Adobe

Adobe Emerge

Problem Statement

Hackathon Problem Statement

AI-Powered SQL Query Generator and Error Corrector

In enterprise environments, databases often consist of vast datasets with intricate schemas and complex relationships. Querying such databases efficiently requires expertise in SQL and deep domain knowledge. Writing SQL queries manually, especially for generating insights, is time-consuming, error-prone, and a significant challenge for non-technical users.

This hackathon challenges participants to build an **AI-powered agent** that can:

1. **Generate accurate SQL queries** from **natural language (NL) inputs** by understanding user intent and database schema complexities.
2. **Allow query correction** that are created either because of syntactical error or using wrong attributes and tables

Solutions will be evaluated based on:

- **SQL Accuracy** –Correctness of generated SQL queries.
- **Output Accuracy** –Correctness of results produced by SQL queries.
- **Execution Success Rate** –Ratio of successfully executed queries.
- **Token Efficiency** –Cost-effectiveness in generating results on the test set.
- **Time Efficiency** –Total execution time across the test set per team.
- **Optimization Strategy** –Use of advanced techniques for efficient query generation.

Participants are encouraged to leverage **contextual understanding, NLP models, and AI techniques** to build innovative solutions that simplify database querying for users across industries.

Setup Manual

1. **Install PostgreSQL** on your system and use the SQL script provided by the Adobe team to set up the database.
2. **Set up a Groq account** and generate an API key. Each team member is advised to create their own due to rate limits (refer to the guide for model-specific limits).
3. **Access the training dataset**, which includes two CSV files:
    a. One for **Natural Language (NL) to SQL** conversion.
    b. One for **Incorrect SQL to Correct SQL** transformation.
4. **Train your agentic framework** using these datasets.
5. **Refer to the Submission Format** section to ensure all required collaterals are included in your final submission.
6. Please use the boiler plate code to refer to submission format: https://drive.google.com/file/d/1Jv33yWQK4YNGEGE0Dt3o8KPWopOPDwSv/view?usp=sharing

Postgres setup

Use the following sets to setup postgres

https://www.w3schools.com/postgresql/postgresql_install.php

Setup database

https://www.convert-in.com/import-postgresql-script

To load the database use the following script:
https://drive.google.com/file/d/1MuwvclZeyTKzRURuwDMTNuK3Tp5k_sll/view?usp=sharing

Setup for AI Key (GROQ)

1. Log in to **Groq**
2. Sign up / Create an account
3. Go to API Keys, and click on create API Key
4. Give the API name (like IITDHackathon) and copy the secret key that is generated (be sure to copy it, it won't be shown again)
5. The rate limits for each model can be found **here**

You are allowed to use a model up to 7B parameters

Training data

CSV for NL to SQL

https://drive.google.com/file/d/1Kf2CdRokU8a7v6hQOKTRh82Pzt71vYLw/view?usp=sharing

CSV for Query Correction

https://drive.google.com/file/d/16lufvuIG9_Nj5XZe1O0FLhzmoCoej6Or/view?usp=sharing

Submission Format

1. Python code that will be used for evaluation the code on test set.
2. A Readme briefly describing the approach used to reach the solution and explicitly mentioning the tech stack used.
3. A Low-Level Design for your architecture.
4. For the top teams who qualify, they will be required to share a new GROQ API key (different from the one used at the time of training).

Submission format per task

Format for CSV for NL to SQL task

https://drive.google.com/file/d/1efODPM78UbYkQgNeZczS4cLpaHR0W12S/view?usp=sharing

Format for CSV for query correction

https://drive.google.com/file/d/1jhHclpBLL2lwdhe--ufl9q2DG3UtZXP7/view?usp=sharing

You are allowed to use a model up to 7B parameters.

Code, Create, Conquer –Let the Hackathon Begin!