Talk to Database

CMPE585 – Advanced Topics in Natural Language Processing

Spring Semester

Final Project

Yeşim Yiğitbaşı

124750004

Dept. of Computer Engineering

yesim.yigitbasi@bilgiedu.net

1. Introduction

This project explores a system that can analyze and answer user questions about a structured database using natural languages. The system combines semantic embeddings, large language models and SQL databases and allows its users to ask free form questions as it can generate SQL queries and produce clear outputs. This way non-technical users can understand and explore complex databases. This project, Talk to Database connects data querying and natural language processing.

The core of the system is built with Python, Streamlit and Groq API for accessing LLMs. Similarity between question and data column are handled by comparing cosine scores using SentenceTransformers, which allows to match user queries with the dataset. Through a web application user can choose a topic so that the system suggests questions that can be asked to the database, or users can ask their own questions. The web app displays the automatically generated SQL query, its results in a table form and an overall summary of the result displayed.

Similar application in the literature is Vanna AI, which is an open-source Python RAG tool for SQL generation and related functionality[1]. Vanna trains a RAG model based on the given data and answers questions. In the training phase it is trained on related documentation, example SQL queries and DDLs. Then embeddings are created and stored as metadata in a vector database. When user asks a question it generates embeddings, finds related documentation, constructs a prompt for LLM to generate an SQL. Later the output of the SQL query can be structured based on what is asked.

