



# [BUILD] MEETUP

Connect & Build with Developers Near You

## SNOWFLAKE WORKSHOP



```
n.file  
o_comp  
le.put  
rite=True,  
  
# Add model  
session.add  
('test')  
session.ad  
session.ad  
session.ad  
session.ad  
  
Add Python packages  
session.add_packages('snowflake-snowpark-python',  
'joblib','c  
  
chetools.cache  
loadClass_mapping(  
):
```

# BRO-MANCE



# Agenda

- > Snowflake Overview (15min)
- > Hands-On Lab
  - Snowflake Platform Fundamentals (45min)
  - Cortex Analyst (1h15)
  - Cortex Search Integration (Optional)
  - Snowflake Intelligence (30min)



# Daily Revenue Data Product

```
-- Dimension table: product_dim
product_dim (
    product_id INT,
    product_line VARCHAR
);
```

```
-- Fact table: daily_revenue
daily_revenue (
    date DATE,
    revenue FLOAT,
    cogs FLOAT,
    forecasted_revenue FLOAT,
    product_id INT,
    location_id INT
);
```

```
-- Dimension table: location_dim
location_dim (
    location_id INT,
    sales_region VARCHAR,
    state VARCHAR
);
```

# PRODUCT_ID	A PRODUCT_LINE
1	Electronics
2	Clothing
3	Home Appliances
4	Toys
5	Books

000	DATE	REVENUE	COGS	FORECASTED_REVENUE	PRODUCT_ID	LOCATION_ID
1	2022-11-09	1175.18	2045.42	2438.94	3	1
2	2022-11-10	4297.86	1199.4	2894.6	5	6
3	2022-11-11	4755.99	1202.93	1353.3	3	6
4	2022-11-12	1176.32	1674.98	1058.28	5	1

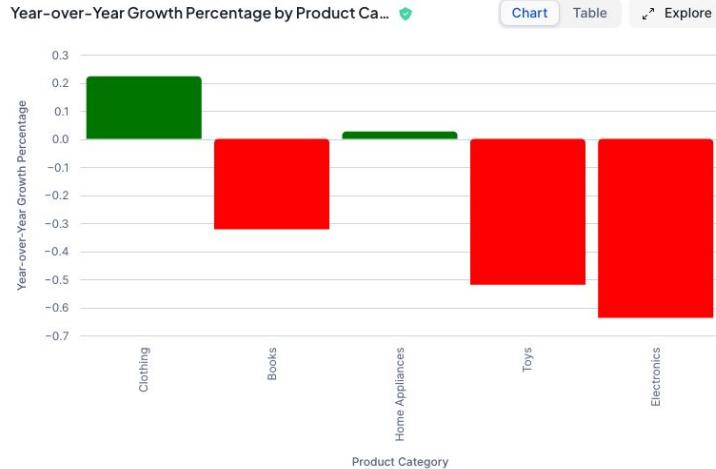
# LOCATION_ID	A SALES_REGION	A STATE
1	North America	New York
2	North America	California
3	North America	Texas
4	North America	Ontario
5	North America	Quebec
6	Europe	England
7	Europe	Germany



# Snowflake Intelligence

Sales revenue for product categories sold in Europe in 2024 & YoY %

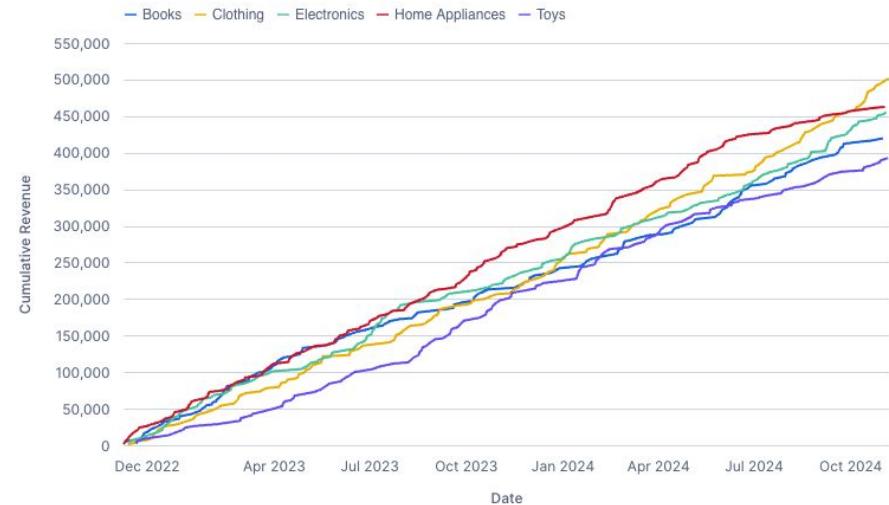
Growth



cumulative revenue per category

Cumulative Revenue Over Time by Product Category

Chart Table Explore



# TRIAL ACCOUNT



© 2025 Snowflake Inc. All Rights Reserved

# Create New Trial Account

<https://signup.snowflake.com/>

Choose your Snowflake edition [Learn more](#)

Enterprise (Most popular) 

Choose your cloud provider  
Snowflake runs on top of the cloud platform you're already working with.

 Microsoft Azure  Amazon Web Services  Google Cloud Platform

Region

EU (Frankfurt) 

I have read and agree to the [Snowflake Self Service On Demand Terms](#).

[Back](#) [Get started](#)



Work with data

- Projects
- Ingestion
- Transformation
- AI & ML
- Monitoring
- Marketplace

Horizon Catalog

- Catalog
- Data sharing
- Governance & security

Manage

- Compute
- Admin

\$400 credits left  ...  
Trial ends in 30 days

[Upgrade](#)

30 days  
\$400 credits



# Workshop

<https://quickstarts.snowflake.com/>

Getting Started with Snowflake Intelligence

30 min

Updated Oct 10, 2025

START

Getting Started with Cortex Analyst:  
Augment BI with AI

47 min

Updated Oct 10, 2025

START

Build a Retrieval Augmented Generation  
(RAG) based LLM assistant using  
Streamlit and Snowflake Cortex Search

86 min

Updated Oct 10, 2025

START

[https://github.com/sfc-gh-qcharriere/workshops/tree/main/cortex\\_analyst](https://github.com/sfc-gh-qcharriere/workshops/tree/main/cortex_analyst)

- Understand the core concepts of the Snowflake Platform
- Understand the core concepts and capabilities of Snowflake Cortex Analyst
- Set up and configure a Cortex Analyst environment
- Create semantic models for their data
- Interact with data using natural language queries
- Generate insights and visualizations through Snowflake Intelligence



