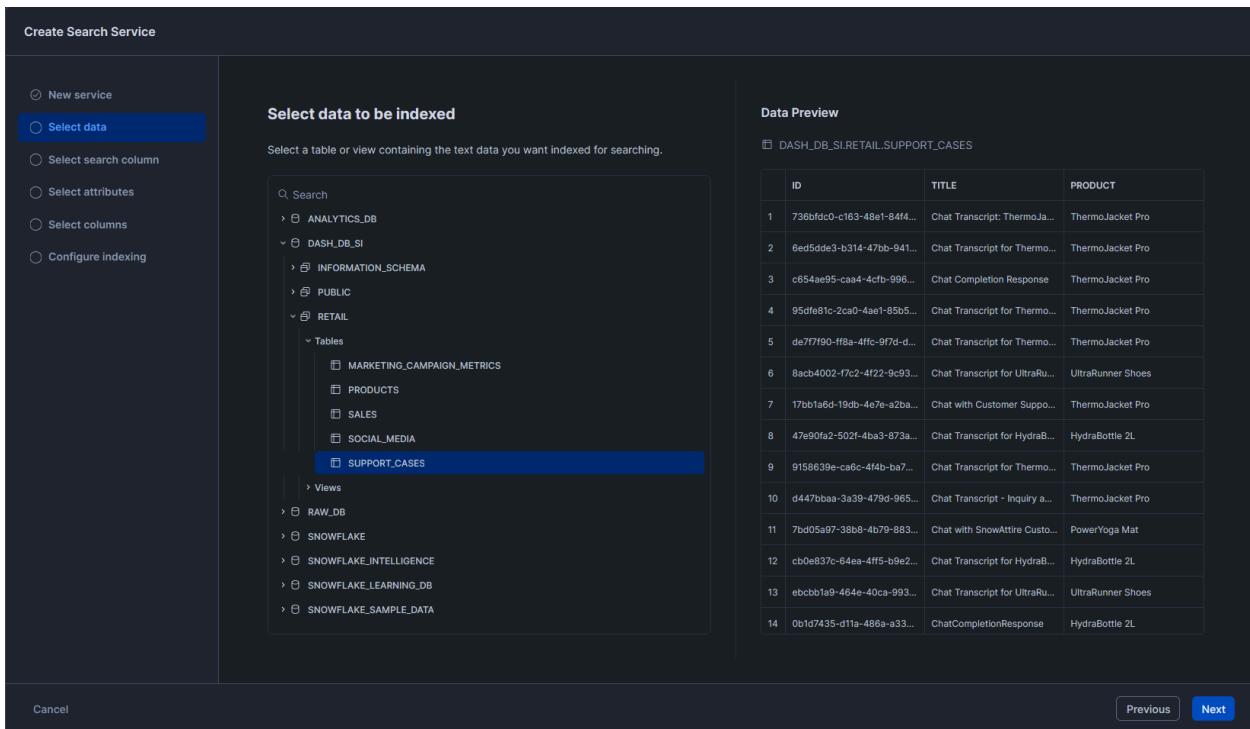
- Click "Create" in the top right corner



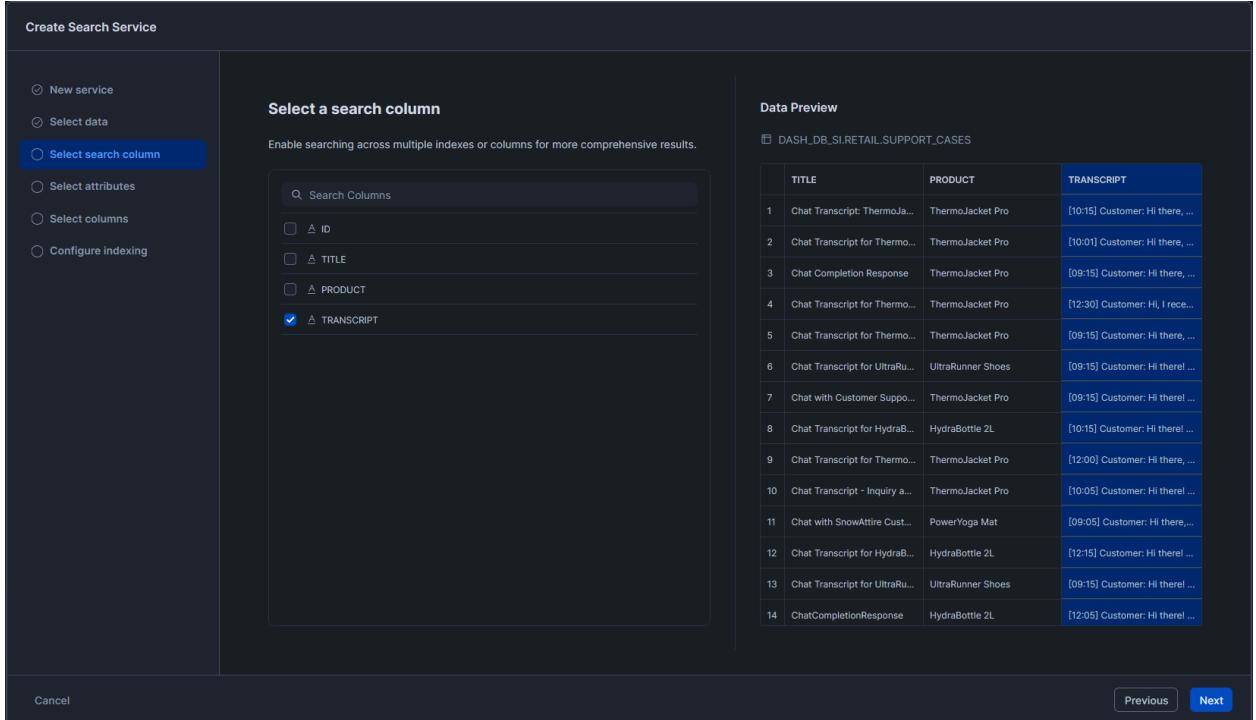
- Select "DASH\_DB\_SI.RETAIL" for Database and Schema, and "Support\_Cases" for Name. Then select "Next: Select data"



- Select the "SUPPORT\_CASES" table as the data to be indexed. Then select "Next: Select search column"



6. Select “TRANSCRIPT” as the search column. Then select “Next: Select attributes”

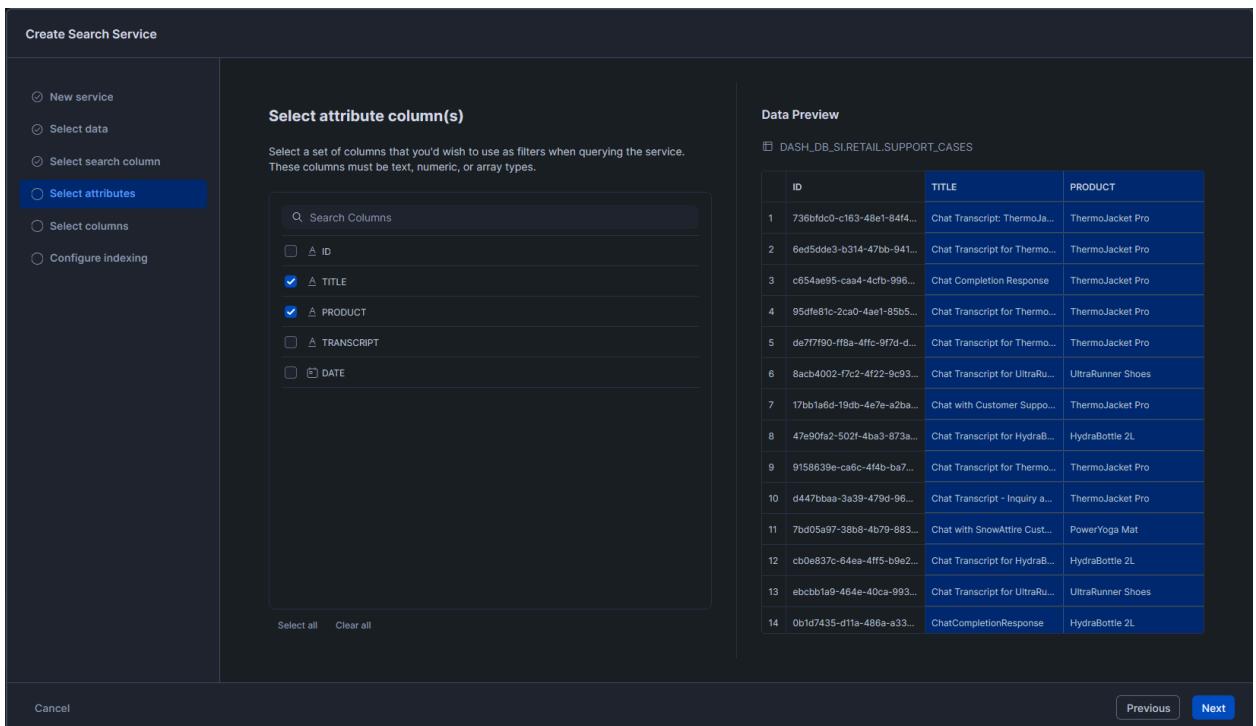


**Select a search column**

Enable searching across multiple indexes or columns for more comprehensive results.

	TITLE	PRODUCT	TRANSCRIPT
1	Chat Transcript: ThermoJa...	ThermoJacket Pro	[10:15] Customer: Hi there, ...
2	Chat Transcript for Thermo...	ThermoJacket Pro	[10:01] Customer: Hi there, ...
3	Chat Completion Response	ThermoJacket Pro	[08:15] Customer: Hi there, ...
4	Chat Transcript for Thermo...	ThermoJacket Pro	[12:30] Customer: Hi, I rece...
5	Chat Transcript for Thermo...	ThermoJacket Pro	[09:15] Customer: Hi there, ...
6	Chat Transcript for UltraRu...	UltraRunner Shoes	[09:15] Customer: Hi therel ...
7	Chat with Customer Suppo...	ThermoJacket Pro	[09:15] Customer: Hi therel ...
8	Chat Transcript for HydraB...	HydraBottle 2L	[10:15] Customer: Hi therel ...
9	Chat Transcript for Thermo...	ThermoJacket Pro	[12:00] Customer: Hi there, ...
10	Chat Transcript - Inquiry a...	ThermoJacket Pro	[10:05] Customer: Hi therel ...
11	Chat with SnowAttire Cust...	PowerYoga Mat	[09:05] Customer: Hi there, ...
12	Chat Transcript for HydraB...	HydraBottle 2L	[12:15] Customer: Hi therel ...
13	Chat Transcript for UltraRu...	UltraRunner Shoes	[09:15] Customer: Hi therel ...
14	ChatCompletionResponse	HydraBottle 2L	[12:05] Customer: Hi therel ...

7. Select “TITLE, PRODUCT” as the attribute column(s). Then select “Next: Select search column”



**Select attribute column(s)**

Select a set of columns that you'd wish to use as filters when querying the service. These columns must be text, numeric, or array types.

	ID	TITLE	PRODUCT
1	736bfdc0-c163-48e1-844...	Chat Transcript: ThermoJa...	ThermoJacket Pro
2	6ed5dde3-b314-47bb-941...	Chat Transcript for Thermo...	ThermoJacket Pro
3	c654ae95-caa4-4cfb-996...	Chat Completion Response	ThermoJacket Pro
4	95dfe81c-2ca0-4ae1-85b5...	Chat Transcript for Thermo...	ThermoJacket Pro
5	de77f90-f8a-4ffc-97d-d...	Chat Transcript for Thermo...	ThermoJacket Pro
6	8acbf4002-f7c2-4f22-9c93...	Chat Transcript for UltraRu...	UltraRunner Shoes
7	17bb1a6d-19db-4e7e-a2ba...	Chat with Customer Suppo...	ThermoJacket Pro
8	47e90fa2-502f-4ba3-873a...	Chat Transcript for HydraB...	HydraBottle 2L
9	9158639e-ca6c-4f4b-ba7...	Chat Transcript for Thermo...	ThermoJacket Pro
10	d447bbaa-3a39-479d-96...	Chat Transcript - Inquiry a...	ThermoJacket Pro
11	7bd05a97-38b8-4b79-883...	Chat with SnowAttire Cust...	PowerYoga Mat
12	cbe0837c-64ea-4ff5-b9e2...	Chat Transcript for HydraB...	HydraBottle 2L
13	ebcbb1a9-464e-40ca-993...	Chat Transcript for UltraRu...	UltraRunner Shoes
14	0b1d7435-d11a-486a-a33...	ChatCompletionResponse	HydraBottle 2L

8. Select "Select all" in the bottom left as column to include in the service. Then select "Next: Configure indexing"

The screenshot shows the 'Create Search Service' wizard. On the left, a sidebar lists options: New service, Select data, Select search column, Select attributes, Select columns (which is selected), and Configure indexing. The main area is titled 'Select columns to include in the service'. It contains a note about including columns in the search index. Below is a list of columns with checkboxes, where 'ID', 'TITLE', 'PRODUCT', 'TRANSCRIPT', and 'DATE' are checked. At the bottom are 'Select all' and 'Clear all' buttons. To the right is a 'Data Preview' section showing a table of data from 'DASH\_DB\_SI.RETAIL\_SUPPORT\_CASES' with columns PRODUCT, TRANSCRIPT, and DATE.