[1] https://vanna.ai

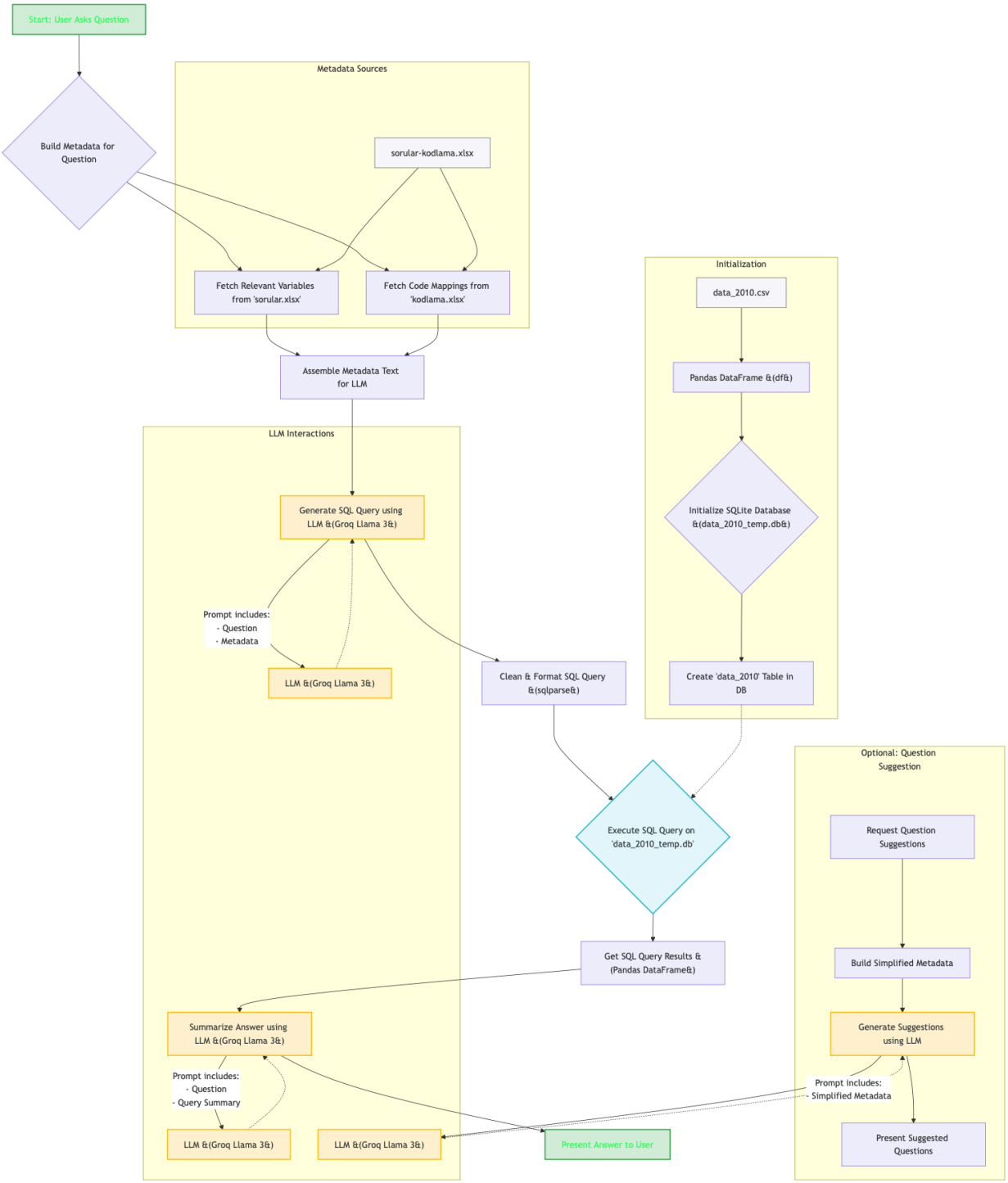

Figure 1 – Flowchart

2. Methodology

2.1. Dataset

The dataset used for this project is a single table exported as a csv document which has 260 columns and 25640 rows. The dataset contains survey data from the year 2010 about a questions and their answers about variety of topics. Each column represents a question in the survey and each row represents a person that is answering the questions.

In a separate excel file with two sheets, the records in the dataset are explained in detail. First sheet matches column names with the actual survey questions and more information about the type of data stored in the column. Second sheet matches coded values in the dataset with the actual answers to survey.

2.2. Frontend

The frontend was built using Streamlit for users to interact easily with their database. The users can choose among 7 different topics ("political", "social", "habits", "economic", "demographic", "education", "environmental") to get suggestions about what they can ask about the data. After the choosing a topic the user can ask for question suggestiong based on the topic they chose. The users are not limited to the suggested questions, they can ask other questions as well. After the question is entered the users can click on run query and display the generated SQL query, the results in table form and a short summary.

2.3. Backend

When the user chooses a topic and asks to get question suggestions, the first 5 columns that match the chosen topic are retrieved. Finding these columns are done with a hybrid approach. First, prefiltering is done and embeddings are created for the labels in questions sheet in excel file and also for the chosen topic. For the embedding model, SentenceTransformer (all-MiniLM-L6-v2) encoder is used. Then the cosine similarity between the topic embeddings and the label embeddings are compared. The first 75 labels with the highest cosine similarity are then sent to the LLM to determine the most similar 5 columns. These top most similar columns are used to first create a metadata with the questions and the coded values and their meanings. A prompt along with the metadata is given to the LLM, the model used here is called llama3-70b-8192. The LLM generates ten relevant question suggestions that can be answered with a SQL query.

The user can use the suggested question or write their own question. Either way after the question is entered the most relevant columns to the question is matched with the cosine similarity again. It returns multiple columns since the question can be related to multiple columns. Based on the returned columns metadata is created from the excel file to be given to the LLM for SQL query

generation. The same LLM is used for query generation and LLM is prompted along with the metadata. The SQL query is then executed and results are displayed in a table form in the UI. A summary of the question and the output is also generated my prompting the LLM.

The system uses the Groq API to interact with the llama3-70b-8192 model, a highly capable LLM optimized for reasoning and code generation. Prompt engineering plays a crucial role in shaping the output. Prompts include top 5 most relevant dataset variables (based on semantic similarity), labels, types, and value-label mappings. Prompts are kept concise and detailed to reduce hallucination and improve output accuracy.

