1. What is a Stored Procedure?

- A **stored procedure** is a **precompiled block of SQL statements** stored in the database that can be **executed by calling it**.
- A Stored Procedure is a precompiled block of SQL code that you can save and reuse.

---

2. Why use Stored Procedures?

1. Reusable logic
2. Better performance (precompiled)
3. Reduces network traffic
4. Secure (can hide logic from end users)
5. Useful for business rules and automation

---

3. Syntax in SQL Server:

```
DELIMITER //

CREATE PROCEDURE procedure_name(IN param1 INT, OUT param2
VARCHAR(100))

  BEGIN

  -- SQL statements

  END //

DELIMITER ;
```

- `DELIMITER //` → Temporarily changes the default `;` so the procedure body doesn't terminate early.
- `IN` → Input parameter
- `OUT` → Output parameter
- `INOUT` → Both input and output

In MySQL, the syntax is:

```
DELIMITER //
CREATE PROCEDURE procedure_name()
BEGIN
-- SQL statements
END //
DELIMITER ;
```

To run Stored Procedures Execute Below Query:

```
CALL GetAllEmployees();
```

4. Step-by-Step Example

Step 1: Create Table

```
CREATE TABLE students (
    id INT PRIMARY KEY,
    name VARCHAR(100),
    marks INT
);
```

Step 2: Insert Sample Data

```
INSERT INTO students VALUES (1, 'Vijay', 85);
INSERT INTO students VALUES (2, 'Ashok', 92);
INSERT INTO students VALUES (3, 'Kiran', 76);
```

Step 3: Create a Simple Stored Procedure

```
DELIMITER //
CREATE PROCEDURE getAllStudents()
BEGIN
   SELECT * FROM students;
END //
DELIMITER ;
```

Execute it:

```
CALL getAllStudents();
```

---

5. Procedure with Input Parameter

```
DELIMITER //
CREATE PROCEDURE getStudentById(IN stuId INT)
BEGIN
   SELECT * FROM students WHERE id = stuId;
END //
DELIMITER ;
```

Call it:

```
CALL getStudentById(2);
```

---

6. Procedure with Output Parameter

```
DELIMITER //

CREATE PROCEDURE getStudentName(IN stuId INT, OUT stuName
VARCHAR(100))

BEGIN

   SELECT name INTO stuName FROM students WHERE id = stuId;

END //

DELIMITER ;
```

Call it and see output:

```
CALL getStudentName(1, @name);

SELECT @name;
```

---

8. Drop/Delete Stored Procedure

```
DROP PROCEDURE IF EXISTS getAllStudents;
```

---

9. Advantages Recap

- Reusable, modular
- Secure: Permissions can restrict access
- Fast: Precompiled
- Easy to maintain
- Ideal for business logic in DB layer

Drawbacks / Disadvantages of Stored Procedures:

1. Harder to Debug

Errors inside procedures can be hard to trace.

2. Not Ideal for Complex Business Logic

Loops, conditions, validations — easier and clearer in application code than in SQL syntax.

Best Practices & When to Use Stored Procedures

- **Use when you need to execute the same query multiple times**
  Avoid rewriting complex queries again and again — store once, reuse anytime.
- **Use for performance in bulk operations**
  Ideal for inserting, updating, or deleting thousands of rows in one call to reduce network traffic and improve speed.
- **Use for encapsulating business logic close to the data**
