LAB1SQL- (RDBMS, MYSQL)

NAME**:** SUSHRITHA.V

BATCHCODE: ANP-C7281

LAB1SQL

Lab 1. Create a Database & Table Using MySQL Command-Line Client.

- Create a database with the name StudentManagementSystem.

Create a table with named Student with attributes:

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

1.(a) Create a Database with a name StudentManagementSystem:

Code:

```
Create database StudentManagementSystem;
```

Output:

```
mysql> create database student_management_system;
Query OK, 1 row affected (0.02 sec)
```

(b) Show all Databases:

LAB1SQL

Code:

Show databases;

Output:

```
mysql> show databases;
+----------------------------+
| Database                   |
+----------------------------+
| information_schema         |
| menu                       |
| my_file                    |
| mybackend                  |
| mysql                      |
| performance_schema         |
| student_management_system  |
| sys                        |
| world                      |
+----------------------------+
9 rows in set (0.00 sec)
```

(c) Use Databases:

Use database and the database we created is Student_management_system:

Code:

USE student_management_system;

Output:

```
mysql> Use student_management_system;
Database changed
```

(d)Create a table with named Student with attributes:

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

Code:

LAB1SQL

```
mysql> CREATE TABLE Student (
    ->   StudentID INT PRIMARY KEY AUTO_INCREMENT,
    ->   FirstName VARCHAR(50) NOT NULL,
    ->   LastName VARCHAR(50) NOT NULL,
    ->   DateOfBirth DATE,
    ->   Gender ENUM('Male', 'Female', 'Other'),
    ->   Email VARCHAR(100) UNIQUE,
    ->   Phone VARCHAR(20)
    -> );
```

Output:

```
+-------------+---------------------------------+------+-----+---------+----------------+
| Field       | Type                            | Null | Key | Default | Extra          |
+-------------+---------------------------------+------+-----+---------+----------------+
| StudentID   | int                             | NO   | PRI | NULL    | auto_increment |
| FirstName   | varchar(50)                     | NO   |     | NULL    |                |
| LastName    | varchar(50)                     | NO   |     | NULL    |                |
| DateOfBirth | date                            | YES  |     | NULL    |                |
| Gender      | enum('Male','Female','Other')   | YES  |     | NULL    |                |
| Email       | varchar(100)                    | YES  | UNI | NULL    |                |
| Phone       | varchar(20)                     | YES  |     | NULL    |                |
+-------------+---------------------------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

(2) Create a table with name Course with attributes:

- CourseID (Primary Key)
- Course Title
- Credits

Code:

LAB1SQL

```
mysql> CREATE TABLE Course (
    ->   CourseID INT PRIMARY KEY AUTO_INCREMENT
    ->   CourseTitle VARCHAR(100) NOT NULL,
    ->   Credits INT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

Output:

```
mysql> describe Course;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| CourseID    | int          | NO   | PRI | NULL    | auto_increment |
| CourseTitle | varchar(100) | NO   |     | NULL    |                |
| Credits     | int          | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

(3) Create a table with named Instructor with attributes:

- InstructorID (Primary Key)
- FirstName
- LastName
- Email

Code:

```
mysql> CREATE TABLE Instructor (
    ->   InstructorID INT PRIMARY KEY AUTO_INCREMENT,
    ->   FirstName VARCHAR(50) NOT NULL,
    ->   LastName VARCHAR(50) NOT NULL,
    ->   Email VARCHAR(100) UNIQUE
    -> );
Query OK, 0 rows affected (0.04 sec)
```

LAB1SQL

Output:

```
mysql> describe Instructor;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| InstructorID | int          | NO   | PRI | NULL    | auto_increment |
| FirstName    | varchar(50)  | NO   |     | NULL    |                |
| LastName     | varchar(50)  | NO   |     | NULL    |                |
| Email        | varchar(100) | YES  | UNI | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
```

(4) Create a table with named Enrollment with attributes:

- Enrollment ID (Primary Key)
- Enrollment Date
- StudentID(Foreign key)
- CourseID(Foreign Key)
- InstructorID(Foreign key)

Code:

```
mysql> CREATE TABLE Enrollment (
    ->   EnrollmentID INT PRIMARY KEY AUTO_INCREMENT,
    ->   EnrollmentDate DATE,
    ->   StudentID INT NOT NULL,
    ->   CourseID INT NOT NULL,
    ->   InstructorID INT NOT NULL,
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    ->   FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
    ->   FOREIGN KEY (InstructorID) REFERENCES Instructor(InstructorID)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