2.4. Prompt Engineering

Prompt engineering principles were used to construct the prompts given to the LLMs for the 4 tasks that LLMs were used for, which are picking most relevant columns from the prefiltered columns, suggesting questions based on the metadata, generating the SQL statement from the question and summarizing the query output. The LLM was instructed to have a data analyst persona. Instructions were given specifically for the task such as “Write a correct SQLite query…” or “Suggest 10 diverse questions…”. For the context it was instructed to use the constructed metadata. Data format was described in detail such as “Return a comma separated list of column names…” or “Return only a numbered list…”. For the summary generation, LLM is instructed to give summary in precise and informative tone for a non-technical audience.

2.5. Examples

Here are some of the questions that the model generated depending on the chosen topic. Figures 2-7 are screenshots from the web application that show the outputs of these questions.

- Figure 2 – What is the distribution of education levels among respondents?
- Figure 3 – Is there a correlation between respondents' education levels and their mothers' education levels?
- Figure 4 – What is the distribution of responses to the question "Türkiye'de politik açidan olumlu gelismeler oluyor, Türkiye iyi yolda"?
- Figure 5 – What is the average score of respondents on the Statükoculuk Ekseni?
- Figure 6 – What is the distribution of responses to "Genel olarak saglik durumumdan memnunum" (B1003_s19)?
- Figure 7 – How many individuals consider themselves to be in the middle-income group (3. dilim)?

📊 Talk to your data

Pick a topic:

demographic

💡 Suggest questions

Ask a question about the dataset:

What is the distribution of education levels among respondents?

▶ Run SQL Query

SQL query

```
SELECT "B1003_s3.1",
       COUNT(*)
FROM data_2010
GROUP BY "B1003_s3.1";
```

Results

	B1003_s3.1	COUNT(*)
0	1 → Lise altı	16413
1	2 → Lise	6468
2	3 → Üniversite	2648
3	99 → Cevap yok	110

Summary

Based on the query results, here's a concise explanation of the distribution of education levels among respondents:

The majority of respondents (64.1%) have not completed high school (Lise altı), indicating a relatively low level of education among the surveyed population. About 20.3% have completed high school (Lise), while only around 8.5% hold a university degree (Üniversite). A small percentage (0.4%) did not provide an answer (Cevap yok). This distribution suggests that the respondents may face educational barriers or have limited access to higher education opportunities.

Figure 2

Suggest questions

Ask a question about the dataset:

Is there a correlation between respondents' education levels and their mothers' education levels?

▶ Run SQL Query

SQL query

```
SELECT "B1003_s3.1",
       "B1205_s4",
       COUNT(*) AS COUNT
FROM data_2010
GROUP BY "B1003_s3.1",
         "B1205_s4"
ORDER BY "B1003_s3.1",
         "B1205_s4";
```

Results

	B1003_s3.1	B1205_s4	COUNT
0	1 → Lise altı		1758
1	1 → Lise altı	99	14655
2	2 → Lise		666
3	2 → Lise	99	5802
4	3 → Üniversite		294
5	3 → Üniversite	99	2354
6	99 → Cevap yok		9
7	99 → Cevap yok	99	101

Summary

Based on the query results, there is a strong correlation between respondents' education levels and their mothers' education levels. Here are the key takeaways:

- The majority of respondents (around 70%) have mothers who also have a high school education or lower (Lise altı or Lise). This suggests that there is a strong intergenerational pattern of education levels, with respondents tending to follow in their mothers' educational footsteps.
- There is a significant jump in respondents' education levels when their mothers have a university education (Üniversite). This suggests that having a mother with a university education can be a strong predictor of a respondent's own university education.
- The "no response" category (Cevap yok) is relatively small, indicating that most respondents were able to provide information about their mothers' education levels.
- The correlation is not absolute, as there are some respondents who have higher education levels than their mothers, and vice versa. However, the overall pattern suggests that there is a strong relationship between the two variables.

Figure 3

Suggest questions

Ask a question about the dataset:

What is the distribution of responses to the question "Türkiye'de politik açıdan olumlu gelişmeler oluyor, Türkiye iyi yolda"?