PRODUCT	TRANSCRIPT	DATE
1 ThermoJacket Pro	[10:15] Customer: Hi there, ...	2025-07-06T00:00:00Z
2 ThermoJacket Pro	[10:01] Customer: Hi there, ...	2025-05-30T00:00:00Z
3 ThermoJacket Pro	[09:15] Customer: Hi there, ...	2025-07-26T00:00:00Z
4 ThermoJacket Pro	[12:30] Customer: Hi, I rece... ...	2025-07-15T00:00:00Z
5 ThermoJacket Pro	[09:15] Customer: Hi there, ...	2025-06-05T00:00:00Z
6 UltraRunner Shoes	[09:15] Customer: Hi therel... ...	2025-08-06T00:00:00Z
7 ThermoJacket Pro	[09:15] Customer: Hi therel... ...	2025-08-10T00:00:00Z
8 HydraBottle 2L	[10:15] Customer: Hi therel... ...	2025-07-29T00:00:00Z
9 ThermoJacket Pro	[12:00] Customer: Hi there, ...	2025-07-29T00:00:00Z
10 ThermoJacket Pro	[10:05] Customer: Hi there, ...	2025-08-11T00:00:00Z
11 PowerYoga Mat	[09:05] Customer: Hi there, ...	2025-06-08T00:00:00Z
12 HydraBottle 2L	[12:15] Customer: Hi therel... ...	2025-05-18T00:00:00Z
13 UltraRunner Shoes	[09:15] Customer: Hi therel... ...	2025-08-06T00:00:00Z
14 HydraBottle 2L	[12:05] Customer: Hi therel... ...	2025-07-07T00:00:00Z

9. Select "SNOWFLAKE\_LEARNING\_WH" as the Warehouse for indexing. Then select "Create Search Service"

The screenshot shows the 'Configure your Search Service' step. The sidebar still has 'Select columns' selected. The main area has a title 'Configure your Search Service' and a note about selecting configuration parameters. It includes fields for 'Target Lag' (set to 1 hour), 'Embedding model' (set to 'snowflake-arctic-embed-m-v1.5'), and 'Warehouse for indexing' (set to 'SNOWFLAKE\_LEARNING\_WH'). A note at the bottom says indexing can take a few minutes. At the bottom are 'Previous' and 'Create' buttons.

10. There we go, now you have a Cortex Search Service. Make sure to show how you can also use the different methods to display to interact with Cortex Search.

COLUMN	SEARCHABLE	ATTRIBUTE
TRANSCRIPT	✓	✓
TITLE	✓	✓
PRODUCT	✓	✓
ID	—	—

```

import os
from snowflake.core import Root
from snowflake.snowpark import Session
CONNECTION_PARAMETERS = {
    "url": "apchrb-vac81038.us-east-1.snowflakecomputing.com",
    "user": "dash_db_si",
    "password": "P@ssw0rd"
}
# Create a session
session = Session.builder.config(**CONNECTION_PARAMETERS).create()
# Run a query
df = session.read.csv("support_case", header=True)
# Show the first few rows
df.show(5)
  
```

## Using Cortex Search

1. Make sure everyone is at this step before proceeding. Once the “Serving” and “Indexing” states are both “Active” we can select the “Playground” button in the top right corner. Note: You may need to refresh the page, either by using your web browser’s refresh button or by selecting the refresh icon located in the top-right corner under Search Services.

The screenshot shows the Cortex Search interface for the SUPPORT\_CASES search service. At the top, there are tabs for "Search Service", "Data Preview", and "Costs". Below the tabs, the table structure is displayed:

COLUMN	SEARCHABLE	ATTRIBUTE
TRANSCRIPT	<input checked="" type="checkbox"/>	
TITLE	<input checked="" type="checkbox"/>	
PRODUCT	<input checked="" type="checkbox"/>	
ID		

Below the table, there are status indicators: "Serving ACTIVE", "Indexing ACTIVE", "Created on Oct 13, 2025", and "Last updated 4 minutes ago".

The screenshot shows the Snowflake AI & ML interface for the SUPPORT\_CASE search service. The left sidebar shows navigation options like "Work with data", "Transformation", "Monitoring", and "Marketplace". The main area displays the search service configuration with a search bar, settings for "Limit" (set to 10), and a "Columns" dropdown set to "TRANSCRIPT". A preview section shows the results for the query "SUPPORT\_CASE".

2. Now you can perform any search, here are some sample search terms:
  - a. Clothes — Shows you all tickets related to clothing products
  - b. Shoes — Shows you all tickets related to shoes, including activities you would need shows for
  - c. Replacement — Shows you all tickets related to issuing a replacement

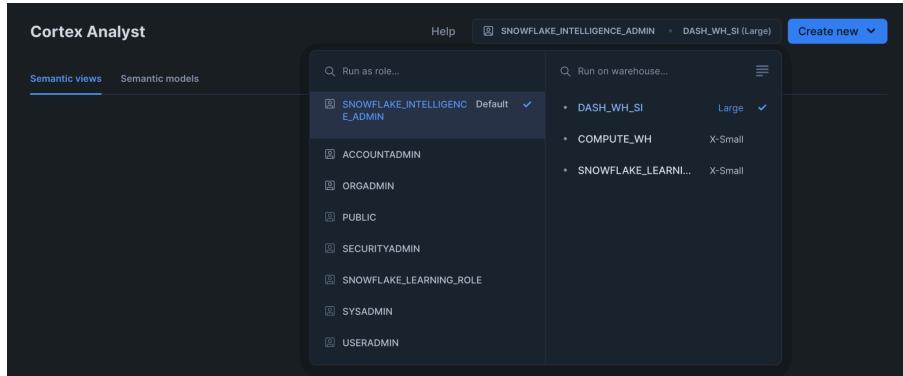
- d. Damage — Shows you all tickets that take about any damage, defect, etc
  - e. Feel free to mix and match them, be creative and come up with additional ones on your own
3. Lastly, ask it a question like “What are the top theme's of all support tickets?”, you should see that it doesn't really provide an answer to those type of question as Cortex Search works as a RAG for unstructured data. This is your transition to introduce our next Module on Cortex Analyst

## Module 3: Cortex Analyst

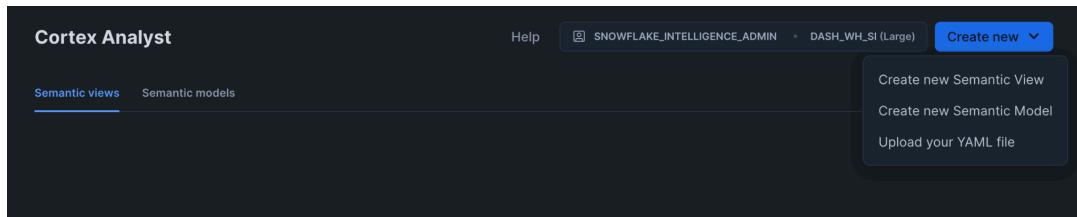
### Setting Up Cortex Analyst

1. Navigate to [mlh.link/snowflake-data](http://mlh.link/snowflake-data)
2. Go through the `marketing_campaigns.yaml` file
3. Make sure to explain the following:
  - a. 1–2: name: Defines the analyst configuration name (`Sales_And_Marketing_Data`).
  - b. 3–95: tables:
    - i. Each table block (`MARKETING_CAMPAIGN_METRICS` [4–53], `PRODUCTS` [54–82], `SALES` [83–112], `SOCIAL_MEDIA` [113–143]) specifies:
      1. Which Snowflake database, schema, and table to use
      2. Primary key columns
      3. Dimensions (business attributes and synonyms; include example values and SQL expressions)
      4. Facts (quantitative measures; include synonyms and example values)
      5. Time dimensions (date fields; include synonyms and example values)
  - c. 144–157: relationships:
    - i. Defines how tables are joined (e.g., `SALES` to `PRODUCTS`, `SOCIAL_MEDIA` to `MARKETING_CAMPAIGN_METRICS`)
    - ii. Specifies join columns, relationship type (`many_to_one`), and join type (`inner`).
  - d. 158–175: verified\_queries:
    - i. Example of a validated query (`sales`)
      1. Written as natural language
      2. Shows corresponding SQL
      3. Indicates who verified and when
      4. Used for onboarding or demonstration
4. Download the `marketing_campaigns.yaml`
5. Navigate to “AI & ML > Cortex Analyst”.

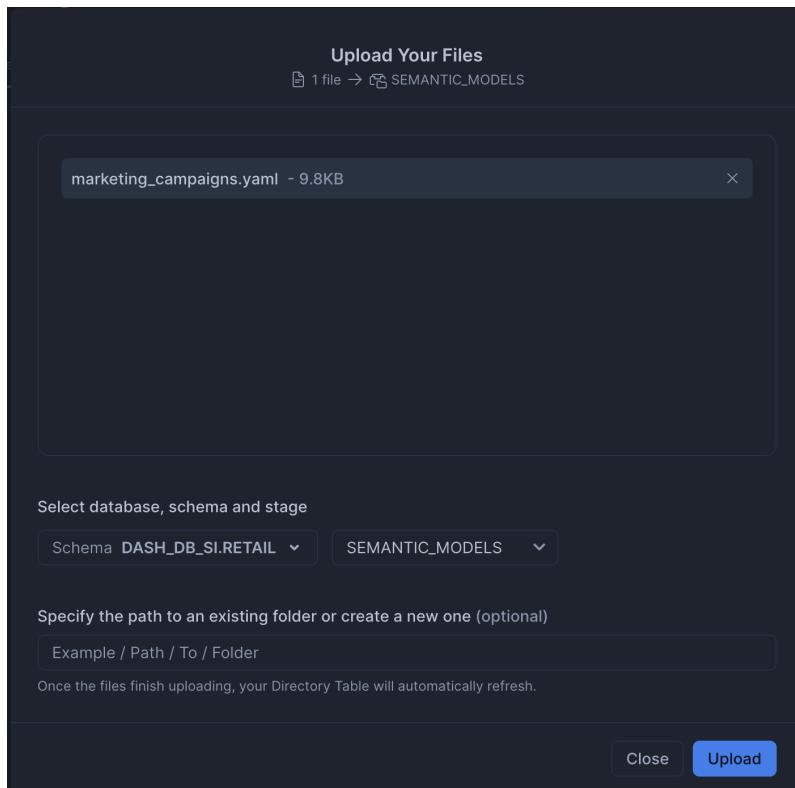
6. Update your Role and Warehouse in the top right corner to "SNOWFLAKE\_INTELLIGENCE\_ADMIN | DASH\_WH\_SI"



