Assignment – 5

-Yash Yadav(107)

```
CREATE DATABASE loop_lab_db;
USE loop_lab_db;

CREATE TABLE employees (     id INT
AUTO_INCREMENT PRIMARY KEY,     name
VARCHAR(50),     department VARCHAR(30),
salary DECIMAL(10,2)
);

INSERT INTO employees (name, department, salary) VALUES
('Anita', 'HR', 25000),
('Bhavesh', 'IT', 32000),
('Chitra', 'Finance', 28000),
('Deepak', 'IT', 40000),
('Esha', 'HR', 35000),
('Farhan', 'Finance', 30000);
```

LOOP

1. Increase salary of first 3 employees by 10%.

```
DELIMITER $$

CREATE PROCEDURE increase_salary()
BEGIN
   DECLARE i INT DEFAULT 1;
   DECLARE total INT DEFAULT 3;

   salary_loop: LOOP
      UPDATE employees
      SET salary = salary * 1.10
      WHERE id = i;

      SET i = i + 1;
      IF i > total THEN
         LEAVE salary_loop;
      END IF;
   END LOOP salary_loop;
END$$

DELIMITER ;
```

CALL increase_salary();

Output:

```
mysql> select * from employees;
+----+---------+------------+----------+
| id | name    | department | salary   |
+----+---------+------------+----------+
|  1 | Anita   | HR         | 27500.00 |
|  2 | Bhavesh | IT         | 35200.00 |
|  3 | Chitra  | Finance    | 30800.00 |
|  4 | Deepak  | IT         | 40000.00 |
|  5 | Esha    | HR         | 35000.00 |
|  6 | Farhan  | Finance    | 30000.00 |
+----+---------+------------+----------+
6 rows in set (0.00 sec)
```

2. Display all employee names using LOOP.

```
DELIMITER $$

CREATE PROCEDURE display_employee_names()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE emp_name VARCHAR(50);

  DECLARE emp_cursor CURSOR FOR SELECT name FROM employees;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

  OPEN emp_cursor;

  emp_loop: LOOP
    FETCH emp_cursor INTO emp_name;
    IF done THEN
      LEAVE emp_loop;
    END IF;
    SELECT emp_name AS 'Employee Name';
  END LOOP emp_loop;

  CLOSE emp_cursor;
END$$

DELIMITER ;

CALL display_employee_names();
```

Output:

```
+---------------+
| Employee Name |
+---------------+
| Anita         |
+---------------+
1 row in set (0.00 sec)

+---------------+
| Employee Name |
+---------------+
| Bhavesh       |
+---------------+
1 row in set (0.00 sec)

+---------------+
| Employee Name |
+---------------+
| Chitra        |
+---------------+
1 row in set (0.00 sec)

+---------------+
| Employee Name |
+---------------+
| Deepak        |
+---------------+
1 row in set (0.00 sec)

+---------------+
| Employee Name |
+---------------+
| Esha          |
+---------------+
1 row in set (0.00 sec)

+---------------+
| Employee Name |
+---------------+
| Farhan        |
+---------------+
1 row in set (0.01 sec)
```

3. Calculate total salary of all employees using LOOP.

```
DELIMITER $$

CREATE PROCEDURE total_salary()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE sal DECIMAL(10,2);
  DECLARE total DECIMAL(10,2) DEFAULT 0;

  DECLARE cur CURSOR FOR SELECT salary FROM employees;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

  OPEN cur;

  read_loop: LOOP
    FETCH cur INTO sal;
    IF done THEN
      LEAVE read_loop;
    END IF;
    SET total = total + sal;
  END LOOP read_loop;

  CLOSE cur;

  SELECT total AS 'Total Salary of Employees';
END$$

DELIMITER ;

CALL total_salary();
```

Output:

```
mysql> CALL total_salary();
+---------------------------+
| Total Salary of Employees |
+---------------------------+
|                 198500.00 |
+---------------------------+
1 row in set (0.00 sec)
```

