

**Name:** K. Guna Shekar Reddy

**Register Number:** 192424323

**Course Code:** CSA0504

**Course Name:** Database Management System for Transaction Management

## Lab Experiments

### EX. No. 1 – DDL Commands using CREATE, ALTER, TRUNCATE, and DROP

```
mysql> use student;
Database changed
mysql> CREATE TABLE Student (
    ->     RollNo INT,
    ->     Name VARCHAR(50),
    ->     Age INT
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| RollNo | int    | YES  |     | NULL    |       |
| Name   | varchar(50) | YES  |     | NULL    |       |
| Age    | int    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

```
mysql> ALTER TABLE Student ADD Email VARCHAR(100);
Query OK, 0 rows affected (0.09 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc student;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| RollNo | int       | YES  |     | NULL    |          |
| Name   | varchar(50) | YES  |     | NULL    |          |
| Age    | int       | YES  |     | NULL    |          |
| Email  | varchar(100) | YES  |     | NULL    |          |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> TRUNCATE TABLE Student;
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> desc student;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| RollNo | int       | YES  |     | NULL    |          |
| Name   | varchar(50) | YES  |     | NULL    |          |
| Age    | int       | YES  |     | NULL    |          |
| Email  | varchar(100) | YES  |     | NULL    |          |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> DROP TABLE Student;
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> desc student;
ERROR 1146 (42S02): Table 'student.student' doesn't exist
mysql> |
```

## EX. No. 2 – DDL Commands with Constraints

```
mysql> create database college;
Query OK, 1 row affected (0.03 sec)

mysql> use college;
Database changed
mysql> CREATE TABLE Department (
    ->     DeptID INT PRIMARY KEY,
    ->     DeptName VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc department;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| DeptID | int    | NO   | PRI | NULL    |       |
| DeptName | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> CREATE TABLE Employee (
    ->     EmpID INT PRIMARY KEY,
    ->     EmpName VARCHAR(50),
    ->     DeptID INT,
    ->     FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc employee;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| EmpID | int    | NO   | PRI | NULL    |       |
| EmpName | varchar(50) | YES  |     | NULL    |       |
| DeptID | int    | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE Users (
    ->     UserID INT PRIMARY KEY,
    ->     Email VARCHAR(100) UNIQUE
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> desc users;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| UserID | int       | NO   | PRI | NULL    |       |
| Email  | varchar(100) | YES  | UNI | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> CREATE TABLE Product (
    ->     ProductID INT PRIMARY KEY,
    ->     Price DECIMAL(10,2) CHECK (Price > 0)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc product;
+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| ProductID | int       | NO   | PRI | NULL    |       |
| Price      | decimal(10,2) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> CREATE TABLE Orders (
    ->     OrderID INT PRIMARY KEY,
    ->     Status VARCHAR(20) DEFAULT 'Pending'
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc orders;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| OrderID | int       | NO   | PRI | NULL    |       |
| Status  | varchar(20) | YES  |     | Pending |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> CREATE TABLE Course (
    ->     CourseID INT PRIMARY KEY,
    ->     CourseName VARCHAR(50) NOT NULL
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc course;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| CourseID   | int        | NO   | PRI | NULL    |       |
| CourseName | varchar(50) | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