▶ Run SQL Query

SQL query

```
SELECT "B1003_s20",
       COUNT(*)
FROM data_2010
GROUP BY "B1003_s20";
```

Results

	B1003_s20	COUNT(*)
0	-1 → Unknown	18322
1	1 → Kesinlikle yanlış	781
2	2 → Yanlış	2070
3	3 → Ne doğru ne yanlış	1780
4	4 → Doğru	2278
5	5 → Kesinlikle doğru	332
6	99 → Cevap yok	76

Summary

Based on the distribution of responses to the question "Türkiye'de politik açıdan olumlu gelişmeler oluyor, Türkiye iyi yolda" (which translates to "Are there positive political developments in Turkey, is Turkey on the right path?"), here are the key takeaways:**

Most respondents are uncertain or neutral: The largest group (around 18,000) chose "Unknown", indicating a lack of confidence or knowledge about the topic. This suggests that many people may not be paying close attention to political developments in Turkey or may not feel informed enough to form an opinion.

Negative opinions outweigh positive ones: While around 2,600 respondents (4 and 5) believe that Turkey is on the right path, around 2,800 (1 and 2) think the opposite, indicating a slightly more pessimistic outlook.

A significant portion are neutral or undecided: Around 1,780 respondents chose "Ne doğru ne yanlış" (neither right nor wrong), which reinforces the idea that many people are uncertain or neutral about Turkey's political developments.

Very few respondents refused to answer or didn't know: The "Cevap yok" (no answer) category is relatively small, indicating that most respondents were willing to provide an opinion.

Overall, these results suggest that there is a lack of consensus on Turkey's political developments, with many people unsure or divided in their opinions.

Figure 4

📊 Talk to your data

Pick a topic:

social

💡 Suggest questions

Ask a question about the dataset:

what is the average score of respondents on the Statükoculuk Ekseni?

Press Enter to apply

▶ Run SQL Query

SQL query

```
SELECT AVG("B1004_statukoculuk")
FROM data_2010;
```

Results

	AVG("B1004_statukoculuk")
0	16.375171

Summary

Based on the query result, the average score of respondents on the Statükoculuk Ekseni (B1004_statukoculuk) is approximately 16.38. This suggests that, on average, respondents tend to lean towards the middle to upper end of the spectrum when it comes to Statükoculuk Ekseni. This could imply that respondents generally have a moderate to positive view or attitude towards Statükoculuk Ekseni is a Turkish term that roughly translates to "Status Quo Axis". In the context of social sciences, this might refer to their stance on maintaining the current state of affairs or their willingness to challenge the existing power structures.

Without additional context or data, it's difficult to say what this average score exactly means or what the implications are. But in general, this result could be an important indicator of public opinion or sentiment towards the status quo, and could be used as a starting point for more in-depth exploration of this topic.

Figure 5

Pick a topic:

habits

💡 Suggest questions

Ask a question about the dataset:

What is the distribution of responses to "Genel olarak sağlık durumumdan memnunum" (B1003_s19)?

▶ Run SQL Query

SQL query

```
SELECT "B1003_s19",
       COUNT(*)
FROM data_2010
GROUP BY "B1003_s19";
```

Results

	B1003_s19	COUNT(*)
0	→ Unknown	18322
1	1 → Kesinlikle yanlış	303
2	2 → Yanlış	1034
3	3 → Ne doğru ne yanlış	1055
4	4 → Doğru	4273
5	5 → Kesinlikle doğru	607
6	99 → Cevap yok	45

Summary

Based on the distribution of responses to the question "Genel olarak sağlık durumumdan memnunum" (B1003_s19), here are the key takeaways:

- The majority of respondents (around 70%) have a positive sentiment towards their overall health, as they either strongly agree (5 → Kesinlikle doğru) or agree (4 → Doğru) with the statement.
- A notable proportion (around 15%) of respondents are neutral, indicating neither satisfaction nor dissatisfaction with their health (3 → Ne doğru ne yanlış).
- A smaller but still significant group (around 10%) of respondents have a negative sentiment, with 2 → Yanlış indicating disagreement and 1 → Kesinlikle yanlış indicating strong disagreement.
- There is a small proportion of respondents who did not provide an answer (45) or marked it as unknown (around 40%).