# SNOWFLAKE OVERVIEW



© 2025 Snowflake Inc. All Rights Reserved

# WHERE DATA DOES MORE



## AI DATA CLOUD

Unified Platform and Connected Ecosystem



Data  
Engineering



Analytics



AI



Applications &  
Collaboration

Fully Managed | Cross-Cloud | Interoperable | Secure | Governed



© 2025 Snowflake Inc. All Rights Reserved

# Every Organization Struggles with Silos

Silos equate to complexity, higher costs, and security risks

**SNOWFLAKE ELIMINATES SILOS TO  
DELIVER A SINGLE DATA FOUNDATION**



# One Platform. Any Architectural Pattern.

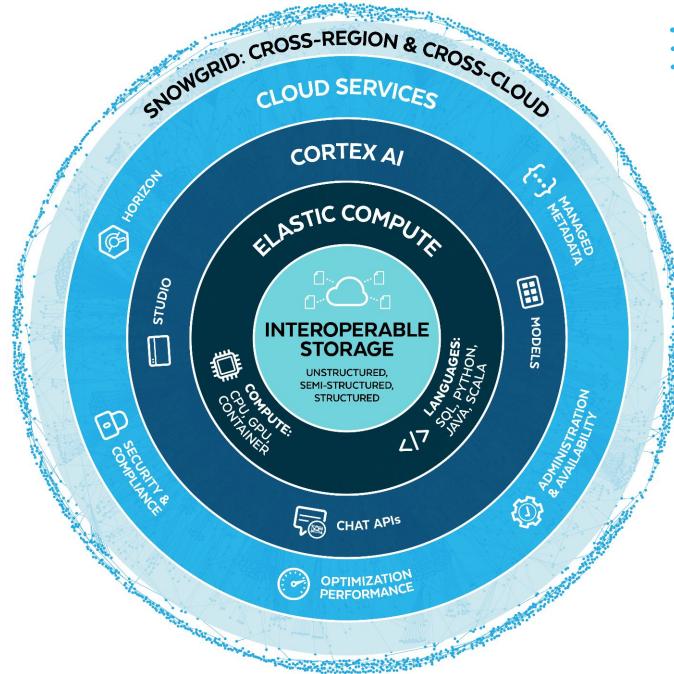
## Data Warehouse

Centralized analytics for structured business data



## Data Lakehouse

Transactional data lake for unified analytics, AI/ML, collaborative workloads



## Data Lake

Unlimited storage for versatile data types and workloads



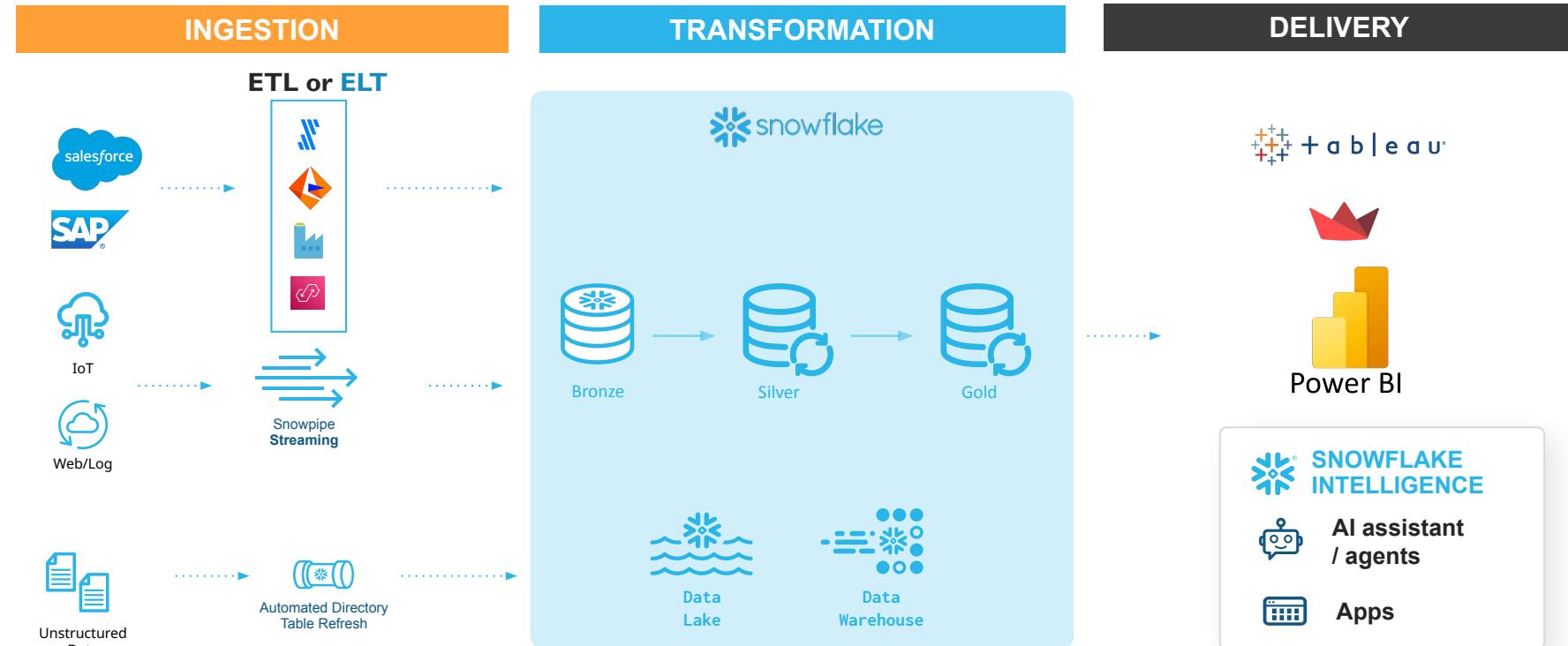
## Data Mesh / Fabric

Distributed and governed, domain-oriented collaboration



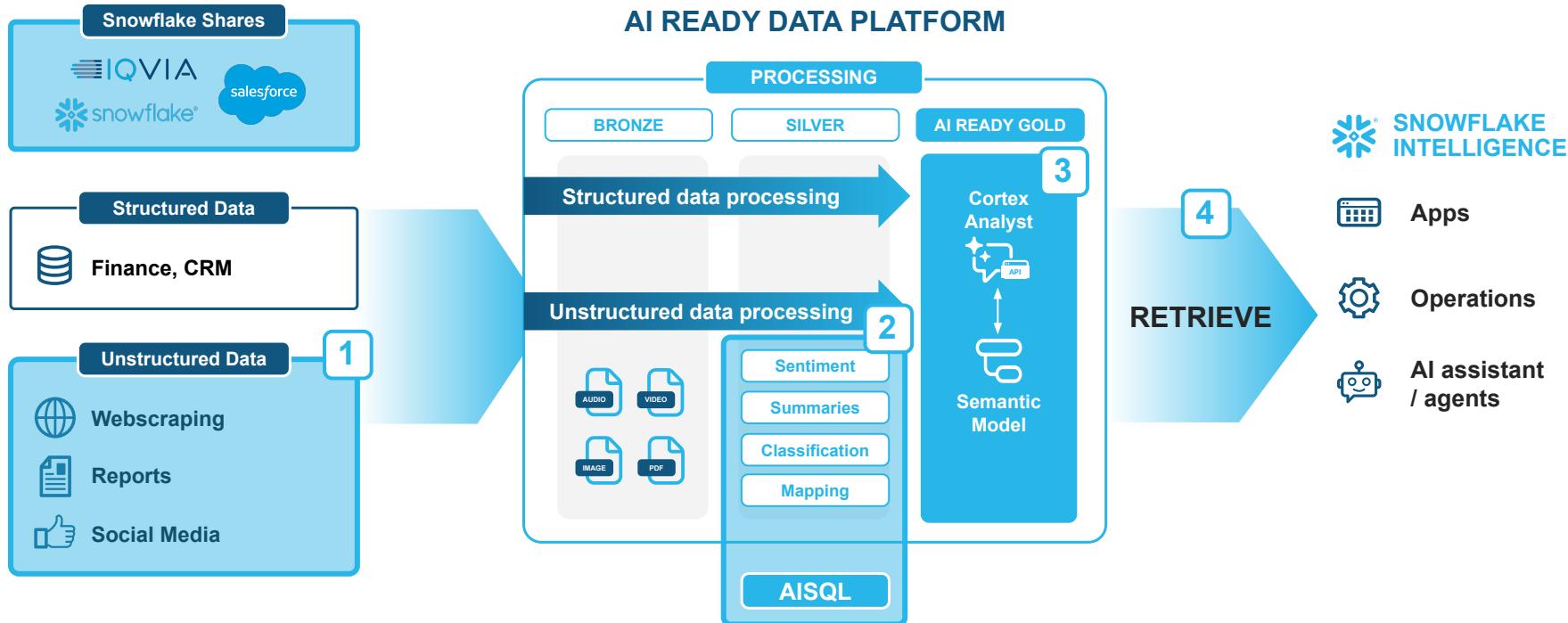
# Snowflake Data Architecture

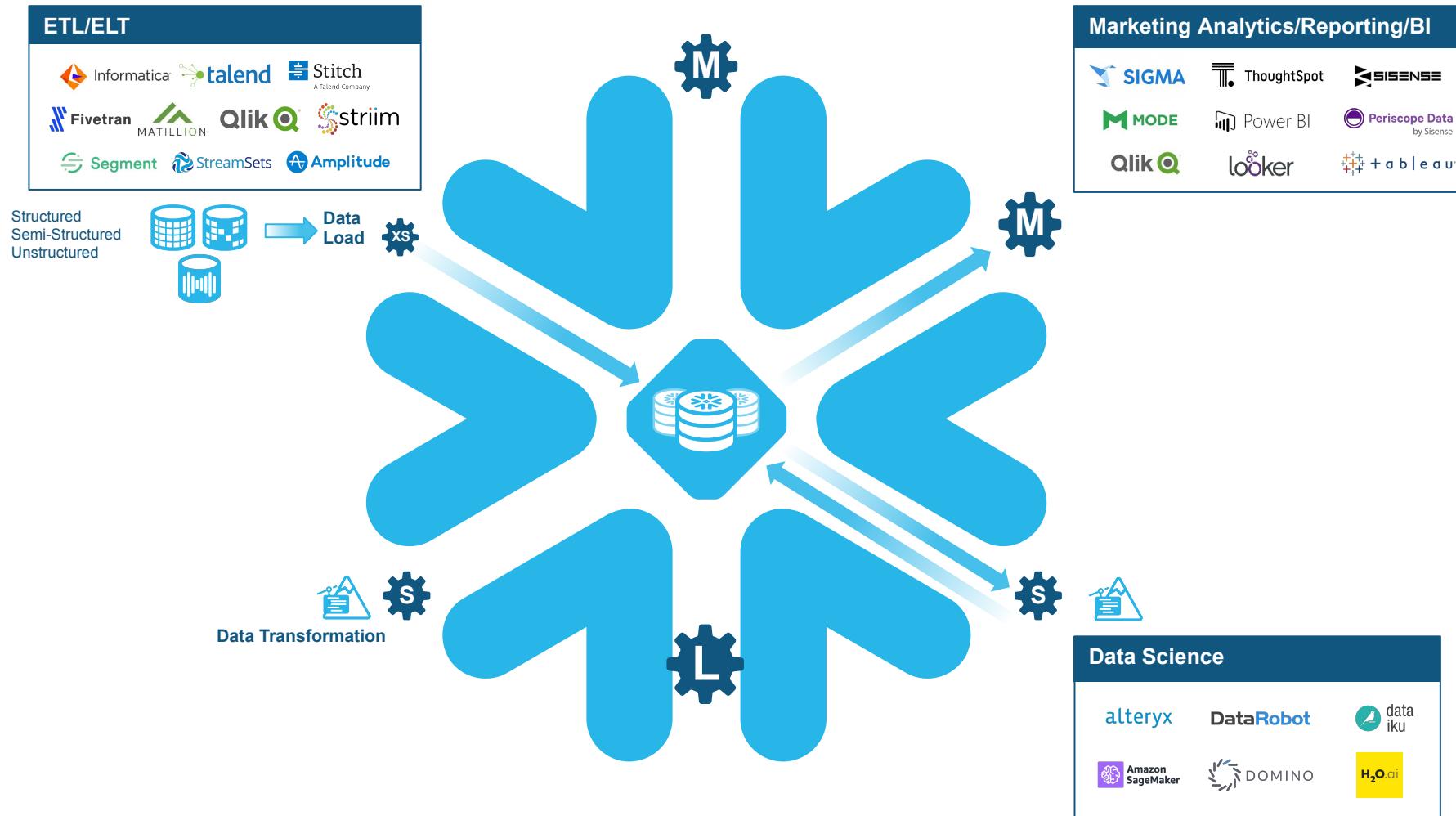
One Platform, Many Workloads

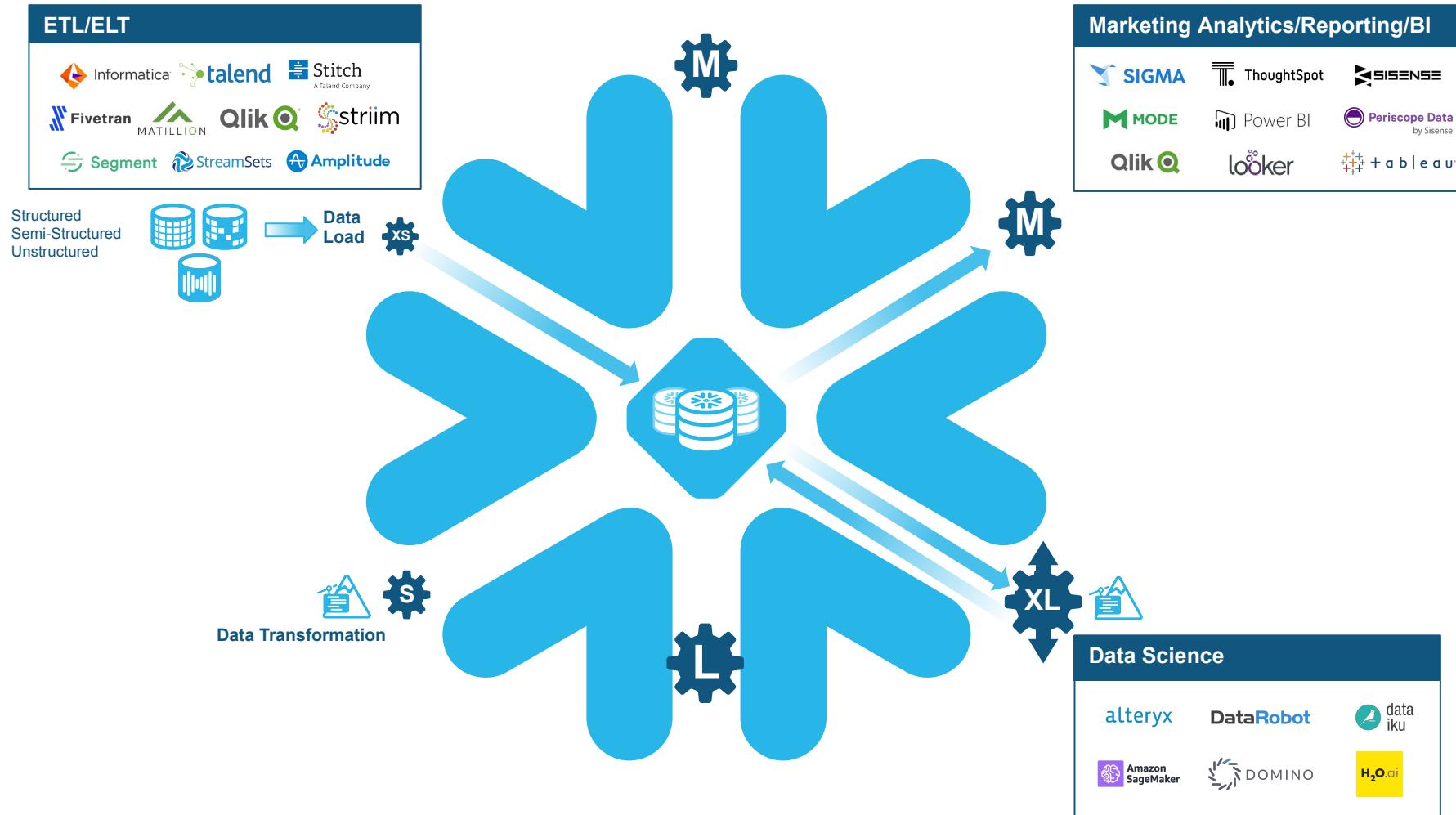


# How to Make Your Data Platform AI Ready: 4 Key Components

Optimized Unstructured Data Ingestion, Unstructured Data Processing Engine, AI Ready Gold Layer & AI Retrieval & Sharing



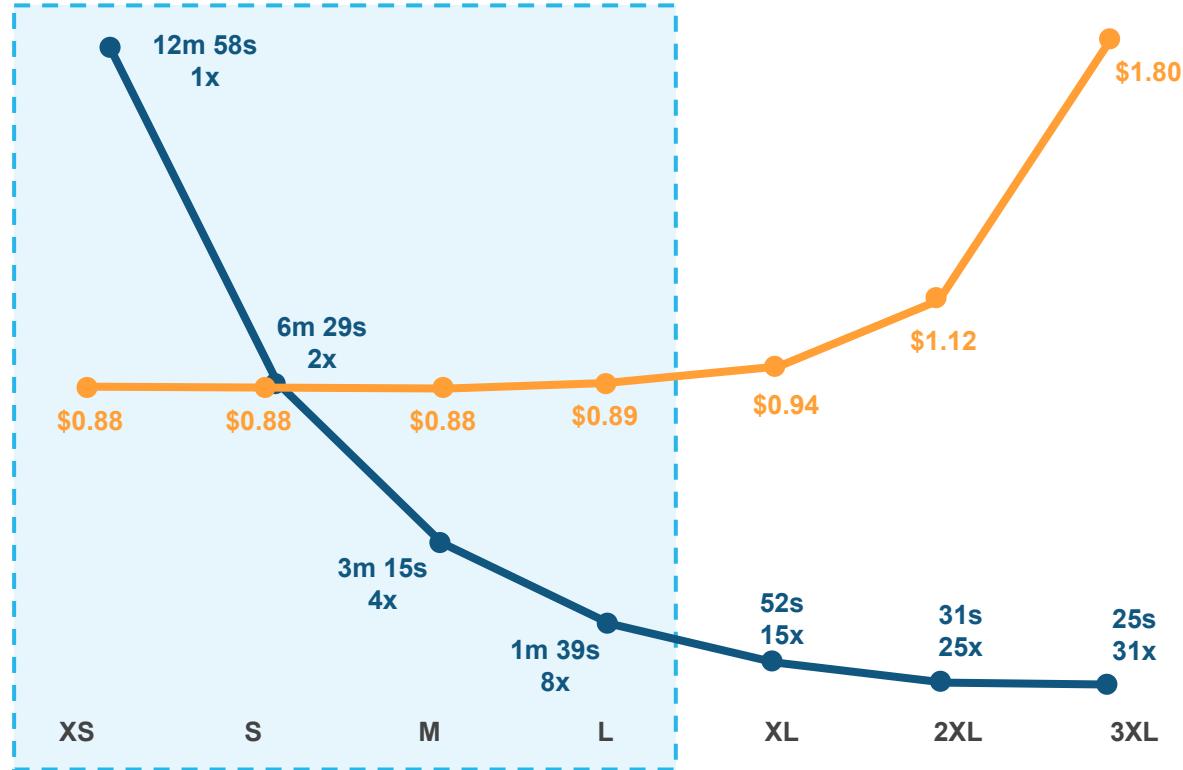




# Scale Up – Loading 1BN Records

- Doubling the number of servers halves the run time
- But you pay per-server, per second of compute
- So you get your answer  
**8X FASTER FOR THE SAME COST**