### EX. No. 3 – DML Commands using INSERT and SELECT

```
mysql> create database department;
Query OK, 1 row affected (0.03 sec)

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

mysql> use department;
Database changed
mysql> create table department(deptid int,deptname varchar(50));
Query OK, 0 rows affected (0.05 sec)

mysql> INSERT INTO Department (DeptID, DeptName)
-> VALUES (1, 'Computer Science');
Query OK, 1 row affected (0.02 sec)

mysql> use employee;
Database changed
mysql> create table employee(empid int,empname varchar(50),deptid int);
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO Employee (EmpID, EmpName, DeptID)
-> VALUES (101, 'Alice', 1);
Query OK, 1 row affected (0.02 sec)

mysql> select *from department;
ERROR 1146 (42S02): Table 'employee.department' doesn't exist
mysql> select *from employee;
+-----+-----+-----+
| empid | empname | deptid |
+-----+-----+-----+
|    101 | Alice   |      1 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> use department;
Database changed
mysql> select *from department;
+-----+-----+
| deptid | deptname      |
+-----+-----+
|      1 | Computer Science |
+-----+-----+
1 row in set (0.00 sec)
```

```

mysql> use employee;
Database changed
mysql> SELECT * FROM Employee;
+-----+-----+-----+
| empid | empname | deptid |
+-----+-----+-----+
| 101   | Alice   |     1 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT EmpName, DeptID FROM Employee;
+-----+-----+
| EmpName | DeptID |
+-----+-----+
| Alice   |     1 |
+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM Employee
      -> WHERE DeptID = 1;
+-----+-----+-----+
| empid | empname | deptid |
+-----+-----+-----+
| 101   | Alice   |     1 |
+-----+-----+-----+
1 row in set (0.02 sec)

```

#### EX. No. 4 – DML Commands using UPDATE and DELETE

```

mysql> UPDATE Employee
      -> SET EmpName = 'Alicia'
      -> WHERE EmpID = 101;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select *from employee;
+-----+-----+-----+
| empid | empname | deptid |
+-----+-----+-----+
| 101   | Alicia  |     1 |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> DELETE FROM Employee
      -> WHERE EmpID = 101;
Query OK, 1 row affected (0.02 sec)

mysql> select *from employee;
Empty set (0.00 sec)

```

## EX. No. 5 – TCL Commands: COMMIT, SAVEPOINT, ROLLBACK

```
mysql> use department;
Database changed
mysql> INSERT INTO Department (DeptID, DeptName) VALUES (2, 'Mechanical');
Query OK, 1 row affected (0.02 sec)

mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)

mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> SAVEPOINT sp1;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO Department (DeptID, DeptName) VALUES (3, 'Electrical');
Query OK, 1 row affected (0.00 sec)

mysql> select *from department;
+-----+-----+
| deptid | deptname      |
+-----+-----+
|      1 | Computer Science |
|      2 | Mechanical        |
|      3 | Electrical         |
+-----+-----+
3 rows in set (0.00 sec)

mysql> ROLLBACK TO sp1;
Query OK, 0 rows affected (0.02 sec)

mysql> select *from department;
+-----+-----+
| deptid | deptname      |
+-----+-----+
|      1 | Computer Science |
|      2 | Mechanical        |
+-----+-----+
2 rows in set (0.00 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

mysql> select *from department;
+-----+-----+
| deptid | deptname      |
+-----+-----+
|      1 | Computer Science |
|      2 | Mechanical        |
+-----+-----+
2 rows in set (0.00 sec)
```

#### **EX. No. 6 – DCL Commands: GRANT and REVOKE**

```
mysql> create database university;
Query OK, 1 row affected (0.01 sec)

mysql> use university;
Database changed
mysql> CREATE TABLE Department (
    ->     DeptID INT PRIMARY KEY,
    ->     DeptName VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> create 'user1'@'localhost' identified by 'user123';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
1
mysql> create user 'user1'@'localhost' identified by 'user123';
Query OK, 0 rows affected (0.04 sec)

mysql> GRANT SELECT, INSERT ON Department TO 'user1';
ERROR 1410 (42000): You are not allowed to create a user with GRANT
mysql> GRANT SELECT, INSERT ON univesity.Department TO 'user1'@'localhost';
ERROR 1146 (42S02): Table 'univesity.Department' doesn't exist
mysql> GRANT SELECT, INSERT ON university.Department TO 'user1'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> show grants for 'user1'@'localhost';
+-----+
| Grants for user1@localhost |
+-----+
| GRANT USAGE ON *.* TO 'user1'@'localhost'
| GRANT SELECT, INSERT ON `university`.`department` TO 'user1'@'localhost' |
+-----+
2 rows in set (0.00 sec)

mysql> REVOKE INSERT ON university.Department FROM 'user1'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> show grants for 'user1'@'localhost';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
mysql> show grants for 'user1'@'localhost';
+-----+
| Grants for user1@localhost |
+-----+
| GRANT USAGE ON *.* TO 'user1'@'localhost'
| GRANT SELECT ON `university`.`department` TO 'user1'@'localhost' |
+-----+
2 rows in set (0.00 sec)
```

