

Introduction to Snowflake Cortex and Notebooks

INTRODUCTION TO GENERATIVE AI IN SNOWFLAKE



James Cha-Earley

Senior Developer Advocate, Snowflake

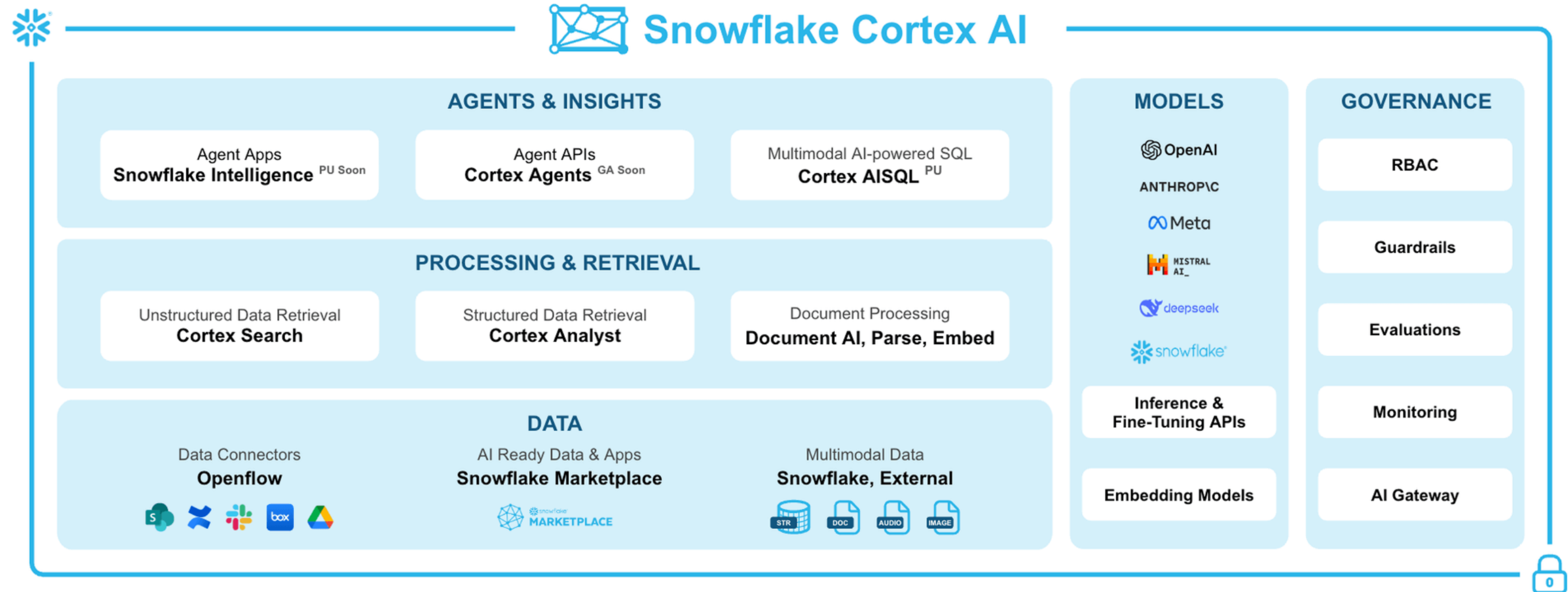
Meet your instructor!

James Cha-Earley

- Senior Developer Advocate at Snowflake



Snowflake Cortex



- Production-ready AI at our fingertips

¹ Image source: <https://www.snowflake.com/en/product/features/cortex/>

Cortex AI

- Native integration across SQL, Python, and notebooks
- Experiment to production within Snowflake
- Built-in governance, policies, and observability