— Cost  
— Secs

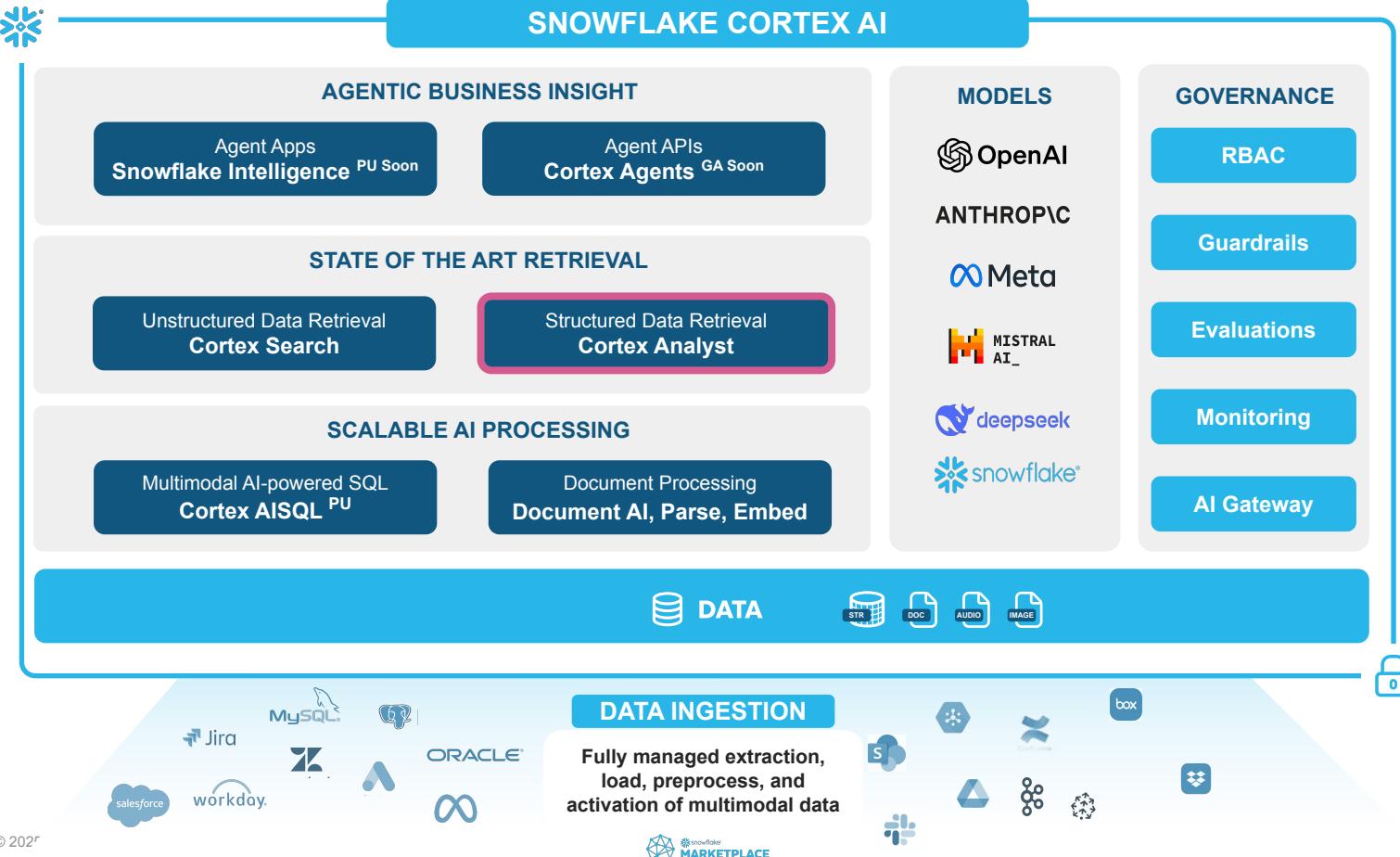


# CORTEX ANALYST



© 2025 Snowflake Inc. All Rights Reserved

# Snowflake Cortex AI is Easy, Connected, Trusted



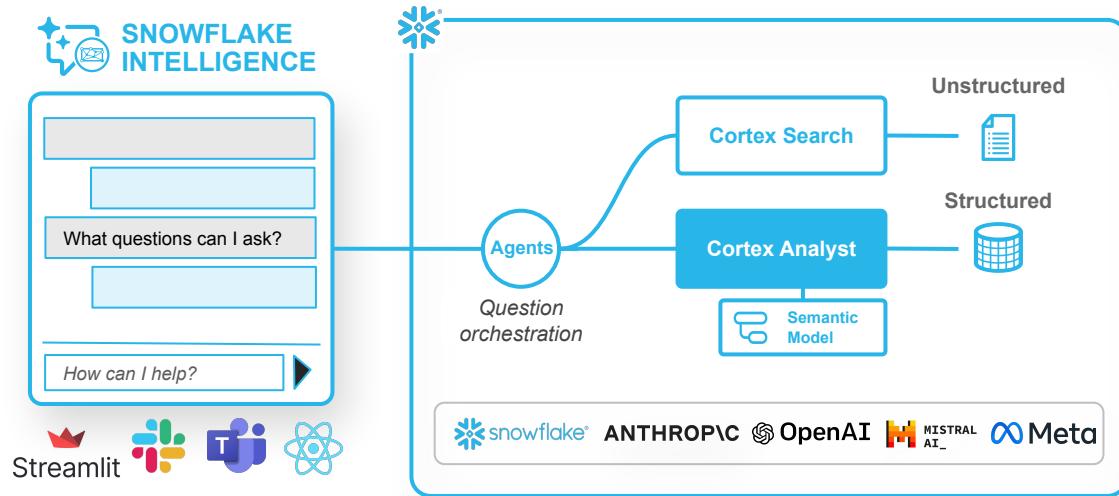
# Cortex Analyst: Self-Service Data Analytics

## Business Users

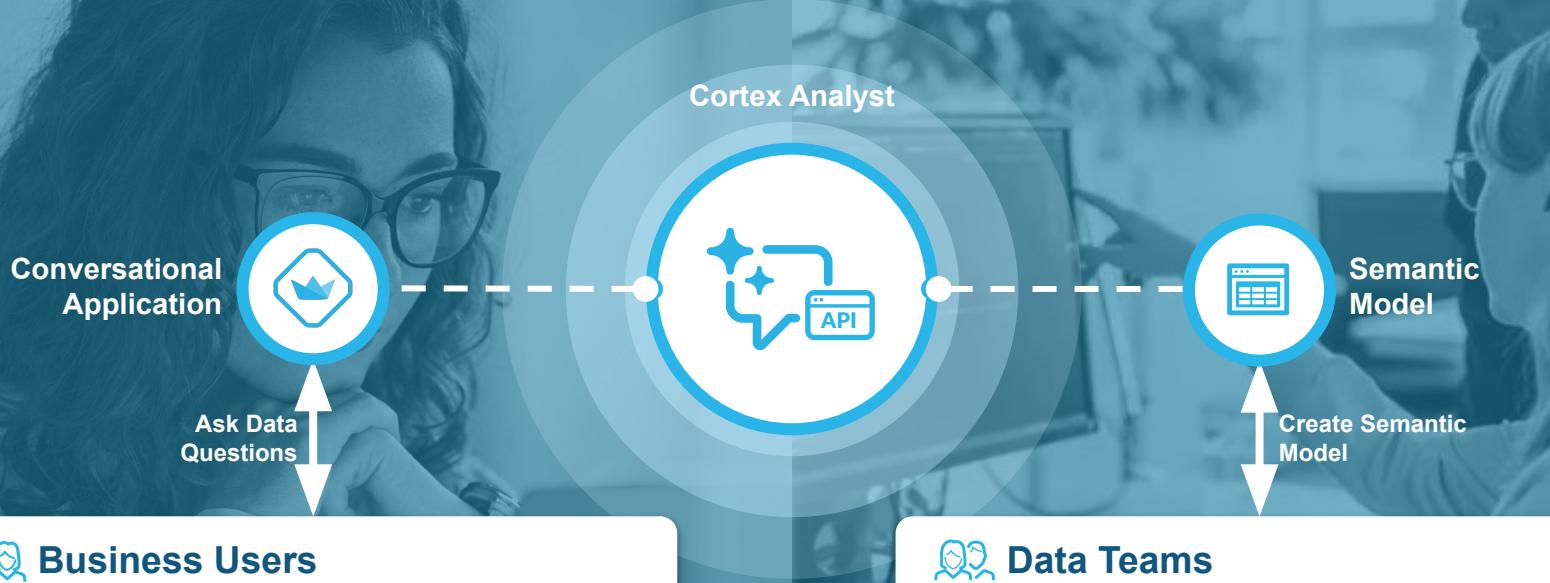
Can answer their data questions via a self-service workflow using a conversational interface. Able to move faster and make data-driven decisions.

## Data Teams

Set up the semantic model and let Cortex Analyst take care of the rest. No longer receive inbound requests, which helps save valuable time.



# Cortex Analyst: Self-Service Data Analytics



## Business Users

Can answer their data questions via a self-service workflow using a conversational interface. Able to move faster and make data-driven decisions.

## Data Teams

Set up the semantic model and let Cortex Analyst take care of the rest. No longer receive inbound requests, which helps save valuable time.

# Increase Precision Using Semantic Model

## Precision

Roughly doubles the Text-to-SQL precision

### Before Semantic Model



Raw schemas are often messy and don't contain the semantic information required for answering data questions.

### With Semantic Model



Our semantic model is a lightweight mechanism to bridge the gap between business users and raw schemas by capturing the missing semantic information that LLMs need to correctly answer user questions.



Provides semantic information about the schema

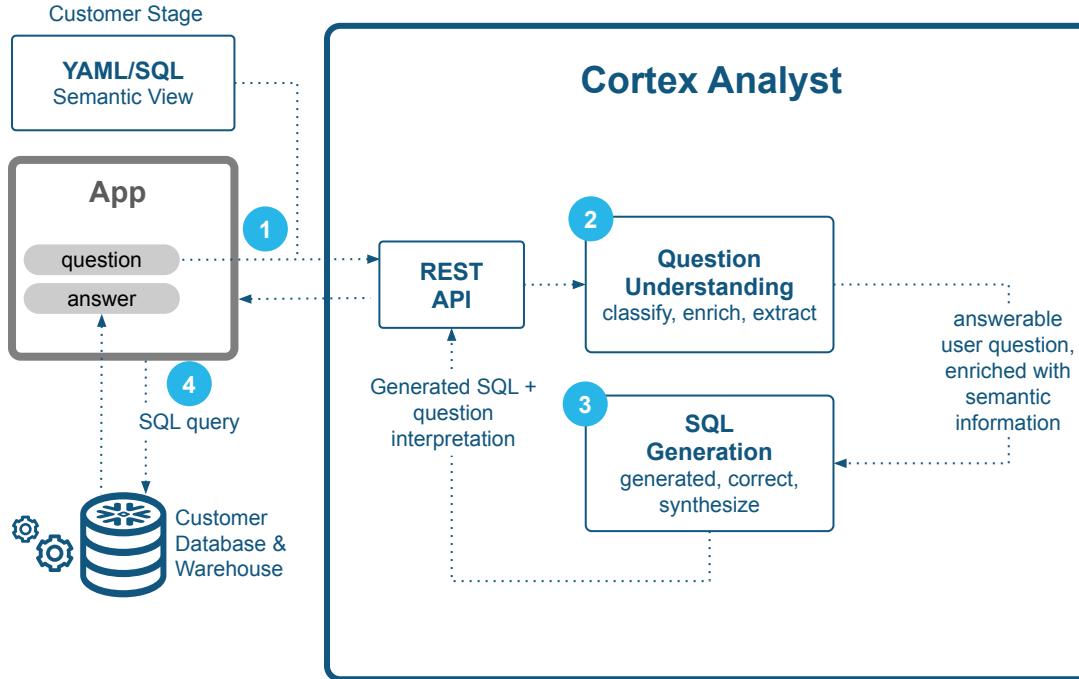
- Descriptions
- Synonyms
- Instructions



Enable some lightweight data modeling

- Expose only relevant columns
- Define common metrics
- Filtering conditions, etc

# How Does Cortex Analyst Work?



## 1 Request

Client app submits user question and Semantic View to Cortex Analyst REST API.

