

MySQL Cursor

A cursor in database is a construct which allows you to iterate/traversal the records of a table. In MySQL you can use cursors with in a stored program such as procedures, functions etc.

In other words, you can iterate though the records of a table from a MySQL stored program using the cursors. The cursors provided by MySQL are embedded cursors. They are –

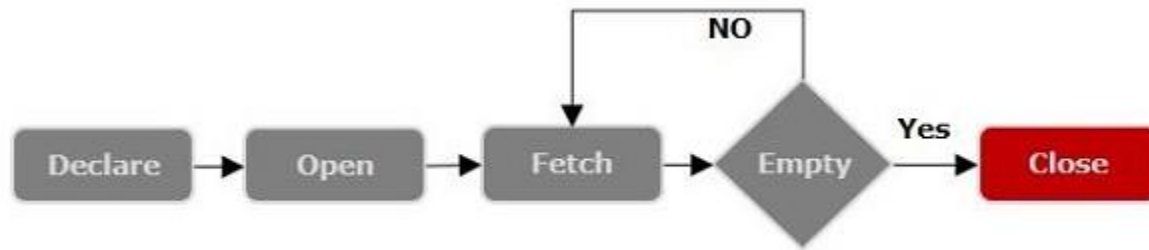
- **READ ONLY** – Using these cursors you cannot update any table.
- **Non-Scrollable** – Using these cursors you can retrieve records from a table in one direction i.e., from top to bottom.
- **Asensitive** – These cursors are insensitive to the changes that are made in the table i.e. the modifications done in the table are not reflected in the cursor.

Which means if we have created a cursor holding all the records in a table and, meanwhile if we add some more records to the table, these recent changes will not be reflected in the cursor we previously obtained.

While Declaring cursors in a stored program you need to make sure these (cursor declarations) always follow the variable and condition declarations.

To use a cursor, you need to follow the steps given below (in the same order)

- Declare the cursor using the *DECLARE* Statement.
- Declare variables and conditions.
- Open the declared cursor using the *OPEN* Statement.
- Retrieve the desired records from a table using the *FETCH* Statement.
- Finally close the cursor using the *CLOSE* statement.



The DECLARE Statement

Using the DECLARE statement you can declare a cursor and associate It with the SELECT statement which fetches the desired records from a table. This SELECT statement associated with a cursor does not allow INTO clause.

Once you declare a cursor you can retrieve records from it using the FETCH statement. You need to make sure the cursor declaration precedes handler declarations. You can create use cursors in a single stored program.

Syntax

Following is the syntax of the MySQL Cursor DECLARE Statement –

DECLARE cursor_name CURSOR FOR select_statement;

Example

Assume we have created a table with name tutorials in MySQL database using CREATE statement as shown below –

```
CREATE TABLE tutorials (  
  ID INT PRIMARY KEY,  
  TITLE VARCHAR(100),  
  AUTHOR VARCHAR(40),  
  DATE VARCHAR(40)  
);
```

Now, we will insert 5 records in tutorials table using INSERT statements

—

```
insert into tutorials values(1, 'Java', 'Krishna', '2019-09-01');
insert into tutorials values(2, 'JFreeCharts', 'Satish', '2019-05-01');
insert into tutorials values(3, 'JavaSprings', 'Amit', '2019-05-01');
insert into tutorials values(4, 'Android', 'Ram', '2019-03-01');
insert into tutorials values(5, 'Cassandra', 'Pruthvi', '2019-04-06');
```

Let us create another table to back up the data —

```
CREATE TABLE backup (
  ID INT,
  TITLE VARCHAR(100),
  AUTHOR VARCHAR(40),
  DATE VARCHAR(40)
);
```

Following procedure backups the contents of the tutorials table to the backup table using cursors —

```
DELIMITER //
CREATE PROCEDURE ExampleProc()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE tutorialID INTEGER;
  DECLARE tutorialTitle, tutorialAuthor, tutorialDate
  VARCHAR(20);
  DECLARE cur CURSOR FOR SELECT * FROM tutorials;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done
  = 1;
  OPEN cur;
  label: LOOP
    FETCH cur INTO tutorialID, tutorialTitle, tutorialAuthor,
    tutorialDate;
    INSERT INTO backup VALUES(tutorialID, tutorialTitle,
    tutorialAuthor, tutorialDate);
```

```
IF done = 1 THEN LEAVE label;
END IF;
END LOOP;
CLOSE cur;
END//
DELIMITER ;
```

You can call the above procedure as shown below –

```
mysql> CALL ExampleProc;
Query OK, 1 row affected (0.78 sec)
```

If you verify the contents of the backup table you can see the inserted records as shown below –

```
mysql> select * from backup;
+-----+-----+-----+-----+
| ID | TITLE      | AUTHOR | DATE       |
+-----+-----+-----+-----+
| 1 | Java       | Krishna | 2019-09-01 |
| 2 | JFreeCharts | Satish | 2019-05-01 |
| 3 | JavaSprings | Amit   | 2019-05-01 |
| 4 | Android    | Ram    | 2019-03-01 |
| 5 | Cassandra  | Pruthvi | 2019-04-06 |
+-----+-----+-----+-----+
5 rows in set (0.08 sec)
```