ChatGroq SQL: Conversational MySQL Interface Using LLMs, K-NN Retrieval and Beam Decoding

Shilpa Tarnalle[1]*,Mohammad Sakib Shaik[2], Sarbjot Singh[3], Shaik Nizamuddin Sufiyan[4], Syed Saqeef Hussain[5]

[1]*Professor, Dept of Information Science and Engineering, Guru Nanak Dev Engineering College, Bidar 585403- Karnataka*
[2,3,4,5] *Department of Information Science and Engineering, Guru Nanak Dev Engineering College, Bidar-585403, Karnataka*
tarnalle.shilpa1988@gmail.com

Abstract

This paper presents an interactive framework for natural language querying of structured relational databases, with a focus on MySQL environments. The proposed system employs a hybrid architecture that integrates large language models (LLMs), retrieval-based mechanisms, and advanced decoding strategies to translate natural language inputs into executable SQL queries. A transformer-based LLM combined with cosine similarity–driven k-nearest neighbors (k-NN) retrieval and Retrieval-Augmented Generation (RAG) ensures schema-aware query generation, followed by parsing, execution-based validation, and ranking to improve reliability. Iterative chain-of-thought prompting enables effective handling of complex queries, while natural language summaries enhance interpretability for non-technical users.

Keywords- Large Language Models, Natural Language Interface, SQL Generation, k-NN Retrieval, Beam Decoding, MySQL, FAISS, LangChain.

1. Introduction

Traditionally, accessing structured data stored in relational databases like MySQL has required users to know SQL and understand how database schemas are organized. However, as businesses increasingly rely on large-scale data analytics for real-time decision-making, the need for simpler and more user-friendly ways to interact with data has become more important. As a result, there is growing interest in developing interfaces that allow non-technical users to access, query, and understand data using natural language instead of specialized technical knowledge. This work introduces a comprehensive system addressing these challenges by integrating a transformer-based LLM with retrieval-augmented generation (RAG), schema-aware k-nearest neighbors (k-NN) retrieval, and robust decoding strategies. The goal is to enable users to query MySQL databases using natural language in a conversational format, producing precise, executable SQL queries and interpretable outputs. The system uses subword tokenization (WordPiece or BPE) and dense vector representations, cosine similarity-based k-NN retrieval to surface contextually relevant schema elements, and beam search decoding to maintain multiple high-probability sequences during inference. Generated SQL statements are parsed, executed, and ranked, while iterative chain-of-thought reasoning handles complex or ambiguous inputs. The result is an accessible, scalable, and intelligent conversational interface for structured database interaction.

2. Related Work

The task of translating natural language queries into SQL, commonly referred to as Text-to-SQL, has evolved through rule-based approaches, statistical learning, and deep learning. Early systems like PRECISE [4] relied on hand-crafted rules and deterministic parsing, which lacked scalability and struggled with linguistic variability. Statistical approaches required large annotated datasets such as ATIS and GeoQuery but lacked generalization capability. Encoder-decoder models using RNNs and Transformers (e.g., Seq2SQL [5], SQLNet) demonstrated strong performance; however, these models still required significant task-specific training data and struggled with complex multi-join queries.

Pretrained LLMs such as BERT [1], GPT-3/4 [2], and Codex [3] have shown strong zero-shot and few-shot SQL generation capabilities. PICARD [8] and SmBoP introduced structured decoding mechanisms and execution-guided generation to improve robustness. However, most LLMs operate with limited schema awareness, often resulting in hallucinated table or

column names. Retrieval-Augmented Generation (RAG) [7] emerged as a hybrid solution, combining dense k-NN retrieval with LLMs by injecting retrieved schema context into the generation prompt. The Spider benchmark [6] established a standard for evaluating cross-domain Text-to-SQL systems. Our work builds on these foundations by integrating FAISS-based k-NN retrieval, RAG via LangChain, and beam decoding into a unified conversational MySQL interface.