## 2 Question understanding and enrichment

Classification agent determines if question can be answered. If no, suggestions of similar questions are returned. If yes, question is enhanced by two additional agents.

## 3 SQL generation & error correction

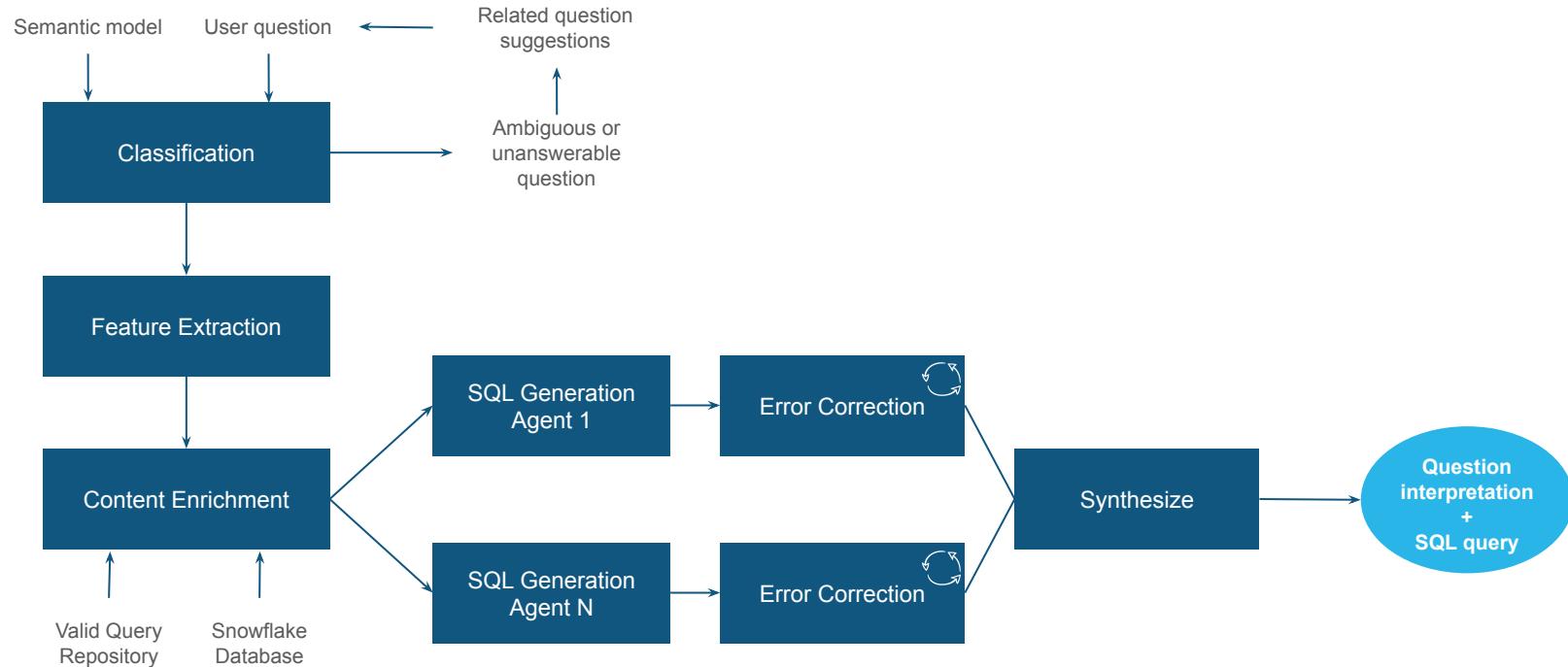
Multiple SQL generation agents run in parallel. For each, the error correction agent checks and corrects SQL. Results are passed to synthesizer agent produces the best SQL query.

## 4 Display response

SQL query and the interpretation of the user question are returned as the API response. This query can be executed or further processed, with the final results presented to the end user.



# Cortex Analyst: Agent Deep Dive



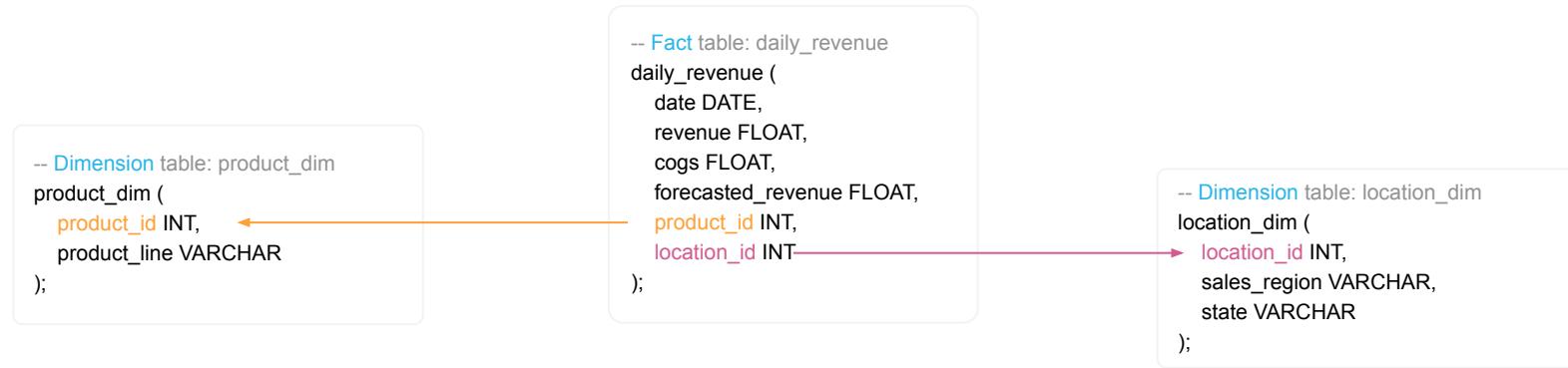
# CORTEX ANALYST HANDS-ON LAB



© 2025 Snowflake Inc. All Rights Reserved

# Set up the Snowflake environment

## Database, Schema, Warehouse, and stage creation



# PRODUCT_ID	▲ PRODUCT_LINE
1	Electronics
2	Clothing
3	Home Appliances
4	Toys
5	Books

000	DATE	REVENUE	COGS	FORECASTED_REVENUE	PRODUCT_ID	LOCATION_ID
1	2022-11-09	1175.18	2045.42	2438.94	3	1
2	2022-11-10	4297.86	1199.4	2894.6	5	6
3	2022-11-11	4755.99	1202.93	1353.3	3	6
4	2022-11-12	1176.32	1674.98	1058.28	5	1

# LOCATION_ID	▲ SALES_REGION	▲ STATE
1	North America	New York
2	North America	California
3	North America	Texas
4	North America	Ontario
5	North America	Quebec
6	Europe	England
7	Europe	Germany



# Set up the Snowflake environment

## Database, Schema, Warehouse

```
/*
• Database, schema, warehouse, and stage creation
--*/  
  
USE ROLE ACCOUNTADMIN;  
  
-- Create demo database  
CREATE OR REPLACE DATABASE cortex_analyst_demo;  
  
-- Create schema  
CREATE OR REPLACE SCHEMA cortex_analyst_demo.revenue_timeseries;  
  
-- Create warehouse  
CREATE OR REPLACE WAREHOUSE cortex_analyst_wh  
WAREHOUSE_SIZE = 'small'  
WAREHOUSE_TYPE = 'standard'  
AUTO_SUSPEND = 60  
AUTO_RESUME = TRUE  
INITIALLY_SUSPENDED = TRUE  
COMMENT = 'Warehouse for Cortex Analyst demo';  
  
-- Create stage for raw data  
CREATE OR REPLACE STAGE raw_data DIRECTORY = (ENABLE = TRUE);
```

```
-- Fact table: daily_revenue  
CREATE OR REPLACE TABLE cortex_analyst_demo.revenue_timeseries.daily_revenue (  
    date DATE,  
    revenue FLOAT,  
    cogs FLOAT,  
    forecasted_revenue FLOAT,  
    product_id INT,  
    region_id INT  
);  
  
-- Dimension table: product_dim  
CREATE OR REPLACE TABLE cortex_analyst_demo.revenue_timeseries.product_dim (  
    product_id INT,  
    product_line VARCHAR  
);  
  
-- Dimension table: region_dim  
CREATE OR REPLACE TABLE cortex_analyst_demo.revenue_timeseries.location_dim (  
    location_id INT,  
    sales_region VARCHAR,  
    state VARCHAR  
);
```



# Ingest the Revenue, Product and Location Data

Load Data Files into Tables

The screenshot shows the Snowflake interface for loading data into tables. On the left, the database navigation pane displays the structure of the 'CORTEX\_ANALYST\_DEMO' database, including 'INFORMATION\_SCHEMA', 'PUBLIC', and 'REVENUE\_TIMESERIES' schemas. Under 'REVENUE\_TIMESERIES', there is a 'Tables' section containing three tables: 'DAILY\_REVENUE', 'LOCATION\_DIM', and 'PRODUCT\_DIM'. The 'DAILY\_REVENUE' table is highlighted with a red box. On the right, the detailed view for the 'DAILY\_REVENUE' table is shown under the 'REVENUE\_TIMESERIES / REVENUE\_TIMESERIES' path. The table details include a 'Table' icon, 'ACCOUNTADMIN' owner, '14 minutes ago' creation time, 0 rows, and 0.0B size. Below the table details are tabs for 'Table Details', 'Columns', 'Data Preview', 'Copy History', 'Data Quality', 'PREVIEW' (which is selected), and 'Lineage'. A large input field for the table's description is present, with the placeholder 'Write a description or [Generate with Cortex](#)'. At the top right of the table detail view are buttons for 'Describe Table', '...', and 'Load Data'.



# CREATE SEMANTIC MODEL

## Snowsight App



# Semantic Model Specification

**SemanticModel**: Collection of logical tables.

- `tables: List[LogicalTable]`

**LogicalTable**: View over a database table or view.

Each logical table is made up of logical columns which are categorized into Dimensions, Time Dimensions and Measure.

- `dimensions: List[Dimension]`
- `time_dimensions: List[TimeDimension]`
- `measures: List[Measure]`
- `filters: List[Filter]`

**Dimension**: A dimension contains categorical values.

**TimeDimension**: A time dimension contains time values.

**Measure**: A measure contains numerical values.

**Filter**: A filter represents a SQL expression used for filtering.

**Verified Queries**: Example questions and queries that answer them.

```
name: Demo
tables:
- name: DAILY_REVENUE
  description: The table contains...
  base_table:
    database: CORTEX_ANALYST_DEMO
    schema: REVENUE_TIMESERIES
    table: DAILY_REVENUE
  dimensions:
    - name: PRODUCT_LINE
      synonyms:
        - product_category
      description: The product line to which a product belongs.
      expr: PRODUCT_LINE
      data_type: TEXT
      cortex_search_service_name: product_line_search_service
  time_dimensions :
    - name: DATE
      synonyms:
        - day
        - calendar_day
      description: The daily revenue date.
      expr: DATE
      data_type: DATE
      sample_values :
        - '2022-11-09'
  measures:
    - name: COGS
      description: Cost of goods sold.
      expr: COGS
      data_type: FLOAT
      sample_values :
        - '2045.42'
```



# Anatomy of a Semantic View

A Semantic View connects business language to data - so AI knows exactly what you mean.

Part	Example	Impact on Accuracy	What It Does
Metadata	name: Customer Orders Model description: Tracks order history	◎	Helps route questions to the right model, especially when you have many.
Tables & Relationships	tables/dimensions: ... Relationships: ...	◎◎	Ensures valid joins, filters, and aggregations without hallucination.
Search Services	search_services: - dimension: customer_name	◎◎◎	Enables fuzzy matching for values like names or venues.
Custom Instructions	custom_instructions: Q1 starts on February 2nd every year	◎◎◎	Adds business rules clarifying information and idiosyncratic SQL.
Verified Queries	verified_queries: - question: ... sql: ...	◎◎◎	Shows correct SQL for real questions. Trains the model to answer similar ones accurately.



# Snowflake AI & ML Studio

## Cortex Analyst - Semantic View

The screenshot shows the AI & ML Studio sidebar with 'Cortex Analyst' selected. Below it, a modal window is open with the title 'Create new Semantic View'. The modal contains three tabs: 'Getting started' (selected), 'Select tables', and 'Select columns'. At the bottom of the modal, there are buttons for 'Cancel' and 'Next'.