Course data

```
1 SELECT *
2 FROM HOTELS.REVIEWS
3 LIMIT 10;
```

Results

Chart

	🕒 DATE	# RATING	📄 DESCRIPTION	📄 HOTEL_NAME	📄 CITY	📄 COUNTRY	📄 LANGUAGE
1	2023-09-21	10	Had an absolutely wonderful stay at the Hyatt Nice	Hyatt Regency Palais	Nice	France	en
2	2023-09-21	6	Very standard hotel. Great location.	Hyatt Regency Palais	Nice	France	en
3	2023-09-21	10	We enjoyed our stay at fiesta America Coral beach.	Grand Fiesta Americana	Cancun	Mexico	en
4	2023-09-21	10	The property was excellent. Very clean. From the ti	Grand Fiesta Americana	Cancun	Mexico	en
5	2023-09-21	10	Excelente hotel super limpieza y buena atencion	Grand Fiesta Americana	Cancun	Mexico	es
6	2023-09-20	8	Esta muy descuidada; es un hotel muy viejo y su as	Fairmont Monte Carlo	Monaco	France	es
7	2023-09-20	10	Awesome place to stay & relatively cheap for the M	Fairmont Monte Carlo	Monaco	France	en
8	2023-09-20	10	Always convenient; easily accessible and well kept	Warwick Geneva	Geneva	Switzerland	en
9	2023-09-19	10	The room was so unique and special. Spectacular f	Grand Fiesta Americana	Cancun	Mexico	en
10	2023-09-18	10	Très bon hôtel comme attendu. On s'occupe de vo	barriere-le-majestic	Cannes	France	fr

Template code

Python ▾ as cell1

```
1 # Import python packages
2 import streamlit as st
3 import pandas as pd
4
5 # We can also use Snowpark for our analyses!
6 from snowflake.snowpark.context import get_active_session
7 session = get_active_session()
8
```

+ Python

+ SQL

+ Markdown

SQL ▾ as cell2

```
1 -- Welcome to Snowflake Notebooks!
2 -- Try out a SQL cell to generate some data.
3 SELECT 'FRIDAY' as SNOWDAY, 0.2 as CHANCE_OF_SNOW
4 UNION ALL
5 SELECT 'SATURDAY', 0.5
6 UNION ALL
7 SELECT 'SUNDAY', 0.9;
```

	SNOWDAY	CHANCE_OF_SNOW
0	FRIDAY	0.2
1	SATURDAY	0.5
2	SUNDAY	0.9

Python ▾ as cell3

```
1 # Then, we can use the python name to turn cell2 into a Pandas dataframe
2 my_df = cell2.to_pandas()
3
4 # Chart the data
5 st.subheader("Chance of SNOW ❄️")
6 st.line_chart(my_df, x='SNOWDAY', y='CHANCE_OF_SNOW')
7
8 # Give it a go!
```

Querying the data

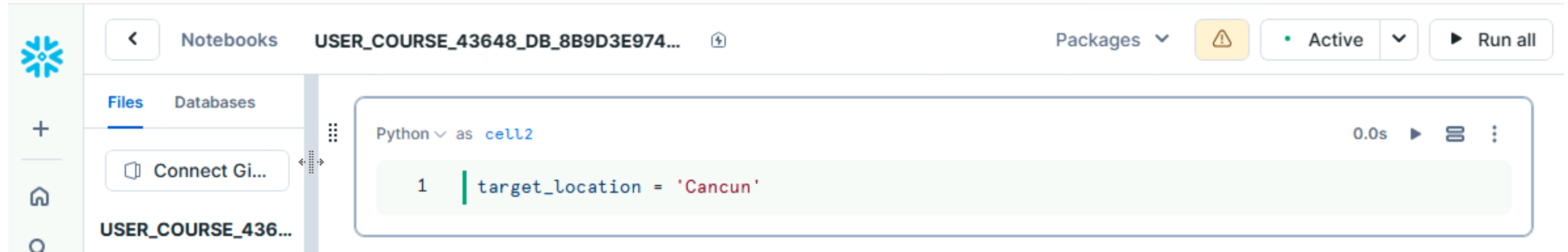
SQL ▾ as cell4

```
1 | DESC TABLE HOTELS.REVIEWS
```

	name	type	kind	null?	default
0	DATE	DATE	COLUMN	Y	None
1	RATING	NUMBER(38,0)	COLUMN	Y	None
2	DESCRIPTION	VARCHAR(16777216)	COLUMN	Y	None
3	HOTEL_NAME	VARCHAR(16777216)	COLUMN	Y	None
4	CITY	VARCHAR(16777216)	COLUMN	Y	None
5	COUNTRY	VARCHAR(16777216)	COLUMN	Y	None
6	LANGUAGE	VARCHAR(16777216)	COLUMN	Y	None

