Project Proposal: Natural Query Language:

1. Project Overview

The NLP-to-SQL Translator aims to simplify the process of querying an SQL database by allowing users to speak or write queries in natural language. The system will leverage a pre-trained language model to understand natural language inputs, generate appropriate SQL queries, and execute them on a connected database to retrieve desired information. This project will demonstrate how users can interact with their databases without needing in-depth SQL knowledge, allowing for more intuitive data retrieval. Additionally, the project demonstrates the feasibility of implementing a generalized translation model or pipeline between natural language and programmatic languages or frameworks, using Structured Query Language (SQL) as the archetype.

2. High-Level Software Architecture

- Architecture Diagram

Alt text

- Application Layer:
  - Frontend User Interface (UI): A simple web interface or command-line tool where users can input natural language queries and view data from the executed SQL queries and/or image from data analysis
  - Backend Service: Communicates with the AI Stack layer and the Database System layer via API calls
    - Its roles include
    1. Pass natural language user input to the LLM
    2. Receive translated SQL query from LLM
    3. Pass SQL query to the database system
    4. Receive results from database
    5. Send images and data to front end UI for viewing by user
    6. Separating the LLM from the actual data to ensure long term support for the data and to explore more reliable cloud options for implementing the AI stack layer, ensuring minimal coupling between LLMs and data for security concerns.
  - Data Analytics Engine: Receives data analysis script (python matplotlib/R script) from LLM and data from database systeam (after retrieving results from the SQL query produced by the LLM) and runs it as part of the backend service to generate an image output, which is passed to the frontend
- AI Stack:
  - LLM Integration (OpenAI API or custom model): takes the user's natural language input and translates it into SQL. The language model (like GPT) should be fine-tuned to understand specific domain language and the database schema
    - modes of fine tuning

            1. Highly engineered prompt
            2. Few shot prompting for better reliability
        - locally hosted options
            1. LLAMA 3.1

- Database System Layer:

    - Database Management System (DBMS)
        - Database Schema Representation: This layer holds the schema of the SQL database, which is provided to the LLM as context to assist in query translation
        - SQL Execution Engine: This module receives the generated SQL query, connects to the database, and executes the query to retrieve results
    - Database
        - Storage location of all data for querying

---

3. Implementation Details

- Sequence Diagram (of main success scenario)

Alt text