Conditional Statements in MySQL

What are Conditional Statements?

- Conditional statements control the flow of execution based on conditions.
- Used inside stored procedures or blocks (BEGIN...END) to make decisions.

Three main types in MySQL:

1. IF statement

2. IF...ELSE statement

3. CASE statement

Syntax - IF Statement

IF Statement Syntax:

```
IF condition THEN
   -- statements
END IF;
```

Use Case:

- Use when you want to execute code only if a condition is true.

Example - IF Statement

Example: Check if a number is positive

```
DELIMITER $$
CREATE PROCEDURE CheckPositive(IN num INT)
BEGIN
   IF num > 0 THEN
      SELECT 'Positive Number' AS Result;
   END IF;
END $$
DELIMITER ;

CALL CheckPositive(5);
```

Syntax - IF...ELSE Statement

IF...ELSE Syntax:

```
IF condition THEN
   -- statements if true
ELSE
   -- statements if false
END IF;
```

Use Case:

- Use when you need to handle both true and false outcomes.

Example - IF...ELSE Statement

Example: Check if a number is even or odd

```
DELIMITER $$
CREATE PROCEDURE CheckEvenOdd(IN num INT)
BEGIN
   IF MOD(num, 2) = 0 THEN
      SELECT 'Even Number' AS Result;
   ELSE
      SELECT 'Odd Number' AS Result;
   END IF;
END $$
DELIMITER ;

CALL CheckEvenOdd(7);
```

Syntax - IF...ELSEIF...ELSE

Syntax:

```
IF condition1 THEN
   -- code block 1
ELSEIF condition2 THEN
   -- code block 2
ELSE
   -- code block 3
END IF;
```

Use Case:

- Use when there are multiple conditions to evaluate.

Example - IF...ELSEIF...ELSE

Example: Categorize marks

```
DELIMITER $$
CREATE PROCEDURE GradeStudent(IN marks INT)
BEGIN
  IF marks >= 90 THEN
    SELECT 'Grade A' AS Grade;
  ELSEIF marks >= 75 THEN
    SELECT 'Grade B' AS Grade;
  ELSEIF marks >= 60 THEN
    SELECT 'Grade C' AS Grade;
  ELSE
    SELECT 'Fail' AS Grade;
  END IF;
END $$
DELIMITER ;

CALL GradeStudent(78);
```

Syntax - CASE Statement

CASE Statement Syntax (Simple CASE):

```
CASE expression
   WHEN value1 THEN statement1
   WHEN value2 THEN statement2
   ...
   ELSE default_statement
END CASE;
```

Use Case:

   - Best for comparing one variable against multiple values.

Example - Simple CASE Statement

Example: Day number to name

```
DELIMITER $$
CREATE PROCEDURE DayName(IN dayNum INT)
BEGIN
   CASE dayNum
      WHEN 1 THEN SELECT 'Sunday' AS Day;
      WHEN 2 THEN SELECT 'Monday' AS Day;
      WHEN 3 THEN SELECT 'Tuesday' AS Day;
      ELSE SELECT 'Invalid Day' AS Day;
   END CASE;
END $$
DELIMITER ;

CALL DayName(2);
```

Searched CASE Syntax

Searched CASE Syntax:

```
CASE
   WHEN condition1 THEN statement1
   WHEN condition2 THEN statement2
   ...
   ELSE default_statement
END CASE;
```

Use Case:

- Use when each condition involves different expressions or logic.

Example - Searched CASE Statement

Example: Temperature category

```
DELIMITER $$
CREATE PROCEDURE TempCategory(IN temp INT)
BEGIN
   CASE
      WHEN temp >= 40 THEN SELECT 'Very Hot' AS Category;
      WHEN temp >= 25 THEN SELECT 'Warm' AS Category;
      WHEN temp >= 10 THEN SELECT 'Cool' AS Category;
      ELSE SELECT 'Cold' AS Category;
   END CASE;
END $$
DELIMITER ;

CALL TempCategory(15);
```

Summary

IF – Simple yes/no check

IF...ELSEIF...ELSE – Multiple condition paths

CASE – Clean alternative to multiple IFs

Always use DELIMITER $$ and END blocks properly in stored procedures