Project Report: IIIT RGUKT RK Valley Student Data Analysis System

Executive Summary

This project implements a natural language interface to the IIIT RGUKT RK Valley student database, allowing users to query student information through conversational language rather than SQL. The system leverages the Gemini API to translate natural language questions into PostgreSQL queries, executes them against the database, and presents the results in a user-friendly format with visualizations when appropriate.

System Architecture

The application follows a modular architecture with the following components:

1. **Metadata Service**: Extracts database schema information and enriches it with human-readable descriptions
2. **Query Processing Service**: Translates natural language to SQL using Gemini API
3. **Query Execution Service**: Safely runs generated SQL against the PostgreSQL database
4. **Response Generation Service**: Converts raw database results into natural language responses
5. **Conversation Management**: Maintains context across multiple queries
6. **Logging Service**: Records queries and responses for analysis and improvement

Database Schema Analysis

The current database schema presents significant challenges for data analysis:

Schema Issues

The database uses a poorly normalized structure with generic column names (col1, col2, etc.) across three tables:

- **table_a**: Contains basic student information (ID, name, hall ticket, DOB, gender)
- **table_b**: Contains academic performance data (ID, semester marks)
- **table_c**: Contains additional student details (ID, parents' names, address, branch, batch)

Data Quality Issues

The data suffers from numerous quality problems:

1. **Inconsistent Formatting**: Student IDs appear in different cases (R200001, r200001)
2. **Non-standardized Values**: Branch/department names have multiple variations (CS, CSE, Computer Science)
3. **Missing Values**: Numerous NULL entries across important fields
4. **Type Inconsistencies**: Numeric values stored as text, dates in various formats
5. **Redundant Data**: Information duplicated across tables without proper normalization

These issues necessitate extensive data cleaning and transformation in queries, significantly increasing complexity.

Query Capabilities

Despite the schema challenges, the system can handle a variety of query types:

Student Performance Queries

- Identifying top performers in specific batches and branches
- Calculating student rankings within their cohort
- Comparing performance across semesters
- Finding average/median/distribution of marks

Demographic Queries

- Distribution of students across branches
- Gender distribution within batches/branches
- Geographic distribution based on addresses

Specific Information Retrieval

- Looking up details for individual students by name
- Finding students with similar attributes
- Retrieving contact and family information

Comparative Analysis

- Performance comparisons between R20 and R21 batches
- Branch-wise performance analysis
- Trend analysis across semesters

Technical Implementation

The system employs several advanced techniques to overcome the database limitations:

1. **Robust SQL Generation**: Carefully crafted prompts that explain the schema quirks to Gemini
2. **Pattern Matching**: Complex LIKE expressions with multiple variations to handle inconsistent data
3. **Null Handling**: COALESCE and NULLIF functions to manage missing data
4. **Case Normalization**: UPPER/LOWER functions to handle inconsistent capitalization
5. **Fallback Mechanisms**: Default queries when Gemini cannot generate appropriate SQL
6. **Visualization Generation**: Automatic chart creation based on query type and result structure

Challenges and Solutions

Challenge 1: Poor Schema Design

Solution: Enhanced metadata with detailed descriptions and relationships, allowing Gemini to understand the logical structure despite poor naming.

Challenge 2: Inconsistent Data

Solution: Implemented robust pattern matching and standardization in SQL queries to handle variations.

Challenge 3: Complex Query Translation

Solution: Developed specialized prompts with examples and fallback mechanisms for common query patterns.

Challenge 4: Performance Issues

Solution: Implemented query optimization techniques, caching, and timeout management for complex queries.

Recommendations for Improvement

1. **Database Restructuring**: Normalize the schema with proper table and column names
2. **Data Cleaning**: Standardize values, fix inconsistencies, and fill missing data where possible
3. **Index Creation**: Add appropriate indexes to improve query performance
4. **Schema Documentation**: Create comprehensive data dictionary and relationship diagrams
5. **Query Templates**: Develop more pre-built query templates for common question types

Conclusion

The IIIT RGUKT RK Valley Student Data Analysis System successfully transforms a poorly structured database into a usable information system through natural language queries. Despite significant data quality and schema challenges, the implementation of advanced NLP techniques and robust SQL generation allows users to extract meaningful insights from the student data.

The system demonstrates how modern AI approaches can bridge the gap between legacy data systems and user-friendly interfaces, making valuable institutional data accessible to non-technical users.