**Midterm Report: Natural Language Interface (NLI) to an RDBMS**

**1. Title**

ChatDB: A Natural Language Interface for Relational Databases

**2. Team Details**

* Name: Yuchen Zhu
* Role: Sole Developer (One-Person Group)

**3. Implementation Questions**

**Tech Stack Used**

The following tools, frameworks, and libraries are being used for the project:

* Backend: Python with FastAPI
* Database Connector: pymysql for MySQL/PostgreSQL, pymongo for MongoDBNLP Processing: OpenAI API (GPT-4)
* Regex Handling: re for simple pattern matching
* SQL Execution: SQLAlchemy for MySQL/PostgreSQL
* NoSQL Query Execution: PyMongo for MongoDB

**Query Syntax Implementation Plan**

To convert natural language queries into SQL statements, the system follows these steps:

* User Input: The user enters a natural language query.
* NLP Processing: The query is sent to GPT-4 for interpretation.
* Database Selection: The system determines whether the query is best suited for MySQL, PostgreSQL, or MongoDB.
* Query Generation: The system generates appropriate SQL or NoSQL queries based on database selection
* Execution & Output: The SQL is executed in MySQL/PostgreSQL, and results are returned to the user.
* Validation & Security: The system prevents SQL injection by validating generated queries before execution.

**Database Selection**

* **MySQL (RDBMS): Suitable for traditional structured data and basic SQL queries.**
* **PostgreSQL (RDBMS): Best for complex queries, JSON data support, and ACID transactions.**
* **MongoDB (NoSQL): Suitable for dynamic, unstructured data and flexible schema storage.**

**4. Planned Implementation**

The implementation aligns with the original proposal. The following progress has been made:

* Database setup completed (MySQL, PostgreSQL, and MongoDB instances are configured).
* Basic API development started (initial API endpoints for schema exploration and query execution).
* Integration with OpenAI API (natural language queries successfully converted to SQL queries).
* Next Steps:
  + Implementing database selection logic for SQL vs. NoSQL queries.
  + Improve query validation to reduce errors in SQL generation.
  + Implement data modification capabilities (INSERT, UPDATE, DELETE).
  + Conduct testing with larger real-world datasets.

**5. Project Status**

Finished:

* Schema exploration is implemented (users can ask about tables and columns).
* Basic query conversion is functional (natural language → SQL works for SELECT queries).
* MongoDB integration set up, but NoSQL query translation still in progress.

Ongoing:

* Improving accuracy of query transformation.
* Implement data modification features for both SQL and NoSQL databases.

**6. Challenges Faced**

Database Selection Complexity

* Issue: Determining the best database (SQL vs. NoSQL) for a given query.
* Solution: Designing a classification model to route queries appropriately.

Query Accuracy Issues

* Issue: Some natural language queries were not being correctly translated into SQL.
* Solution: Began improving prompt engineering and refining query validation.

API Rate Limits

* Issue: OpenAI API has usage limits, making testing slower.
* Solution: Started implementing local query caching to minimize API calls.

**7. Timeline**

|  |  |  |
| --- | --- | --- |
| Milestone | Task | Deadline |
| Week 4-5 (Feb 22 - Mar 7) | Implement schema exploration & query translation | Midterm Report (Mar 7) |
| Week 6-8 (Mar 8 - Apr 4) | Add data modification & optimize accuracy |  |
| Week 9-10 (Apr 5 - Apr 20) | Final testing & debugging |  |
| Week 11 (Apr 21 - Apr 23) | In-Class Demo |  |
| Week 12 (Apr 24 - May 9) | Prepare and submit final report | May 9 |