

Project – Stage 2

Database Normalization and Implementation

Objective

The objective of Stage 2 is to delve deeper into database management using **PostgreSQL and Command Line Interface (CLI) based functionalities**. In this stage, you will focus on table design, normalization, schema definition, and implementation of various data manipulation operations.

As part of the project, you will need to implement a full-fledged system with user-faced interfaces through a CLI (command-line interface). The functionalities mentioned below should have a CLI based interface in the system to perform the necessary action. You will need to ensure that the database system meets specified requirements and functionalities to support efficient data management.

Stage 2 Tasks:

1. Table Design and Implementation:

- **Conversion of ERD into Tables using DDL Commands:**
 - Convert the Entity-Relationship Diagram (ERD) into tables using Data Definition Language (DDL) commands in PostgreSQL.
 - Provide the schema and the DDL commands used to create the tables.
- **Normalization:** Applied normalization techniques to ensure tables are in Boyce-Codd Normal Form (BCNF), minimizing redundancy and ensuring data integrity.
- **Key Mapping:** Appropriately mapped keys to establish relationships between tables, ensuring referential integrity.
- **Schema Definition:** Defined the schema for the database, specifying table structures, attributes, and relationships.

2. CLI Based User Interface:

- Build a CLI interface to include options for numbers 1 through 10 and beyond. These options will be used to specify which functionality of the SQL query to perform.
- Document each query and the functionality associated with each number option.
- When a user selects an option, execute the corresponding SQL query or functionality.
- After executing each query or functionality, capture screenshots of the query execution result. Ensure that these screenshots are clear and properly labeled.

3. **Make sure to have the following functionalities are implemented:** The database system you will be building must include the following in the querying operations.

- **Insert Data:** Implemented functionality to add new records to tables.
 - Example: **INSERT INTO TableName (Column1, Column2, ...) VALUES (Value1, Value2, ...);**
- **Delete Data:** Provided functionality to remove specific records from tables.
 - Example: **DELETE FROM TableName WHERE Condition;**
- **Update Data:** Supported modification of existing records in tables.
 - Example: **UPDATE TableName SET Column1 = NewValue WHERE Condition;**
- **Search Data:** Enabled finding records based on specified criteria.
 - Example: **SELECT * FROM TableName WHERE Condition;**
- **Aggregate Functions:** Implemented calculations like sum, average, count, min, and max on data.
 - Example: **SELECT SUM(Column) FROM TableName;**
- **Sorting:** Enabled arranging query results based on specified columns.
 - Example: **SELECT * FROM TableName ORDER BY Column ASC/DESC;**
- **Joins:** Combined data from multiple tables using relationships.
 - Example: **SELECT * FROM Table1 INNER JOIN Table2 ON Table1.Key = Table2.Key;**
- **Grouping:** Grouped query results based on specified columns.
 - Example: **SELECT Column, COUNT(*) FROM TableName GROUP BY Column;**
- **Subqueries:** Supported nested operations within queries.
 - Example: **SELECT * FROM TableName WHERE Column IN (SELECT Column FROM AnotherTable);**
- **Transactions:** Ensured consistency and reliability of database operations.
 - Example: **BEGIN TRANSACTION; ... COMMIT; or ROLLBACK;**
- **Error Handling:** Implemented mechanisms to catch and handle exceptions gracefully during database operations.
 - Example: **TRY...CATCH block in SQL Server or EXCEPTION block in PL/SQL.**

An example of the CLI Interface you can have:

```
Welcome to the Database CLI Interface!

Please select an option:
1. Insert Data
2. Delete Data
3. Update Data
4. Search Data
5. Aggregate Functions
6. Sorting
7. Joins
8. Grouping
9. Subqueries
10. Transactions
11. Error Handling
12. Exit

Enter your choice (1-12):
```

Submission Requirements:

- Submit a PDF document containing:
 - PART 1: Schema and DDL commands used to create tables from the ERD. Screenshots of the results from executing DDL commands. DML commands along with screenshots showcasing the results. Queries such as SELECT * statements to display all the rows inserted into the tables.
 - PART 2: The normalized tables and the normalization step you have followed to ensure that tables are in BCNF.
 - PART 3: Document the queries you have used for the functionalities and document the appropriate screenshots.
 - PART 4: Ensure there is a CLI interface for using the queries in the backend.

Evaluation Criteria:

1. **Evaluate** clarity and correctness of normalization steps, ensuring tables are in BCNF.
2. Assess completeness and accuracy of queries documented with relevant screenshots.
3. Examine functionality and usability of CLI interface for query execution and backend interaction.

Important Notes:

- This is an **individual** assignment.
- **Academic Integrity Violation:**
 - In this open project environment, it is strictly prohibited to substantially duplicate ideas or code from other projects. If we find any, we'll have to check for possible academic misconduct.
 - Be careful not to copy code from the internet. If you do, it could mean failing the project and facing the academic integrity committee.
- Reach out to the instructor or any of the TAs for any clarification or guidance.
- Ensure **proper citation** if external sources are used.

Deadline for Stage 2 Submission: April 8th, 2024