```
CREATE WAREHOUSE IF NOT EXISTS SonicAI_WH
WITH
WAREHOUSE_TYPE = 'STANDARD'
WAREHOUSE_SIZE = 'XSMALL'
MAX_CLUSTER_COUNT = 1
MIN_CLUSTER_COUNT = 1
AUTO_SUSPEND = 600
COMMENT = 'This is SonicCortex small warehouse';

USE ROLE accountadmin;
USE WAREHOUSE SONICAI_WH;

CREATE OR REPLACE DATABASE SonicEmbeddingsDB;

USE SonicEmbeddingsDB;

DROP table SONICDAILYOBJECTS;

CREATE SCHEMA IF NOT EXISTS SonicSchema
COMMENT = 'This is my schema';
Use schema SonicSchema;

CREATE TABLE SonicDailyObjects (
obj_id INT PRIMARY KEY,
obj_name VARCHAR(255) NOT NULL
);

INSERT INTO SONICDAILYOBJECTS (obj_id, obj_name)
VALUES
(1, 'Chair'),
(2, 'Table'),
(3, 'Book'),
(4, 'Pen'),
(5, 'Phone'),
(6, 'Computer'),
(7, 'Car'),
(8, 'House'),
(9, 'Tree'),
(10, 'Flower'),
(11, 'Food'),
(12, 'Water'),
(13, 'Clothes'),
(14, 'Shoes'),
(15, 'Key'),
(16, 'Lamp'),
(17, 'Bed'),
(18, 'Door'),
(19, 'Window'),
(20, 'Clock'),
(21, 'Beaver'),
(22, 'Hunt'),
(23, 'Galaxy');

select * from SONICDAILYOBJECTS where obj_name like 'P%'; -- Returns objects starting with 'P'
select * from SONICDAILYOBJECTS where obj_name like 'Writing'; -- Doesn't return 'Pen'
select * from SONICDAILYOBJECTS where obj_name like 'animal'; -- Doesn't return 'Beaver'

-- use SNOWFLAKE.CORTEX.EMBED_TEXT_768 to create embeddings:
-- This new table everydayobjects_embeddings contains the original objects along with their vector embeddings(a new data -- type in
Snowflake).

CREATE OR REPLACE TABLE SonicDailyObjects_embeddings AS
SELECT *, SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', obj_name) AS obj_embedding
FROM SONICDAILYOBJECTS;


SELECT
```

```
obj_name,
VECTOR_COSINE_SIMILARITY(
obj_embedding,
SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', 'tree')
) AS similarity
FROM SonicDailyObjects_embeddings
ORDER BY similarity DESC
LIMIT 5; -- This will return the top 5 objects most closely related to the word 'tree' semantically.


-- USING RAG to do chunking

-- This concept extends to larger text documents. Let's create a DocumentChunks table and embed the chunks:


-- Table Creation

CREATE OR REPLACE TABLE DocumentChunks (
chunk_id INT PRIMARY KEY,
document_name VARCHAR(255),
page_number INT,
chunk_text TEXT );


-- Sample Data Insertion
INSERT INTO DocumentChunks (chunk_id, document_name, page_number, chunk_text)
VALUES
(1, 'Sales_Report_Q1_2024.pdf', 1, 'This report summarizes the sales performance for the first quarter of 2024. The company achieved a
15% increase in revenue compared to the same period last year.'),
(2, 'Sales_Report_Q1_2024.pdf', 2, 'Key factors contributing to the growth include the launch of new products and expansion into new
markets.'),
(3, 'Product_Brochure.pdf', 1, 'Our innovative product X is designed to revolutionize the way you work. It offers a seamless user experience
and powerful features to enhance productivity.'),
(4, 'Technical_Manual.pdf', 5, 'Troubleshooting Guide: If you encounter an error message E101, please check the connection cables and
restart the device.'),
(5, 'Marketing_Strategy_2024.pdf', 3, 'The marketing team will focus on social media campaigns and influencer partnerships to increase
brand awareness.');

Select * from DocumentChunks;


create or replace table documentembeddings as select *, SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m',
CHUNK_TEXT) AS object_embedding from documentchunks;

select * from documentembeddings;

SELECT
CHUNK_TEXT,
VECTOR_COSINE_SIMILARITY(
object_embedding,
SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', 'sales growth in Q1')
) AS similarity
FROM DocumentEmbeddings
ORDER BY similarity DESC
LIMIT 1; --Change to from Limit 5 to limit 1


-- We can then find the chunk most similar to a query and summarize it:

SELECT
chunk_text AS Original_Chunk,
SNOWFLAKE.CORTEX.SUMMARIZE(chunk_text) AS Cortex_Summary
FROM (
SELECT
CHUNK_TEXT,
VECTOR_COSINE_SIMILARITY(
object_embedding,
SNOWFLAKE.CORTEX.EMBED_TEXT_768('snowflake-arctic-embed-m', 'sales growth in Q1')
```

```
) AS similarity
FROM DocumentEmbeddings
ORDER BY similarity DESC
LIMIT 1
) AS MostSimilarChunk;

-- This query will search the document chunks for text related to "sales growth in Q1", then return the original chunk along with a summary generated by Snowflake Cortex.

-- Key Takeaways:
-- Embeddings unlock the ability to search and analyze text based on meaning.
-- Snowflake Cortex makes it easy to generate and work with embeddings.
-- Vector similarity search enables finding semantically related information.
-- Applications range from simple object comparisons to complex document analysis and RAG based applications.
```