4. Insert 3 new temporary employees into the table using LOOP.

```
DELIMITER $$
```

```
CREATE PROCEDURE insert_temp_employees()
BEGIN
  DECLARE i INT DEFAULT 1;

  insert_loop: LOOP
    INSERT INTO employees (name, department, salary)
    VALUES (CONCAT('TempEmployee', i), 'Temp', 20000 + (i * 1000));

    SET i = i + 1;

    IF i > 3 THEN
      LEAVE insert_loop;
    END IF;
  END LOOP insert_loop;
END$$

DELIMITER ;

CALL insert_temp_employees();
```

Output:

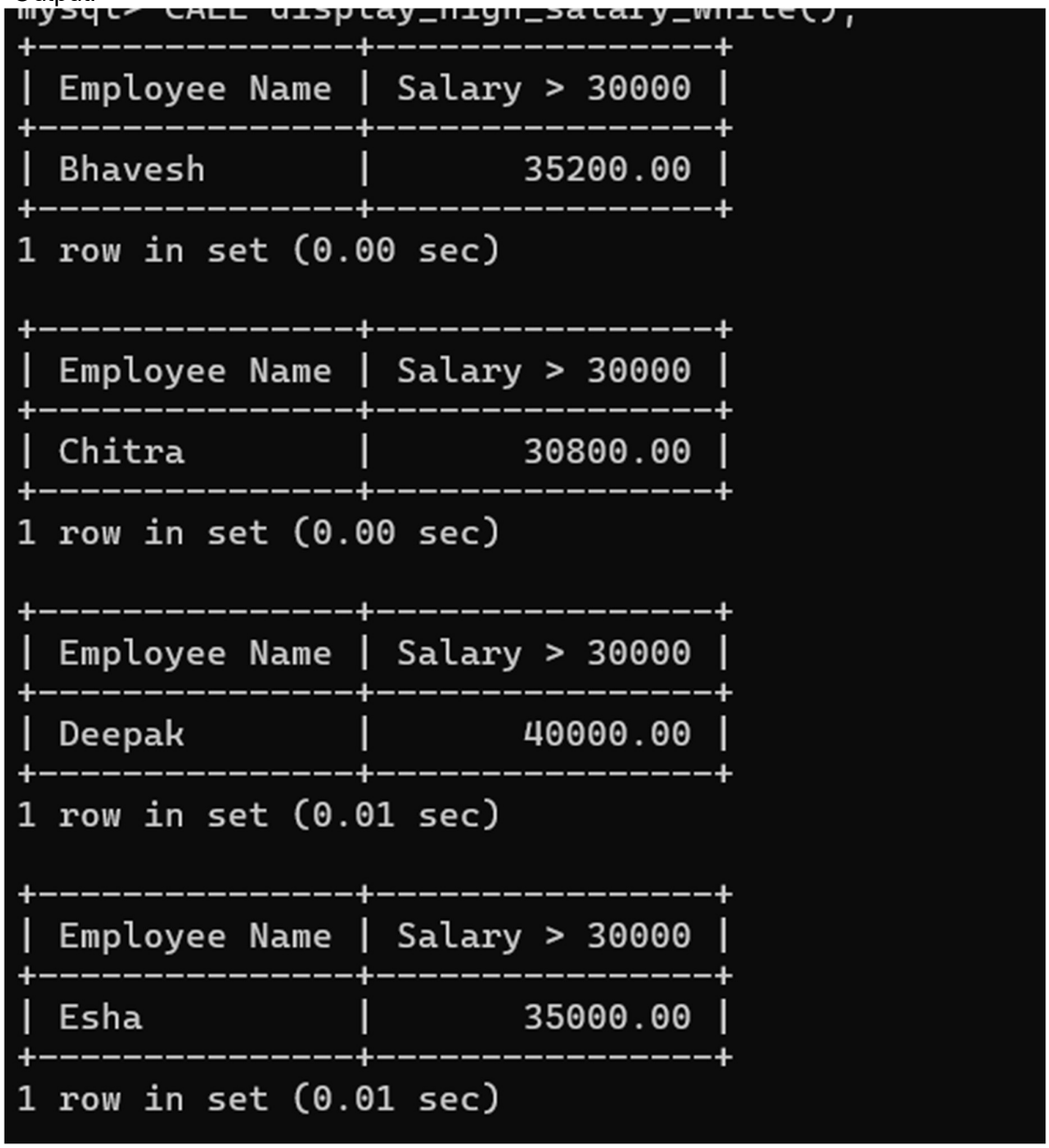
```
mysql> select * from employees;
+----+---------------+------------+----------+
| id | name          | department | salary   |
+----+---------------+------------+----------+
|  1 | Anita         | HR         | 27500.00 |
|  2 | Bhavesh       | IT         | 35200.00 |
|  3 | Chitra        | Finance    | 30800.00 |
|  4 | Deepak        | IT         | 40000.00 |
|  5 | Esha          | HR         | 35000.00 |
|  6 | Farhan        | Finance    | 30000.00 |
|  7 | TempEmployee1 | Temp       | 21000.00 |
|  8 | TempEmployee2 | Temp       | 22000.00 |
|  9 | TempEmployee3 | Temp       | 23000.00 |
| 10 | TempEmployee1 | Temp       | 21000.00 |
| 11 | TempEmployee2 | Temp       | 22000.00 |
| 12 | TempEmployee3 | Temp       | 23000.00 |
| 13 | TempEmployee1 | Temp       | 21000.00 |
| 14 | TempEmployee2 | Temp       | 22000.00 |
| 15 | TempEmployee3 | Temp       | 23000.00 |
+----+---------------+------------+----------+
15 rows in set (0.00 sec)
```