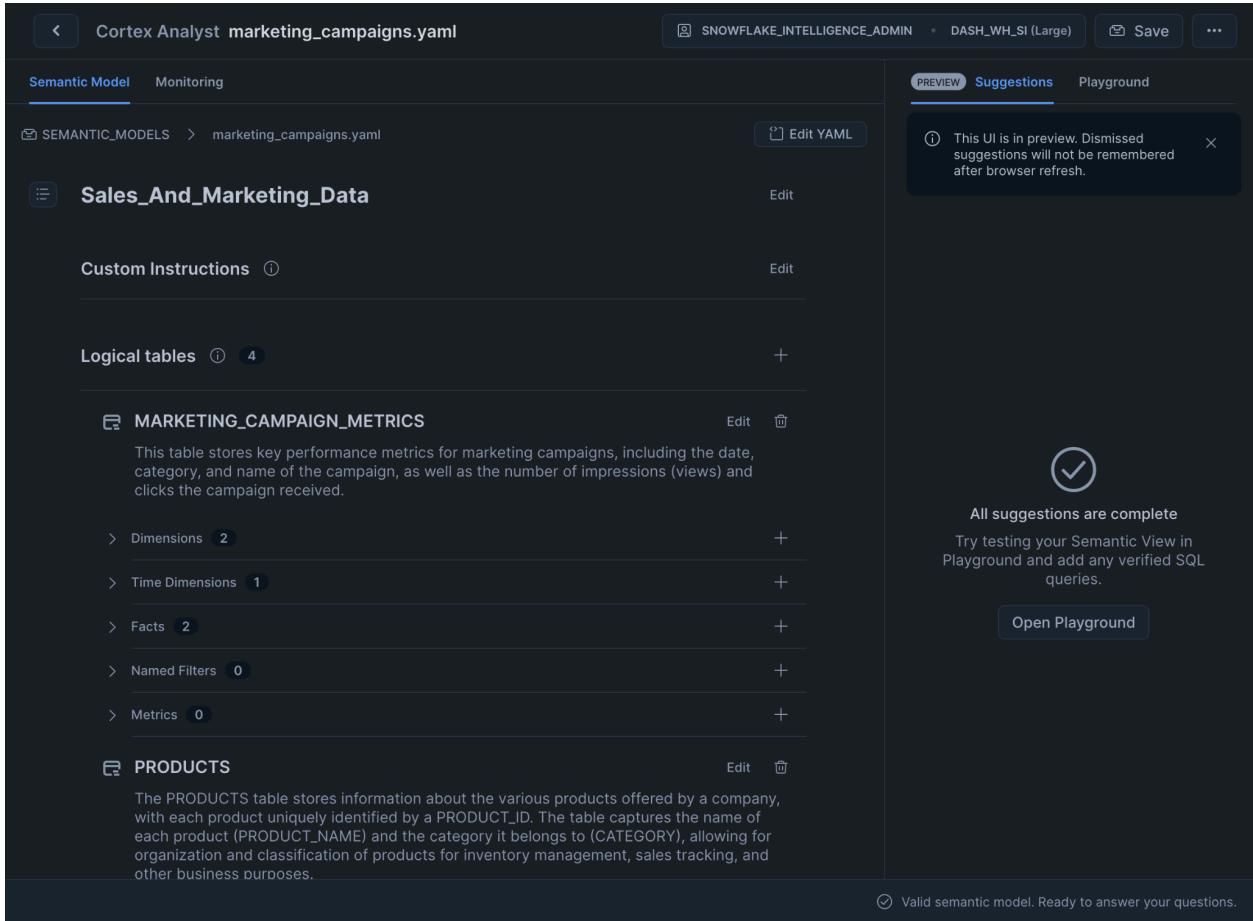
7. Click on "Create new" in the top right corner and select "Upload your YAML file"



8. Select "DASH\_DB\_SI.RETAIL > SEMANTIC\_MODELS" for database, schema, and stage. Then select "Upload".



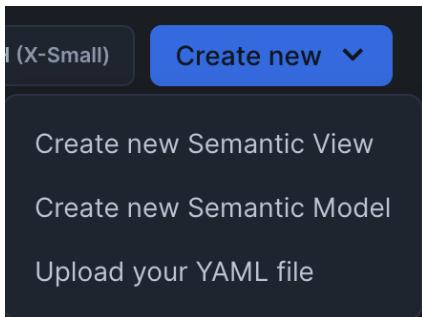
9. Scroll through and showcase the different components that the YAML file is generating



The screenshot shows the Cortex Analyst interface for managing semantic models. The main area displays a semantic model named "marketing\_campaigns.yaml". It lists two logical tables: "MARKETING\_CAMPAIGN\_METRICS" and "PRODUCTS". Under "MARKETING\_CAMPAIGN\_METRICS", there are sections for Dimensions (2), Time Dimensions (1), Facts (2), Named Filters (0), and Metrics (0). Under "PRODUCTS", there is a brief description of the table's purpose. On the right side, a sidebar indicates that all suggestions are complete and encourages testing in the playground. At the bottom right, a message says "Valid semantic model. Ready to answer your questions."

#### Optional: Showcase and Explain Semantic Views & Models

1. Navigate back to "AI & ML > Cortex Analyst".



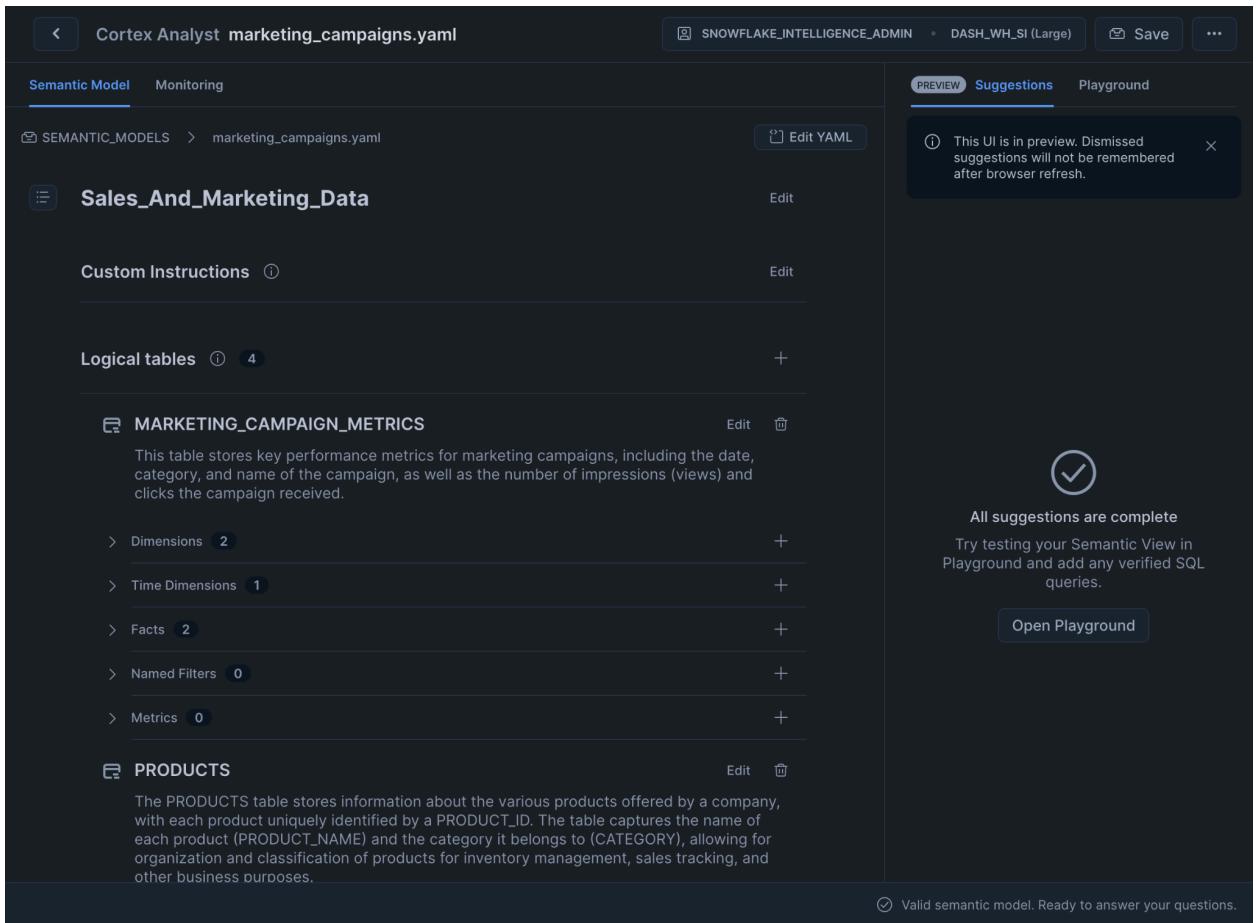
The screenshot shows a dropdown menu titled "Create new" in the Cortex Analyst interface. The menu includes three options: "Create new Semantic View", "Create new Semantic Model", and "Upload your YAML file".

2. Showcase how you can create each one and make sure to explain:
  - a. Semantic View: SQL-defined view that encapsulates business logic such joins and calculations over one or more base tables.

- b. Semantic Model: Defines entities, attributes, hierarchies, and measures.
- c. YAML File: Declarative file that tells Cortex Analyst how to interpret your data model for natural-language queries. Specifies tables, columns (dimensions, facts, time dimensions), synonyms, descriptions, relationships, and verified example queries. This file basically declares a Semantic Model.

## Using Cortex Analyst

1. Make sure everyone is at this step before proceeding. Once the “Valid semantic model. Ready to answer your question is displayed”. You are now ready to click “Open Playground” and start asking questions.



The screenshot shows the Cortex Analyst interface with a dark theme. On the left, there's a navigation bar with 'Cortex Analyst marketing\_campaigns.yaml' and tabs for 'Semantic Model' (which is selected) and 'Monitoring'. Below this, there's a breadcrumb trail: SEMANTIC\_MODELS > marketing\_campaigns.yaml. A 'Edit YAML' button is available. The main content area displays two sections: 'Sales\_And\_Marketing\_Data' and 'MARKETING\_CAMPAIGN\_METRICS'. 'Sales\_And\_Marketing\_Data' has an 'Edit' button. 'MARKETING\_CAMPAIGN\_METRICS' is expanded, showing sub-sections: 'Dimensions' (2), 'Time Dimensions' (1), 'Facts' (2), 'Named Filters' (0), and 'Metrics' (0). Each section has an 'Edit' button. To the right, a sidebar titled 'Suggestions' shows a message: 'This UI is in preview. Dismissed suggestions will not be remembered after browser refresh.' A large green checkmark icon indicates 'All suggestions are complete'. It also says 'Try testing your Semantic View in Playground and add any verified SQL queries.' A 'Open Playground' button is present. At the bottom right, a message says 'Valid semantic model. Ready to answer your questions.'

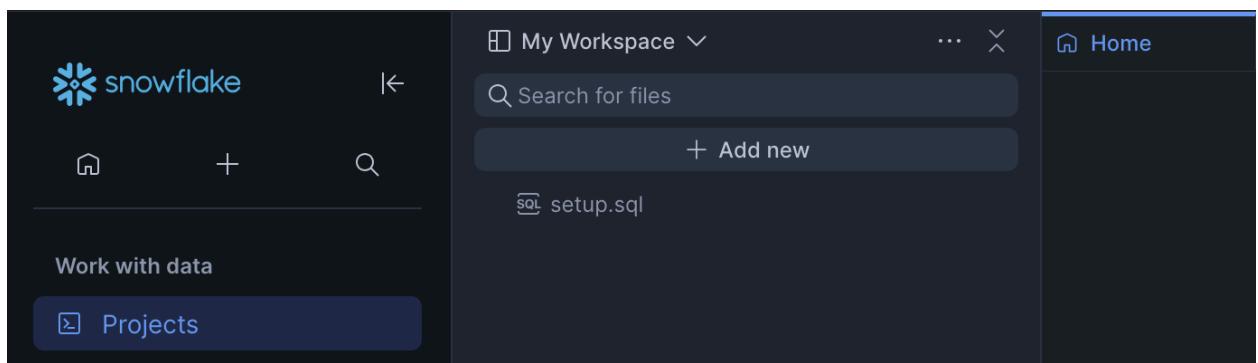
2. Now you can ask any question, here are some sample questions:
  - a. Explain the dataset
  - b. Show me the trend of sales by product category between June 2025 and August 2025

## Module 4: User Defined Functions

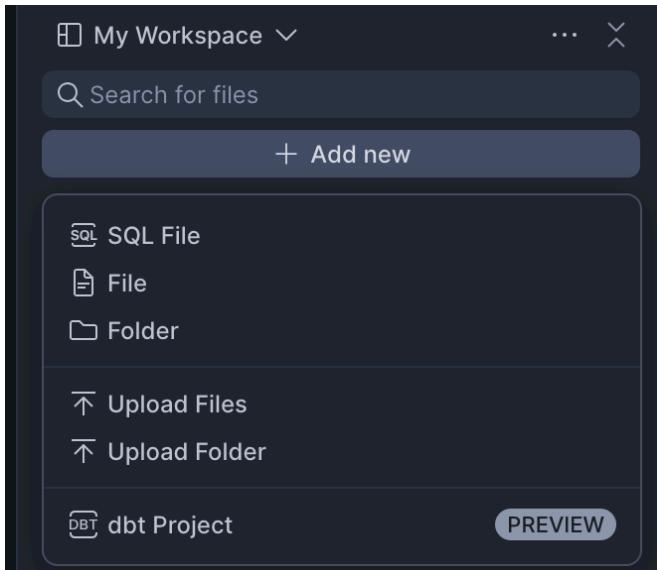
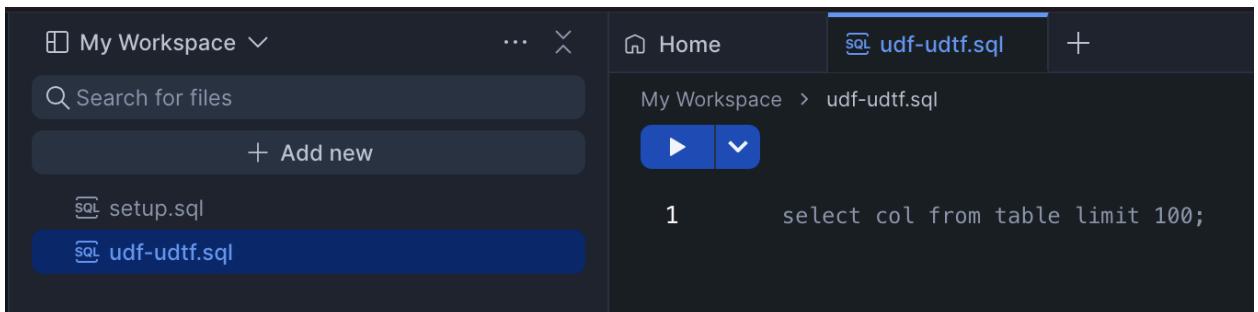
Still need to design example UDF based on scenario  
<https://github.com/MLH/build25-udf-udtf-examples>

