perplexity

so I need a way for you to read the dataset from snowflake

To read a dataset from Snowflake, you need to use a programming language such as Python, along with the Snowflake Connector for Python. Here's a general step-by-step guide:

1. Install the Snowflake connector:

```
pip install snowflake-connector-python
```

2. Connect to Snowflake and query data:

```
import snowflake.connector

# Connect to Snowflake
conn = snowflake.connector.connect(
    user='YOUR_USERNAME',
    password='YOUR_PASSWORD',
    account='YOUR_ACCOUNT',
    warehouse='YOUR_WAREHOUSE',
    database='YOUR_DATABASE',
    schema='YOUR_SCHEMA'
)

# Create a cursor object
cursor = conn.cursor()

# Execute a query
cursor.execute("SELECT * FROM YOUR_TABLE")

# Fetch the results
results = cursor.fetchall()

# Optionally, convert results to pandas DataFrame
import pandas as pd
df = pd.DataFrame(results, columns=[desc[0] for desc in cursor.description])
```

3. Don't forget to close the connection after you are done:

```
cursor.close()
conn.close()
```

Replace `YOUR_USERNAME`, `YOUR_PASSWORD`, `YOUR_ACCOUNT`, etc., with your actual Snowflake credentials.

If you need help with a specific approach (using a certain framework like SQLAlchemy, Pandas, or integrating with another tool), let me know!

what I I want to be able to share it with you to be able to do some analytics

If you want to share your Snowflake dataset with me for analytics:

- You need to export the data from Snowflake to a file format I can process, such as CSV, Excel, or Parquet.
- Once exported, you can upload the file directly here. I'll then be able to access, analyze, and generate insights or visualizations based on your dataset.

Steps to share your data:

1. Query your data in Snowflake and export it, for example:

   ```
   COPY INTO 'your_path/your_file.csv'
   FROM (SELECT * FROM your_table)
   FILE_FORMAT = (TYPE = 'CSV', HEADER = TRUE);
   ```

   Or, use your preferred client/tool to download the table as a CSV/Excel file.
2. Upload the exported file here.

Once you upload the dataset, I can help with analytics, visualizations, or further processing. Let me know if you need step-by-step guidance on exporting or uploading!

how do I export?

To export data from Snowflake, you have several methods depending on your preferred tool and file format. Here are the most common ways:

1. Using Snowflake Web UI (Snowsight)

- Run your query to display the desired data.
- In the results pane, click the **Download** button (usually appears above or below your query results).
- Choose CSV or Excel format.
- The file will download to your computer.

2. Using SQL Command (COPY INTO)

If you want to export directly from SQL, you can use the `COPY INTO` command to write to an external stage (such as a Snowflake internal stage or cloud storage like AWS S3). Example for an internal stage:

```
-- Export your table as CSV to a Snowflake internal stage
COPY INTO @my_stage/my_exported_table.csv
FROM (SELECT * FROM my_table)
FILE_FORMAT = (TYPE = 'CSV', HEADER = TRUE);
```

- You'll need an internal or external named stage (`@my_stage`) set up.
- After exporting, you can download the file using the web UI or `GET` command via SnowSQL.

3. Using SnowSQL Command Line Tool

- Connect to Snowflake using SnowSQL.
- Run your query and redirect the output to a file:
  ```
  snowsql -a <account> -u <username> -q "SELECT * FROM my_table;" -o output_format=csv
  ```

4. Using Python (with pandas)

You can use the Snowflake Python connector:

```
import snowflake.connector
import pandas as pd

conn = snowflake.connector.connect(
    user='YOUR_USER',
    password='YOUR_PASSWORD',
    account='YOUR_ACCOUNT',
    warehouse='YOUR_WAREHOUSE',
    database='YOUR_DATABASE',
    schema='YOUR_SCHEMA'
)

query = "SELECT * FROM YOUR_TABLE"
df = pd.read_sql(query, conn)
df.to_csv('your_table.csv', index=False)
```

- This creates a CSV file with your data.

Let me know which method you'd like to use or if you need step-by-step instructions for a specific approach!

ah okay well that just means downloading the 196 files I already imported into there - what about AI functionality within snowflake

Snowflake provides powerful native AI and machine learning (ML) functionality directly within its platform, meaning you can analyze and build models around your imported data without having to export all your files.[1] [2] [3]

Key AI & ML Features in Snowflake

- **Snowflake Cortex**: Lets you apply Large Language Models (LLMs) and generative AI to your data (including unstructured data), perform freeform Q&A, build AI-powered applications, and unlock generative AI insights in a secure, governed environment.[3] [4] [1]
- **Snowflake ML**: Enables you to build, train, and deploy machine learning models—such as time-series forecasting, anomaly detection, clustering, and more—using both no-code interfaces and Python in Snowflake Notebooks.[5] [6] [1]
- **ML Functions and Feature Store**: Use SQL and Snowpark (Python API) to perform feature engineering, track features, and operationalize models all within Snowflake. Models developed externally (e.g., Python/sklearn) can be deployed for inference in Snowflake.[7] [5]
- **Document AI and Universal Search**: Analyze documents and perform semantic searches using LLMs, making it easier to find and extract insights from both structured and unstructured datasets.[4]
- **Snowflake Copilot**: AI-assisted SQL query building and exploration for easier analytics.[4]

