Lab 1 — AI-Assisted SQL Foundations (with Global Superstore on BigQuery)

When: Week 3 (Thu)
Goal: Use Gemini as a co-pilot to write and understand basic SQL queries. Load the **Global Superstore** dataset into **BigQuery** using Colab, then answer fundamental business questions with AI-assisted SQL.

A. Setup & Data Load (Colab → BigQuery)

1) Create a Colab notebook named Lab1_AI_Assisted_SQL.ipynb.
2) Install libraries: google-cloud-bigquery, pandas-gbq, kagglehub.
3) Authenticate to GCP and set your PROJECT_ID.
4) Download dataset from Kaggle:
`path = kagglehub.dataset_download("anandaramg/global-superstore")`
5) Load CSV (Global Superstore.csv or Orders.csv) into BigQuery as table *sales*.
6) Sanity check with SELECT COUNT(*).

B. Gemini-Assisted SQL (Core Prompts & Queries)

Use Gemini to generate SQL, then run it in Colab. Paste the prompt you used before each query. **1) The "What" Question:**
Business Question: Unique product sub-categories in the West region.
Prompt: Generate query with SELECT DISTINCT Sub_Category WHERE Region='West'. **2) The "How Many" Question:**
Business Question: How many orders were placed in each Ship Mode?
Prompt: Generate query with COUNT(*) GROUP BY Ship_Mode. **3) The "Who is Best" Question:**
Business Question: Top 5 most profitable customers.
Prompt: Generate query with SUM(Profit) GROUP BY Customer_ID ORDER BY DESC LIMIT 5.

C. Challenge — Construct Your Own Prompts

1) What is the average discount for products in the 'Technology' category sold in the 'East' region?
2) How many unique customers has each 'Segment' served?
Author your own prompt → Generate SQL → Run → Interpret results.

D. Reflection (DIVE mindset)

Write short notes:
- **Discover:** First relevant answer.
- **Investigate:** Alternate query/angle.
- **Validate:** Where Gemini's first answer was wrong, how you checked.
- **Extend:** New business question you would ask next.

E. Submit

- Push Lab1_AI_Assisted_SQL.ipynb to your team GitHub repo.
- Submit the repo link on Brightspace.

Troubleshooting Tips

- If CSV load fails, try encoding='latin1'.
- Use pd.to_datetime(..., errors='coerce') for dates.
- Ensure PROJECT_ID and dataset name match.
- Keep queries scoped to control costs.