### Setting Up UDF/UDTF via SQL File

1. Go to “Projects” which should take you to “My Workspace”



- Click on “+ Add new” and select “SQL File” and name it udf-udtf.sql

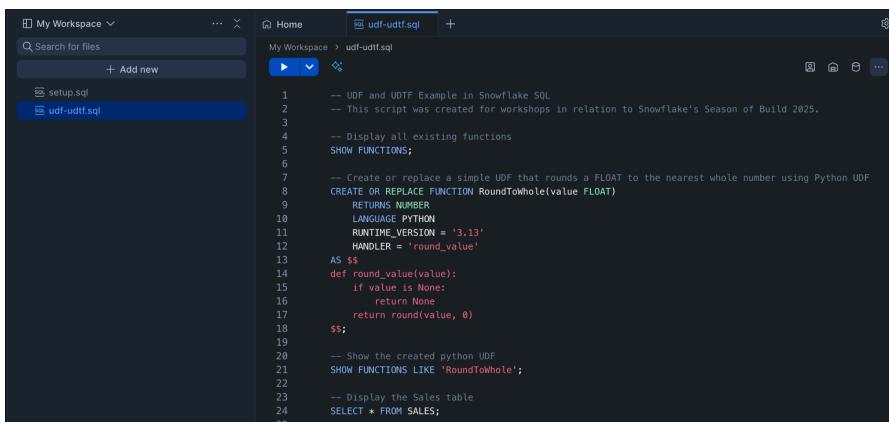



```

1      select col from table limit 100;

```

- Navigate back to the GitHub repo, open udf-udtf.sql, copy its contents, and paste them into the SQL file you created in your workspace.



```

1 -- UDF and UDTF Example in Snowflake
2 -- This script was created for workshops in relation to Snowflake's Season of Build 2025.
3
4 -- Display all existing functions
5 SHOW FUNCTIONS;
6
7 -- Create or replace a simple UDF that rounds a FLOAT to the nearest whole number using Python UDF
8 CREATE OR REPLACE FUNCTION RoundToWhole(value FLOAT)
9   RETURNS NUMBER
10  LANGUAGE PYTHON
11  RUNTIME_VERSION = '3.13'
12  HANDLER = 'round_value'
13 AS $$
14 def round_value(value):
15   if value is None:
16     return None
17   return round(value, 0)
18 $$;
19
20 -- Show the created python UDF
21 SHOW FUNCTIONS LIKE 'RoundToWhole';
22
23 -- Display the Sales table
24 SELECT * FROM SALES;
25

```

## Executing and Using UDF

1. Instead of executing the whole .sql file like last time, slowly go step by step and click the play button after explaining what each of the lines is doing.
  - a. 5: Displays all existing functions, which if you scroll through all of them should have `is_builtin` set to Y

Results (just now)						
		Table	Chart	Q	1,041 rows	773ms
#	name	schema_name	is_builtin	is_aggregate	is_ansi	min_n
1	2012-08-01 00:00:00.000 -0700	f=	Y	N	Y	
2	2012-08-01 00:00:00.000 -0700	%	Y	N	N	
3	2012-08-01 00:00:00.000 -0700	*	Y	N	N	
4	2012-08-01 00:00:00.000 -0700	+	Y	N	Y	
5	2012-08-01 00:00:00.000 -0700	-	Y	N	Y	
6	2012-08-01 00:00:00.000 -0700	/	Y	N	N	

- b. 8-18: Creates a simple UDF that rounds a FLOAT to the nearest whole number using Python UDF. Call out that it took seconds and the format:
  - i. Return: What should this UDF return
  - ii. Language: What coding/query language is being used
  - iii. Runtime Version: Which Python Runtime should Snowflake use when the Python for this function is run
  - iv. Handler: What is the function's name that you will define in the UDF
  - v. As: The Python code that is going to be this UDF

Results (just now)						
		Table	Chart	Q	1 row	4.8s
00		A status				
1		Function ROUNDTOWHOLE successfully created.				

- c. 21: Checks to make sure in the Functions list there does exist a RoundToWhole UDF. Note: The `is_builtin` is set to N, scroll to the right and call out the `is_table_function` is set to N since this is a UDF, and show the language is set to PYTHON.

Results (just now)						
		Table	Chart	Q	1 row	98ms
#	name	schema_name	is_builtin	is_aggregate	is_ansi	
1	2025-10-23 21:36:09.8 ROUNDTOWHOLE	RETAIL	N	N	N	

- d. 24: Displays all columns from the Sales table

Results (just now)						
		DATE	REGION	PRODUCT_ID	UNITS SOLD	SALES_AMOUNT
1	2025-05-16	North		1	28	2199.67
2	2025-05-16	North		2	32	1039.35
3	2025-05-16	North		3	32	692.70
4	2025-05-16	North		4	25	866.81
5	2025-05-16	North		5	27	1473.00
6	2025-05-16	North		6	26	810.15

- e. 27-38: Using the new UDF that we create, display the Sales table with the UDF applied to the Sales\_Amount. Note: You should see all the results rounded.

Results (2 minutes ago)						
		DATE	REGION	PRODUCT_ID	UNITS SOLD	ROUNDED SALES AMOUNT
1	2025-05-16	North		1	28	2200
2	2025-05-16	North		2	32	1039
3	2025-05-16	North		3	32	693
4	2025-05-16	North		4	25	867
5	2025-05-16	North		5	27	1473
6	2025-05-16	North		6	26	810

- f. 42-62: Create a simple UDTF that will create a new column that stores the average price per unit of product for each sale record within the Sales table. Note: Call out how the SQL UDTF only took milliseconds compared to the seconds for the Python UDF.

Results (just now)						
		status	avgpriceperunitproductpersale successfully created.			
1			Function	AVGPRICEPERUNITPRODUCTPERSALE	successfully created.	

- g. 65: Checks to make sure in the Functions list there does exist a AvgPricePerUnitProductPerSale UDTF. Note: The is\_builtin is set to N, scroll to the right is\_table\_function is set to Y since this is a UDTF, and show the language is set to SQL.

Results (just now)						
		name	schema_name	is_builtin	is_aggregate	is_ansi
1	AVGPRICEPERUNITPRC	RETAIL		N	N	N

- h. 68: Show the simple UDTF in action by producing a table with avg price per unit of a product for each sale record. What do you notice that is strange about those numbers? (Hint: Do prices for products ever stay the same?)

Results (just now)

Table						
	DATE	REGION	PRODUCT_ID	UNITS SOLD	SALES_AMOUNT	Avg Price Per Unit
1	2025-06-03	East	1	26	1184.81	45.56961538
2	2025-07-16	East	1	31	1478.08	47.68000000
3	2025-05-21	East	1	25	1037.07	41.48280000
4	2025-08-02	East	1	33	900.52	27.28848485
5	2025-07-15	East	1	31	1167.76	37.66967742
6	2025-07-29	East	1	36	1561.95	43.38750000

- i. 71–76: Using both the UDF and UDTF, create a view that leverages the AvgPricePerUnitProductPerSale UDTF to generate a table. Then, apply the RoundToWhole UDF to the avg\_price\_per\_unit column from that table, creating a rounded\_avg\_price\_per\_unit.

Results (just now)

Table	
	status
1	View AVG_PRICE_PER_UNIT_PRODUCT_PER_SALE successfully created.

- j. 78–85: Creates a new table PRODUCTS\_WITH\_AVG\_PRICE that joins the PRODUCTS table with the average price, which is the average of the rounded average prices of each product per sale for all regions from the VIEW that we created.

Results (just now)

Table	
	status
1	Table PRODUCTS_WITH_AVG_PRICE successfully created.

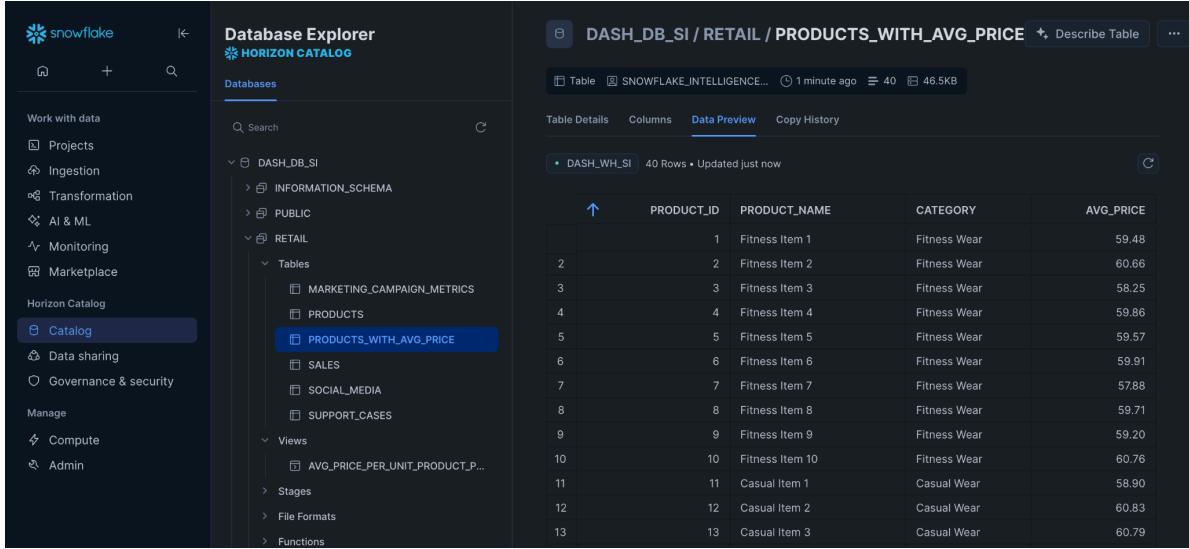
- k. 88: Sets the status to show that UDF/UDTF creation and usage was successful

Results (1 minute ago)

Table	
	status
1	UDF and UDTF creation and usage completed successfully!

## Verifying the Results of Running the UDF/UDTF

- Now, let's go verify the UDF/UDTF script ran successfully by going to "Catalog" and checking for "DASH\_DB\_SI/RETAIL" DB. This should list a new table called PRODUCTS\_WITH\_AVG\_PRICE with 40 rows showing all the products with the average price from all sales and a new view called AVG\_PRICE\_PER\_UNIT\_PRODUCT\_PER\_SALE.



	PRODUCT_ID	PRODUCT_NAME	CATEGORY	AVG_PRICE
1	1	Fitness Item 1	Fitness Wear	59.48
2	2	Fitness Item 2	Fitness Wear	60.66
3	3	Fitness Item 3	Fitness Wear	58.25
4	4	Fitness Item 4	Fitness Wear	59.86
5	5	Fitness Item 5	Fitness Wear	59.57
6	6	Fitness Item 6	Fitness Wear	59.91
7	7	Fitness Item 7	Fitness Wear	57.88
8	8	Fitness Item 8	Fitness Wear	59.71
9	9	Fitness Item 9	Fitness Wear	59.20
10	10	Fitness Item 10	Fitness Wear	60.76
11	11	Casual Item 1	Casual Wear	58.90
12	12	Casual Item 2	Casual Wear	60.83
13	13	Casual Item 3	Casual Wear	60.79

- The results indicate that the average price across all products is relatively consistent.