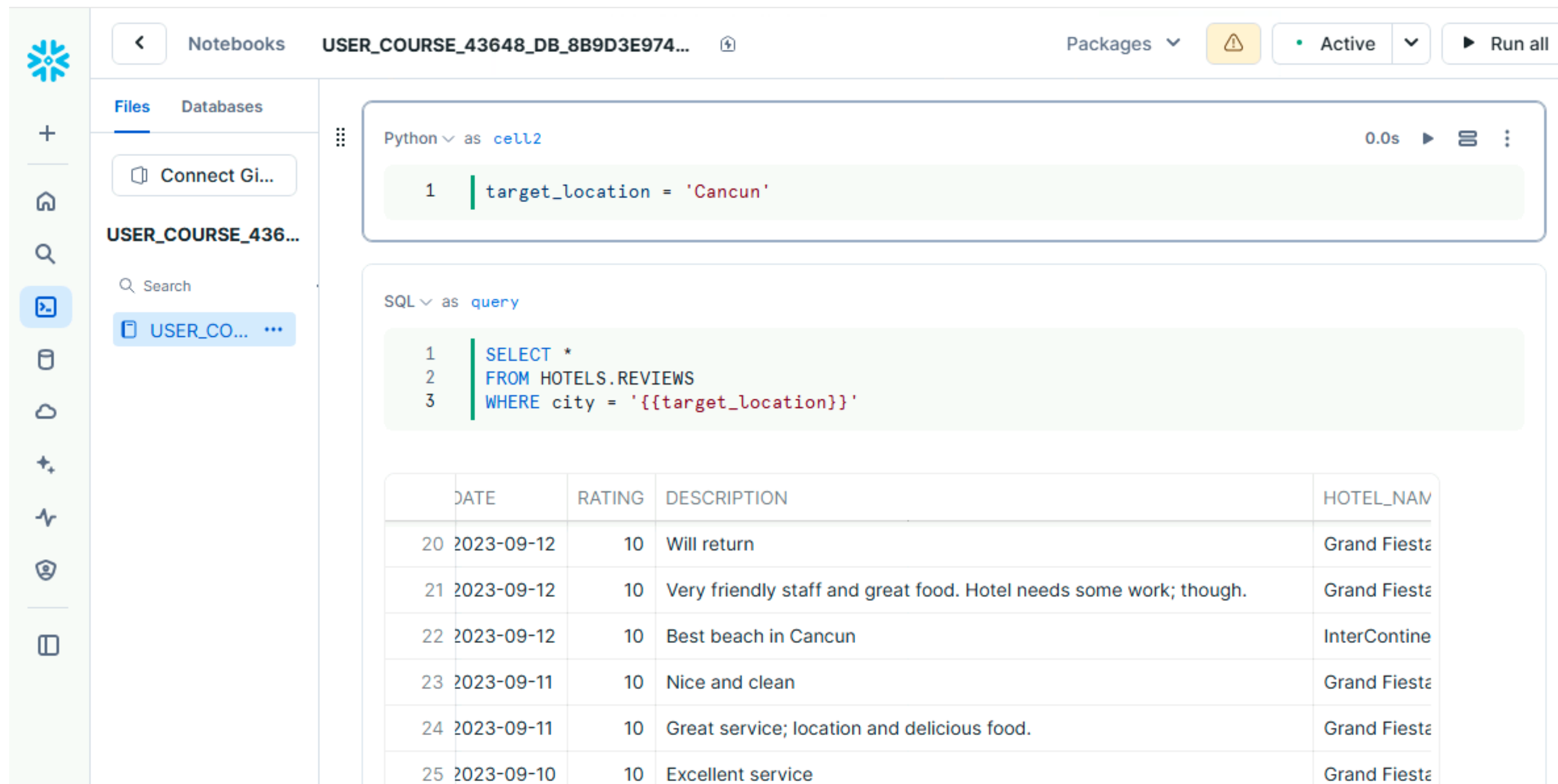
Python variable injection

- Create a Python variable for substitution in a SQL query



The screenshot shows the Snowflake web interface. On the left is a sidebar with the Snowflake logo, a plus sign, a home icon, and a search icon. Below these are tabs for 'Files' and 'Databases'. Under 'Files', there is a 'Connect Gi...' button and a list item 'USER_COURSE_436...'. The main area at the top shows a breadcrumb 'Notebooks' followed by 'USER_COURSE_43648_DB_8B9D3E974...'. To the right of this are 'Packages' with a dropdown arrow, a warning icon, a status indicator 'Active' with a dropdown arrow, and a 'Run all' button. The main workspace contains a code cell labeled 'Python' with a dropdown arrow and 'as cell2'. The cell has a green background and contains the code: `1 target_location = 'Cancun'`. To the right of the code is a '0.0s' timer, a play button, a list icon, and a vertical ellipsis menu.

Python variable injection



The screenshot shows a Snowflake Notebook interface. The top bar includes a back arrow, the text "Notebooks", the notebook name "USER_COURSE_43648_DB_8B9D3E974...", a "Packages" dropdown, a warning icon, a status indicator "Active", and a "Run all" button. The left sidebar contains a vertical menu with icons for home, search, and other notebook functions. The main area is divided into two sections: "Files" and "Databases". Under "Files", there is a "Connect Gi..." button and a search bar. Under "Databases", there is a "USER_CO..." entry. The notebook content is displayed in two cells. The first cell is a Python cell named "cell2" containing the code:

```
1 target_location = 'Cancun'
```

. The second cell is an SQL cell named "query" containing the code:

```
1 SELECT *
2 FROM HOTELS.REVIEWS
3 WHERE city = '{{target_location}}'
```

. Below the SQL code, the results of the query are shown as a table with 5 columns: an index, DATE, RATING, DESCRIPTION, and HOTEL_NAME. The table contains 7 rows of data.