**Create new Semantic View**

- Getting started
- Select tables
- Select columns

**Cancel** **Next**

**ACCOUNTADMIN** • COMPUTE\_WH (X-Small)

**Create new** ▾

- Create new Semantic View
- Create new Semantic Model
- Upload your YAML file

Database	CORTEX_ANALYST_DEMO
Schema	REVENUE_TIMESERIES
Name	<b>REVENUE_TIMESERIES</b>

# Snowflake AI & ML Studio

## Cortex Analyst

Create Semantic View

ACCOUNTADMIN • COMPUTE\_WH (X-Small)

Getting started

Select tables

Select only the tables required to answer the business questions your users want to ask.

All Selected 3

Search

CITIBIKE

CORTEX\_ANALYST\_DEMO

- INFORMATION\_SCHEMA
- PUBLIC
- REVENUE\_TIMESERIES
  - 3 Tables
    - DAILY\_REVENUE
    - LOCATION\_DIM
    - PRODUCT\_DIM

SNOWFLAKE

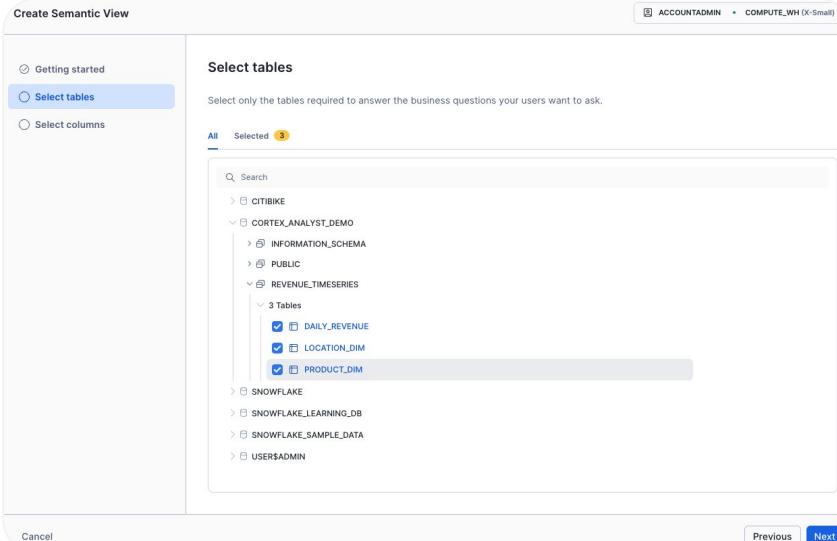
SNOWFLAKE\_LEARNING\_DB

SNOWFLAKE\_SAMPLE\_DATA

USER\$ADMIN

Cancel

Previous Next



Create Semantic View

ACCOUNTADMIN • COMPUTE\_WH (X-Small)

Getting started

Select tables

Select columns

Select only the columns required to answer the business questions your users want to ask.

Columns (11 selected) Expand all

Search columns

CORTEX\_ANALYST\_DEMO.REVENUE\_TIMESERIES.DAILY\_REVENUE

CORTEX\_ANALYST\_DEMO.REVENUE\_TIMESERIES.LOCATION\_DIM

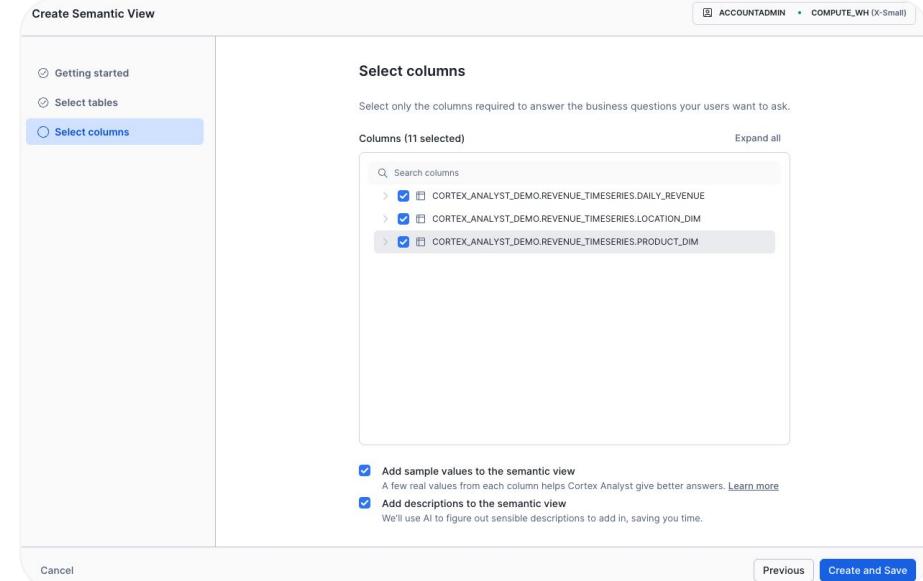
CORTEX\_ANALYST\_DEMO.REVENUE\_TIMESERIES.PRODUCT\_DIM

Add sample values to the semantic view  
A few real values from each column helps Cortex Analyst give better answers. [Learn more](#)

Add descriptions to the semantic view  
We'll use AI to figure out sensible descriptions to add in, saving you time.

Cancel

Previous Create and Save



# Snowflake AI & ML Studio

## Cortex Analyst

<> Edit YAML

Semantic View Monitoring

REVENUE\_TIMESERIES > REVENUE\_TIMESERIES

Edit YML

About Custom Instructions Logical Tables DAILY\_REVENUE PRODUCT\_DIM REGION\_DIM Relationships Verified Queries

REVENUE\_TIMESERIES

Custom Instructions

Logical tables 3

DAILY\_REVENUE

This table stores daily revenue data for a company, including the actual revenue, cost of goods sold (COGS), and forecasted revenue for each product in each region.

Dimensions 2

# PRODUCT\_ID Unique identifier for the product that generated revenue.

PRODUCT\_ID - NUMBER(38,0)

# REGION\_ID Unique identifier for a geographic region where revenue is generated.

REGION\_ID - NUMBER(38,0)

Semantic Model Monitoring

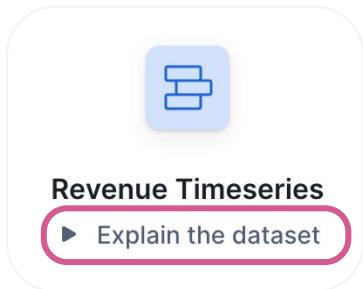
RAW\_DATA > revenue\_timeseries.yaml

Semantic model viewer

```
1 name: Revenue Timeseries
2 description: ''
3 tables:
4   - name: DAILY_REVENUE
5     base_table:
6       database: CORTEX_ANALYST_DEMO
7       schema: REVENUE_TIMESERIES
8       table: DAILY_REVENUE
9     time_dimensions:
10      - name: DATE
11        expr: DATE
12        data_type: DATE
13        sample_values:
14          - '2022-11-09'
15          - '2022-11-10'
16          - '2022-11-11'
17        description: Date of revenue generation, used to track and analyze daily revenue over time.
18      synonyms:
19        - day
20        - calendar_date
21        - timestamp
22        - calendar_day
23        - daily_date
24    measures:
25      - name: COGS
26        expr: COGS
27        data_type: FLOAT
28        sample_values:
29          - '2045.42'
30          - '1199.4'
31          - '1202.93'
32        description: Cost of Goods Sold, representing the direct costs associated with producing and selling a company's products or services.
33      synonyms:
34        - cost of goods sold
```



# Explain the dataset



→



Revenue Timeseries

↶

**Revenue Timeseries**

▶ Explain the dataset

**User**

Explain the dataset

**Cortex Analyst**

Request ID

This semantic data model is designed to help analyze daily revenue, costs, and sales performance across different products and geographic regions. It includes tables for tracking daily revenue and costs, product details, and regional information. You can perform various analyses such as tracking revenue trends over time, comparing sales performance across different product lines or regions, and forecasting future revenue.

What was the total revenue generated last month?

What is the forecasted revenue for the year to date?

What is the total cost of goods sold for the Electronics product line?



# Define Join Columns as Dimensions

The screenshot illustrates the process of defining join columns as dimensions in a data model. It shows three main components: a dimension table, a fact table, and a dimension table with its details expanded.

**Dimension table: product\_dim**

```
-- Dimension table: product_dim  
product_dim (  
    product_id INT,  
    product_line VARCHAR  
)
```

**Fact table: daily\_revenue**

```
-- Fact table: daily_revenue  
daily_revenue (  
    date DATE,  
    revenue FLOAT,  
    cogs FLOAT,  
    forecasted_revenue FLOAT,  
    product_id INT,  
    location_id INT);
```

**Dimension table: location\_dim**

```
-- Dimension table: location_dim  
location_dim (  
    location_id INT,  
    sales_region VARCHAR,  
    state VARCHAR  
)
```

**PRODUCT\_DIM** table details:

- Dimensions: 1
- Time Dimensions: 0
- Facts: 1
  - # PRODUCT\_ID: Unique identifier for a product in the catalog. (Type: NUMBER(38,0))
    - Options: Remove fact, Move to Dimensions (highlighted with a red box), Time Dimensions
  - Named Filters: 0
  - Metrics: 0

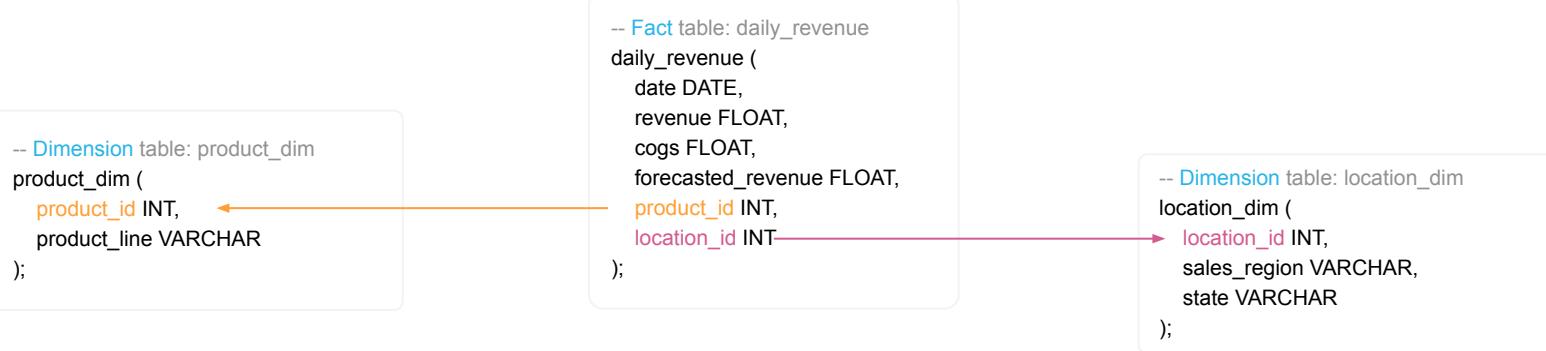
**DIMENSION DETAILS** for # PRODUCT\_ID:

- Cost of Goods Sold (COGS): Cost of Goods Sold (COGS) represents the direct costs associated with producing and selling the company's products or services, such as the cost of raw materials, labor, and overhead, during a specific period of time. (Type: FLOAT)
- FORECASTED\_REVENUE: The forecasted revenue for a given day, representing the predicted amount of money expected to be earned from sales or other sources. (Type: FLOAT)
- LOCATION\_ID: Unique identifier for the physical location where revenue was generated. (Type: NUMBER(38,0))
  - Options: Remove fact, Move to Dimensions (highlighted with a red box), Time Dimensions
- PRODUCT\_ID: Unique identifier for the product that generated revenue. (Type: NUMBER(38,0))
  - Options: Remove fact, Move to Dimensions (highlighted with a red box), Time Dimensions

**Save** button



# Define Relationships



Add relationship

Relationship name:

Left table: DAILY\_REVENUE      Right table: PRODUCT\_DIM

Relationship columns:

Your right column will need to be a primary or unique key for this table. [Define one now](#)

Left column(s): PRODUCT\_ID      Right column(s):  [\(i\)](#) [X](#)

[+ Columns](#)

[Cancel](#) [Add](#)

 Save

Add relationship

Relationship name:

Left table: DAILY\_REVENUE      Right table: LOCATION\_DIM

Relationship columns:

Your right column will need to be a primary or unique key for this table. [Define one now](#)

Left column(s): LOCATION\_ID      Right column(s):  [\(i\)](#) [X](#)

[+ Columns](#)

[Cancel](#) [Add](#)



# Test Join Query

What is the total cost of goods sold for the Electronics product line?