WHILE

5. Display employees having salary greater than ₹30,000.

```
DELIMITER $$

CREATE PROCEDURE display_high_salary_while()
BEGIN
   DECLARE done INT DEFAULT 0;
   DECLARE emp_name VARCHAR(50);
   DECLARE emp_salary DECIMAL(10,2);

   DECLARE cur CURSOR FOR SELECT name, salary FROM employees;
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

   OPEN cur;

   read_loop: WHILE done = 0 DO
      FETCH cur INTO emp_name, emp_salary;
      IF done = 1 THEN
         LEAVE read_loop;
      END IF;

      IF emp_salary > 30000 THEN
         SELECT emp_name AS 'Employee Name', emp_salary AS 'Salary > 30000';
      END IF;
   END WHILE;

   CLOSE cur;
END$$

DELIMITER ;


CALL display_high_salary_while();
```

Output:

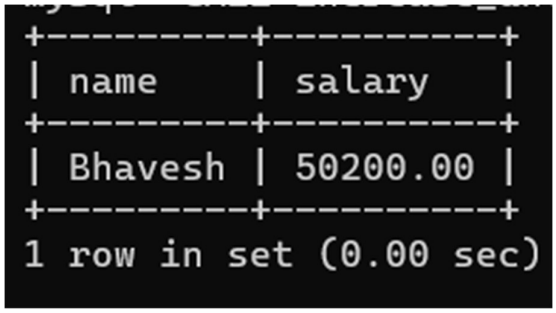


6. Increase one employee's salary incrementally by 5000 until it reaches ₹50,000.

```
DELIMITER $$

CREATE PROCEDURE increase_until_50000_while(IN emp_id INT)
BEGIN
    DECLARE current_salary DECIMAL(10,2);

    -- Get current salary
    SELECT salary INTO current_salary FROM employees WHERE id = emp_id;

    -- Keep increasing salary until it reaches 50,000
    WHILE current_salary < 50000 DO
        SET current_salary = current_salary + 5000;
```

```
        UPDATE employees
        SET salary = current_salary
        WHERE id = emp_id;
    END WHILE;

    -- Show final result
    SELECT name, salary
    FROM employees
    WHERE id = emp_id;
END$$

DELIMITER ;

CALL increase_until_50000_while(2);
```

Output:



7. Count total IT department employees.

```
DELIMITER $$



CREATE PROCEDURE count_it_employees_while()

BEGIN

   DECLARE done INT DEFAULT 0;

   DECLARE dept VARCHAR(30);

   DECLARE count_it INT DEFAULT

  DECLARE cur CURSOR FOR SELECT department FROM employees;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

 OPEN cur;
```

```
    count_loop: WHILE done = 0 DO
        FETCH cur INTO dept;
        IF done = 1 THEN
            LEAVE count_loop;
        END IF;

        IF dept = 'IT' THEN
            SET count_it = count_it + 1;
        END IF;
    END WHILE;
    CLOSE cur;
    SELECT count_it AS 'Total IT Department Employees';
END$$
DELIMITER ;
CALL count_it_employees_while();
```