	DATE	RATING	DESCRIPTION	HOTEL_NAME
20	2023-09-12	10	Will return	Grand Fiesta
21	2023-09-12	10	Very friendly staff and great food. Hotel needs some work; though.	Grand Fiesta
22	2023-09-12	10	Best beach in Cancun	InterContine
23	2023-09-11	10	Nice and clean	Grand Fiesta
24	2023-09-11	10	Great service; location and delicious food.	Grand Fiesta
25	2023-09-10	10	Excellent service	Grand Fiesta

Setting up Cortex

The screenshot displays the Snowflake Cortex interface. At the top, there's a navigation bar with a back arrow, 'Notebooks', 'EXERCISE_1_1', a 'Packages' dropdown menu, a 'Start' button, and several utility icons. Below this, the left sidebar has tabs for 'Files' and 'Databases'. Under 'Files', there's a 'Connect Git Repository' button and a list of files for 'EXERCISE_1_1'. The file 'EXERCISE_1_1.ipynb' is selected. The main area on the right shows a code editor for a Python cell named 'cell1'. The code imports 'streamlit' and 'pandas', and sets up a Snowflake session.

Navigation: < Notebooks EXERCISE_1_1 Packages Start

Files Databases

Connect Git Repository

EXERCISE_1_1

Search +

- environment.yml ...
- EXERCISE_1_1.ipynb ...

Python as cell1

```
1 # Import python packa
2 import streamlit as s
3 import pandas as pd
4
5 # We can also use Sno
6 from snowflake.snowpa
7 session = get_active_
8
```

Summarizing text

```
from snowflake.cortex import summarize
```

```
review = """
```

```
I was very impressed with this hotel, especially the service.
```

```
They make you feel special and you feel like everything is possible.
```

```
Special shout out to the Kidz Club staff.
```

```
Our daughter is very picky and for the first time, she begged to go back there.
```

```
They have a great supervision and they let the kids do fun activities  
such as cooking classes and crafting.
```

```
The food was really good (buffet offered a different choice everyday).
```

```
The all-inclusive program is really worth the price, and the beach was beautiful!
```

```
"""
```

