Stored Procedures in MySQL

Instructor: Sriya Ivaturi

What is a Stored Procedure?

- A Stored Procedure is a pre-written set of SQL statements stored in the database under a name.

  - It can be executed using a single CALL statement.

  - Helps in automating repetitive tasks.

  - Think of it like a reusable SQL function.

- Q1: What is the main difference between a stored procedure and a SQL query?

Why Use Stored Procedures?

- Reusability: Write once, use often.
- Security: Users can execute without seeing internal SQL.
- Efficiency: Reduces network traffic.
- Maintainability: Changes in one place.

Syntax of Stored Procedure

```
DELIMITER $$
CREATE PROCEDURE procedure_name([parameters])
BEGIN
   -- SQL statements
END $$
DELIMITER ;
```

- Q3: Why is the delimiter changed?

- A: To avoid conflict with semicolons inside the procedure.

Basic Procedure (No Parameters)

```
DELIMITER $$
CREATE PROCEDURE SayHello()
BEGIN
   SELECT 'Hello from MySQL!';
END $$
DELIMITER ;

CALL SayHello();
```

- Q4: What happens if we call SayHello before creating it?

- A: MySQL will return: PROCEDURE SayHello does not exist.

Procedure with IN Parameter

```
DELIMITER $$
CREATE PROCEDURE GreetUser(IN userName
VARCHAR(50))
BEGIN
   SELECT CONCAT('Hello, ', userName, '!') AS Greeting;
END $$
DELIMITER ;

CALL GreetUser('Ravi');
```

- Q5: What does the IN keyword do?

- A: Allows passing a value into the procedure.

Procedure with Two IN Parameters

```
DELIMITER $$
CREATE PROCEDURE AddTwoNumbers(IN a INT, IN b INT)
BEGIN
   SELECT a + b AS Sum;
END $$
DELIMITER ;
```

- CALL AddTwoNumbers(10, 20);

Procedure with OUT Parameter

```
DELIMITER $$
CREATE PROCEDURE SquareNumber(IN input INT, OUT result INT)
BEGIN
   SET result = input * input;
END $$
DELIMITER ;
```

- CALL SquareNumber(5, @out);
- SELECT @out;

- Q6: What does OUT do?

- A: Sends result out of the procedure to a session variable.

Procedure with INOUT Parameter

```
DELIMITER $$
CREATE PROCEDURE DoubleValue(INOUT num INT)
BEGIN
   SET num = num * 2;
END $$
DELIMITER ;
```

- SET @val = 10;
- CALL DoubleValue(@val);
- SELECT @val;

- Q7: What does INOUT do?

- A: It receives and modifies a value.

Mixed Parameters

```
DELIMITER $$
CREATE PROCEDURE ProcessMarks(
    IN name VARCHAR(50),
    IN mark1 INT,
    IN mark2 INT,
    OUT total INT,
    OUT average DECIMAL(5,2))
BEGIN
   SET total = mark1 + mark2;
   SET average = total / 2;
END $$
DELIMITER ;
```

- CALL ProcessMarks('Amit', 80, 90, @tot, @avg);
- SELECT @tot, @avg;

- Q8: Can we have multiple OUTs?

- A: Yes.

Procedure Calling Another Procedure

```
DELIMITER $$
CREATE PROCEDURE OuterProcedure()
BEGIN
   CALL SayHello();
   CALL AddTwoNumbers(5, 7);
END $$
DELIMITER ;
```

- CALL OuterProcedure();

- Q9: Can we nest procedures?

- A: Yes.

Summary

- - Procedures encapsulate SQL logic
- - Use IN, OUT, INOUT
- - Improve security and performance
- - Use DELIMITER properly
- - Nesting supported

Practice Challenges

- 1. Greet user and show current date.
- 2. Swap two numbers using INOUT.
- 3. Calculate area and perimeter of rectangle.