LOOP

====

Label:loop

IF condition THEN

LEAVE label

END IF;

body of loop

inc / dec

END loop;

Need of cursor

==============

=> select sal INTO temp from employee where id=3;

temp=> 40000

=>select sal INTO temp from employee;

temp => 45000,60000,40000,65000 => ERROR

To process the set of result row by row given by the select

statement we use cursor.

CURSOR

========

cursor is temporary memory location provided by DBMS

to store the set of results return by the select statement.

so that we can process the result row by row.

In MySQL, a cursor allows row-by-row processing of the result sets. A cursor is used for the result set and returned from a query. By using a cursor, you can iterate or step through the results of a query and perform certain operations on each row. The cursor allows you to iterate through the result set and then perform the additional processing only on the rows that require it.

CURSOR select sal INTO temp from employee;

40000 40000

65000 65000

60000 <============ 60000

45000 45000

SYNTAX

========

declare cursor

---------------

DECLARE cursor\_name CURSOR FOR SELECT statement;

OPEN cursor\_name;

label:LOOP

FETCH cursor\_name INTO variable\_list;

logic

END LOOP;

CLOSE cursor\_name;

CURSOR select name,sal INTO temp from employee;

harry,45000 harry,45000

mac,60000 mac,60000

hari,40000 <============ hari,40000

shree,65000 shree,65000

create procedure abc()

-> BEGIN

-> DECLARE i int;

-> SET i=1;

-> myloop:LOOP

-> IF i>=5 THEN

-> LEAVE myloop;

-> END IF;

-> select i;

-> SET i=i+1;

-> END LOOP;

-> END //

EX:

delimiter //

create procedure emp\_grade2()

begin

declare s float;

declare n varchar(20);

declare cur cursor for select name,sal from employee;

open cur ;

myloop :Loop

fetch cur into n,s ;

if s>=6000 then

select n as name,s as salary,'platinum' as grade;

else

select n as name, s as salary ,'gold' as grade;

end if;

end loop;

close cur;

end //

call emp\_grade2();

* Read-Only: Using cursors, you can iterate over a result set row by row but you cannot make changes to the table or update any table.
* Non-Scrollable: Cursors in MySQL allow you to iterate and retrieve data from the result set in only one direction, that is from top to bottom. You cannot go up and down as you want while using cursors in MySQL.
* **Exception Handling with Cursors:**
  + Managing errors that may occur during the execution of cursor operations.
  + Ensuring robust and fault-tolerant code by handling exceptions gracefully.
* **Cursor-Related Exceptions:**
  + NO\_DATA\_FOUND: Raised when a fetch operation does not return any row.
  + TOO\_MANY\_ROWS: Raised when a SELECT INTO statement returns more than one row.
  + INVALID\_CURSOR: Raised when an operation is performed on an unopened or already closed cursor.

**Handling Exceptions:**

* **BEGIN...EXCEPTION Block:**
  + PL/SQL block structure to handle exceptions.

Syntax:  
BEGIN

-- Code to execute

EXCEPTION

WHEN exception\_name THEN

-- Handle exception

* + END;

delimiter //

create procedure emp\_grade6()

begin

declare s float;

declare done int default false;

declare cur cursor for select sal from employee;

declare continue handler for not found set done=true;

open cur ;

myloop :Loop

fetch cur into s ;

if done then

leave myloop;

end if;

if s>=6000 then

select s as salary,'platinum' as grade;

else

select s as salary ,'gold' as grade;

end if;

end loop;

close cur;

end //

call emp\_grade6();