Demo Use-Case 2- Hello RAG in Snowflake Cortex

This project will walk you through building a **Retrieval-Augmented Generation (RAG)** workflow **entirely inside Snowflake** using **Cortex LLM + Embeddings + SQL**.

We'll go from data setup to an AI-powered query answering system — no external APIs or Python required.

Project Goal

Build an intelligent **Question Answering System** that:

- Stores internal documents (policies, FAQs, product info, etc.)
- Embeds them using **Cortex embedding models**
- Finds the most relevant documents for a user's query
- Uses **Cortex LLM (COMPLETE function)** to generate a natural-language answer

All inside Snowflake SQL.

Architecture Overview

Step	Component	Snowflake Feature Used
1	Data ingestion	CREATE TABLE, INSERT
2	Vector embedding creation	SNOWFLAKE.CORTEX.EMBED_TEXT_768

Step	Component	Snowflake Feature Used
3	Semantic similarity search	VECTOR_COSINE_SIMILARITY
4	Answer generation	SNOWFLAKE.CORTEX.COMPLETE
5	Optional summarization	SNOWFLAKE.CORTEX.SUMMARIZE

Step 1: Create Sample Knowledge Base

```
CREATE OR REPLACE TABLE COMPANY_DOCS AS
SELECT 1 AS DOC_ID, 'Employees are allowed to work remotely up to two days per week.' AS CONTENT UNION ALL
SELECT 2, 'Annual leave requests should be submitted at least two weeks in advance.' UNION ALL
SELECT 3, 'The company provides full health insurance coverage to all permanent employees.' UNION ALL
SELECT 4, 'Employees must complete mandatory cybersecurity training every year.' UNION ALL
SELECT 5, 'Performance reviews are conducted twice a year: mid-year and end-of-year.';
```

You now have a basic knowledge base with five HR/policy documents.

Step 2: Create Embeddings for All Documents

```
CREATE OR REPLACE TABLE COMPANY_DOCS_EMBEDDINGS AS
SELECT
  DOC_ID,
  CONTENT,
  SNOWFLAKE.CORTEX.EMBED_TEXT_768('e5-base-v2', CONTENT) AS EMBEDDING
FROM COMPANY_DOCS;
```

This creates 768-dimensional embeddings for each document.

Run this to confirm:

```
SELECT * FROM COMPANY_DOCS_EMBEDDINGS LIMIT 2;
```

Step 3: Run Semantic Search (Vector Similarity)

Now test with a user query, for example:

“Does the company allow remote work?”

```
WITH QUERY_EMB AS (
  SELECT SNOWFLAKE.CORTEX.EMBED_TEXT_768('e5-base-v2', 'Does the company allow
remote work?') AS VECTOR
)
SELECT
  d.DOC_ID,
  VECTOR_COSINE_SIMILARITY(d.EMBEDDING, q.VECTOR) AS SIMILARITY,
  d.CONTENT
FROM COMPANY_DOCS_EMBEDDINGS d, QUERY_EMB q
ORDER BY SIMILARITY DESC
LIMIT 3;
```

You should see the remote work policy document (DOC_ID = 1) ranked at the top.

Step 4: Generate an Answer using Cortex LLM

Now we'll feed the top-matched documents into an **LLM prompt** for final answer generation.

```
WITH QUERY AS (
  SELECT 'Does the company allow remote work?' AS QUESTION
),
RELEVANT_DOCS AS (
  SELECT
    d.CONTENT
  FROM COMPANY_DOCS_EMBEDDINGS d, QUERY q,
       LATERAL (
        SELECT SNOWFLAKE.CORTEX.EMBED_TEXT_768('e5-base-v2', q.QUESTION) AS
VECTOR
       ) qe
  ORDER BY VECTOR_COSINE_SIMILARITY(d.EMBEDDING, qe.VECTOR) DESC
  LIMIT 3
),
CONTEXT AS (
  SELECT LISTAGG(CONTENT, '\n') AS DOC_CONTEXT FROM RELEVANT_DOCS
)
SELECT SNOWFLAKE.CORTEX.COMPLETE(
  'mistral-large',
  CONCAT(
```

```
  'You are an HR assistant. Use the following context to answer the question clearly.\n\n',
  'Context:\n', (SELECT DOC_CONTEXT FROM CONTEXT), '\n\n',
  'Question: ', (SELECT QUESTION FROM QUERY), '\nAnswer:'
 )
) AS GENERATED_ANSWER;
```

Expected Output:

The company allows employees to work remotely up to two days per week.

Step 5: Optional — Add Summarization or Sentiment

You can summarize long retrieved docs before passing to the LLM:

```
SELECT SNOWFLAKE.CORTEX.SUMMARIZE(CONTENT) AS SHORT_SUMMARY
FROM COMPANY_DOCS
WHERE DOC_ID = 1;
```

Or analyze tone:

```
SELECT SNOWFLAKE.CORTEX.SENTIMENT('I really enjoy the flexible work policy!') AS
SENTIMENT;
```

Step 6: Validate RAG Workflow

Step	Action	Function Used	Example
1	Ingest documents	CREATE TABLE	HR policies
2	Embed documents	EMBED_TEXT_768	Converts text → vectors

Step	Action	Function Used	Example
3	Retrieve relevant docs	VECTOR_COSINE_SIMILARITY	Finds top-K context
4	Generate final answer	COMPLETE	LLM answers user query
5	(Optional) Summarize	SUMMARIZE	Simplify context

Real-World Use Cases

Use Case	Description	Snowflake Cortex Function
HR Knowledge Bot	Employees ask HR policy questions	COMPLETE + EMBED_TEXT_768
Customer Support Assistant	Summarize FAQs and return answers	SUMMARIZE + EMBED_TEXT_768
Product Catalog Search	Semantic search for product info	VECTOR_COSINE_SIMILARITY
Document Summarization	Short summaries for legal docs	SUMMARIZE
Internal AI Agent	Combine multiple Cortex functions	COMPLETE, SENTIMENT, TRANSLATE, etc.

Key Cortex Functions Used

Function	Purpose
EMBED_TEXT_768(model, text)	Convert text to vector embeddings
VECTOR_COSINE_SIMILARITY(vec1, vec2)	Compute similarity
COMPLETE(model, prompt)	Generate text / answers
SUMMARIZE(text)	Create short summaries
TRANSLATE(text, target_lang)	Translate content
SENTIMENT(text)	Analyze tone of text

Final Output

After completing all steps, you'll have a **fully working in-database RAG pipeline**:

- No external vector database
- No API keys
- Fully secured, auditable, and scalable inside Snowflake