Overall, the distribution suggests that most people are generally satisfied with their health, but there is a notable minority with negative sentiments or neutral opinions.

Figure 6

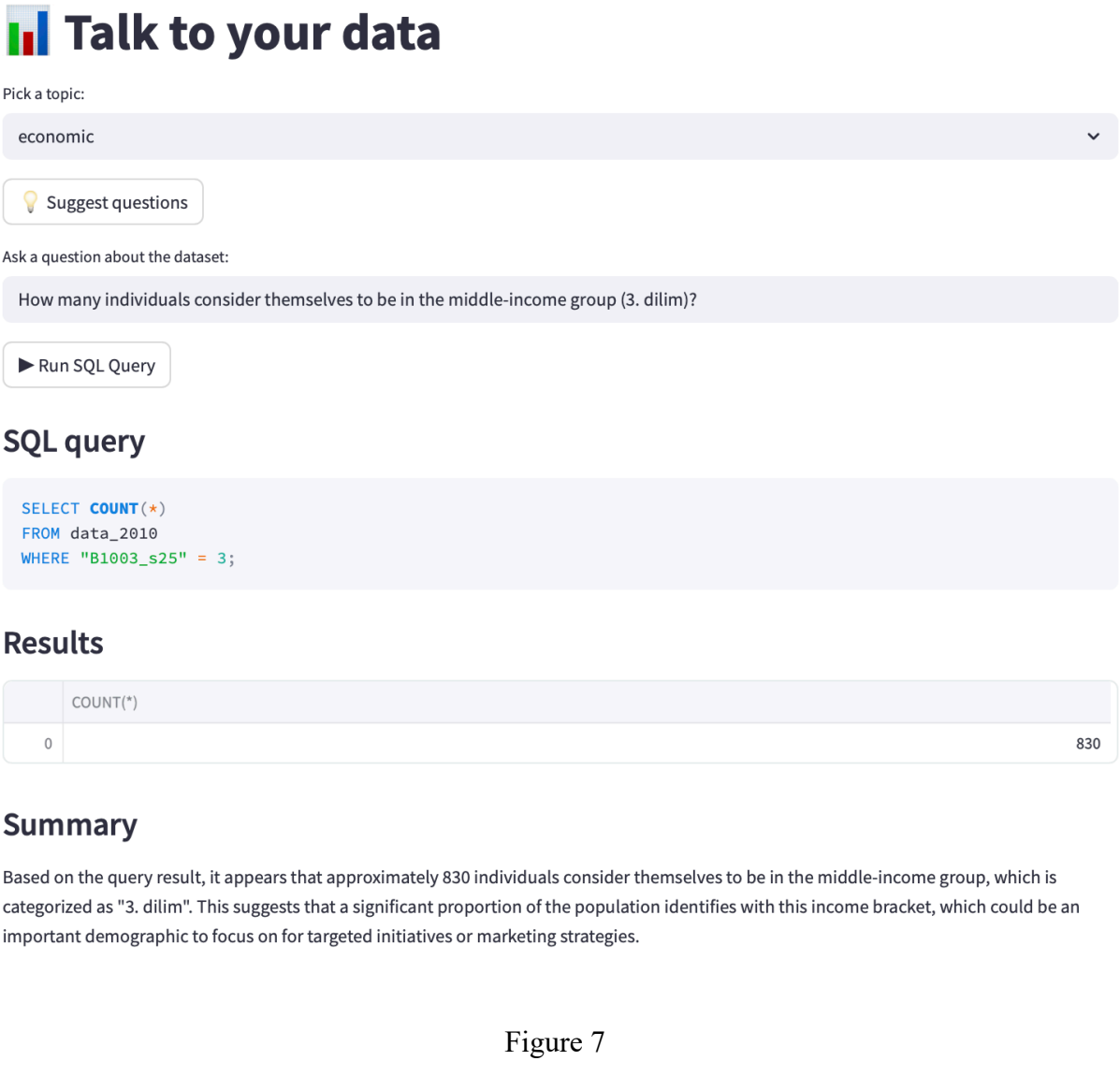
Talk to your data
Pick a topic:
economic
Suggest questions
Ask a question about the dataset:
How many individuals consider themselves to be in the middle-income group (3. dilim)?
Run SQL Query
SQL query
SELECT COUNT(*)
FROM data_2010
WHERE "B1003_s25" = 3;
Results
COUNT(*)
0
830
Summary
Based on the query result, it appears that approximately 830 individuals consider themselves to be in the middle-income group, which is categorized as "3. dilim". This suggests that a significant proportion of the population identifies with this income bracket, which could be an important demographic to focus on for targeted initiatives or marketing strategies.

