Flow Control Statements in MySQL

Instructor: Sriya Ivaturi

What are Flow Control Statements in MySQL

- Flow control statements in MySQL allow us to run blocks of code repeatedly based on conditions.
- Types:
  - LOOP: Basic infinite loop with exit condition inside.
  - WHILE: Runs as long as the condition is TRUE.
  - REPEAT: Runs at least once, checks condition after execution.
- These are similar to control structures in programming languages.
- Use Case: Looping through values, validations, conditional logic in business rules.

Syntax for LOOP

```
[loop_label]: LOOP

   -- Statements to execute repeatedly

   IF exit_condition THEN

       LEAVE [loop_label];  -- Exit the loop

   END IF;

END LOOP [loop_label];
```

Using LOOP in a Stored Procedure

```
DELIMITER $$
CREATE PROCEDURE SumFiveNumbers()
BEGIN
   DECLARE total INT DEFAULT 0;
   DECLARE counter INT DEFAULT 1;

   loop_label: LOOP
       SET total = total + counter;
       SET counter = counter + 1;

       IF counter > 5 THEN
           LEAVE loop_label;
       END IF;
   END LOOP loop_label;

   SELECT total AS SumResult;
END $$
DELIMITER ;

CALL SumFiveNumbers();
```

Syntax Of WHILE

```
[while_label]: WHILE search_condition DO

   -- Statements to execute while the condition is TRUE

END WHILE [while_label];
```

Using WHILE in a Stored Procedure

```
DELIMITER $$
CREATE PROCEDURE FactorialCalc(IN num INT)
BEGIN
   DECLARE result INT DEFAULT 1;
   DECLARE i INT DEFAULT 1;

   WHILE i <= num DO
       SET result = result * i;
       SET i = i + 1;
   END WHILE;

   SELECT result AS Factorial;
END $$
DELIMITER ;

CALL FactorialCalc(5);
```

Syntax Of REPEAT

```
[repeat_label]: REPEAT

    -- Statements to execute

UNTIL search_condition

END REPEAT [repeat_label];
```

Using REPEAT in a Stored Procedure

```
DELIMITER $$
CREATE PROCEDURE RepeatExample()
BEGIN
   DECLARE i INT DEFAULT 1;

   REPEAT
       SELECT CONCAT('Current Value: ', i);
       SET i = i + 1;
   UNTIL i > 5
   END REPEAT;
END $$
DELIMITER ;

CALL RepeatExample();
```

Flow Control in a FUNCTION (WHILE)

```
DELIMITER $$
CREATE FUNCTION IsPrime(n INT)
RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
   DECLARE i INT DEFAULT 2;

   IF n < 2 THEN
       RETURN 'Not Prime';
   END IF;

   WHILE i <= SQRT(n) DO
       IF MOD(n, i) = 0 THEN
           RETURN 'Not Prime';
       END IF;
       SET i = i + 1;
   END WHILE;

   RETURN 'Prime';
END $$
DELIMITER ;

SELECT IsPrime(7); -- Output: Prime
```

Question Slide

Q: What is the main difference between WHILE and REPEAT loops in MySQL?

A) WHILE checks condition before execution, REPEAT checks after.

B) REPEAT runs only once.

C) WHILE always runs at least once.

D) Both behave the same.

Answer Slide

Answer: A) WHILE checks condition before execution,

REPEAT checks after.

Explanation:

- WHILE loop might never run if the condition is FALSE at the start.

- REPEAT loop will always run at least once.

Question Slide

```sql
DELIMITER $$
CREATE PROCEDURE TestLoop()
BEGIN
   DECLARE counter INT DEFAULT 1;
   DECLARE text_out VARCHAR(100) DEFAULT '';

   WHILE counter < 5 DO
       SET text_out = CONCAT(text_out, counter);
       SET counter = counter + 2;
   END WHILE;

   SELECT text_out;
END $$
DELIMITER ;

CALL TestLoop();
```

Options:
A) 1234
B) 135
C) 13
D) 12

Answer Slide

Answer: C) 13

Explanation:

- counter starts at 1

- First loop: text_out = "1", counter = 3

- Second loop: text_out = "13", counter = 5 (loop stops)

Summary and Use Cases

- Summary of Flow Control Statements:
  - LOOP: Flexible, exit using LEAVE.
  - WHILE: Entry-controlled loop, good for known logic checks.
  - REPEAT: Exit-controlled loop, runs at least once.
- Best used in:
  - Calculations like factorials
  - Summation problems
  - Validations
  - Repetitive DB checks or conditions
- Keep logic simple inside loops to avoid infinite executions.