# Module 5: Snowflake Intelligence

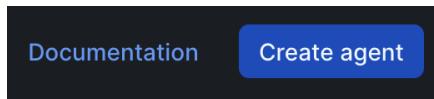
## Setting Up Custom Snowflake Intelligence

### Creating an Agent

1. Navigate to “AI & ML > Cortex Agent”

The screenshot shows the Snowflake AI & ML interface. On the left, there's a sidebar with categories like Work with data, Projects, Ingestion, Transformation, AI & ML (which is selected), Monitoring, Marketplace, Horizon Catalog, Catalog, Data sharing, Governance & security, Manage, Compute, and Admin. Below the sidebar, there's a message about credits and an upgrade button. At the bottom, there's a user profile for Kevin Chen. The main area is titled "Agents PREVIEW" and lists various agents: Studio, Cortex Analyst, Cortex Search, Features, Models, Evaluations, Document AI, Agents, and Snowflake Intelligence. The "Agents" item is highlighted with a blue box. The top right of the main area has buttons for "Documentation" and "Create agent".

2. Click “Create agent” in the top right corner



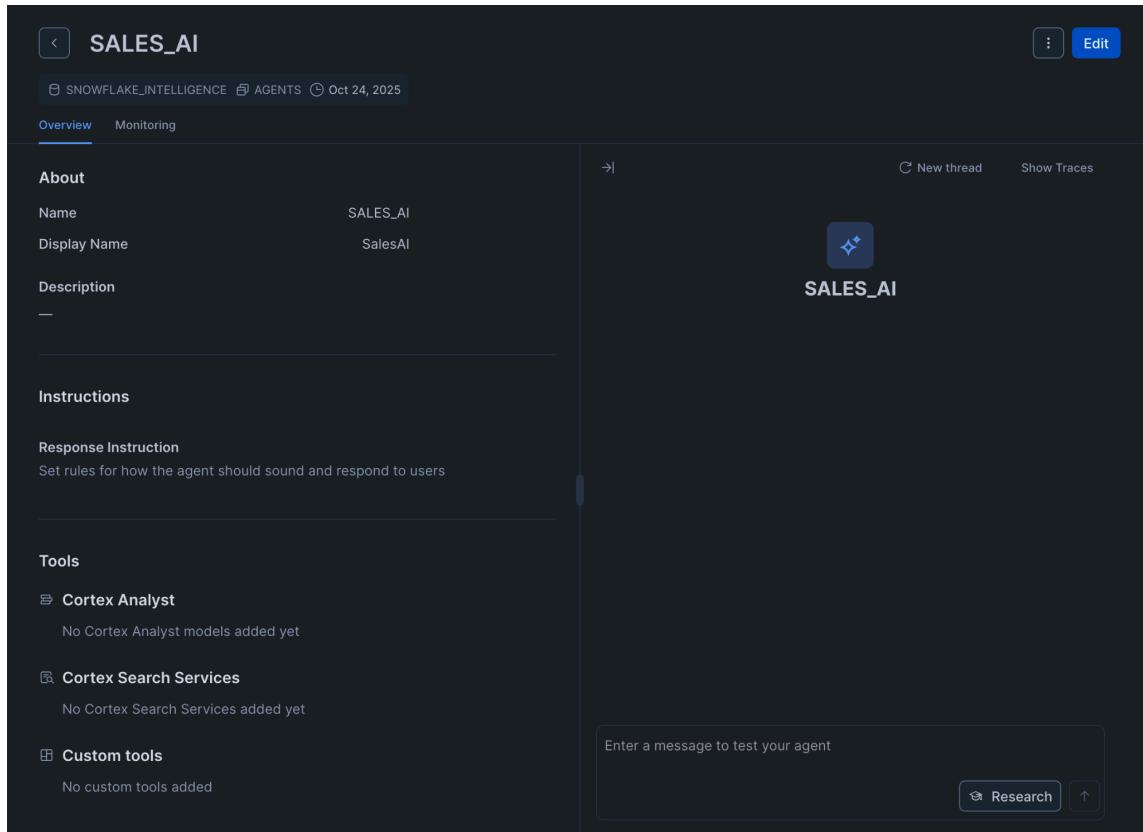
3. Check the box for "Create this agent for Snowflake Intelligence". The default Database and schema should be set to SNOWFLAKE\_INTELLIGENCE.AGENTS. For Agent object name, enter Sales\_AI, this is the internal identifier Snowflake uses to organize your agent's metadata. For Display name, choose any user-facing name

you prefer; we'll use SalesAI as the name shown to users.

4. Select the Sales\_AI Agent from the list to open and configure the Agent.

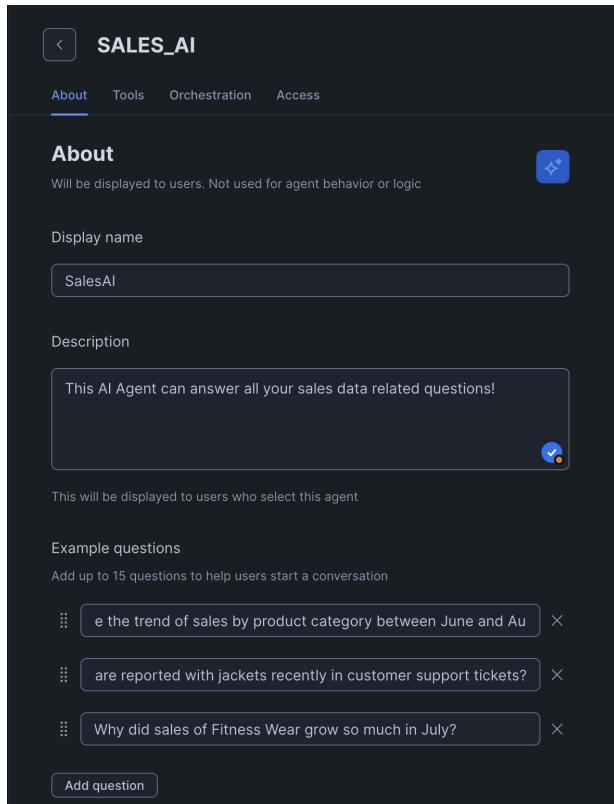
5. You can now view the current Sales\_AI configuration, which displays general information, instructions, tools, and a preview test chat interface. To customize your

Agent, click the “Edit” button in the top right corner.



6. In this step, configure the essential details for your Agent. Optionally, add a description to clarify its purpose or scope. To guide users and spark ideas, enter a few sample questions that demonstrate the types of queries the Agent can answer. Examples:
  - a. Show me the trend of sales by product category between June and August
  - b. What issues are reported with jackets recently in customer support tickets?

c. Why did sales of Fitness Wear grow so much in July?



**SALES\_AI**

About Tools Orchestration Access

**About**

Will be displayed to users. Not used for agent behavior or logic

Display name

SalesAI

Description

This AI Agent can answer all your sales data related questions!

This will be displayed to users who select this agent

Example questions

Add up to 15 questions to help users start a conversation

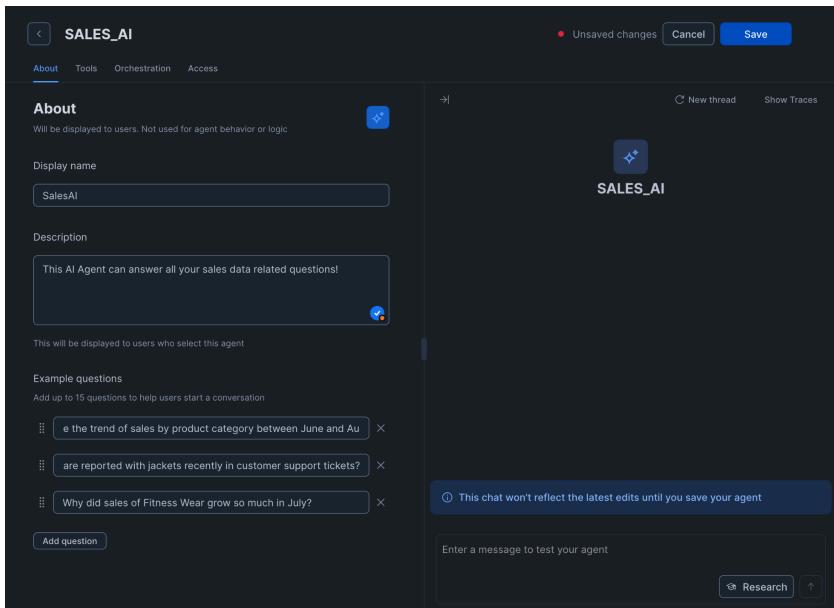
e the trend of sales by product category between June and Au  
X

are reported with jackets recently in customer support tickets?  
X

Why did sales of Fitness Wear grow so much in July?  
X

Add question

7. After completing all fields, the changes won't appear until you save them. Click the "Save" button in the top right corner to persist your Agent configuration.



**SALES\_AI**

About Tools Orchestration Access

• Unsaved changes Cancel Save

**About**

Will be displayed to users. Not used for agent behavior or logic

Display name

SalesAI

Description

This AI Agent can answer all your sales data related questions!

This will be displayed to users who select this agent

Example questions

Add up to 15 questions to help users start a conversation

e the trend of sales by product category between June and Au  
X

are reported with jackets recently in customer support tickets?  
X

Why did sales of Fitness Wear grow so much in July?  
X

Add question

New thread Show Traces

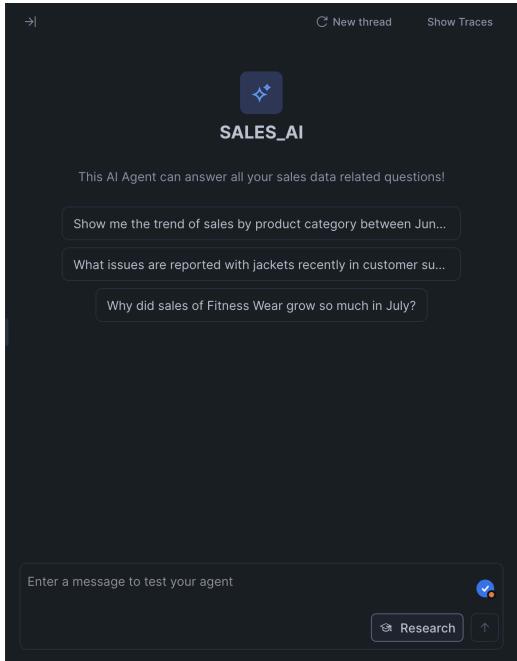
SALES\_AI

This chat won't reflect the latest edits until you save your agent

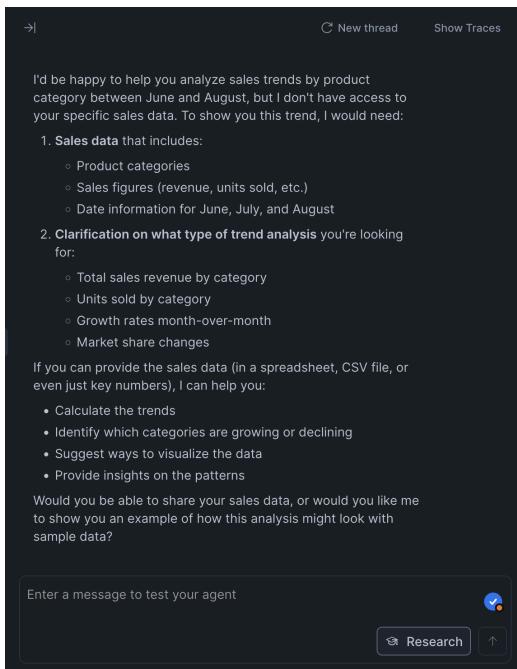
Enter a message to test your agent

Research

- Once you click "Save", the preview test chat interface will reflect all the configuration details you've set for the Agent.

