Team 9 Project Proposal

Explainable Natural Language Query Interface for Relational Databases Using a Multi-Agent System

With the growing adoption of Large Language Models (LLMs), many professionals now use AI tools to assist with tasks such as database querying and SQL code generation. However, these models often produce generic SQL syntax without being able to fully understand the underlying schema, additionally, they operate as black boxes, offering little transparency into how results are derived.

Key Challenges

- Users receive outputs without understanding the reasoning process, this lack of explainability creates uncertainty, particularly when users need to interpret complex queries across diverse and unfamiliar database schemas.
- Unlike prior studies where models benefit from exposure to the same schemas during training, our challenge lies in testing whether a system can generalise not only to new SQL queries, but also to unseen database schemas and domains.

To address these challenges, we propose developing an ***Explainable Natural Language Interface for Database Exploration***, providing a user-friendly interface for database access, enabling users to query complex databases using natural language, lowering the barrier for non-technical users and reducing reliance on SQL expertise.

Key Benefits

- Explainable query process and results, the multi-agent system will be able to provide reasoning process, presenting how the queries are formed and how results are retrieved, significantly boosted user's trust in the system's output. Users can also refine or guide the system's output, creating an collaborative data exploration and decision-making process.
- Generality to the new database/domain, this setup more realistically reflects real-world scenarios, where systems should handle unfamiliar private database and understand given schema, without prior domain-specific training.
- The use of a Multi-Agent System allows for flexibility and scalability, where individual agents can be improved through reinforcement learning / prompt engineering to handle different task and objectives.

Deliverables & Timeline & Costs

Deliverables

This proposal aims to deliver an **Intelligent and Explainable Database Exploration Tool** that enables users to input natural language queries and receive not only the correct results, but also the underlying reasoning process. This includes the identification of relevant tables, highlighted records, and explanations of why these records were selected and presented.

Unlike traditional systems that simply output the result or generate SQL code, this solution prioritises reasoning, transparency, and generalisability across diverse database schemas.
The system will employ a **Multi-Agent System (MAS)** architecture, with each agent performing a specialised role as follows:

- **Agent A: Database Selector**
  Identifies and selects the most relevant database(s) from a pool, using the user's query and schema embeddings to ensure the correct context for downstream processing.
- **Agent B: Table & Column Selector**
  Analyzes the selected database's schema and the user's intent to determine which tables and columns are pertinent to the query, ensuring that only the necessary data is targeted.
- **Agent C: SQL Generator**
  Constructs accurate and executable SQL queries based on the user's natural language input and the tables/columns identified by Agent B, translating intent into precise database instructions.
- **Agent D: Record Retriever & Response Formatter**
  Executes the generated SQL query, retrieves the relevant records, and formats the results. This agent also provides clear explanations and highlights, making the reasoning process and selected data transparent to the user.

The final deliverables by the end of **Week 12** will include:

- A fully functional multi-agent NL-to-SQL system tested on the *Spider v1* dataset (focusing on easy and medium difficulty questions).
- An interactive web-based dashboard where users can input natural language queries and receive relevant tables with highlighted records. Where possible, the interface will also provide clear explanations for why each table, record, or component was selected, making the reasoning process transparent.
- A comprehensive code repository containing modular implementations of each agent, as well as data processing scripts, training logs, and evaluation reports to support reproducibility and further development.

- A final oral presentation, which will include a live demonstration of the system in action, showcasing its features, explainability, and user interface, as well as an explanation of the technical approach and project outcomes.