▶ Run



# Verified Query

Sales revenue for product categories sold in Europe in 2024 & YoY % Growth

▶ Run



# Verified Query

Sales revenue for product categories sold in Europe in 2024 & YoY % Growth

Run

Cortex Analyst

This is our interpretation of your question:

Sales revenue for product categories sold in Europe in 2024 & YoY % Growth

PRODUCT_LINE	CURR_YEAR	PREV_YEAR	CURR_REVENUE	PREV_REVENUE
Clothing	2024	2023	53459.93	43711
Books	2024	2023	39770.41	58668.1
Home Appliances	2024	2023	39233.22	38225.8
Toys	2024	2023	25733.4	53582.38
Electronics	2024	2023	19707.94	54379.27

Semantic query · Physical query (executed)

```
WITH __daily_revenue AS (
  SELECT
    location_id,
    date,
    product_id,
    revenue
  FROM
    cortex_analyst_demo.revenue_timeseries.daily_revenue
),
__location_dim AS (
  SELECT
    location_id,
    sales_region
  FROM
    cortex_analyst_demo.revenue_timeseries.location_dim
```

+ Verified query

Run

Question

Sales revenue for product categories sold in Europe in 2024 & YoY % Growth

Cortex Analyst

Semantic query Physical query (executed)

```
1 WITH yearly_revenue AS (
2   SELECT
3     p.product_line,
4     DATE_PART('YEAR', dr.date) AS year,
5     SUM(dr.revenue) AS total_revenue
6   FROM
7     daily_revenue AS dr
8     LEFT OUTER JOIN location_dim AS ld ON dr.location_id =
9       ld.location_id
10    LEFT OUTER JOIN product_dim AS p ON dr.product_id =
11      p.product_id
12    WHERE
13      ld.sales_region = 'Europe'
14      AND DATE_PART('YEAR', dr.date) IN (2023, 2024)
15    GROUP BY
16      p.product_line,
17      DATE_PART('YEAR', dr.date)
18  ),
19  yoy_growth AS (
20    SELECT
21      curr.product_line,
22      curr.year AS curr_year,
```



# Add Customer Metric and Synonyms

Monthly profit in Europe per category

▶ Run



# Add Customer Metric and Synonyms

Monthly profit in Europe per category

▶ Run

This query calculates the monthly profit in Europe per product category over the entire available time period. Profit is calculated as the **difference** between **revenue** and **cost of goods sold (COGS)**. The data is grouped by product category and month, and the results are sorted by month in descending order and then by product category.



# Add Customer Metric and Synonyms

monthly profit in Europe per category

Run

User  
Monthly profit in Europe per category

Cortex Analyst  Request ID ⏪ ⏴

This is our interpretation of your question:

Monthly profit in Europe per product category over the entire available time period

MONTH	PRODUCT_LINE	PROFIT
2024-11-01	Clothing	2632.07
2024-11-01	Home Appliances	434.14

```
SELECT  
    DATE_TRUNC('MONTH', dr.date) AS month,  
    ld.sales_region,  
    pd.product_line,  
    SUM(dr.revenue - dr.cogs) AS profit
```

By default: Profit = Revenue - Cost

Edit metric ...

A metric is an aggregate calculation defined once for consistency. Example: Average Order Value =  $\text{SUM}(\text{order\_amount}) / \text{COUNT}(\text{order\_id})$ . Cortex Analyst reuses this definition whenever you ask.

Expression + Add column

`SUM(REVENUE - (0.01 * REVENUE) - COGS)`

Metric name

Profit

diamond Generate fields

Metric description (optional)

The profit generated from sales after deducting 1% processing fee and cost of goods sold

Synonyms (optional)

earnings, margin, net income

Cancel Save



# Logical Column and Synonyms

Profit in Europe

▶ Run



# Logical Column and Synonyms

## Profit in Europe

▶ Run

```
SELECT
  SUM(
    dr.revenue - (0.01 * dr.revenue) - dr.cogs
  ) AS total_profit
```

◀ ▶ Edit YAML

```
metrics:
  - name: Profit
    synonyms:
      - earnings
      - margin
      - net income
    description: The profit generated from sales after deducting 1% processing fee and cost of goods sold
    expr: SUM(REVENUE - (0.01 * REVENUE) - COGS)
```

ⓘ Profit

Edit ...

The profit generated from sales after deducting 1% processing fee and cost of goods sold

SUM(REVENUE - (0.01 \* REVENUE) - COGS)



# Add Named Filters & Synonyms

List **large** orders

▶ Run



# Add Named Filters & Synonyms

List **large** orders

▶ Run

Apologies, but the question 'List large orders' is **too vague**. It does not specify what constitutes a 'large order' (e.g., based on revenue, profit, or quantity) or any specific criteria for filtering or sorting the orders.

List orders with revenue greater than \$1000

List orders with profit greater than \$500

List the top 10 orders by revenue

List the top 10 orders by profit



# Add Named Filters & Synonyms

Allow additional business terminology to be recognized

list large orders

▶ Run

Cortex Analyst

Request ID

The question 'List large orders' is too vague and does not specify what constitutes a 'large order'. It could refer to orders with high revenue, high profit, or some other measure. Additionally, it does not specify any time period or other criteria for filtering the orders.

List orders with revenue greater than \$1000

List orders with profit greater than \$500

List orders with revenue greater than \$1000 in the last month

List orders with profit greater than \$500 in the region of Texas

## Edit named filter

...

A filter is a saved condition to reuse in queries. Example: Defining active\_customers as status = 'active' and user\_type = 'Customer' lets you ask things like, "Average spend for active customers?" and Cortex Analyst will use this filter more deterministically

Expression ⓘ

+ Add column

REVENUE >= 1150

Filter name

large\_order

Generate fields

Filter description (optional)

Orders with revenue equal to or greater than 1150

Synonyms (optional) ⓘ

big orders, high value orders, large orders

Cancel

Save



# Add Named Filters & Synonyms

Sales in **DAPS** region?

▶ Run



# Add Named Filters & Synonyms

Sales in **DAPS** region?

▶ Run

I apologize, but the question 'What are the sales in the DAPS region?' is **unclear** because '**DAPS** region' is not a defined region in the provided schema. The schema includes regions identified by region\_id and sales\_region, but 'DAPS' does not match any of the provided values.

What are the sales in the North America region?

What are the sales in the Europe region?

What are the sales in the Asia region?



# Add Named Filters & Synonyms

Allow additional business terminology to be recognized

sales in DAPS region

▶ Run

Cortex Analyst

Request ID

The question is unclear because 'DAPS region' is not defined in the given schema. The schema includes regions identified by 'region\_id' and 'sales\_region', but 'DAPS region' does not match any of these identifiers.

What are the sales in the Europe region?

What are the sales in the North America region?

What are the sales in the Asia region?

Named Filters 1 +

Edit named filter

Expression ⓘ + Add column

STATE in ('Germany', 'Austria', 'Poland', 'Switzerland')

Filter name

DAPS

Generate Fields

Filter description (optional)

A filter to include only countries within the DAPS region

Synonyms (optional)

Enter synonyms

Cancel Save



# Add Named Filters & Synonyms

Sales in **DAPS** region?

▶ Run

What are the sales in the **DAPS** region over the entire available time period?

WHERE  
r.state IN ('Germany', 'Austria', 'Poland', 'Switzerland')

<> Edit YAML

filters:  
- expr: STATE in ('Germany', 'Austria', 'Poland', 'Switzerland')  
name: DAPS  
description: DAPS region

Named Filters 1 +

# DAPS

"Germany, Austria, Poland, Switzerland"

Edit

...

DAPS region

STATE in ('Germany', 'Austria', 'Poland', 'Switzerland')



# Custom Instructions



# Custom Instructions in the Classifier

Expand the types of instructions you can give Cortex Analyst

**Before:** Custom Instructions ONLY for SQL Generation

- Tell Cortex Analyst how to generate correct SQL
- Eg: “The financial year starts Feb 1st”

**Now:** Custom Instructions for Classification AND SQL Generation

- Additionally tell Cortex Analyst how to classify SQL. Customers can specify how to reject or disambiguate.
- REJECT example: “Reject questions relating to guns and violence”
- DISAMBIGUATE example: “Suggest looking at data in the last month when the question doesn’t specify a time range”



# Custom Instructions

Expand the types of instructions you can give Cortex Analyst

Ensure that all numeric columns are rounded to 1 decimal point in the output.

For any percentage or rate calculation, multiply the result by 100.

If no date filter is provided, apply a filter for the last year.

## Percentage of Revenue per Region.

Cortex Analyst

This is our interpretation of your question:

Over Last Year.

Request ID

What is the percentage of revenue per region over the last year?

SALES_REGION	TOTAL_REVENUE	REVENUE_PERCENTAGE
North America	18104.7	23.4
Europe	16252.9	21
South America	15540.1	20.1
Asia	15154.9	19.6
Africa	12203.4	15.8

Rounded & Percentage!

```
ROUND(
    (rr.total_revenue / NULLIF(tr.overall_revenue, 0)) * 100,
    1
) AS revenue_percentage
```