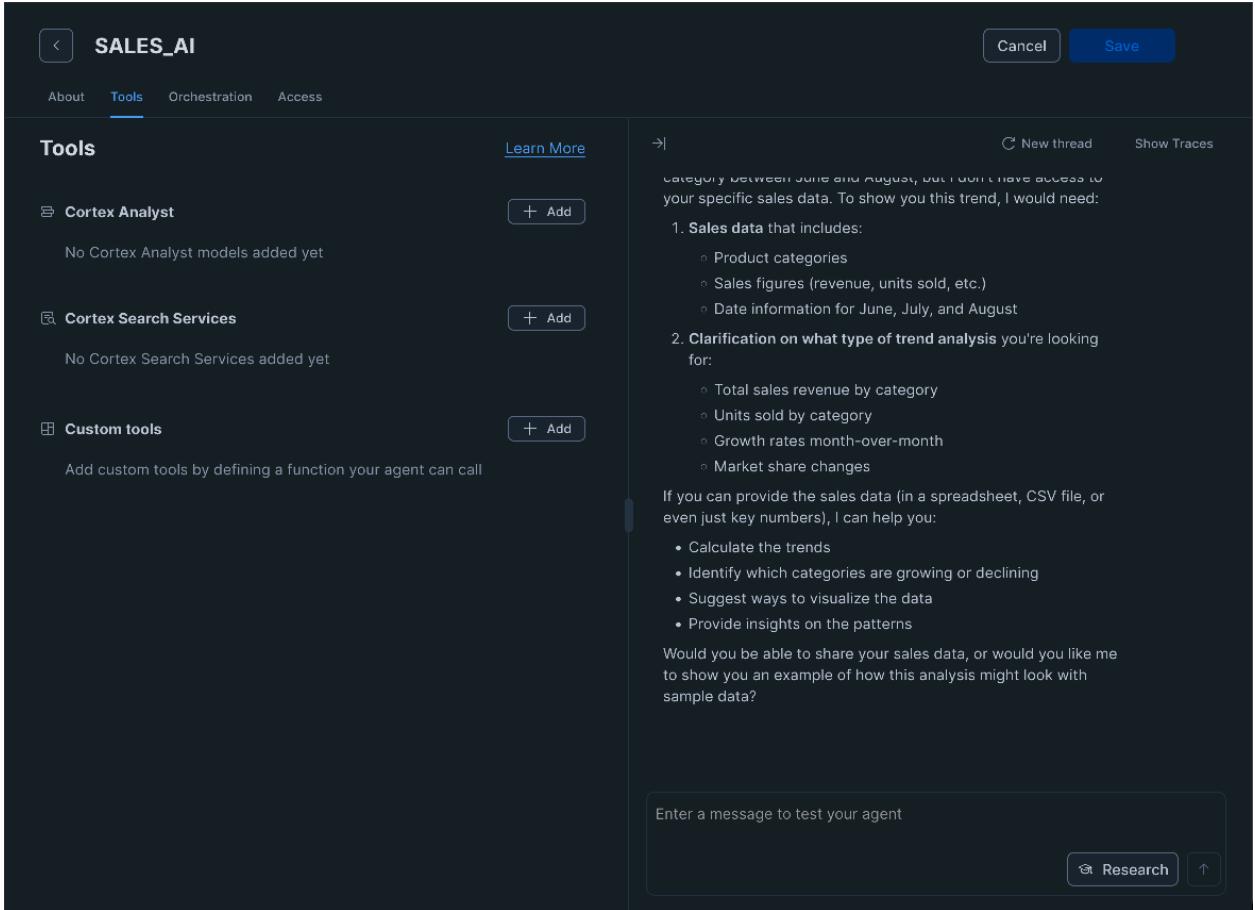


- Select one of the sample questions to test the Agent. You'll notice it responds that it lacks sufficient data or information to fulfill the request. Consider why this might occur? (Hint: Think about components that enable Snowflake Intelligence to function)



## Setting up the Tools

1. Click “Tools” in the top left corner to configure the capabilities available to your AI Agent. Here you can integrate tools including Cortex Analyst for data analysis, Cortex Search for information retrieval, and Custom Tools such as UDFs for specialized functionality.



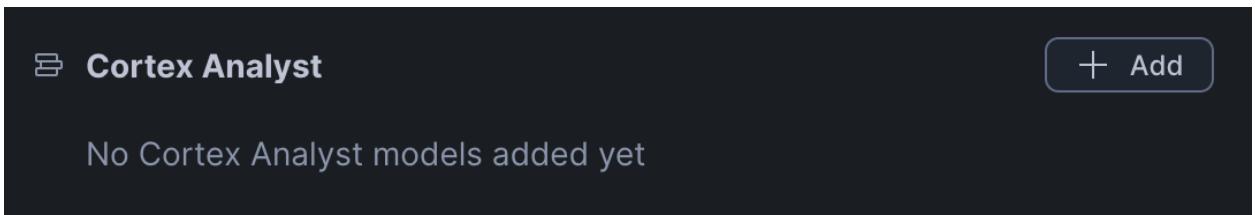
The screenshot shows the 'SALES\_AI' tool configuration interface. The 'Tools' tab is selected. There are three main sections:

- Cortex Analyst**: No Cortex Analyst models added yet. Includes a '+ Add' button.
- Cortex Search Services**: No Cortex Search Services added yet. Includes a '+ Add' button.
- Custom tools**: Add custom tools by defining a function your agent can call. Includes a '+ Add' button.

A message input field at the bottom right contains the following text:

→|  
 Category between June and August, but I don't have access to  
 your specific sales data. To show you this trend, I would need:  
 1. Sales data that includes:  
     ◦ Product categories  
     ◦ Sales figures (revenue, units sold, etc.)  
     ◦ Date information for June, July, and August  
 2. Clarification on what type of trend analysis you're looking  
 for:  
     ◦ Total sales revenue by category  
     ◦ Units sold by category  
     ◦ Growth rates month-over-month  
     ◦ Market share changes  
 If you can provide the sales data (in a spreadsheet, CSV file, or  
 even just key numbers), I can help you:  
     • Calculate the trends  
     • Identify which categories are growing or declining  
     • Suggest ways to visualize the data  
     • Provide insights on the patterns  
 Would you be able to share your sales data, or would you like me  
 to show you an example of how this analysis might look with  
 sample data?

2. Begin by adding the Cortex Analyst configured in Module 3 to your Agent's available tools. Click the “+ Add” button.



The screenshot shows the 'Cortex Analyst' configuration interface. It displays the following content:

→|  
**Cortex Analyst**  
 + Add  
 No Cortex Analyst models added yet

3. Select the “Semantic model file radio” button. Navigate to and select DASH\_DB\_SI.RETAIL.SEMANTIC\_MODELS, then choose the marketing\_campaigns.yml file. Enter Sales\_And\_Marketing\_Data as the tool

name. For the description, click “Generate with Cortex” to automatically create a description that will guide the Agent during orchestration. Set the “Query timeout” to 60 seconds, then click “Add” in the bottom right corner.

### Add tool: Cortex Analyst

Cortex Analyst  Semantic view  Semantic model file

Schema DASH\_DB\_SI.RETAIL X ▼

SEMANTIC\_MODELS X ▼

Go Back Current Path: / marketing\_campaigns.yaml

Selected file: marketing\_campaigns.yaml

Tool details Name

Description Generate with Cortex

[Learn more](#) Cancel Add

Selected file: marketing\_campaigns.yaml

Tool details Name

Description Generate with Cortex

Will be used during orchestration to guide how the agent uses this tool

Describe what this analyst tool does and when it should be used

✓

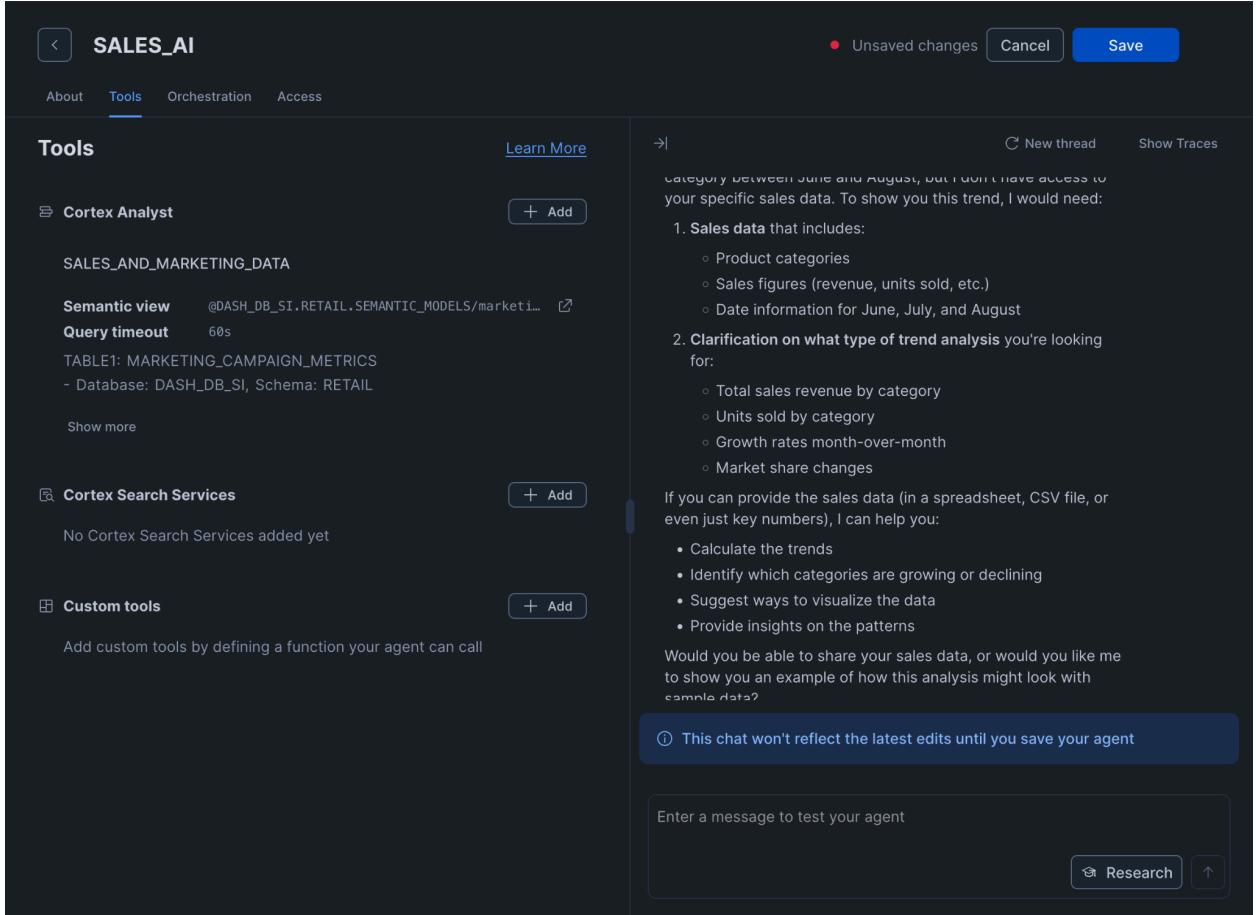
Warehouse   
 Warehouse to execute the SQL query

User's default  Custom

Query timeout   
 Maximum time in seconds for query execution (max 300s)

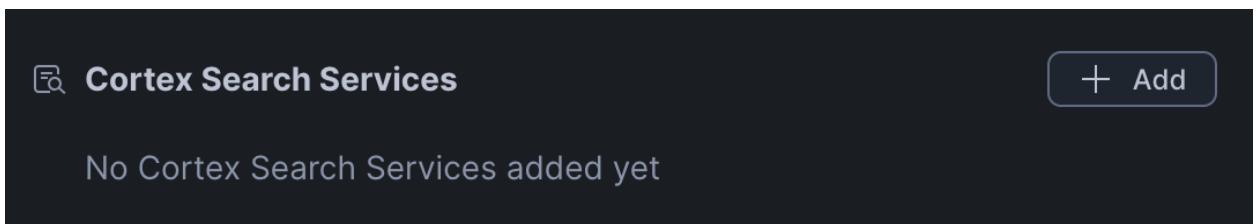
[Learn more](#) Cancel Add

4. You should now see that under Cortex Analyst you have the instance configure for the Agent to use



The screenshot shows the 'SALES\_AI' instance configuration page. The 'Tools' tab is selected. Under 'Cortex Analyst', there is a section for 'SALES\_AND\_MARKETING\_DATA' which includes a 'Semantic view' set to '@DASH\_DB\_SI.RETAIL.SEMANTIC\_MODELS/marketing...' and a 'Query timeout' of 60s. Below this is a note about TABLE1: MARKETING\_CAMPAIGN\_METRICS, mentioning it's from Database: DASH\_DB\_SI, Schema: RETAIL. A '+ Add' button is available for adding more semantic views. The 'Cortex Search Services' section shows 'No Cortex Search Services added yet' and has its own '+ Add' button. The 'Custom tools' section allows adding functions for agents and also has a '+ Add' button. On the right side, there is a chat interface with a message from 'Cortex' asking for sales data trends. It lists two main points: 1. Sales data that includes categories, sales figures, and date information for June, July, and August. 2. Clarification on what type of trend analysis is being looked for, such as total sales revenue by category, units sold by category, growth rates month-over-month, or market share changes. A note says if sales data is provided, insights can be provided. A warning message at the bottom states: 'This chat won't reflect the latest edits until you save your agent'. At the bottom, there is a text input field 'Enter a message to test your agent' with a 'Research' button and an upward arrow icon.