3. Problem Statement

Accessing and extracting meaningful insights from relational databases such as MySQL has traditionally required expertise in SQL and familiarity with the underlying database schema. This dependency on technical staff creates a bottleneck for business leaders and domain experts. Because they cannot query databases directly, data-driven decision-making is often delayed while waiting for technical teams to translate their questions. While LLMs demonstrate impressive code generation, directly applying them to SQL query generation presents several challenges: (1) Because general-purpose language models are not trained on your specific database structure, they often produce queries that follow SQL rules but refrence; (2) natural language queries are often ambiguous, requiring contextual grounding; (3) one-shot LLM-based systems lack iterative refinement capability; (4) greedy decoding leads to brittle SQL outputs without exploring a diverse solution space; and (5) non-technical users need human-readable summaries, not just raw query results

4. Proposed Methodology

The proposed system integrates LLMs, semantic retrieval, context-aware generation, and robust decoding across several key stages as illustrated in Fig. 1.

A. User Input and Preprocessing

The query is tokenized using WordPiece or Byte Pair Encoding (BPE), and tokens are converted into dense embedding vectors via a pretrained Hugging Face embedding model.

B. Schema and Intent Retrieval Using k-NN (FAISS)

A cosine similarity-based k-NN retrieval mechanism using FAISS (Facebook AI Similarity Search) compares the user query embedding against a pre-indexed vector database containing table names, column names, and previously asked query intents. The top-k most semantically relevant schema elements are selected and used as additional context for the LLM, enabling schema-aware generation and eliminating hallucinated references.

C. Retrieval-Augmented Generation (RAG) via LangChain

Retrieved schema elements and contextual information are injected into the LLM prompt using a RAG architecture implemented via LangChain. This grounding step ensures that the model's generation is aligned with the actual database structure and domain semantics, significantly reducing the risk of generating queries referencing non-existent tables or columns.

D. SQL Generation via LLM with Beam Decoding

A transformer-based LLM (Llama 3.3 via the Groq inference platform) generates SQL queries based on the enriched RAG prompt. Beam search decoding explores multiple high-probability SQL sequences simultaneously rather than selecting a single greedy output, increasing the robustness of the final query. Optional top-k or top-p (nucleus) sampling adds diversity. Groq's hardware-optimized inference achieves up to 185 output tokens per second for 70B-class models, ensuring real-time conversational responsiveness (see Fig. 3).

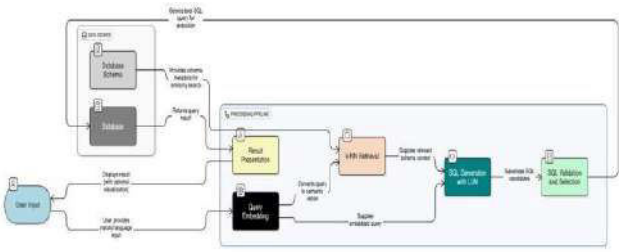

Fig. 1. System Architecture and Processing Pipeline of ChatGroq SQL

E. SQL Parsing, Validation and Ranking

Each generated SQL candidate is parsed to ensure syntactic validity, then executed against the MySQL database. Queries resulting in errors or empty results are filtered out. This process ensures the user receives only accurate and executable query results.

F. Chain-of-Thought Reasoning and NL Summary

For complex or ambiguous queries, chain-of-thought prompting decomposes the problem into logical sub-steps before synthesis, significantly improving the LLM's handling of multi-part natural language inputs.

Multi-turn conversational context is preserved across sessions for enhanced intent resolution.

5. Results and Discussion

The ChatGroq SQL system was evaluated using the Chinook MySQL database, a standard benchmark containing tables for customers, invoices, tracks, artists, and albums. As shown in Fig. 2, simple filtering queries (e.g., "list customers from Canada") were handled with single-table SELECT statements, while complex multi-part queries involving outer joins, aggregations, and NULL handling (e.g., listing top customers by total spend including those with zero purchases) were accurately resolved with COALESCE and LEFT JOIN constructs.

FAISS-based k-NN schema retrieval significantly reduced hallucinated column and table names compared to baseline LLM-only generation. Beam decoding consistently outperformed greedy decoding for multi-join queries. The Groq inference platform delivered substantially lower latency, with throughput benchmarks at 185 tokens/second for 70B-class models, far exceeding Anyscale (66 tokens/s), Together.ai (65 tokens/s), Fireworks.ai (40 tokens/s), and AWS Bedrock (21 tokens/s), enabling real-time responsiveness even for complex queries.