Summarizing text

```
result = summarize(text=review)
print(result)
```

```
The hotel delivered excellent service with outstanding Kidz Club staff who engaged even picky children through supervised activities. The daily buffet offered good variety, and the all-inclusive program plus beautiful beach provided great value.
```

Cortex in SQL cells

SQL ▾ as cell1

1

SELECT SNOWFLAKE.CORTEX.SUMMARIZE('I was very impressed with this hotel, especially the service. They make you feel special and you feel like everything is possible. Special shout out to the Kidz Club staff. Our daughter is very picky and for the first time, she begged to go back there. They have a great supervision and they let the kids do fun activities such as cooking classes and crafting. The food was really good (buffet offered a different choice everyday). The all-inclusive program is really worth the price, and the beach was beautiful!')

0

SNOWFLAKE.CORTEX.SUMMARIZE('I WAS VERY IMPRESSED WITH THIS H
The hotel impressed the speaker with exceptional service, making guests f

- No imports needed when run in SQL cells
- Use Python cells for chaining multiple functions

Let's practice!

INTRODUCTION TO GENERATIVE AI IN SNOWFLAKE

Text generation

INTRODUCTION TO GENERATIVE AI IN SNOWFLAKE



James Cha-Earley

Senior Developer Advocate, Snowflake

Managing hotel review responses



¹ Image generated with Google Gemini 2.5 Pro

summarize()

```
long_review = """  
This hotel was nothing short of amazing. The best views; the best location;  
and the best staff. It's clean; conveniently located to other restaurants  
on the strip; although we ate at the restaurants on site and each was so good  
there's really no reason to leave. Ohhh the views. The beach was free of seaweed;  
amazingly clean and plenty of space; it wasn't crowded at all.  
Stay here; you will not regret it. Did I mention how comfortable the beds were.  
Yes save yourself trouble and stay here. It's really perfect.  
"""  
  
from snowflake.cortex import summarize  
summarized_review = summarize(text=long_review)
```

Printing the summary

```
print(summarized_review)
```

```
The hotel exceeded expectations with stunning views, exceptional staff,  
comfortable beds, and top-notch on-site dining, making it a perfect and  
highly recommended stay.
```

Extracting from a database

SQL ∨ as `cell1`

```
1 SELECT DESCRIPTION
2 FROM HOTELS.REVIEWS
3 ORDER BY date DESC
4 LIMIT 1;
```

Python ∨ as `cell2`

```
1 import pandas as pd
2
3 latest_review = cell1.to_pandas()
4 latest_review = latest_review['DESCRIPTION'].iloc[0]
5 print(latest_review)
```

Had an absolutely wonderful stay at the Hyatt Nice!

Generating a reply

```
from snowflake.cortex import complete

prompt = f"Write a short response to this hotel review: {latest_review}."

# Pass the prompt
response = complete(prompt=prompt,
```

Generating a reply

```
from snowflake.cortex import complete

prompt = f"Write a short response to this hotel review: {latest_review}."

# Specify the model
response = complete(prompt=prompt,
                    model='llama3.1-8b',
```

Generating a reply

```
from snowflake.cortex import complete

prompt = f"Write a short response to this hotel review: {latest_review}."

# Set the temperature - control predictability of output
response = complete(prompt=prompt,
                    model='llama3.1-8b',
                    options={
                        'temperature':0.3,
```

Generating a reply

```
from snowflake.cortex import complete

prompt = f"Write a short response to this hotel review: {latest_review}."

# Limit the number of tokens with max_tokens
response = complete(prompt=prompt,
                    model='llama3.1-8b',
                    options={
                        'temperature':0.3,
                        'max_tokens':150
                    })
```

Viewing the response

```
print(response)
```

```
Thank you so much for sharing your wonderful experience at the Hyatt Nice!  
We're thrilled to hear that you had a fantastic stay with us.
```

- Fast, flexible, authentic

Text generation with AI_COMPLETE()

```
SELECT
  DESCRIPTION,
  AI_COMPLETE(
    'llama3.1-8b',
    PROMPT('Write a short response to this hotel review: {0}.', DESCRIPTION),
    {'temperature': 0.3, 'max_tokens': 150}
  ) AS completion
FROM HOTELS.REVIEWS
ORDER BY date DESC
LIMIT 1;
```