Timeline

Proposal deadline: End of **Week 4**
Project deadline: End of **Week 12**

Week(s)	Task Name	Description
W3–W4	Research Tools & Frameworks	Investigate multi-agent system architectures (e.g., LangChain, custom pipelines), evaluate how agents will communicate and be orchestrated, and determine the most effective frameworks and tools for building and integrating the pipeline.
W3–W4	Project Proposal & Scoping	Collaborate with the client to clarify project objectives, deliverables, and success criteria. Refine the technical and functional scope, document requirements, and submit project proposal. Ensure all team members understand the project vision and their roles moving forward.
W5	Develop Individual Agents	Train and test each agent in isolation (Schema Intelligence, NL Understanding, Database/Table Selection, SQL Generation, Record Discovery). Link schema embeddings to Database Selector agent.
W6	Pipeline Integration (Initial)	Connect agents in a sequential pipeline; establish data flow between components (architecture may not yet be fully functional).
W7	End-to-End CLI Prototype	Achieve a working command-line version of the full pipeline capable of handling basic queries.
W8	UI Development & Integration	Build and connect a web-based dashboard to the backend pipeline.
W8–W10	Iterative Improvements	Improve accuracy; explore Spider v2 dataset support; implement query memory features.
W10–W12	Finalisation & Reporting	Prepare presentation, polish codebase and UI, rehearse demonstration.

Costs

There will be no costs borne by the team. The only anticipated expense is for API keys for Large Language Models (e.g., ChatGPT), which will be provided by the client. Only **one API key** is expected to be required for development and testing.

Gantt Chart

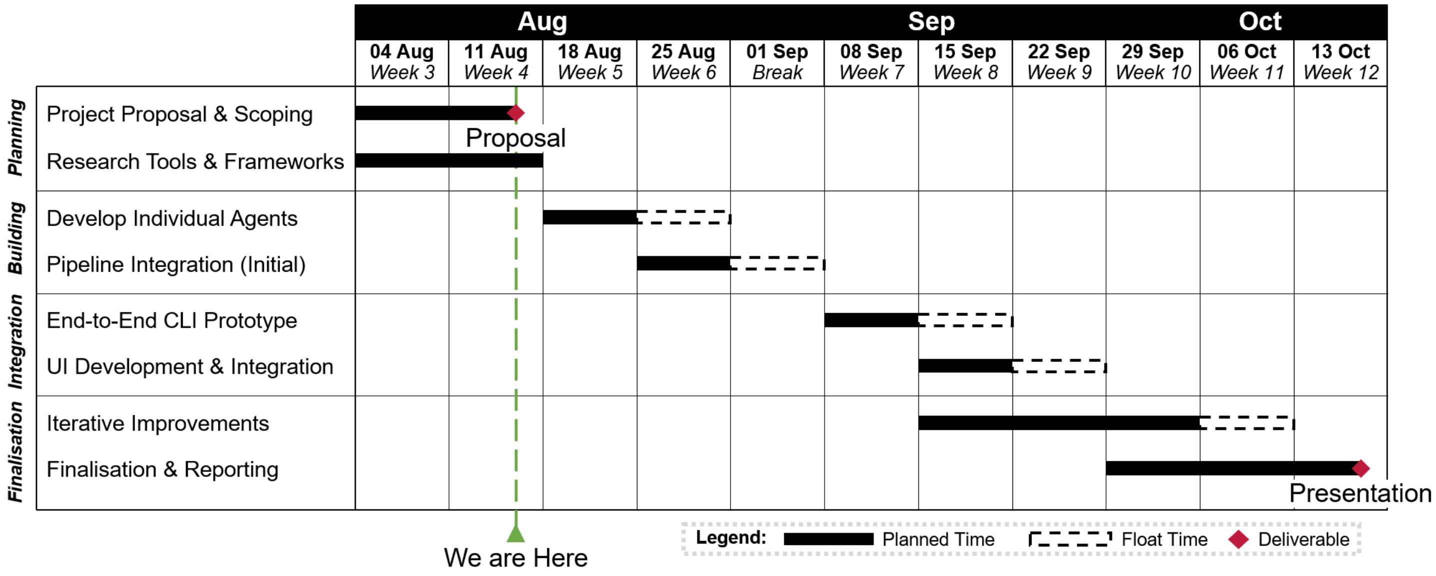

Figure 1. Gantt chart for Weeks 3–12 of the project.

METHODS (5 marks)

Preliminary interrogation of the data

Choice of different data exploration/analytical techniques to test

How to make the process understandable and the output usable

Visualisation to communicate the processes/results

How to use collaborative tools to ensure data and code version controls