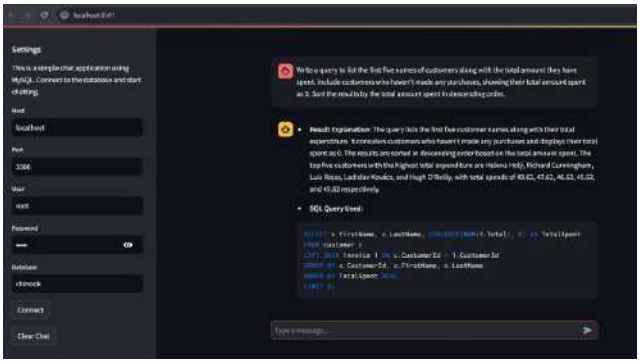

Fig. 2. ChatGroq SQL Interface Showing Complex Multi-Join Query with NL Summary

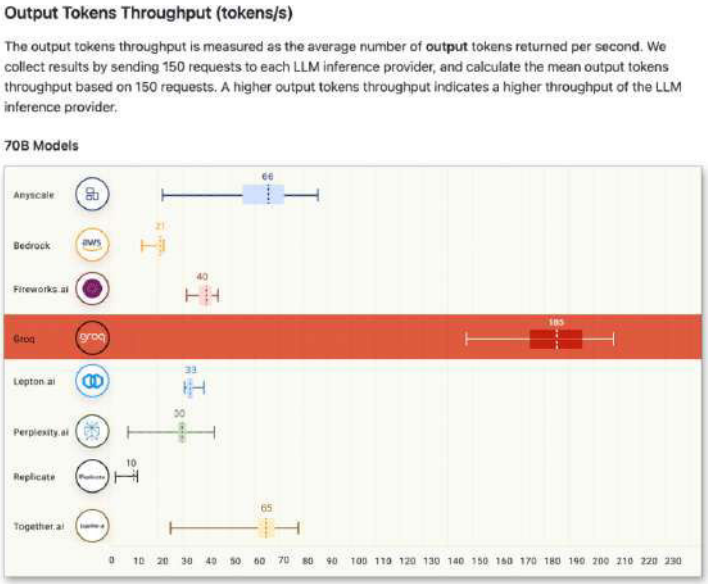

Fig. 3. Groq Output Token Throughput (tokens/s) Comparison Across LLM Inference Providers

The modular system architecture—with agent.py (core logic), llm.py (Groq integration), db.py (MySQL connection), and app.py (Streamlit UI)—enabled clean separation of concerns and ease of extension. Environment variable management ensured secure credential handling. Dynamic database connections supported runtime configuration without restarts. Natural language summaries proved highly effective for non-technical users, clearly explaining both query logic and result interpretation. Future enhancements include multi-database support (PostgreSQL, MongoDB), voice-based querying, and data visualization dashboards for chart and graph outputs.

6. Conclusion

This paper presented ChatGroq SQL, a conversational MySQL interface that democratizes access to structured relational data through natural language. By combining transformer-based LLMs (Llama 3.3 via Groq) with FAISS-based k-NN schema retrieval, Retrieval-Augmented Generation via LangChain, and beam decoding strategies, the system generates accurate, schema-aware, and executable SQL queries from plain English inputs. The integration of iterative chain-of-thought prompting handles complex multi-part queries, while natural language summaries enhance interpretability for non-technical stakeholders.

Experimental evaluation on the Chinook benchmark demonstrated high query accuracy across both simple and complex scenarios. The Groq inference

platform's exceptional throughput (185 tokens/s) ensured real-time conversational responsiveness. The proposed framework is modular, scalable, and readily extensible to other relational databases and enterprise data environments. This work establishes a solid basis for future advancements in natural language database interfaces, with upcoming enhancements such as support for multipule databases, voice-enabled interaction, and visual data exploration, aiming to expand access to data-driven insights for users of all technical skill levels.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2018.

[2] T. B. Brown et al., "Language Models are Few-Shot Learners," arXiv:2005.14165, 2020.

[3] M. Chen et al., "Evaluating Large Language Models Trained on Code," arXiv:2107.03374, 2021.

[4] J. M. Zelle and R. J. Mooney, "Learning to Parse Database Queries using Inductive Logic Programming," in Proc. AAAI, 1996, pp. 1050–1055.

[5] V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning," arXiv:1709.00103, 2017.

[6] T. Yu et al., "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," arXiv:1809.08887, 2018.

[7] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Proc. NeurIPS, 2020.