Delivery Cadet Challenge

Challenge Overview

In this challenge, you will design and build an **AI-powered data exploration agent** capable of answering **natural language questions over a structured database**. The agent will dynamically generate and execute SQL queries, return results conversationally through a ChatGPT-style interface, and optionally visualize results and run advanced Python analysis.

This challenge is designed to showcase your ability to **design and build modern agentic systems end-to-end**, integrate cutting-edge AI tooling, and solve real-world data problems in a clean, scalable way.

You will receive a dataset and a pre-built UI. Your task is to build the **AI agent backend**, connect all components, and demonstrate your solution in a short walkthrough video.

What You Will Build

Part 1. Data Model & Database Setup

You will receive a set of **CSV files** that together represent a **data model**.

A data model defines how multiple datasets (tables) relate to each other, including keys, relationships, and business meaning.

Your tasks:

- Load the provided CSV files into a **local SQL database** (SQLite, MySQL, or PostgreSQL)
- Write a **Python script** that:
  - Creates tables
  - Identify primary and foreign keys in tables if exist
  - Loads data
  - Can be reused for different datasets in the future

Part 2. AI Agent with LangGraph

You will build an **AI agent using LangGraph** that can:

- Accept **natural language questions**
- Convert those questions into **valid SQL queries**

- Execute queries against your database through the use of a tool
- Return results in a conversational format

Key requirements:

- Must be built using **LangGraph**
- Must use **LangSmith** to visualize execution traces
- Prompting must be **dataset-agnostic**
- The system must be easy to adapt to new databases
- **The system must have a guardrail** that obscures any personal name from the final response.

Part 3. User Interface Integration

You will connect your agent to a **pre-built chat UI**, allowing a user to:

- Ask questions conversationally
- See the agent's answers and results in real time

The UI will behave like a ChatGPT-style interface powered by your agent.

Extra Points (Optional Enhancements)

You will receive additional credit if you implement:

- A **plotting tool** that allows the agent to visualize SQL query results
- A **Python execution tool** for deeper analysis and transformations
- A feature where the agent proactively uncovers and presents **interesting insights** about the dataset

---

What We Are Looking For

We are looking for engineers who enjoy building real systems, not just experimenting with prompts.

We value people who can:

- Design and build **robust, innovative agentic systems**
- Navigate technical ambiguity and **solve real engineering problems**

- Write **clean, modular, scalable code**
- Use modern AI and data tools with confidence
- Demonstrate strong **engineering judgment and curiosity**

We care deeply about **how you think, structure problems, and evolve solutions**, not just whether the demo works.

Technical Requirements

Your solution must:

1. Built using Python
2. Be built using **LangGraph**
3. Use **LangSmith** for trace visualization
4. Run **locally** using:
   - LangGraph
   - LangGraph Agent Chat UI (Note you may need to add additional features to display images correctly.)
5. Be **dataset-agnostic** (no dataset-specific hardcoded prompts)
6. Allow easy swapping of datasets without major refactoring

Recommended Technology (Not Mandatory but please provide justification of your choices in the video)

LLM Providers with free tokens

- Cerebras Cloud
- Home - GroqCloud

Databases

- SQLite
- MySQL
- PostgreSQL

Visualization

- Plotly

- Matplotlib
- Seaborn
- Altair

Python Execution

- Pyodide

Example Questions:

Easy (Single Table Queries)

- What are the top 3 most popular products by total quantity sold?

- How many customers are there in each continent?

- What payment methods are used and how often is each one used?

Medium (Two-Table Joins)

- Which country generates the highest total revenue? Show all countries ranked by revenue.

- Who are the top 5 customers by total spending? Show their names and total amount spent.

- Which franchises have received the most customer reviews? List the top 5 franchise names with their review counts.

Hard (Multi-Table Joins + Visualizations)

- Create a bar chart showing total revenue by supplier ingredient. Which ingredients are associated with the highest-selling franchises?

- Visualize daily sales trends over time. Create a line chart showing total revenue per day and identify any patterns.

- Create a visualization comparing revenue by franchise size (S, M, L, XL, XXL). Also show the average transaction value for each size category.

Expert (Window Functions with PARTITION BY)

- For each country, rank the products by total revenue and show only the top-selling product in each country. Use a window function with PARTITION BY.

- Calculate the running cumulative revenue per day, and create a visualization showing both daily revenue and the cumulative total over time.

- For each transaction, calculate how its total price compares to the average transaction value for that franchise (as a percentage). Show the top 10 transactions that most exceeded their franchise average.

Final Deliverables

You must submit:

1. Working Codebase

- Full solution in a **local Git repository**
- Clear setup instructions in a README.md

2. Short Demo Video (5–10 Minutes)

Your video should demonstrate:

- Database setup and data loading
- The agent answering real user questions
- UI interaction
- Optional visualization and Python tools
- Optional examples of insights discovered by the agent

We would also like you to include a brief explanation of why you chose the technologies you used (database, visualization library, etc.), along with a discussion of any limitations you observed in your system and your ideas for how it could be improved.

How to Submit

1. Push your local Git repository to a **private GitHub repository**
2. Add the following GitHub username as a collaborator: QuangNguyen2609 (https://github.com/QuangNguyen2609) to your repository as a collaborator (how to add a collaborator to you're your GitHub repo)
3. Submit your **video link and GitHub repository link** when requested