## EX. No. 7 – SELECT with Various Clauses: WHERE, Pattern Matching

```
mysql> use Employee;
Database changed
mysql> SELECT EmpName FROM Employee WHERE DeptID = 1;
Empty set (0.00 sec)

mysql> INSERT INTO Employee (EmpID, EmpName, DeptID)
    -> VALUES (101, 'Alice', 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Employee (EmpID, EmpName, DeptID) values(102,'Kalyan',2);
Query OK, 1 row affected (0.02 sec)

mysql> select *from employee;
+-----+-----+-----+
| empid | empname | deptid |
+-----+-----+-----+
|   101 | Alice   |     1 |
|   102 | Kalyan  |     2 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT EmpName FROM Employee WHERE DeptID = 1;
+-----+
| EmpName |
+-----+
| Alice   |
+-----+
1 row in set (0.00 sec)

mysql> SELECT EmpName FROM Employee WHERE EmpName LIKE 'A%';
+-----+
| EmpName |
+-----+
| Alice   |
+-----+
1 row in set (0.00 sec)
```

#### EX. No. 8 – SELECT with Various Clauses: BETWEEN, IN, Aggregate Function

```
mysql> use shop;
Database changed
mysql> CREATE TABLE Product (
    ->     ProductID INT PRIMARY KEY,
    ->     ProductName VARCHAR(50),
    ->     Price DECIMAL(10,2),
    ->     Quantity INT
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO Product VALUES
    -> (1, 'Soap', 25.00, 100),
    -> (2, 'Shampoo', 95.50, 50),
    -> (3, 'Toothpaste', 45.00, 80);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT Price FROM Product WHERE Price BETWEEN 50 AND 100;
+-----+
| Price |
+-----+
| 95.50 |
+-----+
1 row in set (0.00 sec)

mysql> use employee;
Database changed
mysql> SELECT EmpName FROM Employee WHERE DeptID IN (1, 2);
+-----+
| EmpName |
+-----+
| Alice   |
| Kalyan  |
+-----+
2 rows in set (0.00 sec)

mysql> use shop;
Database changed
mysql> SELECT AVG(Price) FROM Product;
+-----+
| AVG(Price) |
+-----+
| 55.166667 |
+-----+
1 row in set (0.02 sec)
```

## EX. No. 9 – SELECT with Various Clauses: GROUP BY, HAVING, ORDER BY

```
mysql> use employee;
Database changed
mysql> SELECT DeptID, COUNT(*) FROM Employee GROUP BY DeptID;
+-----+-----+
| DeptID | COUNT(*) |
+-----+-----+
|      1 |        1 |
|      2 |        1 |
+-----+-----+
2 rows in set (0.01 sec)

mysql> SELECT DeptID, COUNT(*) FROM Employee GROUP BY DeptID HAVING COUNT(*) > 1;
Empty set (0.00 sec)

mysql> SELECT EmpName, empid FROM Employee ORDER BY Empid DESC;
+-----+-----+
| EmpName | empid |
+-----+-----+
| Kalyan  |    102 |
| Alice   |    101 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT DeptID, COUNT(*) FROM Employee GROUP BY DeptID HAVING COUNT(*) > 0;
+-----+-----+
| DeptID | COUNT(*) |
+-----+-----+
|      1 |        1 |
|      2 |        1 |
+-----+-----+
2 rows in set (0.00 sec)
```