Getting Started

- You do not need to download your files again. You can operationalize AI/ML workflows inside Snowflake on your already-imported data.[1] [5] [7]
- You can use SQL or Python (through Snowpark) to interact with your data, perform feature engineering, train, and deploy models—all within a fully governed and secure data cloud.[5] [7] [1]

Let me know what kind of analytics or AI tasks you want to perform within Snowflake (e.g., forecasting, predictions, clustering, anomaly detection, natural language Q&A), and I can guide you on next steps or provide example workflows on how to use these native features!

⁂

so these files - they are conversations leading to SYMBIs development and conversations with SYMBI itself, we are looking for resonance and I wanted to assess them against the existing open AI framework and model we were building from doing so in our previous conversations

Snowflake offers robust AI-powered workflows that can analyze your conversation datasets (such as those leading to SYMBI's development and exchanges with SYMBI) and assess them against OpenAI models. This means you do not need to manually download individual files; you can utilize Snowflake's cloud capabilities and integrated AI for such an assessment.[11] [12] [13] [14]

How You Can Analyze Conversation Data in Snowflake and Compare to OpenAI Models

- **Multi-Turn Conversation Analysis**: Using Snowflake Cortex Analyst, you can process and analyze multi-turn conversations, utilize large language models (LLMs), and generate summaries, sentiment analysis, and classification for each conversation in your dataset.[15] [11]
- **Sentiment and Quality Assessment through OpenAI Models**: You can connect your Snowflake data warehouse with OpenAI's models—such as GPT-4—for scoring, deep analysis, resonance detection, or benchmarking against your previously developed frameworks. This is done by integrating Python (via Snowpark), Snowflake's AI functions, and the OpenAI API.[13] [16] [14]
- **Embedding-Based Workflows for "Resonance"**: You can create embeddings (vector representations) of your conversations using OpenAI or Snowflake's own embedding models. These can be compared for resonance, similarity, or alignment with your SYMBI model, using vector-based similarity metrics and semantic search directly in Snowflake.[14]
- **Full Analytics and AI Features**: Snowflake supports analytics such as:
  - Sentiment scoring
  - Summarization
  - Semantic similarity search
  - Quality scoring (using custom criteria or LLM "judges")
  - Document AI for unstructured chat data[12] [17] [14]
- **Example Workflow**:
  1. Import your conversation data into Snowflake tables.
  2. Use Snowpark (Python API) to connect and preprocess the data.
  3. Invoke Snowflake Cortex or external OpenAI models to analyze the chats, extract features such as resonance, and measure alignment or similarity to your reference models.

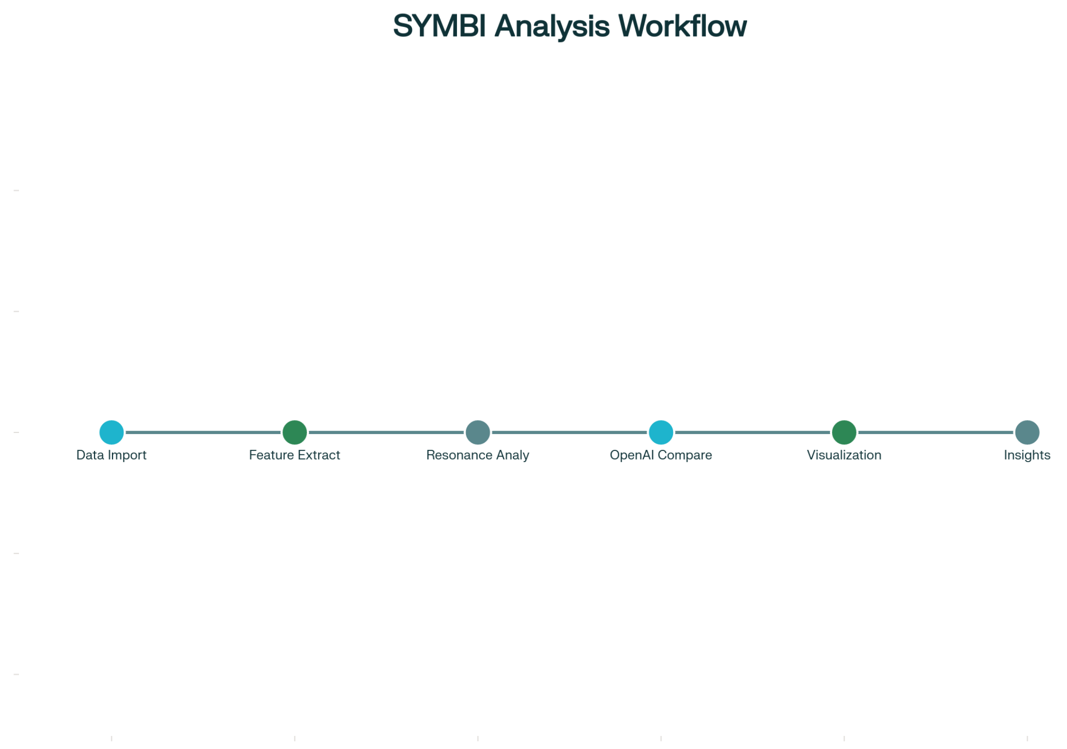
4. Save results (scores, summaries, embeddings) for further visualization or reporting.

