

SQL LAB-2

RDBMS, MYSQL

NAME: Thummala Pavani

ID: AF0366728

QUESTIONS

Lab 1. Day-2 Lab

(PN: ChatGPT exercise is mandatory)

Lab 1. Use the Database and table from Day 1 lab. Insert 5 records in each table and retrieve data from all tables and display.

- Create a database with the name StudentManagementSystem.

Code:

```
mysql> create database StudentManagementSystem;  
Query OK, 1 row affected (0.07 sec)
```

Output:

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| studentmanagementsystem |  
| sys |  
+-----+  
5 rows in set (0.06 sec)
```

i) A table with named Students with attributes:

StudentID (Primary Key), FirstName, LastName, DateOfBirth, Gender, Email, Phone

Code:

```
mysql> Use Studentmanagementsystem;  
Database changed
```

Insert five records:

```
mysql> INSERT INTO Student (StudentID, FirstName, LastName, DateOfBirth, Gender, Email, Phone) VALUES
-> (101, 'Adi', 'Sharma', '2000-01-01', 'Male', 'adi@gmail.com', '9876543210'),
-> (102, 'Ishitha', 'Iyer', '2001-02-02', 'Female', 'Ishitha@gmail.com', '9123456789'),
-> (103, 'Raman', 'Bhalla', '2002-03-03', 'Male', 'Bhalla@gmail.com', '9282726252'),
-> (104, 'Ruhi', 'Khan', '2003-04-04', 'Female', 'Ruhi@gmail.com', '9325649871'),
-> (105, 'Vidyuth', 'Sahay', '2004-05-05', 'Male', 'Vidyuth@gmail.com', '9563214789');
```

Display the data which is retrieved from the table Students:

```
+-----+-----+-----+-----+-----+-----+-----+
| StudentID | FirstName | LastName | DateOfBirth | Gender | Email | Phone |
+-----+-----+-----+-----+-----+-----+-----+
| 101 | Adi | Sharma | 2000-01-01 | Male | adi@gmail.com | 9876543210 |
| 102 | Ishitha | Iyer | 2001-02-02 | Female | Ishitha@gmail.com | 9123456789 |
| 103 | Raman | Bhalla | 2002-03-03 | Male | Bhalla@gmail.com | 9282726252 |
| 104 | Ruhi | Khan | 2003-04-04 | Female | Ruhi@gmail.com | 9325649871 |
| 105 | Vidyuth | Sahay | 2004-05-05 | Male | Vidyuth@gmail.com | 9563214789 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

2) A table with name Course with attributes:

CourseID (Primary Key), CourseTitle, Credits

Insert five records:

```
mysql> INSERT INTO Course (CourseID, CourseTitle, Credits) VALUES
-> (201, 'Mathematics', 3),
-> (202, 'Physics', 4),
-> (203, 'Chemistry', 4),
-> (204, 'Biology', 3),
-> (205, 'Computer Science', 3);
```

Display the data which is retrieved from the table Course:

```
+-----+-----+-----+
| CourseId | CourseTitle | Credits |
+-----+-----+-----+
| 201 | Mathematics | 3 |
| 202 | Physics | 4 |
| 203 | Chemistry | 4 |
| 204 | Biology | 3 |
| 205 | Computer Science | 3 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

3) A table with named Instructor with attributes:

InstructorID (Primary Key), FirstName, LastName, Email

Insert five records:

```
mysql> INSERT INTO Instructor (InstructorID, FirstName, LastName, Email) VALUES
-> (301, 'Albert', 'Einstein', 'albert.einstein@example.com'),
-> (302, 'Marie', 'Curie', 'marie.curie@example.com'),
-> (303, 'Isaac', 'Newton', 'isaac.newton@example.com'),
-> (304, 'Charles', 'Darwin', 'charles.darwin@example.com'),
-> (305, 'Alan', 'Turing', 'alan.turing@example.com');
```

To display the data retrieved from the Instructor table:

```
+-----+-----+-----+-----+
| InstructorID | FirstName | LastName | Email |
+-----+-----+-----+-----+
| 301 | Albert | Einstein | albert.einstein@example.com |
| 302 | Marie | Curie | marie.curie@example.com |
| 303 | Isaac | Newton | isaac.newton@example.com |
| 304 | Charles | Darwin | charles.darwin@example.com |
| 305 | Alan | Turing | alan.turing@example.com |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

4) A table with named Enrollment with attributes:

EnrollmentID (Primary Key), EnrollmentDate, StudentID(Foreignkey),

CourseID(Foreign Key), InstructorID(Foreign key)

Insert five records:

```
mysql> INSERT INTO Enrollment (EnrollmentID, EnrollmentDate, StudentID, CourseID, InstructorID) VALUES
-> (401, '2023-01-01', 101, 201, 301),
-> (402, '2023-01-02', 102, 202, 302),
-> (403, '2023-01-03', 103, 203, 303),
-> (404, '2023-01-04', 104, 204, 304),
-> (405, '2023-01-05', 105, 205, 305);
```