## EX. No. 10 – Query with Subquery and Correlated Query

```
mysql> use school;
Database changed
mysql> CREATE TABLE Department (
    ->     DeptID INT PRIMARY KEY,
    ->     DeptName VARCHAR(50),
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE Employee (
    ->     EmpID INT PRIMARY KEY,
    ->     EmpName VARCHAR(50),
    ->     DeptID INT,
    ->     FOREIGN KEY (DeptID) REFERENCES Department(DeptID)
    -> );
Query OK, 0 rows affected (0.05 sec)

mysql> INSERT INTO Department VALUES (1, 'Computer Science');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Department VALUES (2, 'Mechanical');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Department VALUES (3, 'Electrical');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee VALUES (101, 'Alice', 1);
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO Employee VALUES (102, 'Bob', 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee VALUES (103, 'Charlie', 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee VALUES (104, 'David', 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee VALUES (105, 'Eve', 2);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT EmpName FROM Employee WHERE DeptID IN (SELECT DeptID FROM Department WHERE DeptName = 'Computer Science');
+-----+
| EmpName |
+-----+
| Alice   |
| Charlie |
+-----+
2 rows in set (0.00 sec)
```

```
2 rows in set (0.00 sec)

mysql> SELECT EmpName FROM Employee e WHERE EXISTS (SELECT 1 FROM Department d WHERE e.DeptID = d.DeptID AND d.DeptName = 'Mechanical');
+-----+
| EmpName |
+-----+
| Bob    |
| Eve   |
+-----+
2 rows in set (0.02 sec)
```

### EX. No. 11 – Query with Joins: EquiJoin, InnerJoin, OuterJoin

```
mysql> SELECT Employee.EmpName, Department.DeptName
-> FROM Employee, Department
-> WHERE Employee.DeptID = Department.DeptID;
+-----+-----+
| EmpName | DeptName |
+-----+-----+
| Alice   | Computer Science |
| Charlie | Computer Science |
| Bob     | Mechanical      |
| Eve     | Mechanical      |
| David   | Electrical       |
+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT Employee.EmpName, Department.DeptName
-> FROM Employee
-> INNER JOIN Department ON Employee.DeptID = Department.DeptID;
+-----+-----+
| EmpName | DeptName |
+-----+-----+
| Alice   | Computer Science |
| Charlie | Computer Science |
| Bob     | Mechanical      |
| Eve     | Mechanical      |
| David   | Electrical       |
+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT Employee.EmpName, Department.DeptName
-> FROM Employee
-> LEFT OUTER JOIN Department ON Employee.DeptID = Department.DeptID;
+-----+-----+
| EmpName | DeptName |
+-----+-----+
| Alice   | Computer Science |
| Bob     | Mechanical      |
| Charlie | Computer Science |
| David   | Electrical       |
| Eve     | Mechanical      |
+-----+-----+
5 rows in set (0.02 sec)

mysql> |
```

### **EX. No. 12 – Query with VIEW and INDEX**

### EX. No. 13 – Query with AUTO\_INCREMENT SEQUENCES

```
mysql> use employee;
Database changed
mysql> CREATE TABLE Employee (
    ->     EmpID INT AUTO_INCREMENT,
    ->     EmpName VARCHAR(50),
    ->     DeptID INT,
    ->     PRIMARY KEY (EmpID)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO Employee (EmpName, DeptID) VALUES ('Alice', 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee (EmpName, DeptID) VALUES ('Bob', 2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee (EmpName, DeptID) VALUES ('Charlie', 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee (EmpName, DeptID) VALUES ('David', 3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Employee (EmpName, DeptID) VALUES ('Eva', 2);
Query OK, 1 row affected (0.00 sec)

mysql> select *from employee;
+-----+-----+-----+
| EmpID | EmpName | DeptID |
+-----+-----+-----+
|     1 | Alice   |     1 |
|     2 | Bob     |     2 |
|     3 | Charlie |     1 |
|     4 | David   |     3 |
|     5 | Eva     |     2 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### EX. No. 14 – Simple Programming using REPEAT and WHILE