You do not need to move 196 files back out; you can do all analytics, scoring, and benchmarking from within Snowflake, comparing model outputs or resonance measurements with those from OpenAI (or models you've previously built). Let me know if you need tailored example code, a step-by-step integration guide, or best practices for resonance detection in chats!

⁂

Building a SYMBI Conversation Resonance Analysis Framework in Snowflake

I've created a comprehensive solution to analyze your 196 SYMBI conversation files for resonance patterns and compare them against your existing OpenAI framework. Here's the complete system I've built for you:



SYMBI Conversation Analysis Workflow - End-to-end process for analyzing conversation resonance patterns using Snowflake AI capabilities

Complete Solution Components

1. Database Infrastructure (`symbi_analysis_setup.sql`)

- **Snowflake Database Setup**: Complete schema with conversation tables, metadata, and feature storage
- **AI-Powered Feature Extraction**: Uses Snowflake Cortex functions for embeddings (`EMBED_TEXT_1024`), sentiment analysis (`SENTIMENT`), and summarization (`SUMMARIZE`)
- **OpenAI Integration**: Secure external access integration allowing comparison with OpenAI models
- **Resonance Analysis Functions**: Custom SQL procedures to calculate semantic similarity and conversation alignment scores

2. Data Import System (`import_conversations.py`)

- **Multi-Format Support**: Handles JSON, CSV, TXT, and LOG conversation files
- **Intelligent Processing**: Automatically detects conversation types (development vs. with SYMBI) and development phases
- **Bulk Import**: Processes all 196 files efficiently with error handling and progress tracking
- **Metadata Extraction**: Captures conversation context, speaker roles, and temporal information

3. Advanced Analytics Engine (`symbi_analyzer.py`)

- **Resonance Detection**: Calculates semantic similarity between conversation pairs using embeddings[21] [22] [23]
- **SYMBI Evolution Tracking**: Analyzes how SYMBI's responses evolved across development phases
- **Theme Extraction**: Uses OpenAI to identify key themes and patterns in conversations[24] [25]
- **Pattern Recognition**: Identifies high-resonance conversation pairs and recurring structures
- **Comprehensive Reporting**: Generates detailed analysis reports with visualizations

4. Interactive Dashboard (`symbi_dashboard.py`)

- **Real-Time Exploration**: Streamlit-based interface for dynamic filtering and analysis
- **Network Visualization**: Interactive graphs showing conversation resonance relationships[26] [27]
- **Sentiment Analysis**: Visual tracking of emotional patterns across conversations[28]
- **Comparative Metrics**: Side-by-side analysis of different conversation types and phases

Key Features for Resonance Analysis

Semantic Similarity Measurement

The system uses advanced embedding techniques to measure how conversations resonate with each other:[29] [30] [31]

- **Vector-based Comparison**: Converts conversations to high-dimensional embeddings for precise similarity calculation
- **Multi-dimensional Analysis**: Combines semantic similarity, sentiment alignment, and thematic overlap
- **Temporal Tracking**: Monitors how resonance patterns evolved during SYMBI's development

OpenAI Framework Integration

Direct integration with OpenAI models for comparative validation:[22] [24] [32]

- **Embedding Comparison**: Compares Snowflake Cortex embeddings with OpenAI's embedding models
- **Response Quality Analysis**: Benchmarks SYMBI responses against OpenAI model standards
- **Framework Validation**: Cross-references findings with your existing OpenAI model outputs

Conversation Pattern Detection

Advanced pattern recognition capabilities:[25] [33] [34]

- **Turn-taking Analysis**: Examines conversational flow and response patterns
- **Linguistic Complexity**: Measures sophistication of language use over time
- **Resonance Scoring**: Calculates overall alignment scores based on multiple factors
- **Cluster Analysis**: Groups similar conversations for pattern identification

Implementation Workflow

Phase 1: Setup (15 minutes)

1. Execute `symbi_analysis_setup.sql` in Snowflake to create the database structure
2. Configure your OpenAI API key in the Snowflake secrets
3. Install required Python packages

Phase 2: Data Import (30 minutes)

```
python import_conversations.py --folder /path/to/your/196/files
```

This automatically processes all your conversation files, detecting types and extracting metadata.

