**SQL Cursor:**

A cursor is responsible to repeat a set of rows returned by a query of the SELECT statement where each row is processed separately. Thus, Cursor is a reserved SQL area from where information in a database can be retrieved.

**Features of MySQL Cursor**

Below are the Features of MySQL Cursor:

* **Read-Only**: You cannot modify any data in the primary table using the cursor.
* **Non-Scrollable**: Rows can be fetched only in the order followed by the SELECT statement, not in the reversed direction and neither rows can be skipped nor jumped to a specific one in the result set.
* **Asensitive**: Cursors are of two kinds: asensitive and insensitive cursors. In an asensitive cursor, the actual data is pointed while in an insensitive cursor, a provisional copy of the data is used. Thus, MySQL cursor is asensitive.

#### 1. Implicit Cursors

* Auto-created, when SQL is executed if there is no explicit cursor used.
* Users or programmers cannot control the information or programs in it
* Associated with INSERT, UPDATE and DELETE types of DML operation statements
* Attributes: SQL%FOUND, SQL%NOTFOUND, %ISOPEN, %ROWCOUNT

#### 2. Explicit Cursors

* User-defined cursors which help to gain more control over the context part.
* It is defined in the declaration area of the SQL block.
* Created on SELECT statements that returns multiple records.
* Attributes: SQL%FOUND, SQL%NOTFOUND, %ISOPEN, %ROWCOUNT.

**Syntax and steps for an explicit cursor:**

CURSOR cursorname IS selectstatement;

* Declaring the cursor to define a result set
* Opening the cursor to establish the result set
* Fetching the cursor to retrieve the data into local variables
* Closing the cursor when done

### Cursor Actions in MYSQL

There are four actions performed by the Cursor in MySQL:

**1. DECLARE**: First, declare a cursor with a name associated with a SELECT statement.

DECLARE cursorname CURSOR FOR SELECTstatement;

**2. OPEN**: Next, open the cursor using the OPEN statement that initializes the result set and allocates memory for the cursor before the rows are fetched from the result set.

OPEN cursorname;

**3. FETCH**: Then, the FETCH statement is used to access one row at a time and move to the next row if available.

FETCH cursorname INTO variables\_list;

**4. CLOSE**: Finally, disable the cursor which releases the memory allocated.

CLOSE cursorname;

**Note:** When the FETCH statement is called, the next row is read in the result set each time. But a time comes when it reaches to the end of the set and no data is found there, so to handle this condition with MYSQL cursor we need to use a NOT FOUND handler.

**Syntax:**

DECLARE CONTINUE HANDLER FOR NOT FOUND SET variable\_name = 1;

Here, variable\_name indicates the end of the result set. Let us also discuss the Cursor Attributes now. They are described below:

* **%FOUND**: If the record is fetched successfully by the recent operation then, it returns ‘TRUE’ otherwise ‘FALSE’.
* **%NOTFOUND**: It returns ‘FALSE’ if the record is retrieved but ‘TRUE’ if not.
* **%ISOPEN**: If the cursor is already opened, it results in ‘TRUE’ else ‘FALSE’
* **%ROWCOUNT**: When DML operations are executed, in a result set it provides the count of rows.

Delimiter $$

create procedure createEmailList(INOUT emailList varchar(4000))

begin

declare flag int default 0;

declare emailAddress varchar(100) default"";

declare curEmail

cursor for

select MailID from sample1;

Declare continue handler

for Not found set flag=1;

open curEmail;

getEmail:Loop

Fetch curEmail into emailAddress;

if flag=1 then

Leave getEmail;

end if;

set emailList=concat(emailAddress,";",emailList);

End Loop getEmail;

close curEmail;

end $$

Delimiter ;

set @emailList="";

call createEmailList(@emailList);

select @emailList;