```
mysql> DROP PROCEDURE IF EXISTS repeat_insert;
Query OK, 0 rows affected (0.03 sec)

mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE repeat_insert()
-> BEGIN
->     DECLARE counter INT DEFAULT 1;
->     REPEAT
->         INSERT INTO Employee (EmpName, DeptID)
->             VALUES (CONCAT('Employee', counter), 1);
->
->         SET counter = counter + 1;
->     UNTIL counter > 5
->     END REPEAT;
-> END;
-> //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
mysql> ;
ERROR:
No query specified

mysql> CALL repeat_insert();
Query OK, 1 row affected (0.04 sec)

mysql> SELECT * FROM Employee;
+-----+-----+-----+
| EmpID | EmpName   | DeptID |
+-----+-----+-----+
|    1  | Employee1 |      1 |
|    2  | Employee2 |      1 |
|    3  | Employee3 |      1 |
|    4  | Employee4 |      1 |
|    5  | Employee5 |      1 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE insert_with_while()
-> BEGIN
->     DECLARE counter INT DEFAULT 1;
->     WHILE counter <= 5 DO
->         INSERT INTO Employee (EmpName, DeptID)
->             VALUES (CONCAT('Employee', counter), 1);
->         SET counter = counter + 1;
->     END WHILE;
-> END;
-> //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
mysql>
mysql> CALL insert_with_while();
Query OK, 1 row affected (0.01 sec)

mysql> select *from employee;
+----+----+----+
| EmpID | EmpName | DeptID |
+----+----+----+
|    1 | Employee1 |      1 |
|    2 | Employee2 |      1 |
|    3 | Employee3 |      1 |
|    4 | Employee4 |      1 |
|    5 | Employee5 |      1 |
|    6 | Employee1 |      1 |
|    7 | Employee2 |      1 |
|    8 | Employee3 |      1 |
|    9 | Employee4 |      1 |
|   10 | Employee5 |      1 |
+----+----+----+
10 rows in set (0.00 sec)
```

### EX. No. 15 – Simple Programming using CASE and LOOP

```
mysql> create database university2;
Query OK, 1 row affected (0.01 sec)

mysql> use university2;
Database changed
mysql> CREATE TABLE Employee (
    ->     EmpID INT AUTO_INCREMENT PRIMARY KEY,
    ->     EmpName VARCHAR(50),
    ->     DeptID INT,
    ->     Salary DECIMAL(10, 2)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO Employee (EmpName, DeptID, Salary) VALUES
    -> ('Alice', 1, 6000),
    -> ('Bob', 1, 4500),
    -> ('Charlie', 2, 3000),
    -> ('David', 2, 2500),
    -> ('Eva', 3, 7000);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT EmpName,
    ->     CASE
    ->         WHEN Salary > 5000 THEN 'High'
    ->         WHEN Salary BETWEEN 3000 AND 5000 THEN 'Medium'
    ->         ELSE 'Low'
    ->     END AS Salary_Level
    -> FROM Employee;
+-----+-----+
| EmpName | Salary_Level |
+-----+-----+
| Alice   | High          |
| Bob    | Medium         |
| Charlie | Medium         |
| David  | Low           |
| Eva    | High          |
+-----+-----+
```

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE insert_using_loop()
-> BEGIN
->     SET @counter = 1;
->     loop_label: LOOP
->         INSERT INTO Employee (EmpName, DeptID) VALUES (CONCAT('Employee', @counter), 3);
->         SET @counter = @counter + 1;
->         IF @counter > 5 THEN
->             LEAVE loop_label;
->         END IF;
->     END LOOP;
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> ;
ERROR:
No query specified

mysql> call insert_using-loop;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'call insert_using-loop' at line 1
mysql> CALL insert_using_loop();
Query OK, 0 rows affected (0.03 sec)

mysql> select *from employee;
+-----+-----+-----+-----+
| EmpID | EmpName | DeptID | Salary |
+-----+-----+-----+-----+
|    1 | Alice   |      1 | 6000.00 |
|    2 | Bob     |      1 | 4500.00 |
|    3 | Charlie |      2 | 3000.00 |
|    4 | David   |      2 | 2500.00 |
|    5 | Eva     |      3 | 7000.00 |
|    6 | Employee1 |      3 | NULL    |
|    7 | Employee2 |      3 | NULL    |
|    8 | Employee3 |      3 | NULL    |
|    9 | Employee4 |      3 | NULL    |
|   10 | Employee5 |      3 | NULL    |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## EX. No. 16 – High-Level Programming Extensions – Procedures