To display the data retrieved from the Enrollment table:

EnrollmentID	EnrollmentDate	StudentID	CourseID	InstructorID
401	2023-01-01	101	201	301
402	2023-01-02	102	202	302
403	2023-01-03	103	203	303
404	2023-01-04	104	204	304
405	2023-01-05	105	205	305

5 rows in set (0.00 sec)

5) A table with named Score with attributes:

ScoreID (Primary Key), CourseID (Foreign key), StudentID (Foreign Key),
DateOfExam, CreditObtained

Insert the five records:

```
mysql> INSERT INTO Score (ScoreID, CourseID, StudentID, DateOfExam, CreditObtained) VALUES  
-> (501, 201, 101, '2024-05-10', 2), -- Credits obtained within the range of course credits  
-> (502, 202, 102, '2024-05-11', 3),  
-> (503, 203, 103, '2024-05-12', 3),  
-> (504, 204, 104, '2024-05-13', 2),  
-> (505, 205, 105, '2024-05-14', 2);
```

To display the data retrieved from the Score table:

ScoreID	CourseID	StudentID	DateOfExam	CreditObtained
501	201	101	2024-05-10	2
502	202	102	2024-05-11	3
503	203	103	2024-05-12	3
504	204	104	2024-05-13	2
505	205	105	2024-05-14	2

5 rows in set (0.00 sec)

6) A table with named Feedback with attributes:

FeedbackID (Primary Key), StudentID (Foreign key), Date, InstructorName, Feedback

Insert the records:

```
mysql> INSERT INTO Feedback (FeedbackID, StudentID, Date, InstructorName, Feedback) VALUES
-> (601, 101, '2024-05-15', 'Albert Einstein', 'Excellent course!'),
-> (602, 102, '2024-05-15', 'Marie Curie', 'Very informative.'),
-> (603, 103, '2024-05-15', 'Isaac Newton', 'Great teaching style.'),
-> (604, 104, '2024-05-15', 'Charles Darwin', 'Loved the practicals.'),
-> (605, 105, '2024-05-15', 'Alan Turing', 'Highly recommended.');
```

To display the data retrieved from the Feedback table:

FeedbackID	StudentID	Date	InstructorName	Feedback
601	101	2024-05-15	Albert Einstein	Excellent course!
602	102	2024-05-15	Marie Curie	Very informative.
603	103	2024-05-15	Isaac Newton	Great teaching style.
604	104	2024-05-15	Charles Darwin	Loved the practicals.
605	105	2024-05-15	Alan Turing	Highly recommended.

5 rows in set (0.00 sec)

ChatGPT Exercise

Generating a normalized table using ChatGPT as a continuation of our earlier ChatGPT scenario.

Put the below statement in the message box and see what ChatGPT says:

I have created a database of emergency contact information. Kindly help me by providing normalized tables with understanding.

- 1)Employee (EmployeeID, FirstName, LastName, DateOfBirth, Gender, Phone)
- 2)EmergencyContact (ContactID, EmployeeID, ContactName, Relationship)
- 3)ContactDetails (ContactID, ContactType, ContactValue)

The Employee table stores information about each employee, including their name, date of birth, gender, and phone number. The primary key is EmployeeID, which uniquely identifies each employee.

The EmergencyContact table stores information about each employee's emergency contact, including their name and relationship to the employee. The primary key is ContactID, which is also a foreign key that references the ContactDetails table. The EmployeeID attribute is a foreign key that references the Employee table, indicating which employee this emergency contact belongs to.

The ContactDetails table stores the actual contact information for each emergency contact, including their email address and physical address. The ContactType attribute indicates whether the ContactValue is an email address or a physical address. The primary key is ContactID, which is also a foreign key that references the EmergencyContact table.

Here is an example of how you can create the ContactDetails table:

```
CREATE TABLE ContactDetails (  
    ContactID INT NOT NULL PRIMARY KEY,  
    ContactType VARCHAR(10) NOT NULL,  
    ContactValue VARCHAR(100) NOT NULL,  
    FOREIGN KEY (ContactID) REFERENCES EmergencyContact(ContactID)  
);
```

Sure, I'd be happy to help you normalize your database of emergency contact information. Here's an example of how you might structure the tables.

Employee Table

- EmployeeID (Primary Key, Auto-Increment)
- FirstName
- LastName
- DateOfBirth
- Gender
- Phone

EmergencyContact Table

- ContactID (Primary Key, Auto-Increment)
- EmployeeID (Foreign Key, References Employee(EmployeeID))
- ContactName
- Relationship
- ContactEmail
- ContactAddress

In this design, the Employee table stores information about each employee, and the EmergencyContact table stores information about each employee's emergency contact.

The EmployeeID field in the EmergencyContact table is a foreign key that references the Employee table, indicating which employee this emergency contact belongs to.

By normalizing the database in this way, we avoid redundancy and ensure that each piece of information is stored in only one place. This makes the database easier to maintain, update, and query.