Figure 7

3. Analysis

The system performs well in suggesting questions and almost all the time it generates relevant questions to the chosen topics. Embedding based comparison works well but it is not perfect. Since the semantic similarity is based on vector distance, relevance matching did not match the topic good enough. That is why a hybrid approach was used which increased the performance in matching columns to the topic. Based on the topic, even though most of the time it can find the correct columns, it might rarely match unrelated columns as well. When it comes to the questions, the model might give topic related questions that cannot be answered with an SQL query.

Answering questions entered by the user follows a similar approach but it is only embedding based. Cosine similarity comparison is works well for matching columns with questions since this time it is comparing two sentences instead of a word but it is not perfect. Since the semantic similarity is based on vector distance, relevance matching might not fully understand the user's intent. The use of semantic similarity via SentenceTransformers provides an intuitive mapping between user queries and the underlying data schema.

Furthermore, although the LLM (LLaMA 3 via Groq API) is capable, it sometimes generates invalid SQL queries with minor syntax errors or inaccuracies, such as referencing non-existent columns or missing table constraints. To address this, a function is used to detect and correct obvious SQL issues, which mitigates some but not all problems. Other than the syntax issues, model can generate queries that does not relate to the question asked. So it can essentially write a valid query but the query might be unrelated to the question asked.

The current design with embedding the entire column metadata and querying it for every question is efficient for small-scale datasets. However, as dataset size and schema complexity grow, the embedding and similarity-matching steps could not perform as well. Giving more columns, metadata and longer prompts might theoretically improve the systems performance however, when I tried the token limitations were exceeded.

As long as a valid output is returned from the SQL query the output and the summary is displayed. If asked in the questions, the groupings and counts are easily interpretable from the table. The summaries are generated based on the question itself and of course the output. These summaries are often well explained, even when the output is odd. It can explain and give an idea to the user about what could not be generated or what can essentially be wrong about the question. If a bad response is generated whether it is from SQL query itself or the summaries, re-executing the code might result in better outputs.

4. Conclusion

Through the use of SentenceTransformers and the LLaMA 3 model via Groq API, the system is able to suggest relevant questions, generate appropriate SQL queries, and summarize results in a user-friendly format. Most suggested questions are logically related to the selected topic, demonstrating the model's ability to infer semantic meaning. This empowers users without SQL knowledge to explore and understand structured data efficiently.

The most crucial part about the whole system is the SQL generation from a user question, and it was observed that the system struggled at that part the most. As it is done in Vanna AI training the LLM on a set of valid question and output pairs might result in better performance for the system. These pairs can be question-columns, question-SQL query, question-summary and etc. Another aspect to improve would be keeping a cache for the questions asked and the outputs. This would decrease the reliance on the LLM itself as the API would not have to be called several times for the same question.

While the system is functional and effective for its scope, limitations such as semantic mismatches, occasional SQL generation errors, and scalability concerns highlight areas for future improvement. Incorporating real-time schema analysis, better error handling, and performance optimization for larger datasets would enhance its reliability and applicability.

This project demonstrates how natural language processing and large language models can be integrated with traditional SQL databases to enable intuitive data querying. By allowing users to ask questions in natural language and returning both the SQL query and the answer, the system reduces the technical barrier to database interaction.