Answers

1. Write a SQL query to **create a stored procedure without any parameters** that displays all employees from the Emp table.

```
DELIMITER $$

CREATE PROCEDURE GetAllEmployees()

BEGIN

   SELECT * FROM Emp;

END $$

DELIMITER ;
```

2. Write a SQL query to **create a stored procedure with an IN parameter** that accepts a department ID and displays all employees belonging to that department.

```
DELIMITER $$

CREATE PROCEDURE GetEmployeesByDept(IN p_DeptId INT)

BEGIN

   SELECT * FROM Emp

   WHERE DeptId = p_DeptId;

END $$

DELIMITER ;
```

3. Write a SQL query to **create a stored procedure with an OUT parameter** that returns the total number of employees in the Emp table.

```
DELIMITER $$

CREATE PROCEDURE GetTotalEmployees(OUT totalEmp INT)

BEGIN

   SELECT COUNT(*) INTO totalEmp FROM Emp;

END $$

DELIMITER ;
```

4. **Write a SQL function that accepts an employee's salary as input and returns a grade based on the following conditions:**

If salary ≥ 80,000 → Grade = 'A'

If salary ≥ 50,000 and < 80,000 → Grade = 'B'

If salary ≥ 30,000 and < 50,000 → Grade = 'C'

Otherwise → Grade = 'D'

Use appropriate **IF / IF-ELSE / CASE statements** inside the function to implement this logic.

```
DELIMITER $$

CREATE FUNCTION GetSalaryGrade(salary DECIMAL(10,2))
RETURNS CHAR(1)
DETERMINISTIC
BEGIN
   DECLARE grade CHAR(1);
   IF salary >= 80000 THEN
      SET grade = 'A';
   ELSEIF salary >= 50000 THEN
      SET grade = 'B';
   ELSEIF salary >= 30000 THEN
      SET grade = 'C';
   ELSE
      SET grade = 'D';
   END IF;

   RETURN grade;
END $$
DELIMITER ;
```

5. Write a stored procedure that uses an **explicit cursor** to fetch and display the details of all employees whose salary is greater than 60,000 from the Emp table. Make sure to DECLARE, OPEN, FETCH, and CLOSE the cursor properly.

```
DELIMITER $$
CREATE PROCEDURE GetHighSalaryEmployees()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE e_id INT;
  DECLARE e_name VARCHAR(100);
  DECLARE e_salary DECIMAL(10,2);
  DECLARE emp_cursor CURSOR FOR
    SELECT EmpId, EmpName, Salary FROM Emp WHERE Salary > 60000;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

  OPEN emp_cursor;

  read_loop: LOOP
    FETCH emp_cursor INTO e_id, e_name, e_salary;
    IF done THEN
      LEAVE read_loop;
    END IF;
    SELECT e_id AS EmpId, e_name AS Name, e_salary AS Salary;
  END LOOP;

  CLOSE emp_cursor;
END $$
DELIMITER ;
```

6. Write a trigger on the Emp table that checks before inserting a new employee record: If the Salary is less than 10,000, prevent the insertion and raise an error message "Salary too low".

```
DELIMITER $$

CREATE TRIGGER CheckSalaryBeforeInsert

BEFORE INSERT ON Emp

FOR EACH ROW

BEGIN

  IF NEW.Salary < 10000 THEN

    SIGNAL SQLSTATE '45000'

    SET MESSAGE_TEXT = 'Salary too low';

  END IF;

END $$

DELIMITER ;
```

7. Write a stored procedure in SQL to **print numbers from 1 to 10** using a **WHILE loop**.

```
DELIMITER $$


CREATE PROCEDURE PrintNumbers()

BEGIN

   DECLARE i INT DEFAULT 1;

   WHILE i <= 10 DO

      SELECT i AS Number;

      SET i = i + 1;

   END WHILE;

END $$


DELIMITER ;
```

8. Write a stored procedure to print the multiplication table of 2 using a loop

```
DELIMITER $$

CREATE PROCEDURE PrintTableOf2()

BEGIN

   DECLARE i INT DEFAULT 1;

   WHILE i <= 10 DO

      SELECT CONCAT('2 x ', i, ' = ', 2 * i) AS Result;

      SET i = i + 1;

   END WHILE;

END $$

DELIMITER ;
```

9. Write a function **to** check whether a number is even or odd.

```
DELIMITER $$

CREATE FUNCTION CheckEvenOdd(num INT)

RETURNS VARCHAR(10)

DETERMINISTIC

BEGIN

   DECLARE result VARCHAR(10);

   IF MOD(num, 2) = 0 THEN

      SET result = 'Even';

   ELSE

      SET result = 'Odd';

   END IF;

   RETURN result;

END $$

DELIMITER ;
```

10. Write a user-defined function to calculate the factorial of a given number.

```
DELIMITER $$

CREATE FUNCTION GetFactorial(n INT)
RETURNS BIGINT
DETERMINISTIC
BEGIN
   DECLARE fact BIGINT DEFAULT 1;
   DECLARE i INT DEFAULT 1;

   WHILE i <= n DO
      SET fact = fact * i;
      SET i = i + 1;
   END WHILE;

   RETURN fact;
END $$

DELIMITER ;
```