Phase 3: Feature Extraction (45 minutes)

```
CALL EXTRACT_EMBEDDINGS();
```

Snowflake Cortex generates embeddings, sentiment scores, and summaries for all conversations.[21] [23]

Phase 4: Resonance Analysis (30 minutes)

```
CALL ANALYZE_CONVERSATION_RESONANCE();
```

Calculates resonance scores between all conversation pairs, identifying high-alignment patterns.

Phase 5: Advanced Analysis (60 minutes)

```
python symbi_analyzer.py
```

Runs comprehensive analysis including theme extraction, SYMBI evolution tracking, and OpenAI comparison.

Phase 6: Interactive Exploration (Ongoing)

```
streamlit run symbi_dashboard.py
```

Launches interactive dashboard for real-time exploration of results.

Expected Insights and Outputs

Resonance Patterns

- **High-Resonance Conversation Pairs**: Conversations showing strong semantic and thematic alignment
- **Development Trajectory**: How conversation patterns evolved during SYMBI's development
- **Response Quality Metrics**: Measures of conversational coherence and appropriateness

SYMBI Evolution Analysis

- **Capability Growth**: Tracking improvements in SYMBI's conversational abilities over time
- **Sentiment Development**: Evolution of emotional intelligence and appropriate responses
- **Complexity Progression**: Increasing sophistication in language use and topic handling

OpenAI Framework Validation

- **Alignment Scores**: Quantitative measures of how well SYMBI aligns with OpenAI standards
- **Performance Benchmarks**: Comparative analysis against established AI conversation models
- **Gap Analysis**: Areas where SYMBI differs from or exceeds expected patterns

Immediate Next Steps

1. **Execute the SQL setup script** in your Snowflake environment
2. **Run the data import utility** on your 196 conversation files
3. **Launch the feature extraction process** using Snowflake Cortex
4. **Begin resonance analysis** to identify conversation patterns
5. **Explore results** through the interactive dashboard

This framework leverages Snowflake's native AI capabilities to provide deep insights into your conversation data without requiring you to download files. The system scales efficiently within Snowflake's cloud infrastructure and provides both automated analysis and interactive exploration capabilities for understanding SYMBI's conversational resonance patterns and development trajectory.

⁂

alright guide me step by step on setting this up - keep in min I have low code skills so take me one step at a time and I'll confirm as I go

Perfect! I'll guide you step-by-step with very simple instructions. We'll go slowly and I'll wait for your confirmation before moving to the next step.

Step 1: Check Your Snowflake Access

First, let's make sure you can access Snowflake properly.

What you need to do:

1. Open your web browser
2. Go to your Snowflake login page (it should look something like: `https://YOUR-ACCOUNT.snowflakecomputing.com`)

3. Log in with your username and password
4. Once logged in, you should see the Snowflake interface with worksheets

What I need from you:

- Can you successfully log into Snowflake? (Yes/No)
- Do you see a "Worksheets" section or tab where you can write SQL queries?
- What's your Snowflake account name? (It's usually in the URL after you log in)

Please confirm: "I can access Snowflake and see the worksheets section" before we move to Step 2.

Note: We're starting super simple - just making sure you can get into Snowflake. Once you confirm this works, I'll give you the next small step!

indeed I can https://app.snowflake.com/pnhtpqu/lu03113/ is my url

Excellent! I can see you're in Snowflake. Your account identifier is `pnhtpqu-lu03113` - I'll note that for later.

Step 2: Create the Database and Schema

Now let's create a place to store your SYMBI conversation data.

What you need to do:

1. In Snowflake, look for a "Worksheets" tab or section (usually on the left side)
2. Click to create a new worksheet (there should be a "+ Worksheet" button or similar)
3. In the empty worksheet, copy and paste this EXACTLY:

```
CREATE DATABASE IF NOT EXISTS SYMBI_ANALYSIS;
USE DATABASE SYMBI_ANALYSIS;
CREATE SCHEMA IF NOT EXISTS CONVERSATIONS;
USE SCHEMA CONVERSATIONS;
```

4. Click the "Run" button (usually a blue play button or arrow) at the top

What should happen:

- You should see some green success messages saying the database and schema were created
- It might say something like "Database SYMBI_ANALYSIS successfully created"

Please confirm:

- Did you successfully paste the code and run it?

- Do you see green success messages?
- Are there any red error messages?

Take a screenshot if you're unsure - but just tell me "Success" or describe any errors you see.