Output:

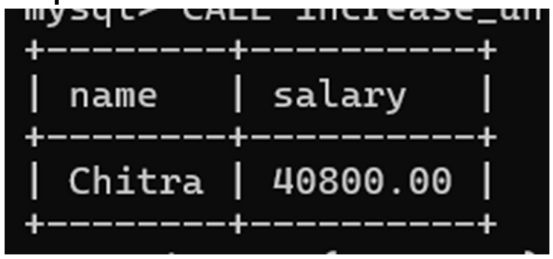
```
mysql> CALL count_it_employees_whi
+-------------------------------+
| Total IT Department Employees |
+-------------------------------+
|                             2 |
+-------------------------------+
1 row in set (0.00 sec)
```

REPEAT

8. Add ₹5,000 to one employee’s salary until it exceeds ₹40,000.
   DELIMITER $$

```
CREATE PROCEDURE increase_until_40000_repeat(IN emp_id INT)
BEGIN
  DECLARE current_salary DECIMAL(10,2);

  -- Get current salary
  SELECT salary INTO current_salary
  FROM employees
  WHERE id = emp_id;

  REPEAT
    SET current_salary = current_salary + 5000;

    UPDATE employees
    SET salary = current_salary
    WHERE id = emp_id;
  UNTIL current_salary > 40000
  END REPEAT;

  -- Show final result
  SELECT name, salary
  FROM employees
  WHERE id = emp_id;
END$$

DELIMITER ;

CALL increase_until_40000_repeat(3);
```

Output:



9. Insert employees until 10 total records exist.

```
DELIMITER $$

CREATE PROCEDURE insert_until_10_repeat()
BEGIN
  DECLARE total_count INT;

  SELECT COUNT(*) INTO total_count FROM employees;

  REPEAT
    SET total_count = total_count + 1;
```

```
    INSERT INTO employees (name, department, salary)
    VALUES (CONCAT('TempEmployee', total_count), 'Temp', 25000);

    SELECT COUNT(*) INTO total_count FROM employees;
  UNTIL total_count >= 10
  END REPEAT;

  -- Show final table
  SELECT * FROM employees;
END$$

DELIMITER ;

CALL insert_until_10_repeat();
```

Output:

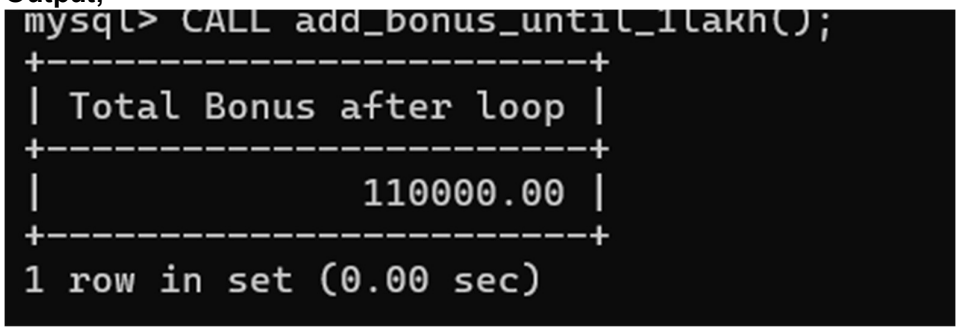
id	name	department	salary
1	Anita	HR	27500.00
2	Bhavesh	IT	50200.00
3	Chitra	Finance	40800.00
4	Deepak	IT	40000.00
5	Esha	HR	35000.00
6	Farhan	Finance	30000.00
7	TempEmployee1	Temp	21000.00
8	TempEmployee2	Temp	22000.00
9	TempEmployee3	Temp	23000.00
10	TempEmployee1	Temp	21000.00
11	TempEmployee2	Temp	22000.00
12	TempEmployee3	Temp	23000.00
13	TempEmployee1	Temp	21000.00
14	TempEmployee2	Temp	22000.00
15	TempEmployee3	Temp	23000.00
16	TempEmployee16	Temp	25000.00

