Stored Procedures

Stored procedure is a reusable block of SQL statements stored in the database.

It is executed using parameters instead of rewriting queries again and again.

Naming Stored Procedures

- Give clear and meaningful names
- No strict rules for case sensitivity
- Uppercase or lowercase does not matter
- Name should explain what procedure does

Structure of a Stored Procedure

```
CREATE PROCEDURE procedure_name(parameters)
BEGIN
statements
END
```

Why we use Stored Procedures

- Reusable logic
- Reduces repeated SQL writing
- Improves readability
- Improves performance
- Adds security (users don't need direct table access)

Parameterizing Stored Procedures

Parameters allow passing values dynamically at runtime.

IN Parameter

- Used to pass input value
- Value is read-only inside procedure

Example idea:
Pass customer_id and fetch payment details.

OUT Parameter

- Used to return value from procedure
- Result is stored in a variable using @

Important:
For output parameter, store value in `@variable_name`

Calling OUT Parameter Procedure

```
CALL procedure_name(input_value, @output_value);
SELECT @output_value;
```

IN + OUT Together

- IN → input to procedure
- OUT → output from procedure

Used when procedure calculates and returns a result.

Stored Procedure Scope

- Stored permanently in database

- Can be reused across sessions
- Can be accessed by multiple users (with permission)

Dynamic Stored Procedures

Dynamic stored procedures create SQL statements at runtime.

Why Dynamic Stored Procedures

- Table names cannot be parameterized normally
- Dynamic SQL allows flexible query generation
- Reuse same logic for different tables

Dynamic SQL Concept

- SQL query is built as a string
- Query is executed dynamically

Steps in Dynamic SQL Procedure

1. Create SQL query as string
2. PREPARE statement
3. EXECUTE statement
4. DEALLOCATE prepared statement

Dynamic SQL Keywords

- SET → store SQL query
- PREPARE → prepare statement
- EXECUTE → run query
- DEALLOCATE PREPARE → free memory

Dynamic Stored Procedure Use Case

- Count rows from different tables
- Create flexible reporting logic
- Runtime table selection

Static vs Dynamic Stored Procedure

Static:

- Fixed query
- Parameters for values only

Dynamic:

- Query structure changes
- Table names or columns passed at runtime

Final Understanding Line

Stored procedure = reusable logic
Dynamic stored procedure = reusable logic + runtime SQL creation

So, procedure is reusable logic, and the dynamic is values are added at runtime.