Output in Snowflake notebook

SQL ▾ as cell6 •

1.0s

```
1  SELECT
2      DESCRIPTION,
3      AI_COMPLETE(
4          'llama3.1-8b',
5          PROMPT('Write a short response to this hotel review: {0}.', DESCRIPTION),
6          {'temperature': 0.3, 'max_tokens': 150}
7      ) AS completion
8  FROM HOTELS.REVIEWS
9  ORDER BY date DESC
10 LIMIT 1;
```

	DESCRIPTION	COMPLETION
0	Had an absolutely wonderful stay at the Hyatt Nice!	We're thrilled to hear that you had an absolutely

- `AI_COMPLETE()` is available out of the box

Text generation in Snowflake

-- Coding in SQL cells

```
AI_COMPLETE(  
  'llama3.1-8b',  
  PROMPT('Write a short response  
         to this hotel review: {0}.',  
         DESCRIPTION),  
  {'temperature': 0.3,  
   'max_tokens': 150}  
)
```

- Generate text for a batch of rows

Coding in Python cells

```
complete(  
  model='llama3.1-8b',  
  prompt = f"""Write a short response  
            to this hotel review:  
            {latest_review}.""",  
  options={'temperature':0.3,  
           'max_tokens':150}  
)
```

- Use to chain multi-step workflows

Let's practice!

INTRODUCTION TO GENERATIVE AI IN SNOWFLAKE

Text classification

INTRODUCTION TO GENERATIVE AI IN SNOWFLAKE



James Cha-Earley

Senior Developer Advocate, Snowflake

Challenges of classifying text

The problem

- Reviews can contain any information
- Manual categorizing is time-consuming and not scalable

The solution

- Automate using Snowflake Cortex!
- `classify_text()`

Defining categories

```
# Define categories
categories = ["overall_experience", "location", "staff", "food_beverages",
             "facilities"]
```

Classifying text

```
from snowflake.cortex import classify_text

category = classify_text(
    str_input="The check-in was smooth and the staff were very friendly.",
    categories=categories
)

print(category)
```

```
{
  "label": "staff"
}
```


Converting outputs to the dictionary

```
print(type(category))
```

```
<class 'str'>
```

```
import json
```

```
category_dict = json.loads(category)  
print(type(category_dict))
```

```
<class 'dict'>
```

Scaling the workflow

- Classification pipeline categorizing reviews for a given month

```
# Python code
```

```
month = 5
```

```
-- SQL query
```

```
SELECT *
```

```
FROM HOTELS.REVIEWS
```

```
WHERE EXTRACT(month FROM date) = '{{month}}'
```

Applying `classify_text()`

```
df = cell2.to_pandas()

def classification(text):
    result = classify_text(
        str_input=text,
        categories=labels
    )
    result_dict = json.loads(result)
    return result_dict["label"]
```

Applying classify_text()

```
# Apply classification to the DataFrame
df["category"] = reviews["DESCRIPTION"].apply(classification)

# Print the first row's predicted category
print(df["category"].head(1))
```

```
0    overall_experience
Name: category, dtype: object
```

Sentiment analysis

- Only interested about reviews sentiment

SQL  as `cell8`

```
1  SELECT AI_SENTIMENT('The check-in was smooth and the staff were very friendly.')
```

	AI_SENTIMENT('THE CHECK-IN WAS SMOOTH AND THE STAFF WERE VER
0	{ "categories": [{ "name": "overall", "sentiment": "positive" }] }

Sentiment analysis

SQL ▾ as cell7

1.1s ▶ ☰ ⋮

```
1 SELECT AI_SENTIMENT('The check-in was smooth and the staff were very  
friendly.')[0]['sentiment']
```

	AI_SENTIMENT('THE CHECK-IN WAS SMOOTH AND THE STAFF WERE VER
0	"positive"

- Available sentiments: positive , negative , neutral , mixed , and unknown

Let's practice!

INTRODUCTION TO GENERATIVE AI IN SNOWFLAKE