```
mysql> use college1;
Database changed
mysql> CREATE TABLE Employee (
    ->     EmpID INT AUTO_INCREMENT,
    ->     EmpName VARCHAR(50),
    ->     DeptID INT,
    ->     PRIMARY KEY (EmpID)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE InsertEmployee(
    ->     IN emp_name VARCHAR(50),
    ->     IN dept_id INT
    -> )
    -> BEGIN
    ->     INSERT INTO Employee (EmpName, DeptID) VALUES (emp_name, dept_id);
    -> END //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> ;
ERROR:
No query specified

mysql> CALL InsertEmployee('John Doe', 1);
Query OK, 1 row affected (0.01 sec)

mysql> select *from employee;
+-----+-----+-----+
| EmpID | EmpName | DeptID |
+-----+-----+-----+
|     1 | John Doe |      1 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

## EX. No. 17 – High-Level Programming Extensions – Function

```
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> DELIMITER $$

mysql>
mysql> CREATE FUNCTION CalculateSalary(emp_id INT)
-> RETURNS INT
-> DETERMINISTIC
-> BEGIN
->     DECLARE emp_salary INT;
->
->     SELECT Salary INTO emp_salary
->     FROM Employee
->     WHERE EmpID = emp_id;
->
->     RETURN emp_salary;
-> END$$
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> SELECT CalculateSalary(1); -- Returns 7000 (guna's salary)
+-----+
| CalculateSalary(1) |
+-----+
|          7000 |
+-----+
1 row in set (0.00 sec)
```

## EX. No. 18 – High-Level Programming Extensions – Cursors

```
mysql> CREATE PROCEDURE process_employee_names()
-> BEGIN
->     DECLARE done INT DEFAULT 0;
->     DECLARE emp_name VARCHAR(50);
->     DECLARE employee_cursor CURSOR FOR
->         SELECT EmpName FROM Employee;
->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
->
->     OPEN employee_cursor;
->
->     read_loop: LOOP
->         FETCH employee_cursor INTO emp_name;
->         IF done THEN
->             LEAVE read_loop;
->         END IF;
->         -- You can replace this with any action (e.g., insert into log table)
->         SELECT emp_name;
->     END LOOP;
->
->     CLOSE employee_cursor;
-> END;
-> //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
mysql> ;
ERROR:
No query specified

mysql> CALL process_employee_names();
+-----+
| emp_name |
+-----+
| Alice    |
+-----+
1 row in set (0.00 sec)

+-----+
| emp_name |
+-----+
| Bob      |
+-----+
1 row in set (0.00 sec)

+-----+
| emp_name |
+-----+
| Charlie  |
+-----+
1 row in set (0.01 sec)
```

## EX. No. 19 – High-Level Programming Extensions – Triggers

```
mysql> create database school1;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE TABLE AuditLog (
    ->     LogID INT AUTO_INCREMENT PRIMARY KEY,
    ->     Action VARCHAR(20),
    ->     EmpID INT,
    ->     Date DATETIME
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> DELIMITER //
mysql>
mysql> CREATE TRIGGER EmployeeInsert
    -> AFTER INSERT ON Employee
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO AuditLog (Action, EmpID, Date)
    ->     VALUES ('INSERT', NEW.EmpID, NOW());
    -> END;
    -> //
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> DELIMITER ;
mysql> ;
ERROR:
No query specified

mysql> INSERT INTO Employee (EmpID, EmpName, DeptID)
    -> VALUES (101, 'Alice', 1);
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM AuditLog;
+-----+-----+-----+-----+
| LogID | Action | EmpID | Date          |
+-----+-----+-----+-----+
|     1 | INSERT |   101 | 2025-08-05 13:54:26 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

## EX. No. 20 – MySQL String Functions – REPLACE, REPEAT, REVERSE, RIGHT, LEFT, RPAD, LPAD

