

Computer Engineering Department

King Mongkut's University of Technology Thonburi

CPE 241 Database

Problem Session #02 Simple Database Management

Date: Check PS02 assessment activities on LEB2

Introduction: Simple Database Management

In this second problem session, you will build upon the foundational concepts introduced in PS01. The tasks involve revisiting and refining the data structures you previously designed, followed by resubmitting your updated design. Once the design is finalized, you will create a database in DBMS to store some of the data. Finally, you will practice inserting sample data into your tables and verifying both the table structures and the accuracy of the data inserted. This session will strengthen your understanding of database creation and management.

Task List

1. Refine Data Structure

- Revisit and refine the data structure you designed in PS01.
- Adjust the design to better suit your needs based on feedback or new insights.

2. Create Database and Insert Data

- Set up a database in DBMS using your refined data structure.
- Insert sample data into the database tables.
- Verify that the table structures and the inserted data are correct.

Task 1: Redesign Data Structure

In this task, you will revisit the data structure you designed in PS01 and refine it based on the knowledge you have gained. Consider the following steps:

1. Rethink Your Data Needs

- Review the data you want to store and its purpose.
- Identify any additional data or attributes that need to be included.

2. Determine Data Types

- Define the type of data for each attribute (e.g., integer, string, date).
- Consider using appropriate data types to optimize storage and retrieval.

3. Create a New Design

- Update your tables, ensuring the structure is logical and meets your requirements.
- Include primary keys for unique identification and foreign keys for relationships where applicable.

4. Submit Your Work

- Prepare a clear and organized submission of your updated design.
- Use the format `data_structure(data_to_keep)` to express your design.

Example:

If your goal is to track student grade:

- `students(studentID, firstname, lastname)`
- `courses(courseID, courseName)`
- `grades(studentID, courseID, date, grade)`

This step ensures your data structure is well-defined and ready for database implementation in the next task

Task 2: Create Database and Insert Data

In this task, you will store your refined data structure in a database. This involves creating a database, defining your tables, inserting data, and verifying its accuracy. You can choose between using a graphical tool (GUI), or performing the process manually through the DBMS client.

Option 1: Using GUI

DBMS GUI provides a simple and user-friendly interface for managing databases.

1. Access your selected GUI:

- Log in using the credentials for DBMS server.

2. Create Database:

- Create a new database named PS02.

3. Create Tables:

- Use the GUI to define the following tables:

- **students:** studentID (number(8)), firstname (varchar(30)), lastname (varchar(20))
- **courses:** courseID (number(8)), courseName (varchar(40))
- **grades:** studentID (number(8)), courseID (number(8)), grade (varchar(2))

4. Insert Data:

- Insert sample records into each table using the Insert feature.

5. Verify Data:

- Query the tables to ensure the data is correctly inserted and reflects your design.

6. You must also try to use the SQL editor to run SQL commands for creating database/tables, inserting data, and verifying data (See SQL commands in step no. 9-13, Option 2). This is required to be included in this lab report.

Option 2: Manual Method Using DBMS Client (a case of MySQL)

Follow these steps to manually create and manage your database:

- 1. Connect to Your Linux VM:**

- Start your Linux VM.

- 2. Use SSH from your host OS to connect:**

```
ssh [username]@[VM IP address]
```

- 3. Gain Root Access:**

Switch to the root user:

```
sudo su -
```

- 4. Start MySQL:**

Launch the MySQL client:

```
mysql
```

- 5. Create a Local MySQL User:**

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'username'@'localhost';
FLUSH PRIVILEGES;
```

- 6. Log out from the root user:**

```
EXIT;
```

- 7. Log In with the New User:**

```
mysql -u username -p
```

- 8. Create the Database:**

```
CREATE DATABASE PS02;
USE PS02;
```

- 9. Define Tables:**

```
CREATE TABLE students (
    studentID INT(8),
    firstname VARCHAR(30),
    lastname VARCHAR(20)
);
```

```
CREATE TABLE courses (
    courseID INT(8),
    courseName VARCHAR(40)
);
```

```
CREATE TABLE grades (
    studentID INT(8),
    courseID INT(8),
    grade VARCHAR(2)
);
```

10. Verify Table Structures:

Use the `DESCRIBE` command to confirm the structure of each table:

```
DESCRIBE students;  
DESCRIBE courses;  
DESCRIBE grades;
```

11. Insert Sample Data:

Insert sample records into the tables:

```
INSERT INTO students (studentID, firstname, lastname) VALUES  
(101, 'John', 'Doe');  
INSERT INTO courses (courseID, courseName) VALUES (201,  
'Database Systems');  
INSERT INTO grades (studentID, courseID, grade) VALUES (101,  
201, 'A');
```

12. Insert More Data

Insert into `students` Table

```
INSERT INTO students (studentID, firstname, lastname) VALUES  
(101, 'John', 'Doe');  
INSERT INTO students (studentID, firstname, lastname) VALUES  
(102, 'Jane', 'Smith');  
INSERT INTO students (studentID, firstname, lastname) VALUES  
(103, 'Alice', 'Johnson');  
INSERT INTO students (studentID, firstname, lastname) VALUES  
(104, 'Bob', 'Brown');  
INSERT INTO students (studentID, firstname, lastname) VALUES  
(105, 'Charlie', 'Davis');
```

Insert into `courses` Table

```
INSERT INTO courses (courseID, courseName) VALUES (201,  
'Database Systems');  
INSERT INTO courses (courseID, courseName) VALUES (202, 'Web  
Development');  
INSERT INTO courses (courseID, courseName) VALUES (203, 'Data  
Structures');  
INSERT INTO courses (courseID, courseName) VALUES (204,  
'Algorithms');
```

```
INSERT INTO courses (courseID, courseName) VALUES (205, 'Operating Systems');
```

Insert into **grades** Table

```
INSERT INTO grades (studentID, courseID, grade) VALUES (101, 201, 'A');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (101, 202, 'B');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (102, 203, 'A');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (103, 204, 'B');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (104, 201, 'C');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (105, 202, 'A');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (105, 203, 'B');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (102, 204, 'C');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (103, 205, 'A');
```

```
INSERT INTO grades (studentID, courseID, grade) VALUES (104, 205, 'B');
```

13. Verify Data:

Query the tables to check the inserted data:

```
SELECT * FROM students;
```

```
SELECT * FROM courses;
```

```
SELECT * FROM grades;
```

Final Step: Test and Confirm

Ensure all data and structures are functioning as expected by testing queries and data relationships. This database will serve as the foundation for your subsequent tasks and projects.

Result Example

After inserting the data and querying the tables, the output should look something like this:

students

studentID	firstname	lastname
101	John	Doe
102	Jane	Smith
103	Alice	Johnson
104	Bob	Brown
105	Charlie	Davis

courses

courseID	courseName
201	Database Systems
202	Web Development
203	Data Structures
204	Algorithms
205	Operating Systems

grades

studentID	courseID	grade
101	201	A
101	202	B
102	203	A
103	204	B
104	201	C
105	202	A
105	203	B
102	204	C
103	205	A
104	205	B

Submission Instructions

For this problem session (PS02), you are required to submit the following materials under the Assessment Activities section for PS02 on LEB2. Your submission must include a comprehensive report containing the following elements:

1. New Table Design

- Provide a clear and detailed representation of your redesigned table structures.
- Use the format `data_structure(data_to_keep)` for each table, ensuring to highlight keys (e.g., primary and foreign keys) where applicable.

2. Database Creation Evidence

- Include a screenshot showing the successful creation of your database (e.g., the `CREATE DATABASE PS02` command or equivalent).

3. Table Creation Evidence

- Provide screenshots showing the commands or GUI steps used to create your tables in the database.

4. Table Description

- Use the **DESCRIBE** command (or equivalent) for each table and include screenshots showing the table structures, including column names, data types, and keys.

5. Data Insertion and Selection

- Insert sample data into each table and verify its correctness.
- Include screenshots of the following queries:
 - Data insertion (**INSERT INTO** commands).
 - Data selection (**SELECT * FROM tablename**) showing all records in each table.

6. Report Formatting

- Organize your report clearly with proper headings for each section (e.g., "Table Design," "Database Creation," etc.).
- Use descriptive captions for each screenshot to explain what it demonstrates.

Final Submission

- Save your report in PDF format and ensure it is named appropriately (e.g., **PS02_YourGroupName.pdf**).
- Upload the PDF report to the Assessment Activities section for PS02 on LEB2.