10. Keep adding bonuses until total crosses ₹1,00,000.

```
DELIMITER $$

CREATE PROCEDURE add_bonus_until_1lakh()
BEGIN
```

```
    DECLARE total_bonus DECIMAL(10,2) DEFAULT 0;
    DECLARE bonus DECIMAL(10,2) DEFAULT 10000;

    REPEAT
        SET total_bonus = total_bonus + bonus;
    UNTIL total_bonus > 100000
    END REPEAT;

    SELECT total_bonus AS 'Total Bonus after loop';
END$$

DELIMITER ;

CALL add_bonus_until_1lakh();
```

Output;



Additional (Optional)

LOOP

1. **Display employee names department-wise using LOOP.**
   Loop through all departments and show employee names under each.

```
DELIMITER $$

CREATE PROCEDURE display_names_deptwise()
BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE emp_name VARCHAR(50);
    DECLARE dept VARCHAR(30);
    DECLARE prev_dept VARCHAR(30) DEFAULT '';

    DECLARE cur CURSOR FOR
        SELECT name, department FROM employees ORDER BY department;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

    OPEN cur;
```

```
    dept_loop: LOOP
        FETCH cur INTO emp_name, dept;
        IF done THEN
            LEAVE dept_loop;
        END IF;

        IF dept <> prev_dept THEN
            SELECT CONCAT('Department: ', dept) AS Department_Header;
            SET prev_dept = dept;
        END IF;

        SELECT emp_name AS 'Employee Name';
    END LOOP dept_loop;

    CLOSE cur;
END$$

DELIMITER ;

CALL display_names_deptwise();
```

Output:

```
+--------------------+
| Department_Header  |
+--------------------+
| Department: Finance |
+--------------------+
1 row in set (0.00 sec)

+---------------+
| Employee Name |
+---------------+
| Chitra        |
+---------------+
1 row in set (0.00 sec)

+---------------+
| Employee Name |
+---------------+
| Farhan        |
+---------------+
1 row in set (0.00 sec)

+--------------------+
| Department_Header |
+--------------------+
| Department: HR    |
+--------------------+
1 row in set (0.01 sec)

+---------------+
| Employee Name |
+---------------+
| Anita         |
+---------------+
1 row in set (0.01 sec)

+---------------+
| Employee Name |
+---------------+
| Esha          |
+---------------+
1 row in set (0.01 sec)
```

2. **Grant a ₹2,000 performance bonus to each IT department employee using LOOP.**

Traverse through each employee and update if the department is ‘IT’.

```
DELIMITER $$

CREATE PROCEDURE add_bonus_it_dept()
BEGIN
   DECLARE done INT DEFAULT 0;
   DECLARE emp_id INT;
   DECLARE dept VARCHAR(30);

   DECLARE cur CURSOR FOR SELECT id, department FROM employees;
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

   OPEN cur;

   bonus_loop: LOOP
      FETCH cur INTO emp_id, dept;
      IF done THEN
         LEAVE bonus_loop;
      END IF;

      IF dept = 'IT' THEN
         UPDATE employees
         SET salary = salary + 2000
         WHERE id = emp_id;
      END IF;
   END LOOP bonus_loop;

   CLOSE cur;
END$$

DELIMITER ;


CALL add_bonus_it_dept();


SELECT * FROM employees WHERE department = 'IT';
```

Output:

```
mysql> SELECT * FROM employees WHERE depart
+----+---------+------------+----------+
| id | name    | department | salary   |
+----+---------+------------+----------+
|  2 | Bhavesh | IT         | 52200.00 |
|  4 | Deepak  | IT         | 42000.00 |
+----+---------+------------+----------+
2 rows in set (0.00 sec)
```

3. **Find and display the highest salary among all employees using LOOP.**

```
DELIMITER $$

CREATE PROCEDURE highest_salary_loop()
BEGIN
   DECLARE done INT DEFAULT 0;
   DECLARE sal DECIMAL(10,2);
   DECLARE max_sal DECIMAL(10,2) DEFAULT 0;

   DECLARE cur CURSOR FOR SELECT salary FROM employees;
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

   OPEN cur;

   sal_loop: LOOP
      FETCH cur INTO sal;
      IF done THEN
         LEAVE sal_loop;
      END IF;

      IF sal > max_sal THEN
         SET max_sal = sal;
      END IF;
   END LOOP sal_loop;

   CLOSE cur;

   SELECT max_sal AS 'Highest Salary';
END$$
```

```
DELIMITER ;
CALL highest_salary_loop();
```