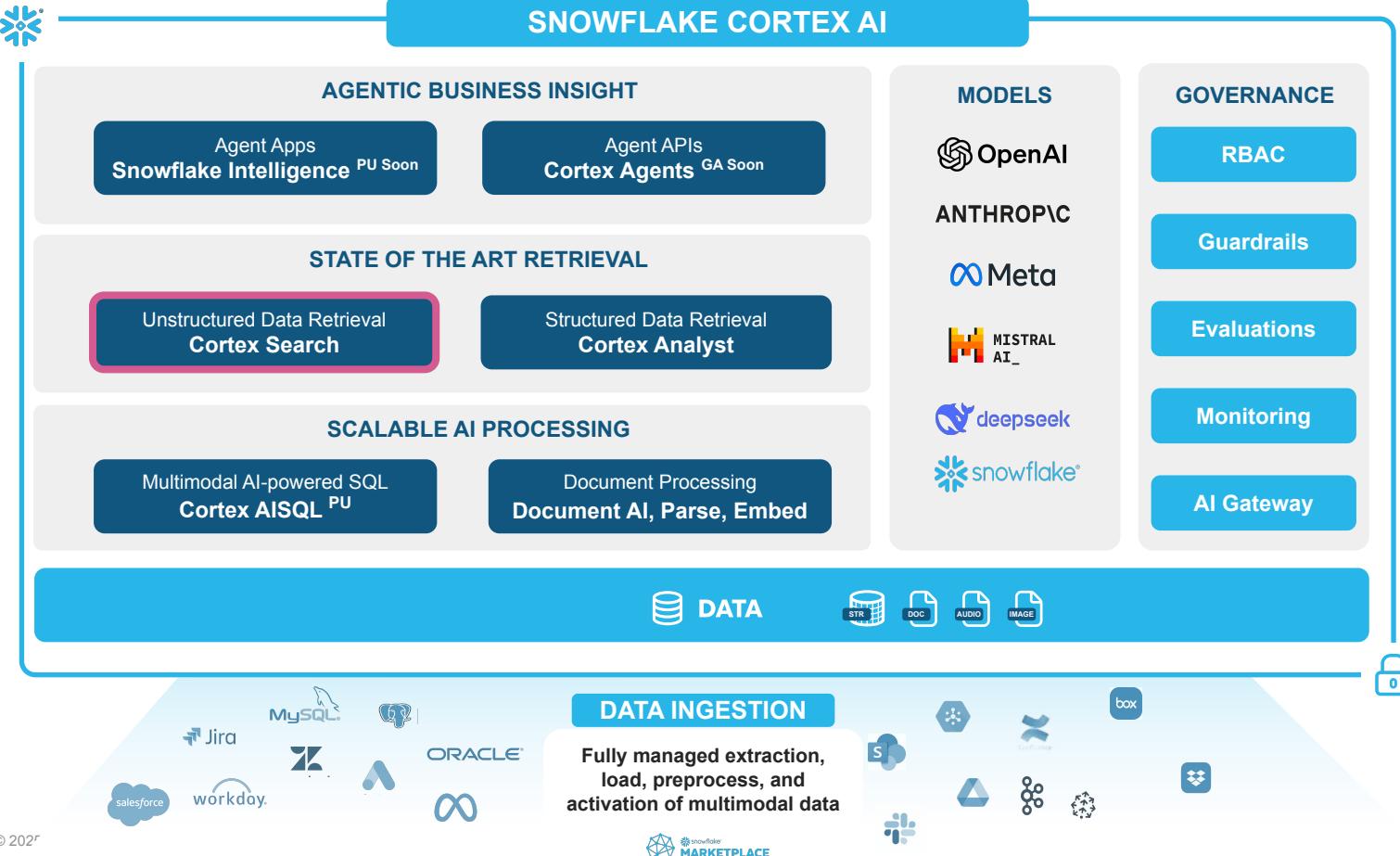


# INTEGRATE CORTEX SEARCH



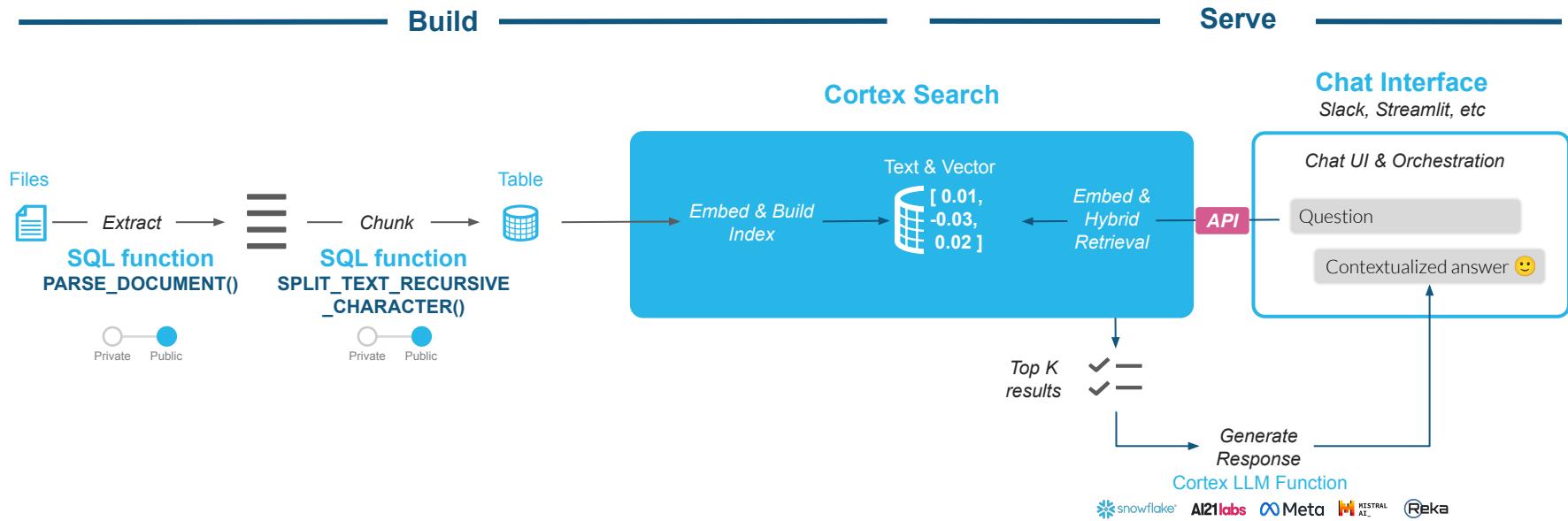
© 2025 Snowflake Inc. All Rights Reserved

# Snowflake Cortex AI is Easy, Connected, Trusted



# RAG Evolution in Snowflake

Accelerate End-to-End RAG Development in Snowflake



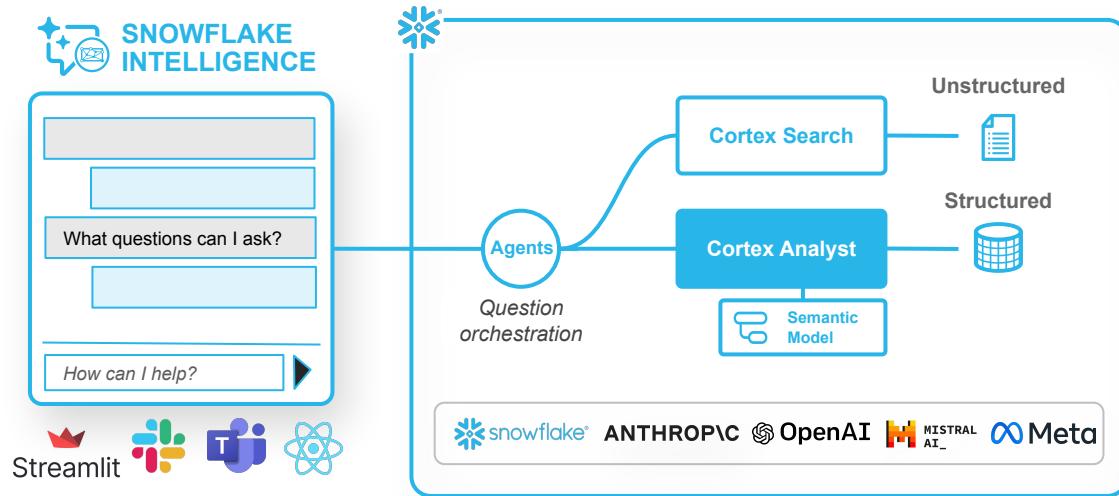
# Cortex Analyst: Self-Service Data Analytics

## Business Users

Can answer their data questions via a self-service workflow using a conversational interface. Able to move faster and make data-driven decisions.

## Data Teams

Set up the semantic model and let Cortex Analyst take care of the rest. No longer receive inbound requests, which helps save valuable time.



# Handling High Cardinality

Dealing with high-cardinality columns, such as proper nouns, presents a unique challenge. Cortex Analyst addresses this with the integration of Cortex Search, Hybrid Search (Vector + Keyword): Distinct proper nouns (e.g., product names, categories, customer names, ...) are stored as embeddings in a **vector store**.

1. Spelling Validation: Approximate names are matched against the vector store to find the closest entities.
2. Agent Integration: The agent dynamically validates proper nouns before constructing SQL queries.

**What it is:** Integration with Cortex Search allows for **dynamic retrieval of exact or similar values** for high-cardinality data fields, improving SQL query accuracy.

**Why use it:** This **minimizes data duplication**, keeps the **semantic model lean**, and **enhances SQL accuracy** for dimensions with frequently changing or complex data values.

**How to use it:** Configure Cortex Analyst to **connect with the Cortex Search service** defined on desired dimensions.

```
dimensions:  
  - name: PRODUCT_LINE  
    description: The category or classification of the product.  
    expr: PRODUCT_LINE  
    data_type: VARCHAR(16777216)  
    sample_values:  
      - Electronics  
      - Clothing  
      - Home Appliances  
cortex_search_service:  
  database: CORTEX_ANALYST_DEMO  
  schema: REVENUE_TIMESERIES  
  service: PRODUCT_LINE_SEARCH_SERVICE
```



# Creating a Cortex Search Service in a Single SQL Command

The screenshot shows the Snowflake UI interface. On the left is a sidebar with various icons. The main area is a code editor with the following SQL command:

```
SQL ▾ as create_cortex_search_service •  
1 CREATE CORTEX SEARCH SERVICE my_svc  
2 ON transcript_text  
3 ATTRIBUTES call_timestamp  
4 WAREHOUSE = my_wh  
5 TARGET_LAG = '1 minute'  
6 AS (  
7     SELECT  
8         transcript_text  
9             , call_timestamp  
10        FROM transcripts  
11    );
```

Annotations with colored boxes point to specific parts of the code:

- Search column**: Text column you want to search over. Points to the `ON transcript_text` line.
- Metadata filter columns**: Define attributes to optionally filter search queries on (e.g., date). Points to the `ATTRIBUTES call_timestamp` line.
- Indexing configuration**: Define warehouse to use to build index, target freshness lag. Points to the `WAREHOUSE = my_wh` and `TARGET_LAG = '1 minute'` lines.
- Indexed data**: Define the indexed data source declaratively with a SQL query. Points to the `AS` block containing the `SELECT` statement.

The result is a continuously, incrementally updated search index that handles all complexity around vector embeddings, source data updates, and serving infrastructure



# Without Cortex Search

Sales revenue for product line having **clothes**

▶ Run



# Without Cortex Search

Sales revenue for product line having **clothes**

▶ Run

What is the sales revenue for the product line that includes **clothes** over the entire available time period?

Sample values	Fetch	Value
Electronics	C	X
Clothing	+ Value	X
Home Appliances		X

WHERE  
pd.product\_line = 'Clothing'



# Without Cortex Search

Sales revenue in **book** product line

▶ Run



# Without Cortex Search

Sales revenue in **book** product line

▶ Run

The schema does not provide information about a '**book** product line', and it is not clear which product line the user is referring to.

Sample values	C Fetch	+	Value
Electronics	X		
Clothing	X		
Home Appliances	X		

What is the total sales revenue for the product line 'Book' over the entire available time period?

START_DATE	END_DATE	TOTAL_REVENUE
null	null	null



# Cortex Search Service

Enables runtime string retrieval and enhances literal usage in SQL queries, optimizing high-cardinality or dynamic dimensions.

```
CREATE OR REPLACE CORTEX SEARCH SERVICE
product_line_search_service
ON product_dimension
WAREHOUSE = cortex_analyst_wh
TARGET_LAG = '1 hour'
AS (
    SELECT DISTINCT product_line AS product_dimension
    FROM product_dim
);
```

Connect Cortex Search (optional) ⓘ

+ Search Service

Sample values

Fetch	Value
Electronics	x
Clothing	x
Home Appliances	x

Connect Cortex Search Service

CORTEX\_ANALYST\_DEMO.REVENUE\_TIMESERIES

Search Service

PRODUCT\_LINE\_SEARCH\_SERVICE

Cancel Connect



# With Cortex Search

Overall sales revenue in **book** and **elec cat**

▶ Run



# With Cortex Search

Overall sales revenue in **book** and **elec cat**

▶ Run

What is the overall sales revenue for the product lines that include **books** and **electronics** over the entire available time period?

```
WHERE  
    p.product_line IN ('Books', 'Electronics')
```

It is not using LIKE + % to find a partial match but using **Vector search** to locate the actual names!

Allows users to make mistakes & not have to be precise with their search criteria & still get proper results.

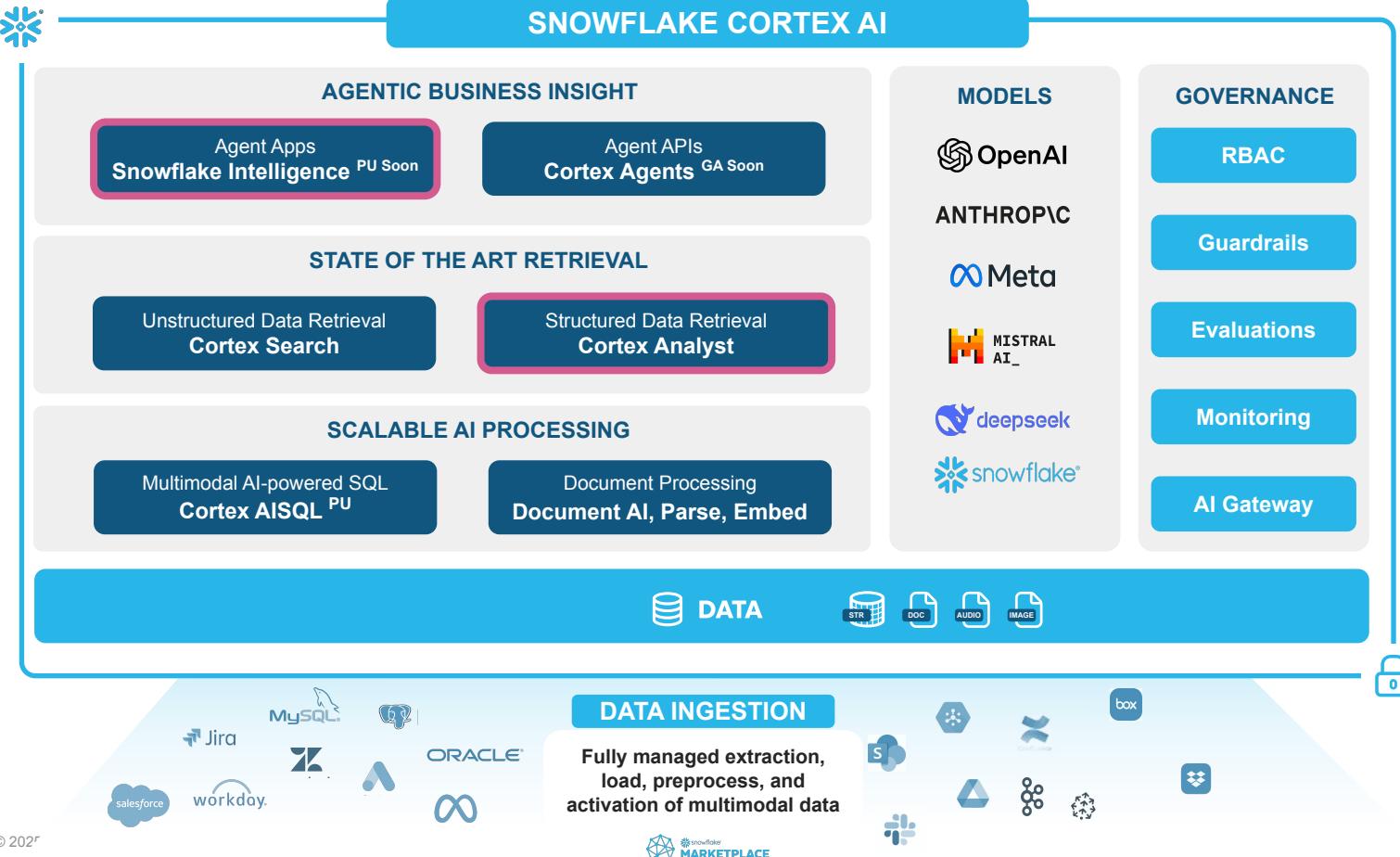


# SNOWFLAKE INTELLIGENCE



© 2025 Snowflake Inc. All Rights Reserved

# Snowflake Cortex AI is Easy, Connected, Trusted



# Introducing Snowflake Intelligence

Deep research on your enterprise data

## Agentic Deep Research

- Natural language interface to get insights from **structured & unstructured** data
- Integrated with Snowflake **governance** and access controls
- Enable **business users** to **self-serve** answers freeing data teams
- **No dashboards or SQL** needed

The screenshot shows the Snowflake Intelligence web application. At the top left is the 'intelligence' logo. On the left is a sidebar with icons and text for Home, Agents, Projects, and Recent. At the bottom of the sidebar are Help & support and a user profile for Jennifer Reynolds. The main area has a greeting 'Good afternoon, Jennifer.' and a question input field 'How may I assist you?'. Below are four example questions:

- Q Are there any leads I need to follow up with today?
- Q What use cases have not been updated?
- Q What's the email and phone number for the account?
- Q Has this lead responded to any outreach recently?



