Summer 2024 CS 687 Capstone Project

Proposal  
Using RAG to Chat with a SQL Database

Joshua Emery

Advisor: Sherin Aly

MS in Computer Science  
School of Technology & Computing (STC)

City University of Seattle (CityU)

emeryjoshua@cityuniversity.edu, alysherin@city.edu

**Keywords:** Retrieval Augmented Generation, RAG, SQL, Langchain, Chatbot, LLM, Generative AI

**1.** **INTRODUCTION**

**Problem Statement**

Large Language Models (LLM) have enabled users to generate previously impossible outputs from simple plain text inputs or prompts. Everything from images, videos, documents can now be easily created without specialized training on the part of the user.

It is possible to provide additional context to LLM responses using a technique called Retrieval Augmented Generation (RAG). RAG has been deployed to enable users to chat with textbooks, company policies, or nearly any sort of reference document to supplement the available data to the LLM. These solutions can be fairly simple as the format of the specialized text is similar to the format that the LLM was trained on. It is natural language. Would it be possible to enable our users to chat with data that is not in such an accessible format?

Proper querying of a SQL database can be a skill intensive process requiring trained expertise. Should a client request a new report or filtering option in a deployed full stack application it can lead to creating of new queries, API endpoints and UI. Adding this functionality can be expensive and time consuming. Would it be possible to simplify this process through LLMs. Could RAG techniques enhance the accuracy of an LLM solution enough for these results to be viable? What transformations would a traditional SQL based, relational database store need to undergo in order to accomplish this?

This project aims to add a specialized chatbot to an existing full stack web application. The chatbot would primarily enable communication between the user and a relational database allowing for retrieval of information from the application data in simple English.

**Motivation**

The overwhelming majority of the worlds application data is still housed in relational SQL based databases. The format of these databases does not easily translate to trainable LLM data. During this development process multiple techniques will be researched. Perhaps the LLM is used only to write SQL Queries which are then executed to retrieve the data? Should the data be transformed into a NOSQL or Vector solution first and then queried by the LLM? If we are able to enable easy communication from plain text to SQL it will allow non-technical users unprecedented access to data, they did not previously have.

**Value to the Student**

As the sole volunteer web developer for the Ray of Hope homeless shelter in Auburn Washington, I would like to enable my users to access previously unavailable information from the SQL Server based database without needed for new development.

**Alignment to Program Outcomes**

This project will need considerations around data privacy and ethical use of Artificial Intellgence. Developed solution must respect user data and operate with transparency. In-depth knowledge of the following topics will be necessary for successful completion:

SQL Databases

Full Stack Development

LLMS

RAG Techniques

Vector Embeddings

A comprehensive report and presentation will be provided where the development process, topics researched, and results will be communicated. We will address the significant challenge of simplifying SQL database communications through LLMs. This is an innovative approach to long standing problem in software development.

**2.** **REFERENCE**

Gupta, M. (2024). *LangChain in Your Pocket : LangChain Essentials: from Basic Concepts to Advanced Applications.* (1st ed.). Packt Publishing, Limited.

Kanjee, R. (2024). *Building smart chatbots with LangChain.* ([First edition].). Packt Publishing.