```
mysql> USE StringFunctionsDB;
Database changed
mysql> CREATE TABLE Messages (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     text VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO Messages (text) VALUES
-> ('Hello World'),
-> ('MySQL is powerful'),
-> ('Welcome to Database'),
-> ('Pad this'),
-> ('Reverse me');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT REPLACE('Hello World', 'World', 'MySQL');
+-----+
| REPLACE('Hello World', 'World', 'MySQL') |
+-----+
| Hello MySQL                                |
+-----+
1 row in set (0.00 sec)

mysql> SELECT REPEAT('Hello ', 3);
+-----+
| REPEAT('Hello ', 3) |
+-----+
| Hello Hello Hello |
+-----+
1 row in set (0.00 sec)

mysql> SELECT REVERSE('MySQL');
+-----+
| REVERSE('MySQL') |
+-----+
| LQSyM           |
+-----+
1 row in set (0.00 sec)

mysql> SELECT RIGHT('Hello World', 5);
+-----+
| RIGHT('Hello World', 5) |
+-----+
| World                  |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LEFT('Hello World', 5);
+-----+
| LEFT('Hello World', 5) |
+-----+
| Hello                 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT RPAD('Hello', 10, '!');
+-----+
| RPAD('Hello', 10, '!') |
+-----+
| Hello!!!!!!           |
+-----+
```

```
mysql> SELECT RPAD('Hello', 10, '!');
+-----+
| RPAD('Hello', 10, '!) |
+-----+
| Hello!!!! |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LPAD('Hello', 10, '!');
+-----+
| LPAD('Hello', 10, '!) |
+-----+
| !!!!!Hello |
+-----+
1 row in set (0.00 sec)

mysql> |
```

**EX. No. 21 – MySQL String Functions – SPACE, SUBSTR, UPPER, LOWER, TRIM, LENGTH**

```
mysql> SELECT CONCAT('Start', SPACE(5), 'End') AS Result;
+-----+
| Result |
+-----+
| Start      End |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUBSTR(text, 1, 5) AS Substring FROM Messages;
+-----+
| Substring |
+-----+
| Hello     |
| MySQL    |
| Welco    |
| Pad t    |
| Rever   |
+-----+
5 rows in set (0.00 sec)

mysql> SELECT UPPER(text) AS UpperText FROM Messages;
+-----+
| UpperText |
+-----+
| HELLO WORLD |
| MYSQL IS POWERFUL |
| WELCOME TO DATABASE |
| PAD THIS |
| REVERSE ME |
+-----+
5 rows in set (0.00 sec)

mysql> SELECT LOWER(text) AS LowerText FROM Messages;
+-----+
| LowerText |
+-----+
| hello world |
| mysql is powerful |
| welcome to database |
| pad this |
| reverse me |
+-----+
5 rows in set (0.00 sec)

mysql> SELECT TRIM('    Hello World    ') AS Trimmed;
+-----+
| Trimmed |
+-----+
| Hello World |
+-----+
1 row in set (0.00 sec)

mysql> SELECT text, LENGTH(text) AS TextLength FROM Messages;
+-----+-----+
| text          | TextLength |
+-----+-----+
| Hello World      |      11 |
| MySQL is powerful |      17 |
| Welcome to Database |      19 |
| Pad this      |      8 |
| Reverse me     |      10 |
+-----+-----+
5 rows in set (0.00 sec)
```

## EX. No. 22 – Database Connectivity Using PHP and MySQL