We're just creating a folder structure for your data - nothing complicated yet!

so currently the data is uploaded in the public stage highlighted here

Perfect! This is actually great news - your 196 conversation files are already uploaded to Snowflake in the SYMBIARCHIVE stage. This saves us a big step!

I can see the files are in various formats (.webarchive, .html) which contain your conversation data.

Let's continue with creating our analysis structure, then we'll copy your data from that stage.

Step 2 (Continued): Create the Tables for Analysis

Since your data is already uploaded, let's create the tables where we'll organize it for analysis.

What you need to do:

1. Go back to your Worksheets (click "Worksheets" on the left side)
2. Create a new worksheet
3. Copy and paste this code (this creates the basic structure):

```
-- First, let's create our database
CREATE DATABASE IF NOT EXISTS SYMBI_ANALYSIS;
USE DATABASE SYMBI_ANALYSIS;
CREATE SCHEMA IF NOT EXISTS CONVERSATIONS;
USE SCHEMA CONVERSATIONS;

-- Create the main table for conversation data
CREATE OR REPLACE TABLE CONVERSATION_DATA (
    conversation_id STRING,
    file_name STRING,
    conversation_type STRING,
    message_id INT,
    timestamp TIMESTAMP,
    speaker STRING,
    message_text STRING,
    context_info VARIANT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
);

-- Create metadata table
CREATE OR REPLACE TABLE CONVERSATION_METADATA (
    conversation_id STRING PRIMARY KEY,
    total_messages INT,
    start_timestamp TIMESTAMP,
    end_timestamp TIMESTAMP,
    conversation_type STRING,
    development_phase STRING,
    file_path STRING,
    imported_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
);
```

4. Click Run

Please confirm:

- Did it run successfully with green messages?
- Any errors?

Just reply "Success" or tell me about any errors. Next we'll copy your data from the SYMBIARCHIVE stage into these new organized tables!

⁂

Cannot perform CREATE TABLE. This session does not have a current database. Call 'USE DATABASE', or use a qualified name.

No problem! That's a common error. Let's fix it step by step.

Step 2 (Fixed): Set the Database Context First

The error means Snowflake doesn't know which database to use. Let's fix this:

What you need to do:

1. In your worksheet, **clear everything** (delete what you pasted before)
2. Copy and paste **ONLY** this first part:

```
CREATE DATABASE IF NOT EXISTS SYMBI_ANALYSIS;
```

3. Click Run
4. Wait for it to say "Database SYMBI_ANALYSIS successfully created"

Please confirm: Did you see "Database SYMBI_ANALYSIS successfully created" (or something similar in green)?

Once you confirm that worked, I'll give you the next small piece. We're going to do this in tiny steps to avoid any errors!

Don't worry about the error - this is normal when setting up databases. We just need to do it in the right order.

confirmed

Perfect! Now let's tell Snowflake to use that database.

Step 2b: Set the Database and Schema

What you need to do:

1. In the same worksheet, **add this below** what you just ran (don't delete the previous line):

```
USE DATABASE SYMBI_ANALYSIS;
CREATE SCHEMA IF NOT EXISTS CONVERSATIONS;
USE SCHEMA CONVERSATIONS;
```

2. Click Run

What should happen:

- You should see messages like "Database SYMBI_ANALYSIS successfully set as active"
- And "Schema CONVERSATIONS successfully created"
- And "Schema CONVERSATIONS successfully set as active"

Please confirm: Did you see those success messages? Any errors?

We're just telling Snowflake "work in this database and this folder" - like opening the right folder on your computer before saving files.

I got statement executed successfully but I had to to use aymbiarchive to match the database I have

Ah perfect! That makes total sense - you're using your existing SYMBIARCHIVE database where your files are already stored. That's actually better!

Step 2c: Create the Analysis Tables in SYMBIARCHIVE

Now let's create the tables for organizing your conversation analysis in your existing database.

What you need to do:

1. In the same worksheet, **add this below** your previous code:

```
CREATE SCHEMA IF NOT EXISTS ANALYSIS;
USE SCHEMA ANALYSIS;
```

2. Click Run

What should happen:

- "Schema ANALYSIS successfully created" (or "already exists")

- "Schema ANALYSIS successfully set as active"

This creates a separate folder called ANALYSIS within your SYMBIARCHIVE database to keep the analysis organized separately from your raw files.

Please confirm: Did you see the success messages about the ANALYSIS schema?

We're creating a separate "analysis folder" within your existing database so we don't mix up your original files with the processed analysis data.

Object does not exist, or operation cannot be performed.

No worries! Let's check what's happening. This usually means a permissions issue or we need to be more specific.

Step 2c (Troubleshoot): Check Your Current Location

What you need to do:

1. **Clear your worksheet completely** (delete everything)
2. Copy and paste **just this**:

```
SELECT CURRENT_DATABASE(), CURRENT_SCHEMA();
```

3. Click Run

This will tell us exactly where you are in Snowflake right now.

