STORED FUNCTIONS

Introduction

A **Function** in MySQL is a stored program that always returns a single value. Unlike procedures, functions can be used directly inside SQL queries such as SELECT, WHERE, and other expressions.

They are mainly used for calculations, validations, and returning computed results.

Key Points

- Must return exactly one value using RETURN.
- Can only use **IN** parameters (no OUT or INOUT).
- Cannot perform transaction control (COMMIT, ROLLBACK).
- Defined with CREATE FUNCTION and executed automatically in queries.

Syntax

```
DELIMITER $$
CREATE FUNCTION function_name(parameter datatype, ...) RETURNS return_datatype
DETERMINISTIC
BEGIN
  -- SQL statements
  RETURN value;
END $$
DELIMITER ;
```

DETERMINISTIC means the function always returns the same result for the same input.

TASK 0: Create Table and Insert Data

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(50),
    dept VARCHAR(20),
    salary DECIMAL(10,2),
    join_date DATE
);
INSERT INTO employees VALUES
(1, 'Rahul', 'HR', 35000, '2021-01-15'),
(2, 'Priya', 'IT', 50000, '2020-07-20'),
(3, 'Amit', 'Finance', 45000, '2019-05-12'),
(4, 'Sneha', 'IT', 60000, '2022-03-01'),
(5, 'Vikas', 'HR', 40000, '2021-11-23');
```

TASK 1: Function to get Annual Salary

```
DELIMITER $$
CREATE FUNCTION GetAnnualSalary(emp INT) RETURNS
DECIMAL(12,2)
DETERMINISTIC
BEGIN
   DECLARE annual DECIMAL(12,2);
   SELECT salary*12 INTO annual FROM employees WHERE emp_id = emp;
   RETURN annual;
END $$
DELIMITER ;
SELECT emp_name, GetAnnualSalary(emp_id) AS Annual_Salary FROM
employees;
```

TASK 2: Function to Get Employee Department

```
DELIMITER $$
CREATE FUNCTION GetDepartment(emp INT) RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
  DECLARE dep VARCHAR(20);
  SELECT dept INTO dep FROM employees WHERE emp_id = emp;
  RETURN dep;
END $$
DELIMITER ;
```

```
SELECT emp_name, GetDepartment(emp_id) AS Department FROM
employees;
```

TASK 3: Function to Find Experience in Years

```
DELIMITER $$
CREATE FUNCTION GetExperience(emp INT) RETURNS INT
DETERMINISTIC
BEGIN
  DECLARE exp INT;
  SELECT TIMESTAMPDIFF(YEAR, join_date, CURDATE()) INTO exp
  FROM employees WHERE emp_id = emp;
  RETURN exp;
END $$
DELIMITER ;
```

```
SELECT emp_name, GetExperience(emp_id) AS Experience_Years FROM
employees;
```

TASK 4: Function to Check High Earner (>= 50000)

```
DELIMITER $$
CREATE FUNCTION IsHighEarner(emp INT) RETURNS TINYINT
DETERMINISTIC
BEGIN
  DECLARE sal DECIMAL(10,2);
  SELECT salary INTO sal FROM employees WHERE emp_id = emp;
  IF sal >= 50000 THEN
    RETURN 1;
  ELSE
    RETURN 0;
  END IF;
END $$
DELIMITER ;
SELECT emp_name, IsHighEarner(emp_id) AS HighEarner FROM employees;
```

TASK 5: Function to Get Join Month Name

```
DELIMITER $$
CREATE FUNCTION GetJoinMonth(emp INT) RETURNS VARCHAR(15)
DETERMINISTIC
BEGIN
  DECLARE m VARCHAR(15);
  SELECT MONTHNAME(join_date) INTO m FROM employees WHERE
emp_id = emp;
  RETURN m;
END $$
DELIMITER ;
SELECT emp_name, GetJoinMonth(emp_id) AS JoinMonth FROM employees;
```

SPOT QUESTIONS

1. Write a function fn_TotalEmployees() that returns the total number of employees.
2. Write a function fn_MaxSalary() that returns the maximum salary.
3. Write a function fn_YearsInCompany(emp INT) that returns how many years the employee has worked.
4. Write a function fn_IsInDept(emp INT, deptName VARCHAR(20)) that returns 1 if employee belongs to deptName, else 0.
5. Write a function fn_JoinYear(emp INT) that returns the year of joining for a given employee.