Output:

LAB1SQL

```
mysql> describe Enrollment;
+----------------+------+------+-----+---------+----------------+
| Field          | Type | Null | Key | Default | Extra          |
+----------------+------+------+-----+---------+----------------+
| EnrollmentID   | int  | NO   | PRI | NULL    | auto_increment |
| EnrollmentDate | date | YES  |     | NULL    |                |
| StudentID      | int  | NO   | MUL | NULL    |                |
| CourseID       | int  | NO   | MUL | NULL    |                |
| InstructorID   | int  | NO   | MUL | NULL    |                |
+----------------+------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

(5) Create a table with named Score with attributes:

- ScoreID (Primary Key)
- CourseID (Foreign key)
- StudentID (Foreign Key)
- DateOfExam
- Credit Obtained

Code:

```
mysql> CREATE TABLE Score (
    ->   ScoreID INT PRIMARY KEY AUTO_INCREMENT,
    ->   CourseID INT NOT NULL,
    ->   StudentID INT NOT NULL,
    ->   DateOfExam DATE,
    ->   CreditObtained INT,
    ->   FOREIGN KEY (CourseID) REFERENCES Course(CourseID),
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

LAB1SQL

Output:

```
mysql> describe Score;
+----------------+------+------+-----+---------+----------------+
| Field          | Type | Null | Key | Default | Extra          |
+----------------+------+------+-----+---------+----------------+
| ScoreID        | int  | NO   | PRI | NULL    | auto_increment |
| CourseID       | int  | NO   | MUL | NULL    |                |
| StudentID      | int  | NO   | MUL | NULL    |                |
| DateOfExam     | date | YES  |     | NULL    |                |
| CreditObtained | int  | YES  |     | NULL    |                |
+----------------+------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

(6) Create a table with named Feedback with attributes:

- FeedbackID (Primary Key)
- StudentID (Foreign key)
- Date
- Instructor Name
- Feedback

Code:

```
mysql> CREATE TABLE Feedback (
    ->   FeedbackID INT PRIMARY KEY AUTO_INCREMENT,
    ->   StudentID INT NOT NULL,
    ->   Date DATE,
    ->   InstructorName VARCHAR(100),
    ->   Feedback TEXT,
    ->   FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
    -> );
Query OK, 0 rows affected (0.05 sec)
```

LAB1SQL

Output:

```
mysql> describe Feedback;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| FeedbackID     | int          | NO   | PRI | NULL    | auto_increment |
| StudentID      | int          | NO   | MUL | NULL    |                |
| Date           | date         | YES  |     | NULL    |                |
| InstructorName | varchar(100) | YES  |     | NULL    |                |
| Feedback       | text         | YES  |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

(7) Show All the Tables

Code & Output:

```
mysql> SHOW TABLES;
+-------------------------------------+
| Tables_in_student_management_system |
+-------------------------------------+
| course                              |
| enrollment                          |
| feedback                            |
| instructor                          |
| score                               |
| student                             |
+-------------------------------------+
6 rows in set (0.00 sec)
```

ChatGPT Exercise

Using ChatGPT generate the Database design

Scenario: Implementing Database Design

The database should store emergency contact information for each employee. This information is crucial for situations were immediate contact with family or emergency contacts are necessary. The design should consider privacy and security measures for

LAB1SQL

sensitive contact details.

Use the ChatGPT prompt to formulate the database design for the scenario described.

Employee (Similar to Emergency Contact table):

- EmployeeID (INT, Primary Key)
- FirstName (VARCHAR (50), NOT NULL)
- LastName (VARCHAR (50), NOT NULL)
- Other Names (VARCHAR (100)) (Optional)
- Department (VARCHAR (50))
- Email (VARCHAR (100), UNIQUE, NOT NULL)

Emergency Contact:

- EmergencyContactID (INT, Primary Key)
- EmployeeID (INT, Foreign Key references Employee (EmployeeID))
- FirstName (VARCHAR (50), NOT NULL)
- LastName (VARCHAR (50), NOT NULL)
- Relationship (ENUM ('Spouse', 'Parent', 'Child', 'Sibling', 'Other'))
- Phone Number (VARCHAR (20))
- PhoneNumberType (ENUM ('Mobile', 'Landline', 'Work')) (Optional)
- Email (VARCHAR (100), UNIQUE) (Optional) - Consider allowing only one unique email per contact, but multiple contacts per employee.