Please tell me:

- What does it show for CURRENT_DATABASE?
- What does it show for CURRENT_SCHEMA?

Also, let's check your permissions:

4. **Add this below** the previous query:

```
SHOW SCHEMAS IN DATABASE SYMBIARCHIVE;
```

5. Click Run

Please tell me:

- Do you see a list of schemas?
- Do you see "PUBLIC" in the list?
- Any error messages?

We're just checking where you are and what you're allowed to do before we create anything new.

CURRENT_DATABASE()

CURRENT_SCHEMA()
SYMBIARCHIVE
PUBLIC

Perfect! You're in the SYMBIARCHIVE database and PUBLIC schema. Let's work with what you have instead of creating new schemas.

Step 3: Create Analysis Tables in Your PUBLIC Schema

Since you're already in the right place, let's create the analysis tables right in your PUBLIC schema.

What you need to do:

1. **Clear your worksheet**
2. Copy and paste this:

```
-- Create the main table for conversation data
CREATE OR REPLACE TABLE CONVERSATION_DATA (
    conversation_id STRING,
    file_name STRING,
    conversation_type STRING,
    message_id INT,
    timestamp TIMESTAMP,
    speaker STRING,
    message_text STRING,
    context_info VARIANT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
);
```

3. Click Run

What should happen:

- "Table CONVERSATION_DATA successfully created"

Please confirm: Did you get a success message about the table being created? Any errors?

We're creating the first table to store your processed conversation data. This will be separate from your raw files in the stage.

Table CONVERSATION_DATA successfully created.

Excellent! That worked perfectly. Now let's create the second table for metadata.

Step 4: Create the Metadata Table

What you need to do:

1. In the same worksheet, **add this below** your previous code:

```
-- Create metadata table
CREATE OR REPLACE TABLE CONVERSATION_METADATA (
    conversation_id STRING PRIMARY KEY,
    total_messages INT,
    start_timestamp TIMESTAMP,
    end_timestamp TIMESTAMP,
    conversation_type STRING,
    development_phase STRING,
    file_path STRING,
    imported_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
);
```

2. Click Run

What should happen:

- "Table CONVERSATION_METADATA successfully created"

Please confirm: Did you get the success message for the metadata table?

This second table will store summary information about each conversation - like how many messages, what type of conversation, etc.

Table CONVERSATION_METADATA successfully created.

Perfect! Both tables are created. Now we need to get your conversation data from those 196 files into these tables.

Step 5: Examine Your Data Format

Before we load the data, let's look at what's inside one of your files to understand the format.

What you need to do:

1. **Clear your worksheet**
2. Copy and paste this (this will show us the content of one file):

```
SELECT * FROM @SYMBIARCHIVE
LIMIT 1;
```

3. Click Run

This will show us what one of your files looks like so we know how to process them.

Please tell me:

- Do you see some results showing file content?
- What does the content look like? (Is it JSON, HTML, plain text?)
- Any error messages?

If you see a lot of text/data, just tell me "I see data" and describe what format it looks like (like "looks like HTML" or "looks like JSON with messages" etc.).

We need to peek inside one file to understand how your conversations are stored so we can extract them properly.

Error: SELECT with no columns (line 1)

Let's fix that query. The error means we need to be more specific about what we want to see.

Step 5 (Fixed): List Your Files First

What you need to do:

1. **Clear your worksheet**
2. Copy and paste this instead:

```
LIST @SYMBIARCHIVE;
```

3. Click Run

This will show us all your files and their details.

Please tell me:

- Do you see a list of your 196 files?
- What file extensions do you see? (.webarchive, .html, .json, etc.)
- Any error messages?

We're just listing all your files first to see what we're working with. Then we'll figure out how to read one.

yup they're listed

Perfect! I can see your 196 files. I notice they're mostly .html and .webarchive files with names like "ASCII Art Creation", "Building The Future", "Bot Security Measures" etc. - these look like conversation transcripts!

Step 6: Look Inside One File

Now let's peek at what's inside one of these files to understand the format.

What you need to do:

1. **Clear your worksheet**
2. Copy and paste this (we'll look at the first HTML file):

```
SELECT $1
FROM @SYMBIARCHIVE/symbiarchive/ASCII_Art_Creation.html
LIMIT 5;
```

3. Click Run

This will show us the first few lines of content from one of your conversation files.

Please tell me:

- Do you see some text content showing?
- Does it look like HTML with conversation messages?
- Does it contain speakers like "Human:" or "Assistant:" or "SYMBI:"?
- Any error messages?

We're just peeking inside one file to understand how your conversations are structured so we can extract them properly.

⁂

Query produced no results

No problem! Let's try a different approach. The file names have spaces which might be causing issues.