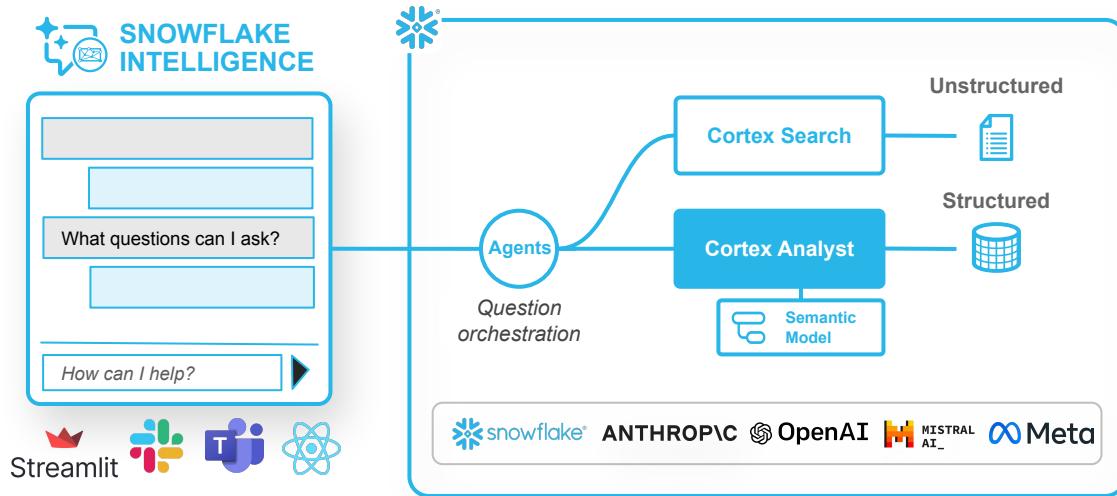
# Cortex Analyst: Self-Service Data Analytics

## Business Users

Can answer their data questions via a self-service workflow using a conversational interface. Able to move faster and make data-driven decisions.

## Data Teams

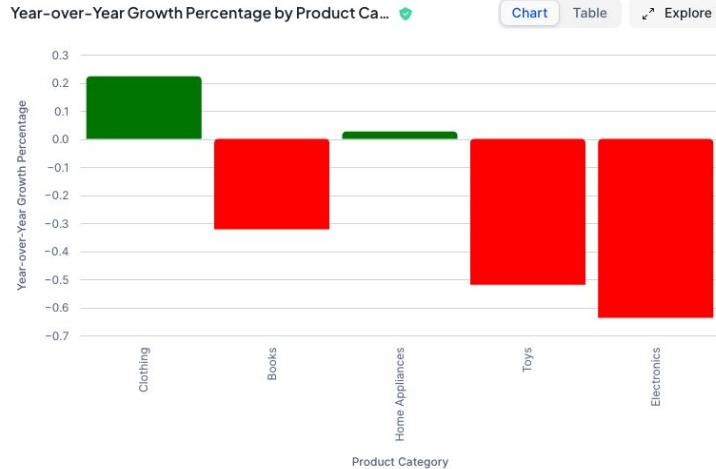
Set up the semantic model and let Cortex Analyst take care of the rest. No longer receive inbound requests, which helps save valuable time.



# Snowflake Intelligence

Sales revenue for product categories sold in Europe in 2024 & YoY %

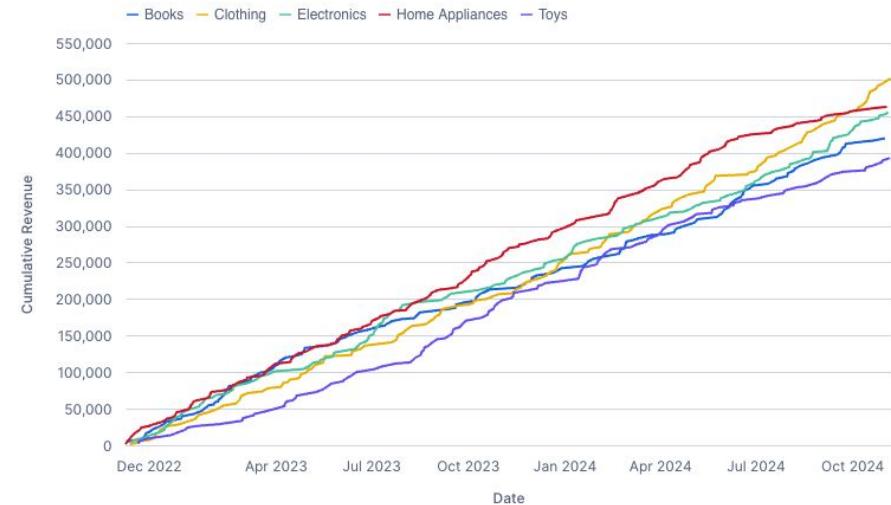
Growth



cumulative revenue per category

Cumulative Revenue Over Time by Product Category

Chart Table Explore



# Snowflake Cortex AI is Connected



## DATA

Multimodal data from all systems of records



## MODELS

No-code interfaces, simple SQL & APIs



## TOOLS & AGENTS

Interoperability via open standards (MCP, A2A)



**Snowflake Cortex AI**

# Tools for Your Agent (Cortex Agents API)

Agents need to invoke tools to take actions and retrieve relevant context.

- **Out-of-the-box tools for common tasks:** web search, charting, SQL execution, etc.
- **3rd-party Data & Semantic Models** through Snowflake Marketplace
- **User-defined custom tools:** write custom functions for agents to execute in Snowflake
- **External tools:** invoke external, MCP-compatible tools simply within a Cortex Agent



## Web Search

Reference up-to-date information to expand the knowledge of your agent to the public internet



## Marketplace

Access AI Ready Data and Apps through Snowflake Marketplace



## Custom Tools

Call code securely in Snowflake to extend your agent to do anything (e.g., UDFs)

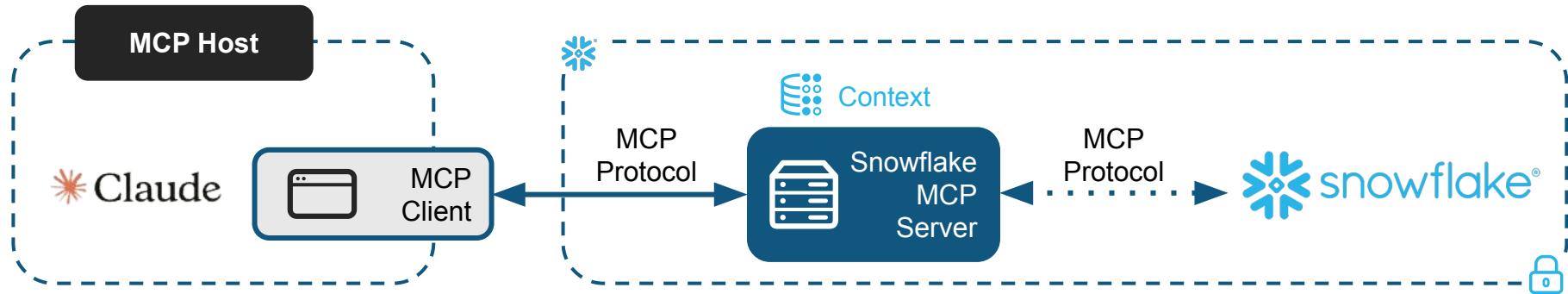


## MCP Servers

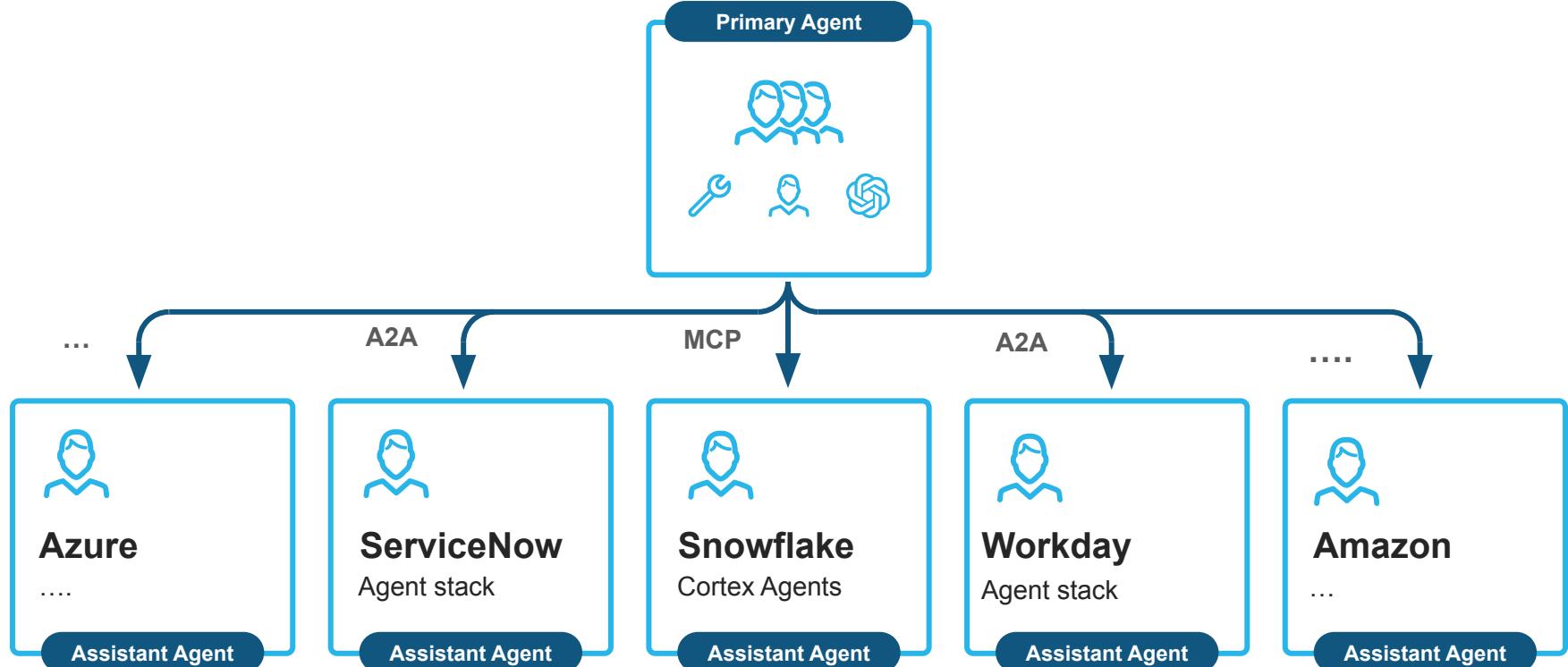
Call externally-hosted, MCP-compatible tools to take actions, bring in context

# Model Context Protocol

Host, client and server relationship within a secure environment



# Multi-Agent



# Snowflake Brings AI to Your Data



## EASY

Apply Powerful  
AI Capabilities across  
Multi-Modal Data Types



## EFFICIENT

Achieve Low Costs  
with Optimized  
Processing



## TRUSTED

Deliver Consistent &  
Accurate Results  
Across Interfaces

# THANK YOU



© 2025 Snowflake Inc. All Rights Reserved