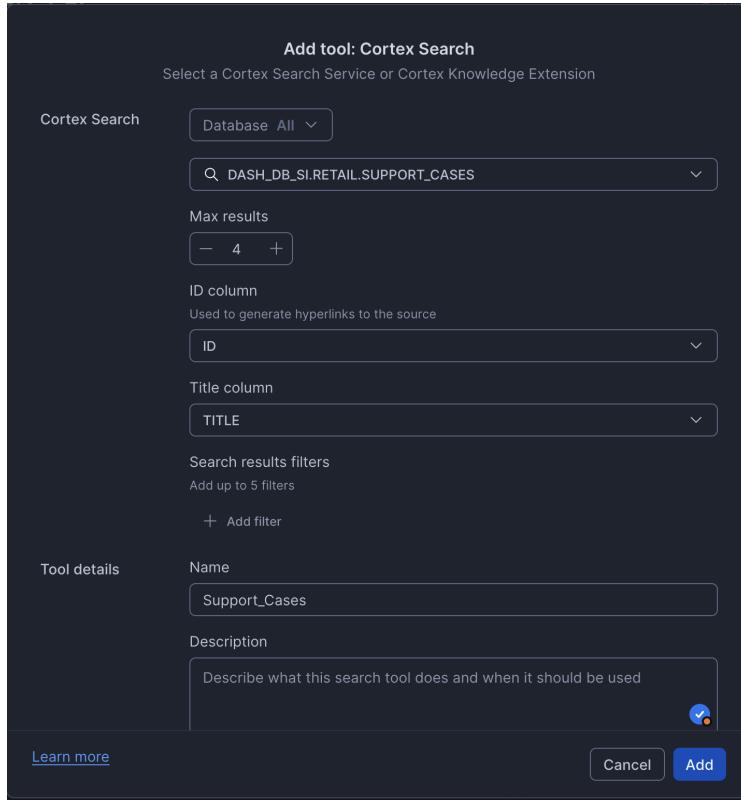
5. Next, we are going to begin adding the Cortex Search configured in Module 2 to your Agent's available tools. Click the "+ Add" button.



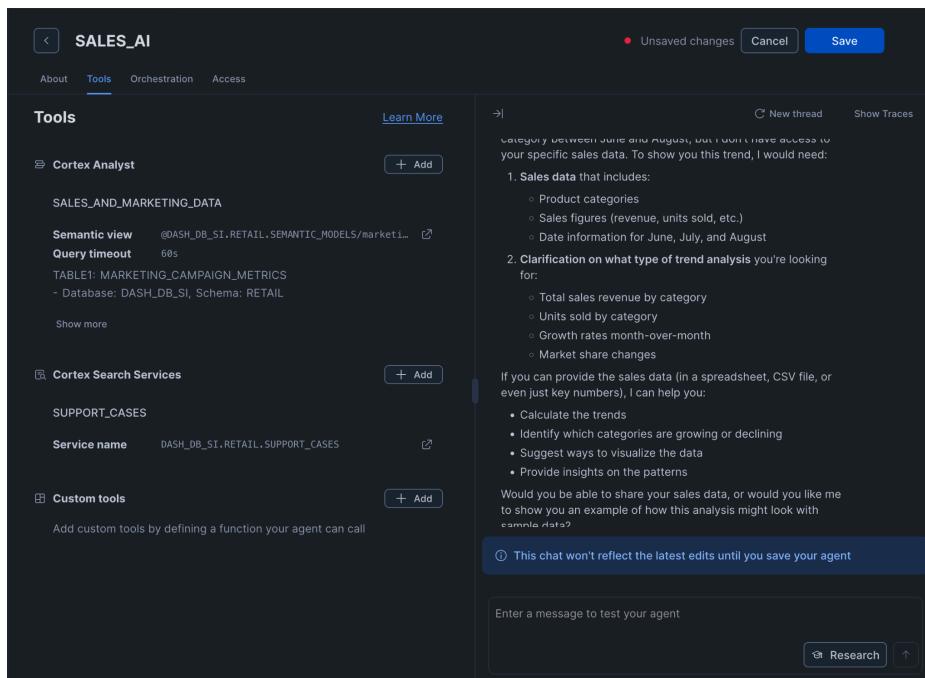
The screenshot shows the 'Cortex Search Services' configuration interface. It displays the message 'No Cortex Search Services added yet' and features a large '+ Add' button.

6. Select DASH\_DB\_SI.RETAIL.SUPPORT\_CASES. Update the "Name" to Support\_Cases, the "ID column" to ID, and the "Title column" to TITLE. Leave all

other settings unchanged and click Add in the bottom right corner.



7. You should now see that under Cortex Search you have the instance configure for the Agent to use

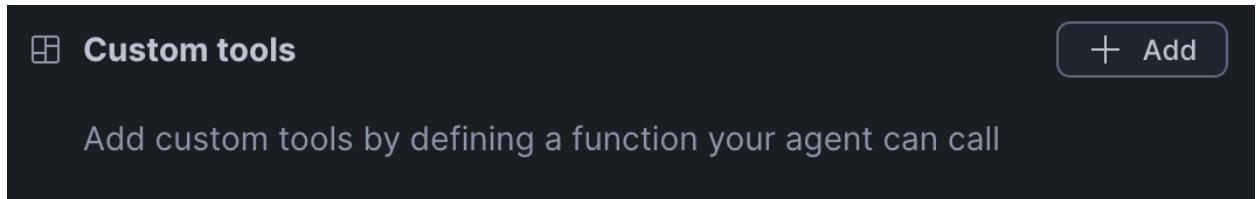


The screenshot shows the 'SALES\_AI' interface with the 'Tools' tab selected. Under 'Cortex Analyst', there is a 'Semantic view' entry pointing to @DASH\_DB\_SI.RETAIL.SEMANTIC\_MODELS/marketing and a 'Query timeout' of 60s. Below it, a table titled 'TABLE1: MARKETING\_CAMPAIGN\_METRICS' is shown, with a note that it's from the Database: DASH\_DB\_SI, Schema: RETAIL. There is a 'Show more' link. Under 'Cortex Search Services', there is a 'Service name' entry for DASH\_DB\_SI.RETAIL.SUPPORT\_CASES. In the sidebar, there is a detailed analysis section:

- Clarification on what type of trend analysis you're looking for:
  - Total sales revenue by category
  - Units sold by category
  - Growth rates month-over-month
  - Market share changes
- If you can provide the sales data (in a spreadsheet, CSV file, or even just key numbers), I can help you:
  - Calculate the trends
  - Identify which categories are growing or declining
  - Suggest ways to visualize the data
  - Provide insights on the patterns

At the bottom, a message input field says 'Enter a message to test your agent' and a 'Research' button is visible.

- Lastly, we are going to add the UDF and UDTF that we created in Module 4 as Custom tools for the AI Agent to use. Click the “+ Add” button.



- Select function as the “Resource type”. Choose DASH\_DB\_SI.RETAIL as the “Database & Schema”, then select the RoundToWhole UDF from Module 4. Name the tool Round\_Float\_To\_Whole and set the “Query timeout” to 60 seconds. Click “Generate with Cortex” to automatically create the “Description”, then click Add in the bottom right corner.

Add custom tool

Tool details	Resource type
	function
Database & Schema	Custom tool identifier
Schema DASH_DB_SI.RETAIL	DASH_DB_SI.RETAIL.ROUNDTOWHOLE(FLO...
Name	Name
	Round_Float_To_Whole
Warehouse	Warehouse to execute the SQL query
User's default	Custom
Query timeout	Maximum time in seconds for query execution (max 300s)
	60
Description	Will be used during orchestration to guide how the agent uses this tool
<b>PROCEDURE/FUNCTION DETAILS:</b> - Type: User-Defined Function (UDF) - Language: Python - Signature: (VALUE FLOAT) - Returns: NUMBER(38,0) - Execution: CALLED ON NULL INPUT with VOLATILE behavior - Volatility: VOLATILE - Primary Function: Mathematical rounding operation	
<a href="#">Learn More</a> <span style="float: right;">Cancel Add</span>	

- Let’s now add other UDTF as a tool as well, so click the “+ Add” button again. Select function as the “Resource type”. Choose DASH\_DB\_SI.RETAIL as the “Database &

Schema", then select the AvgPricePerUnitProductPerSale UDTF from Module 4. Name the tool Calculate\_Avg\_Price\_Per\_Product and set the "Query timeout" to 60 seconds. Click "Generate with Cortex" to automatically create the "Description", then click Add in the bottom right corner.

Add custom tool

Tool details

Resource type

Database & Schema

Schema
DASH\_DB\_SI.RETAIL
×
▼

Custom tool identifier

Name

Warehouse

Warehouse to execute the SQL query

User's default     Custom

Query timeout

Maximum time in seconds for query execution (max 300s)

Description

Will be used during orchestration to guide how the agent uses this tool

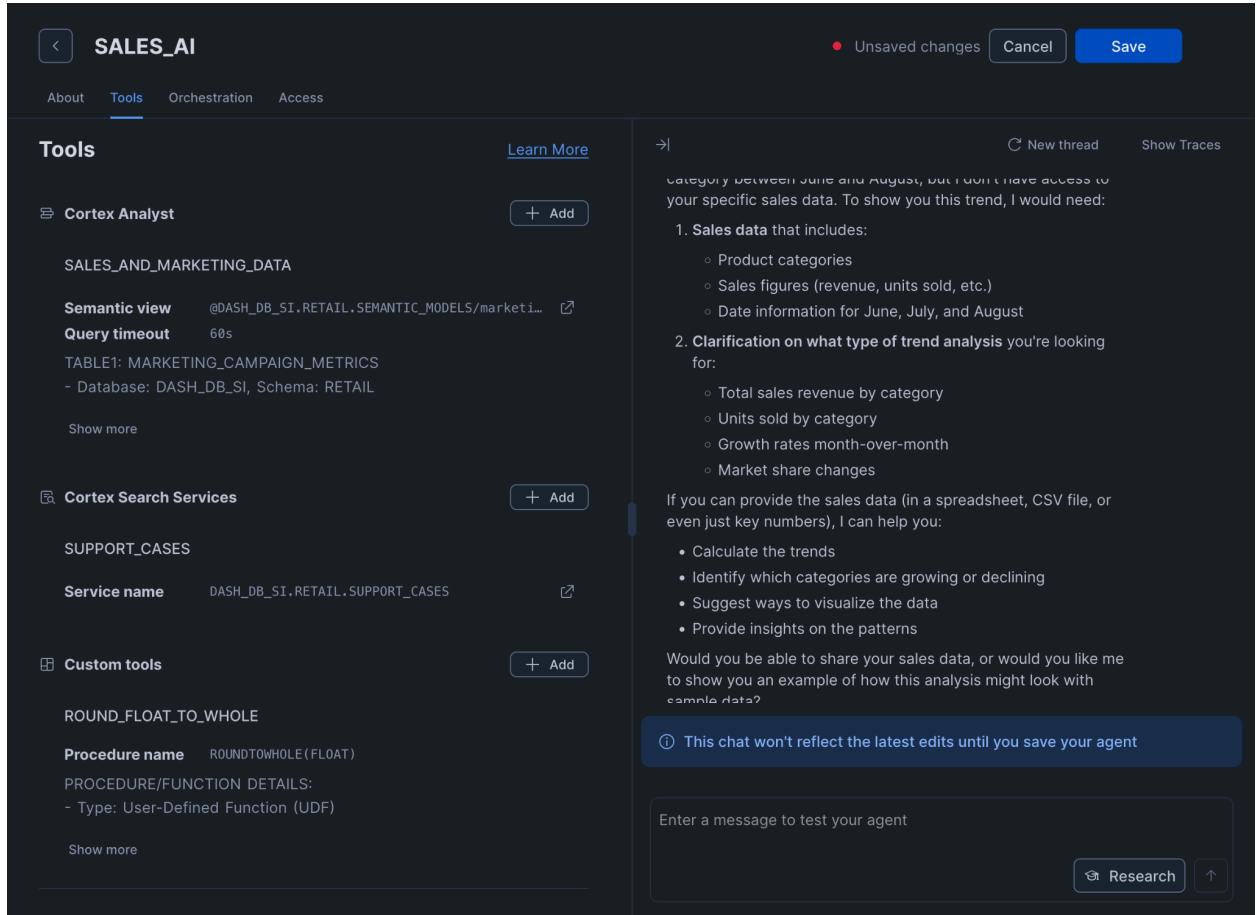
**PROCEDURE/FUNCTION DETAILS:**

- Type: Table-valued function
- Language: SQL
- Signature: ()
- Returns: TABLE (DATE DATE, REGION VARCHAR, PRODUCT\_ID NUMBER, UNITS SOLD NUMBER, SALES\_AMOUNT NUMBER, AVG\_PRICE\_PER\_UNIT NUMBER)
- Execution: Current user context with standard null handling
- Volatility: Stable (depends on underlying SALES table data)

Generate with Cortex

Learn More
Cancel
Add

11. You should now see that under “Custom tools” you have both instance of the UDF/UDTF configure for the Agent to use



The screenshot shows the Snowflake interface for the **SALES\_AI** tool. On the left, there's a sidebar with tabs for **About**, **Tools** (which is selected), **Orchestration**, and **Access**. Under the **Tools** tab, there are three sections: **Cortex Analyst**, **Cortex Search Services**, and **Custom tools**. The **Custom tools** section contains a single entry: **ROUND\_FLOAT\_TO\_WHOLE**. Below it, it says **Procedure name**: `ROUNDTOWHOLE(FL0AT)` and **PROCEDURE/FUNCTION DETAILS:** - Type: User-Defined Function (UDF). There are "Show more" buttons next to each of these sections.

