LOOP, LEAVE and ITERATE statements

There are two statements that allow you to control the loop:

* The LEAVE statement allows you to exit the loop immediately without waiting for checking the condition. The LEAVE statement works like the  break statement in other languages such as PHP, C/C++, Java, etc.
* The ITERATE statement allows you to skip the entire code under it and start a new iteration. The ITERATE statement is similar to the continuestatement in PHP, C/C++, Java, etc.

MySQL also gives you a LOOPstatement that executes a block of code repeatedly with an additional flexibility of using a loop label.

The following is an example of using the LOOP loop statement.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | CREATE PROCEDURE test\_mysql\_loop()  BEGIN  DECLARE x  INT;          DECLARE str  VARCHAR(255);    SET x = 1;          SET str =  '';    loop\_label:  LOOP  IF  x > 10 THEN  LEAVE  loop\_label;  END  IF;    SET  x = x + 1;  IF  (x mod 2) THEN  ITERATE  loop\_label;  ELSE                  SET  str = CONCAT(str,x,',');  END  IF;           END LOOP;             SELECT str;    END; |

* The stored procedure only constructs a string with even numbers e.g., 2, 4, 6, etc.
* We put a loop\_labelloop label before the LOOPstatement.
* If the value of  x is greater than 10, the loop is terminated because of the LEAVE statement.
* If the value of the x is an odd number, the ITERATE statement ignores everything below it and starts a new iteration.
* If the value of the x is an even number, the block in the ELSE statement will build the string with even numbers.

In this tutorial, you have learned various MySQL loop statements that execute a block of code repeatedly based on a condition.