--------LECTURE- 17--------

1️⃣ Simple Stored Procedure (No Parameters)

What it means

➡ A procedure that **does one task** and **does not take any input**.

When to use

- Fixed logic
- Same output every time

Example

```
DELIMITER $$

CREATE PROCEDURE GetAllEmployees()
BEGIN
  SELECT * FROM employees;
END $$

DELIMITER ;
```

Call it

```
CALL GetAllEmployees();
```

📌 **Explanation**

- No input needed
- Just runs the SQL inside it
- Returns all employees

2️⃣ Stored Procedure with IN Parameter

What is IN?

➡ IN means **input value**

➡ You **pass data into the procedure**

When to use

- Filter data
- Search by ID, name, date, etc.

Example

```
CREATE PROCEDURE GetEmployeeById(IN emp_id INT)
BEGIN
  SELECT *
  FROM employees
  WHERE id = emp_id;
END;
```

Call it

```
CALL GetEmployeeById(5);
```

📌 **Explanation**

- emp_id comes **from outside**
- Used inside the query

- Very common in real projects

3️⃣ Stored Procedure with OUT Parameter

What is OUT?

➡ OUT means **output value**

➡ Procedure **returns a value into a variable**

When to use

- Count
- Total
- Status
- Calculated values

Example

```
CREATE PROCEDURE GetEmployeeCount(OUT total_emp INT)
BEGIN
   SELECT COUNT(*) INTO total_emp
   FROM employees;
END;
```

Call it

```
CALL GetEmployeeCount(@count);
SELECT @count;
```

📌 **Explanation**

- Procedure calculates something
- Stores result in total_emp
- Value is saved into @count

4️⃣ Stored Procedure with IN + OUT Together

What this means

➡ Give **input**

➡ Get **output**

Real-life use

- Input: department ID
- Output: number of employees

Example

```
CREATE PROCEDURE GetEmpCountByDept(
   IN dept_id INT,
   OUT emp_count INT
)
BEGIN
   SELECT COUNT(*)
   INTO emp_count
   FROM employees
   WHERE department_id = dept_id;
END;
```

Call it

```
CALL GetEmpCountByDept(2, @total);
SELECT @total;
```

📌 **Explanation**

- dept_id → input
- emp_count → output
- Very common interview example

5️⃣ Stored Procedure with Multiple IN Parameters

When to use

- More filtering conditions

Example

```
CREATE PROCEDURE GetEmployeeDetails(
  IN dept_id INT,
  IN job_title VARCHAR(50)
)
BEGIN
  SELECT *
  FROM employees
  WHERE department_id = dept_id
   AND job = job_title;
END;
```

Call it

```
CALL GetEmployeeDetails(1, 'Manager');
```

6️⃣ Stored Procedure with INOUT Parameter

What is INOUT?

➡ Parameter acts as **input + output**

Example

```
CREATE PROCEDURE IncreaseSalary(INOUT salary INT)
BEGIN
  SET salary = salary + 5000;
END;
```

Call it

```
SET @sal = 30000;
CALL IncreaseSalary(@sal);
SELECT @sal;
```

📌 **Explanation**

- Takes 30000 as input
- Adds 5000
- Returns updated value

🔑 Quick Comparison Table (Very Important for Interviews)

Type	Purpose
Simple Procedure	Fixed logic
IN	Accepts input
OUT	Returns output
IN + OUT	Input + result
INOUT	Modify and return same value

💬 explanation (easy to remember)

Stored procedures can be written without parameters, with IN parameters to accept input, with OUT parameters to return results, or with INOUT parameters to both receive and return values.