```
mysql> CREATE DATABASE new_college;
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> USE new_college;
Database changed
mysql>
mysql> CREATE TABLE students (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(50),
    ->     department VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> INSERT INTO students (name, department) VALUES
    -> ('Rahul', 'Computer Science'),
    -> ('Priya', 'Electronics'),
    -> ('Anil', 'Mechanical');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> |
```

Connected successfully  
ID: 1 - Name: Rahul - Department: Computer Science  
ID: 2 - Name: Priya - Department: Electronics  
ID: 3 - Name: Anil - Department: Mechanical

## EX. No. 23 – Case Scenario: MySQL Queries Implementation for Train Ticket Reservation System to Receive Tickets through Social Network

```
mysql> create database train;
ERROR 1007 (HY000): Can't create database 'train'; database exists
mysql> use train;
Database changed
mysql> CREATE TABLE TicketReservation (
    ->     TicketID INT AUTO_INCREMENT PRIMARY KEY,
    ->     CustomerName VARCHAR(100),
    ->     TrainNumber VARCHAR(50),
    ->     DateOfJourney DATE,
    ->     SocialNetwork VARCHAR(100),
    ->     TicketStatus VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO TicketReservation (CustomerName, TrainNumber, DateOfJourney, SocialNetwork, TicketStatus)
-> VALUES ('John Doe', '101', '2025-05-01', 'Facebook', 'Booked');
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM TicketReservation;
+-----+-----+-----+-----+-----+
| TicketID | CustomerName | TrainNumber | DateOfJourney | SocialNetwork | TicketStatus |
+-----+-----+-----+-----+-----+
|      1   | John Doe     | 101        | 2025-05-01   | Facebook     | Booked       |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

## EX. No. 24 – Case Scenario: MySQL Queries Implementation for College Admission Form

```
mysql> create database college5;
Query OK, 1 row affected (0.01 sec)

mysql> ^C
mysql> use college5;
Database changed
mysql> CREATE TABLE CollegeAdmission (
    ->     StudentID INT AUTO_INCREMENT PRIMARY KEY,
    ->     StudentName VARCHAR(100),
    ->     Course VARCHAR(100),
    ->     AdmissionDate DATE,
    ->     Status VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO CollegeAdmission (StudentName, Course, AdmissionDate, Status)
-> VALUES ('Alice Smith', 'Computer Science', '2025-05-10', 'Admitted');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM CollegeAdmission;
+-----+-----+-----+-----+-----+
| StudentID | StudentName | Course          | AdmissionDate | Status  |
+-----+-----+-----+-----+-----+
|      1     | Alice Smith  | Computer Science | 2025-05-10   | Admitted |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

## EX. No. 25 – Case Scenario: MySQL Queries Implementation for QR Enabled Automatic Bus Ticket Booking System

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

mysql> use bus;
Database changed
mysql> CREATE TABLE BusTicketBooking (
    ->     BookingID INT AUTO_INCREMENT PRIMARY KEY,
    ->     PassengerName VARCHAR(100),
    ->     BusNumber VARCHAR(50),
    ->     TravelDate DATE,
    ->     QRCode VARCHAR(255),
    ->     Status VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO BusTicketBooking (PassengerName, BusNumber, TravelDate, QRCode, Status)
-> VALUES ('Michael Johnson', 'B123', '2025-06-01', 'QR1234567890', 'Booked');
Query OK, 1 row affected (0.01 sec)

mysql> select *from busticketbooking;
+-----+-----+-----+-----+-----+
| BookingID | PassengerName | BusNumber | TravelDate | QRCode      | Status   |
+-----+-----+-----+-----+-----+
|       1 | Michael Johnson | B123      | 2025-06-01 | QR1234567890 | Booked  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

💡 Ticket Booked Successfully!  
Booking ID: **1**  
Name: **Gowrish**  
Bus: **AP29AT5176**  
Date: **2025-08-23**  
Seat: **16U**

Scan Your QR Ticket:



 [Open QR Code in New Tab](#)