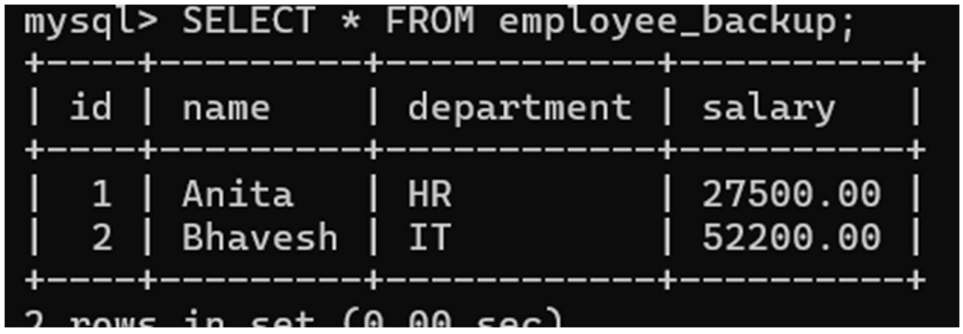
```
mysql> SELECT * FROM employees WHERE department
+----+---------+------------+----------+
| id | name    | department | salary   |
+----+---------+------------+----------+
|  2 | Bhavesh | IT         | 52200.00 |
|  4 | Deepak  | IT         | 42000.00 |
+----+---------+------------+----------+
2 rows in set (0.00 sec)
```

4. **Copy data of first 2 employees into a new table `employee_backup` using LOOP.**

```
-- Create backup table
CREATE TABLE IF NOT EXISTS employee_backup (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50),
    department VARCHAR(30),
    salary DECIMAL(10,2)
);


DELIMITER $$


CREATE PROCEDURE copy_first_two_loop()
BEGIN
    DECLARE i INT DEFAULT 1;
    DECLARE emp_name VARCHAR(50);
    DECLARE dept VARCHAR(30);
    DECLARE sal DECIMAL(10,2);

    copy_loop: LOOP
        SELECT name, department, salary
```

```
        INTO emp_name, dept, sal
        FROM employees
        WHERE id = i;

        INSERT INTO employee_backup (name, department, salary)
        VALUES (emp_name, dept, sal);

        SET i = i + 1;

        IF i > 2 THEN
            LEAVE copy_loop;
        END IF;
    END LOOP copy_loop;
END$$

DELIMITER ;
CALL copy_first_two_loop();
SELECT * FROM employee_backup;
```

Output:

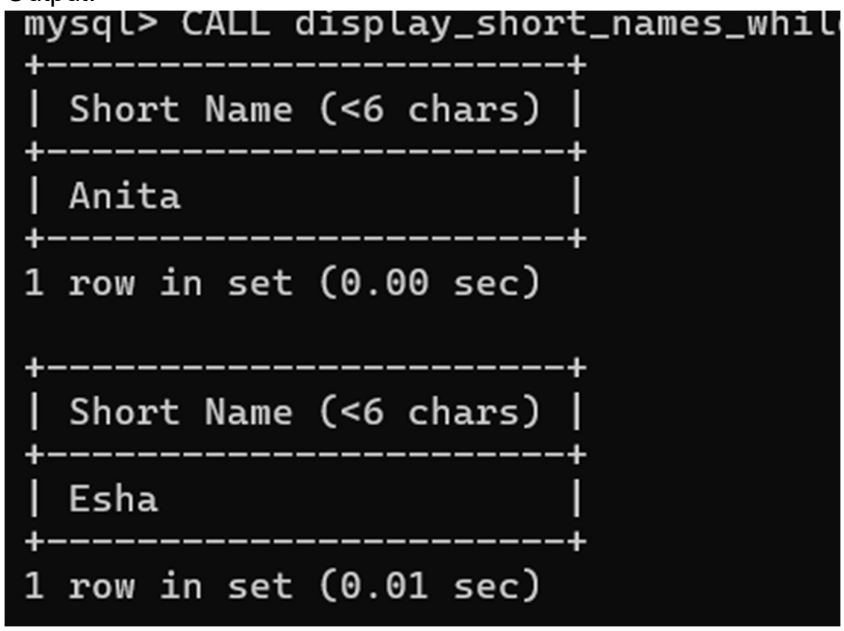


WHILE

5. **Display employees whose name length is less than 6 characters using WHILE.**

```
DELIMITER $$

CREATE PROCEDURE display_short_names_while()
BEGIN
   DECLARE i INT DEFAULT 1;
   DECLARE total INT;

   SELECT COUNT(*) INTO total FROM employees;

   WHILE i <= total DO
      IF LENGTH((SELECT name FROM employees WHERE id = i)) < 6 THEN
         SELECT name AS 'Short Name (<6 chars)'
         FROM employees WHERE id = i;
      END IF;

      SET i = i + 1;
   END WHILE;
END$$

DELIMITER ;

CALL display_short_names_while();
```

Output:



6. **Deduct ₹1,000 from one employee's salary until it reaches ₹25,000 using WHILE.**

```
DELIMITER $$

CREATE PROCEDURE reduce_salary_until_25000_while(IN emp_id INT)
```

```
BEGIN
   DECLARE current_salary DECIMAL(10,2);

   SELECT salary INTO current_salary FROM employees WHERE id = emp_id;

   WHILE current_salary > 25000 DO
      SET current_salary = current_salary - 1000;

      UPDATE employees
      SET salary = current_salary
      WHERE id = emp_id;
   END WHILE;

   SELECT name, salary
   FROM employees
   WHERE id = emp_id;
END$$

DELIMITER ;
CALL reduce_salary_until_25000_while(1);
```

Output:

```
mysql> CALL reduce_salary_unt
+-------+----------+
| name  | salary   |
+-------+----------+
| Anita | 24500.00 |
+-------+----------+
1 row in set (0.01 sec)
```

7. **Calculate average salary of all employees using WHILE.**

```
DELIMITER $$

CREATE PROCEDURE average_salary_while()

BEGIN

   DECLARE i INT DEFAULT 1;

   DECLARE total INT;

   DECLARE sal DECIMAL(10,2);

   DECLARE sum_sal DECIMAL(10,2) DEFAULT 0;

   DECLARE avg_sal DECIMAL(10,2);
```

```
    SELECT COUNT(*) INTO total FROM employees;
      WHILE i <= total DO
       SELECT salary INTO sal FROM employees WHERE id = i;
       SET sum_sal = sum_sal + sal;
       SET i = i + 1;
      END WHILE;
    SET avg_sal = sum_sal / total;
     SELECT avg_sal AS 'Average Salary';
    END$$
  DELIMITER;
 CALL average_salary_while();
```

Output:

```
mysql> CALL average_salary_while(
+----------------+
| Average Salary |
+----------------+
|       27968.75 |
+----------------+
1 row in set (0.00 sec)
```

REPEAT

8. **Keep inserting random temporary employees until 15 total employees exist using REPEAT.**

```
 DELIMITER $$

CREATE PROCEDURE insert_until_15_repeat()
BEGIN
   DECLARE total_count INT;

   SELECT COUNT(*) INTO total_count FROM employees;

   REPEAT
      SET total_count = total_count + 1;
```

```
        INSERT INTO employees (name, department, salary)
        VALUES (CONCAT('TempEmployee', total_count), 'Temp', 20000 + (total_count *
500));

        SELECT COUNT(*) INTO total_count FROM employees;
    UNTIL total_count >= 15
    END REPEAT;

    SELECT * FROM employees;
END$$

DELIMITER ;

CALL insert_until_15_repeat();
```

Output:

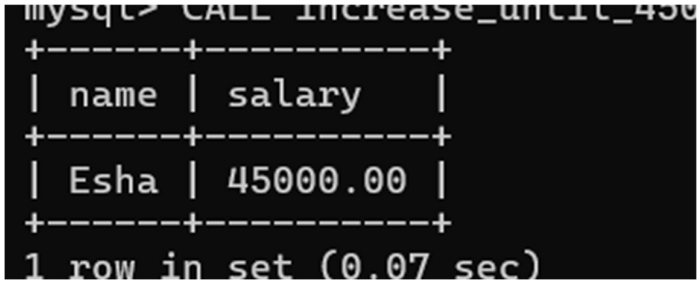
```
+----+----------------+------------+----------+
| id | name           | department | salary   |
+----+----------------+------------+----------+
|  1 | Anita          | HR         | 24500.00 |
|  2 | Bhavesh        | IT         | 52200.00 |
|  3 | Chitra         | Finance    | 40800.00 |
|  4 | Deepak         | IT         | 42000.00 |
|  5 | Esha           | HR         | 35000.00 |
|  6 | Farhan         | Finance    | 30000.00 |
|  7 | TempEmployee1  | Temp       | 21000.00 |
|  8 | TempEmployee2  | Temp       | 22000.00 |
|  9 | TempEmployee3  | Temp       | 23000.00 |
| 10 | TempEmployee1  | Temp       | 21000.00 |
| 11 | TempEmployee2  | Temp       | 22000.00 |
| 12 | TempEmployee3  | Temp       | 23000.00 |
| 13 | TempEmployee1  | Temp       | 21000.00 |
| 14 | TempEmployee2  | Temp       | 22000.00 |
| 15 | TempEmployee3  | Temp       | 23000.00 |
| 16 | TempEmployee16 | Temp       | 25000.00 |
| 17 | TempEmployee17 | Temp       | 28500.00 |
+----+----------------+------------+----------+
```

9. **Keep adding ₹500 increments to a specific employee's salary until it reaches ₹45,000 using REPEAT.**

```
 DELIMITER $$

CREATE PROCEDURE increase_until_45000_repeat(IN emp_id INT)
BEGIN
```

```
    DECLARE current_salary DECIMAL(10,2);

    SELECT salary INTO current_salary FROM employees WHERE id = emp_id;

    REPEAT
      SET current_salary = current_salary + 500;
      UPDATE employees
      SET salary = current_salary
      WHERE id = emp_id;
    UNTIL current_salary >= 45000
    END REPEAT;

    SELECT name, salary
    FROM employees
    WHERE id = emp_id;
  END$$

  DELIMITER ;
  CALL increase_until_45000_repeat(5);
```

Output:



10. **Display employee salaries one by one using REPEAT until all records are printed.**

```
 DELIMITER $$

CREATE PROCEDURE display_salaries_repeat()

BEGIN

  DECLARE i INT DEFAULT 1;

  DECLARE total INT;

  DECLARE emp_name VARCHAR(50);

  DECLARE emp_salary DECIMAL(10,2);

SELECT COUNT(*) INTO total FROM employees;
```

```
    REPEAT
        SELECT name, salary INTO emp_name, emp_salary
        FROM employees WHERE id = i;
       SELECT CONCAT('Employee: ', emp_name, ' | Salary: ₹', emp_salary) AS Details;
        SET i = i + 1;
    UNTIL i > total
    END REPEAT;
END$$
DELIMITER ;
CALL display_salaries_repeat();
```

Output:

```
mysql> CALL display_salaries_repeat();
+-----------------------------------+
| Details                           |
+-----------------------------------+
| Employee: Anita | Salary: ?24500.00 |
+-----------------------------------+
1 row in set (0.00 sec)

+-------------------------------------+
| Details                             |
+-------------------------------------+
| Employee: Bhavesh | Salary: ?52200.00 |
+-------------------------------------+
1 row in set (0.01 sec)

+-----------------------------------+
| Details                           |
+-----------------------------------+
| Employee: Chitra | Salary: ?40800.00 |
+-----------------------------------+
1 row in set (0.01 sec)

+-----------------------------------+
| Details                           |
+-----------------------------------+
| Employee: Deepak | Salary: ?42000.00 |
+-----------------------------------+
1 row in set (0.01 sec)

+----------------------------------+
| Details                          |
+----------------------------------+
| Employee: Esha | Salary: ?45000.00 |
+----------------------------------+
1 row in set (0.01 sec)

+-----------------------------------+
| Details                           |
+-----------------------------------+
| Employee: Farhan | Salary: ?30000.00 |
+-----------------------------------+
```