



# Day 18

## MySQL Loops- (Part 1)

### LOOP Statement





# LOOP Statement in MySQL

- The **LOOP** statement allows you to execute one or more statements repeatedly.

MySQL loops can be only used within Stored Procedures.



# General Syntax :

```
[begin_label:] LOOP  
    statement_list  
END LOOP [end_label]
```



## Key Points:

1. The **LOOP** can have optional labels at the beginning and end of the block.
2. The **LOOP** executes the `statement_list` repeatedly.
3. The `statement_list` may have one or more statements, each terminated by a semicolon (;) statement delimiter.



# LEAVE & ITERATE Statemensts

- The **LEAVE** statement immediately exits the loop.
  - It works like the **break** statement in other programming languages like PHP, C/C++, and Java.
- The **ITERATE** statement is used to skip the current loop iteration and start a new iteration.
  - The **ITERATE** is similar to the **continue** statement in PHP, C/C++, and Java.



# Use Case Example :

**Problem Statement:** To print all even numbers up to n, where n is any user-input number.

**Query :**

```
DELIMITER //
```

```
CREATE PROCEDURE get_even_nos(  
    IN N INT  
)  
BEGIN  
    DECLARE x int;  
    DECLARE output VARCHAR(20);  
  
    SET x = 1;  
    SET output = "";  
  
    loop_label_name : LOOP  
        IF x = N THEN  
            LEAVE loop_label_name;  
        ELSEIF x%2 = 0 THEN  
            SET output = CONCAT(output,x,', ');  
        END IF;  
        SET x = x + 1;  
    END LOOP;  
  
    SELECT output;  
END //
```

```
DELIMITER ;
```

1.N=7

```
CALL get_even_nos(7);
```

Output :

	output
▶	2, 4, 6,

2.N=10

```
CALL get_even_nos(10);
```

Output :

	output
▶	2, 4, 6, 8,