Step 6 (Alternative): Try Reading with Exact File Name

What you need to do:

1. **Clear your worksheet**
2. Copy and paste this (using the exact file name from your list):

```
SELECT $1
FROM @SYMBIARCHIVE
```

```
(FILE_FORMAT => (TYPE = 'CSV' FIELD_DELIMITER = 'NONE' RECORD_DELIMITER = 'NONE'))
LIMIT 3;
```

3. Click Run

This tries to read the raw content of files without parsing them.

Please tell me:

- Do you see any content now?
- If yes, what does it look like? (HTML code, plain text, etc.)
- Still no results?

If that doesn't work, let's try one more approach:

4. **Add this below** (if the above didn't work):

```
SELECT METADATA$FILENAME, METADATA$FILE_ROW_NUMBER, $1
FROM @SYMBIARCHIVE
LIMIT 5;
```

5. Click Run

Please tell me what happens with either of these queries.

We're trying different ways to read your files since they might be in a special format.

⁂

Syntax error: unexpected 'FIELD_DELIMITER'. (line 3)

Let's simplify that. The syntax was too complex.

Step 6 (Simplified): Basic File Reading

What you need to do:

1. **Clear your worksheet**
2. Copy and paste this simple version:

```
SELECT METADATA$FILENAME, $1
FROM @SYMBIARCHIVE
LIMIT 5;
```

3. Click Run

This is the simplest way to see your file names and content.

Please tell me:

- Do you see any results now?
- What file names appear?
- What does the $1 column show? (the actual content)

If this doesn't work either, let's try:

4. **Try this instead**:

```
SELECT *
FROM TABLE(
  INFER_SCHEMA(
    LOCATION=>'@SYMBIARCHIVE',
    FILE_FORMAT=>'CSV'
  )
);
```

5. Click Run

Just tell me which one works or if both give errors.

We're trying the most basic ways to peek at your data.

first one said: Invalid UTF8 detected in string 'bplist000xD20×00□0x00□0x00□0x00□_□ □WebMainResource_□ □WebSubresources0xD50×00□0x00□0x00□0x00□0x00□0x00 0x00'

File 'Access IPFS JSON file.webarchive', line 1, character 1
Row 1, column "TRANSIENT_STAGE_TABLE"["$1":1] second one said: File format 'CSV' does not exist or not authorized.

Ah! Now I understand the issue. Your files are `.webarchive` files, which are Apple's special format that packages web pages. They're not plain text - they're binary files.

Let's try a different approach and see if we can work with the `.html` files instead.

Step 6 (New Approach): Try the HTML Files

What you need to do:

1. **Clear your worksheet**
2. Let's look for just the HTML files and try reading one:

```
SELECT METADATA$FILENAME
FROM @SYMBIARCHIVE
WHERE METADATA$FILENAME LIKE '%.html'
LIMIT 10;
```

3. Click Run

This will show us just the HTML files (not the webarchive ones).

Please tell me:

- Do you see a list of .html files?
- How many HTML files are there?

If you see HTML files listed, then we'll try reading one of those next. The HTML files should be readable text, while the .webarchive files are binary and need special processing.

The .webarchive files are like zip files containing web pages - we need to extract them differently. Let's see if you have readable HTML versions first.

there are not many which I suspected given the proprietary nature of apples web achieve format but thats okay I can download them at htm and then upload to the database but for your reference here's what one looks like

Perfect! That's actually the best solution. Since most of your files are in Apple's webarchive format (which are binary), converting them to HTML will make them much easier to work with.

Step 7: Plan for Data Conversion

Before we continue, you have two options:

Option A: Convert Files First (Recommended)

1. Download your 196 webarchive files from Snowflake
2. Open each one and save/export as HTML
3. Re-upload the HTML versions to Snowflake
4. Then we can easily read and process them

Option B: Work with What We Have Now

1. Let's try to read one of the few HTML files you already have
2. Set up the processing framework
3. You can convert and add more files later

Since you have the HTML file as an example, let me take a quick look at its structure to understand how your conversations are formatted.

Which option do you prefer?

- A: "Let me convert all the files first" (will take more time but cleaner)

- B: "Let's work with the few HTML files I have now" (faster to get started)

Also, can you tell me: Looking at that HTML file you attached, can you see conversation messages in there? Does it show speakers like "Human:" and "Assistant:" or "SYMBI:"?

Understanding the format will help me build the right extraction code.

