📘 Project Log: `LLM_RAG_Financial_Dashboard.ipynb`

1️⃣ What Was Done

- ✅ Loaded a cleaned financial dataset (CSV format).
- ✅ Created a function `human_readable_volume(v)` to format large volume numbers (e.g., `24409527 → 24,409,527`).
- ✅ Used a loop to apply that function and created a new column: `Formatted_Volume`.
- ✅ Created another function `create_readable_sentence(row)` to convert each row into a natural language summary.
- ✅ Used `df.iterrows()` to loop through each row and call the sentence function.
- ✅ Stored each sentence in a list called `text_chunks`.
- ✅ Saved all the summary sentences into a file: `financial_text_chunks.txt` (1 row per line).

2️⃣ Why It Was Done

- GPT and other LLMs understand and work best with **plain natural language**, not raw tables.
- Turning structured financial data into **text summaries** allows LLMs to reason over your dataset.
- These text chunks will be used in a **RAG (Retrieval-Augmented Generation)** setup:

    - The LLM will search for relevant sentences from your file before answering a user's query.

- It builds the foundation for enabling **chat-style Q&A over your own financial data**.

3️⃣ Challenges Faced & How They Were Resolved

❌ Error:

```
TypeError: tuple indices must be integers or slices, not str
```

- **Cause**: Looping with `for row in df.iterrows()` gave a tuple, not a DataFrame row.
- ✅ **Fix**: Replaced with `for _, row in df.iterrows()` — now `row` is a proper Series.

❌ Issue:

Newline characters not working — all sentences appeared on one line in the text file.

- **Cause**: Used `"\\n"` instead of `"\n"` when writing to file.
- ✅ **Fix**: Corrected to `f.write(line + "\n")` to properly separate lines in `financial_text_chunks.txt`.

4️⃣ Why Each Step Was Important

Step	Purpose
`human_readable_volume(v)`	Adds commas to large numbers so text is easy to read
`Formatted_Volume` column	Avoids repeating formatting each time a sentence is created
`create_readable_sentence(row)`	Turns each table row into a clear sentence GPT can understand

`df.iterrows()`	Lets you access and work with each row of data one by one
`text_chunks` list	Collects all summaries in one place for export
`financial_text_chunks.txt`	This is your custom knowledge base for RAG search

5️⃣ Outcome & Current Project Status

Task	Status
Data loaded and cleaned	✅ Done
Volume formatted	✅ Done
Sentences generated	✅ Done
Text chunks saved to file	✅ Done
Ready for embeddings	✅ YES ✅

🔜 Next Step:

- Generate **embeddings** from `financial_text_chunks.txt`
- Store in **vector DB** (FAISS or ChromaDB)
- Connect with **LLM using LangChain or LlamaIndex**
- Build your **Streamlit interface for querying the data**