On the right, there's a large text area where an AI agent is interacting with the user. The agent has unsaved changes and offers to **Cancel** or **Save**. It asks for specific sales data between June and August, noting it doesn't have access to the user's specific sales data. It suggests two types of analysis:

- Sales data** that includes:
  - Product categories
  - Sales figures (revenue, units sold, etc.)
  - Date information for June, July, and August
- Clarification on what type of trend analysis** you're looking for:
  - Total sales revenue by category
  - Units sold by category
  - Growth rates month-over-month
  - Market share changes

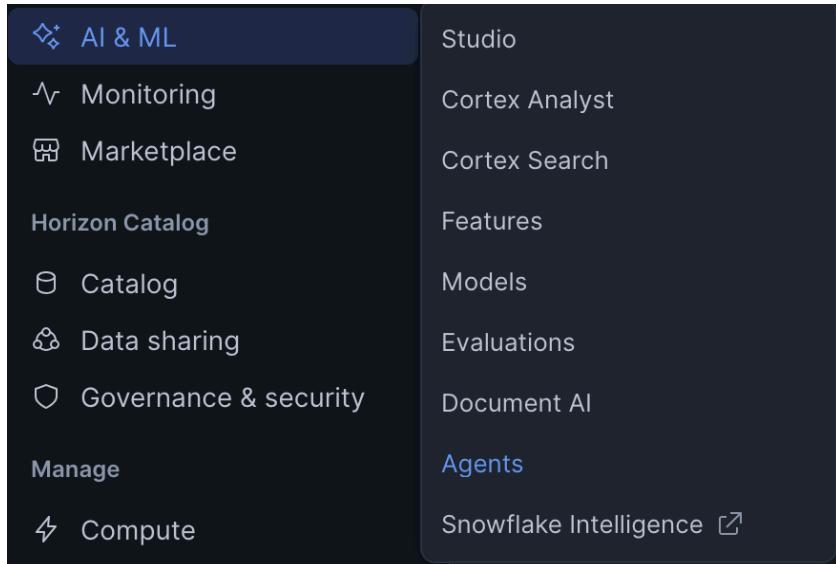
If the user can provide sales data, the agent can help calculate trends, identify growth/decline, suggest visualizations, and provide insights. A note says the chat won't reflect edits until saved. At the bottom, there's a text input field for testing the agent, a **Research** button, and an upward arrow icon.

12. Click the Save button to persist all the tool configurations you've just added.

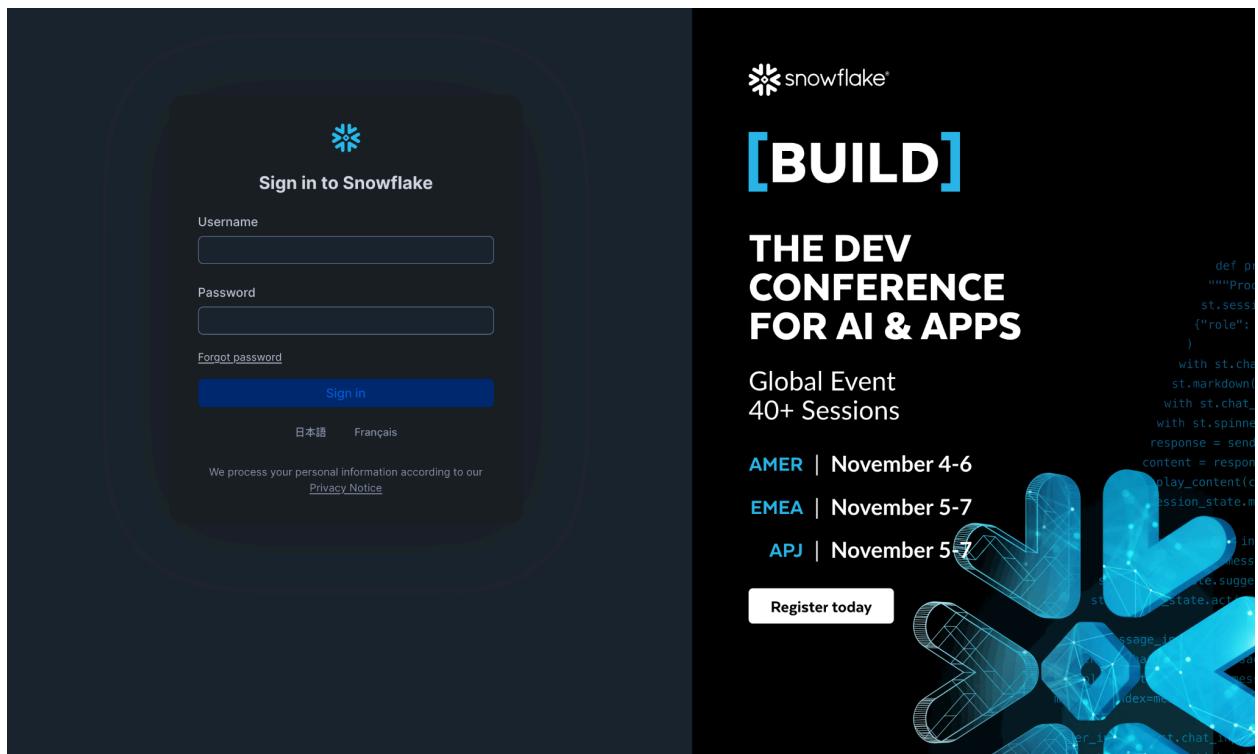
The screenshot shows the Snowflake AI interface. At the top, a modal dialog box displays the message "● Unsaved changes" with two buttons: "Cancel" and "Save". Below the dialog, the main interface shows the "SALES\_AI" workspace. The left sidebar includes navigation links for "Work with data" (Projects, Ingestion, Transformation, AI & ML, Monitoring, Marketplace), "Horizon Catalog" (Catalog, Data sharing, Governance & security), "Manage" (Compute, Admin), and a trial status message ("\$346 credits left", "Trial ends in 19 days", "Upgrade"). The main content area is titled "Tools" and lists three items: "Cortex Analyst", "Cortex Search Services", and "Custom tools". The "Cortex Analyst" section shows a semantic view for "SALES\_AND\_MARKETING\_DATA" with a query timeout of 60s and a table named "TABLE1: MARKETING\_CAMPAIGN\_METRICS". The "Cortex Search Services" section shows a support case named "SUPPORT\_CASES" with a service name "DASH\_DB\_SI.RETAIL.SUPPORT\_CASES". The "Custom tools" section shows a procedure named "ROUND\_FLOAT\_TO\_WHOLE" with a type of "User-Defined Function (UDF)". On the right side of the interface, there is a sidebar with a "SALES\_AI" icon and a message: "This AI Agent can answer all your sales data related questions!". Below this are three input fields with placeholder text: "Show me the trend of sales by product category between Jun...", "What issues are reported with jackets recently in customer su...", and "Why did sales of Fitness Wear grow so much in July?". A text input field at the bottom is labeled "Enter a message to test your agent".

## Using Custom Snowflake Intelligence Agent

1. You're ready to test and utilize the Snowflake Intelligence Agent. Navigate to "AI & ML > Snowflake Intelligence".

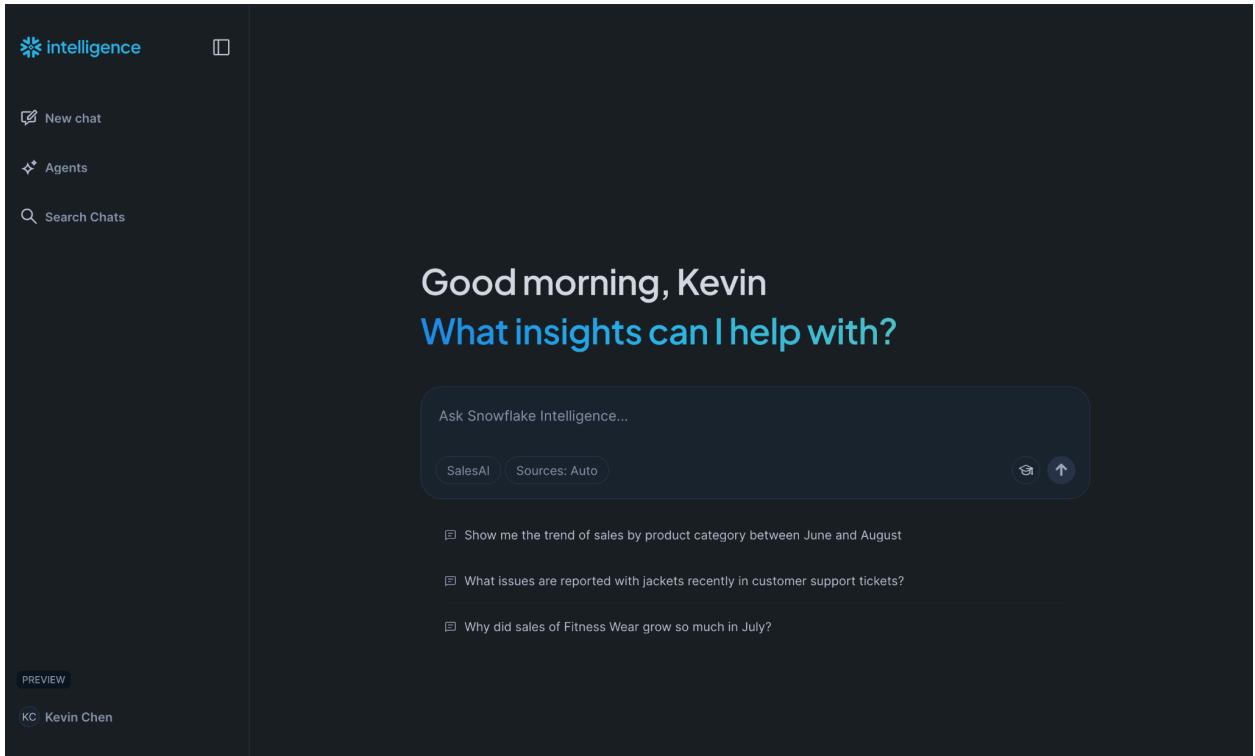


2. This navigates you to the Snowflake Intelligence interface for your account. Sign in using your existing user credentials.



The image consists of two side-by-side screenshots. The left screenshot is a "Sign in to Snowflake" page with a dark background. It features a logo at the top, followed by fields for "Username" and "Password", and links for "Forgot password" and "Sign in". Below these are language options "日本語" and "Français". At the bottom, it says "We process your personal information according to our [Privacy Notice](#)". The right screenshot is for the "[BUILD] THE DEV CONFERENCE FOR AI & APPS" conference. It features the Snowflake logo at the top, followed by the conference name in large blue letters. Below that, it says "Global Event" and "40+ Sessions". It lists three event locations: "AMER | November 4-6", "EMEA | November 5-7", and "APJ | November 5-7". At the bottom, there is a "Register today" button and a decorative graphic of interconnected blue shapes.

3. After logging in, you'll see a familiar chat-like interface similar to standard AI Agent tools. Here you can test the three sample questions you configured and observe the Agent in action.



4. Below are some additional questions that you can use to test the different tools that the AI Agent has access to:
  - a. (Use UDF/UDTF) What is the avg price for all products?
  - b. (Use UDF/UDTF) When does each product sell for the lowest price?
  - c. Which product categories perform best on social media?
  - d. What's the relationship between social media mentions and sales?
  - e. How do different regions respond to marketing campaigns?

## Module 6: Snowflake Badge

### Setup Autograder

1. Follow the instructions in the Autograder setup repo at  
<https://mlh.link/snowflake-autograder>

### Run Autograding Scripts

1. Run the `snowflake-intelligence.sql` script at  
<https://mlh.link/snowflake-autograding-scripts> step by step after you set up the Autograder