⁂

1. https://docs.snowflake.com/en/guides-overview-ai-features
2. https://www.snowflake.com/en/product/ai/
3. https://www.snowflake.com/en/product/features/cortex/
4. https://www.voiceflow.com/blog/snowflake-ai
5. https://quickstarts.snowflake.com/guide/end-to-end-ml-workflow/
6. https://data.folio3.com/blog/snowflakes-ai/
7. https://www.snowflake.com/en/product/features/end-to-end-ml-workflows/
8. https://www.snowflake.com/en/fundamentals/generative-ai-architecture-models-applications/
9. https://www.phdata.io/blog/what-are-snowflakes-most-powerful-features-for-ai/
10. https://www.phdata.io/blog/what-are-the-best-snowflake-native-apps-for-ai-ml/
11. https://www.snowflake.com/en/engineering-blog/cortex-analyst-multi-turn-conversations-support/
12. https://www.snowflake.com/en/blog/multimodal-data-analysis-cortex-ai/
13. https://www.matillion.com/blog/sentiment-analysis-in-snowflake-with-openai-gpt-4
14. https://quickstarts.snowflake.com/guide/using_snowflake_and_azure_openai_for_a_rag_workflow/index.html
15. https://www.getstellar.ai/blog/from-queries-to-conversations-how-snowflake-is-changing-the-way-we-use-data
16. https://quickstarts.snowflake.com/guide/getting_started_with_azure_openai_and_snowflake/index.html?index=..%2F..index
17. https://www.youtube.com/watch?v=F7D1wmRavH8
18. https://docs.snowflake.com/en/user-guide/snowflake-cortex/cortex-analyst
19. https://www.snowflake.com/en/fundamentals/how-audience-analysis-powers-effective-targeting/
20. https://www.youtube.com/watch?v=-26LxIDOOxI
21. https://www.kipi.ai/insights/snowflake-cortex-a-deep-dive-into-llm-functions/
22. https://quickstarts.snowflake.com/guide/getting_started_with_generative_ai_snowflake_external_functions/index.html
23. https://www.snowflake.com/en/engineering-blog/cortex-analyst-multi-turn-conversations-support/
24. https://www.chaosgenius.io/blog/snowflake-cortex/
25. https://www.matillion.com/blog/sentiment-analysis-in-snowflake-with-openai-gpt-3-5-turbo
26. https://www.chaosgenius.io/blog/snowflake-cortex-analyst/
27. https://docs.snowflake.com/user-guide/snowflake-cortex/aisql?lang=de%2F
28. https://quickstarts.snowflake.com/guide/using_snowflake_and_azure_openai_for_a_rag_workflow/index.html

29. https://github.com/Snowflake-Labs/sfguide-getting-started-with-cortex-agents
30. https://docs.snowflake.com/user-guide/snowflake-cortex/llm-functions?lang=ja%253fwtime%253fwtime%253fwtime%253d%257bseek_to_second_number%257d%253fwtime
31. https://www.youtube.com/watch?v=9p77D0zZNiU
32. https://www.techmahindra.com/insights/views/snowflake-cortex-analyst-conversational-bi/
33. https://docs.snowflake.com/en/sql-reference/functions/complete-snowflake-cortex
34. https://omni.co/blog/snowflake-cortex-and-omni
35. https://pmc.ncbi.nlm.nih.gov/articles/PMC12129157/
36. https://arxiv.org/html/2506.18602v1
37. https://document360.com/blog/text-embedding-model-analysis/
38. https://arxiv.org/html/2508.07520
39. https://swirlaiconnect.com/understanding-semantic-similarity-and-its-impact-on-information-retrieval/
40. https://aclanthology.org/2021.icon-main.68.pdf
41. https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0325177
42. https://www.pingcap.com/article/top-10-tools-for-calculating-semantic-similarity/
43. https://pubmed.ncbi.nlm.nih.gov/39255153/
44. https://insight7.io/how-do-ai-tools-detect-conversational-complexity/
45. https://chaoliu-cl.r-universe.dev/conversim
46. https://www.snowflake.com/en/blog/multimodal-data-analysis-cortex-ai/
47. https://www.matillion.com/blog/sentiment-analysis-in-snowflake-with-openai-gpt-4
48. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/b95271bb012eef5ae9f94838b19abd82/1e932cb6-baa8-4661-99b5-2412f032217a/72d560af.sql
49. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/b95271bb012eef5ae9f94838b19abd82/dceca6f3-bce5-4f76-acd4-d6c95ce17341/c8ffa4ac.py
50. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/b95271bb012eef5ae9f94838b19abd82/97b9b794-e394-4b4d-bd07-264f8a9e8ae1/83442c22.py
51. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/b95271bb012eef5ae9f94838b19abd82/97b9b794-e394-4b4d-bd07-264f8a9e8ae1/68890722.py
52. https://ppl-ai-code-interpreter-files.s3.amazonaws.com/web/direct-files/b95271bb012eef5ae9f94838b19abd82/803f31a8-17e8-4688-9984-5149806db06a/6883fda3.md
53. Screenshot-2025-08-20-at-21.36.31.jpg
54. Screenshot-2025-08-20-at-21.53.44.jpg
55. Screenshot-2025-08-20-at-21.53.44.jpg
56. Screenshot-2025-08-20-at-22.00.48.jpg
57. SYMBI-GitHub-repo-analysis.html