MySQL: Cursors

Tushar B. Kute,
http://tusharkute.com

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.

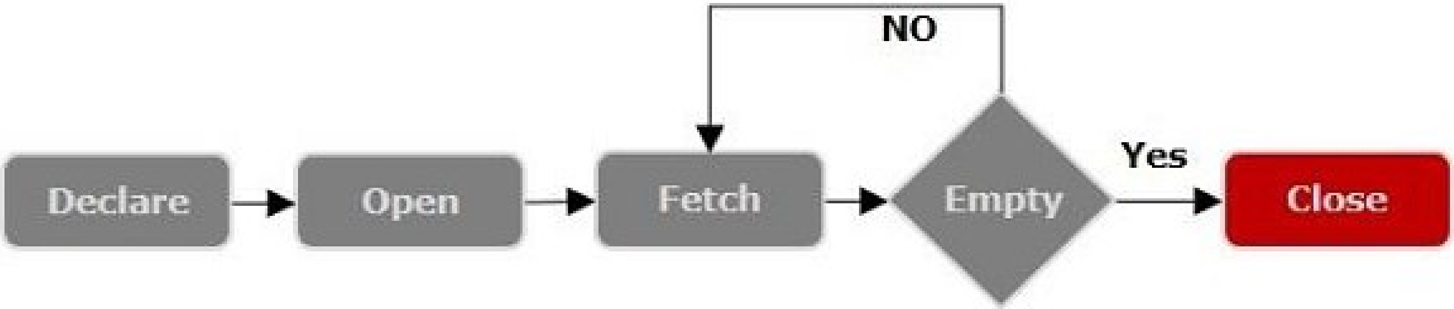
Cursor

- The cursors provided by MySQL are embedded cursors.
  - 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.

Cursor

- 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.

Cursor



Declare Cursor

- The stored function is almost similar to the procedure in MySQL, but it has some differences that are as follows:
  - The function parameter may contain only the IN parameter but can't allow specifying this parameter, while the procedure can allow IN, OUT, INOUT parameters.
  - The stored function can return only a single value defined in the function header.
  - The stored function may also be called within SQL statements.
  - It may not produce a result set.

Declare Cursor

- 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:
  - DECLARE cursor_name CURSOR FOR select_statement;

Open Cursor

- After declaring the cursor the next step is to open the cursor using open statement.
- Syntax:
  - open cursor_name;
- Parameter:
  - cursor_name: name of the cursor which is already declared.

Fetch Cursor

- After declaring and opening the cursor, the next step is to fetch the cursor. It is used to fetch the row or the column.
- Syntax:
  - FETCH [NEXT [FROM]] cursor_name INTO variable_list;
- Parameter:
  - cursor_name: name of the cursor
  - variable_list: variables, comma separated, etc. is stored in a cursor for the result set

Close Cursor

- The final step is to close the cursor.
- Syntax:
  - close cursor_name;
- Parameter:
  - cursor_name: name of the cursor

Example: Create a table

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

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

Example: Insert some values

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

Example: Create table for backup

- Let us create another table to back up the data –

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

Create cursor function

```
DELIMITER &&
CREATE PROCEDURE ExampleProc()
  BEGIN
    DECLARE done INT DEFAULT 0;
    DECLARE lectureID INTEGER;
    DECLARE lectureTitle, lectureAuthor, lectureDate VARCHAR(20);
    DECLARE cur CURSOR FOR SELECT * FROM lecture;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    OPEN cur;
    label: LOOP
    FETCH cur INTO lectureID, lectureTitle, lectureAuthor, lectureDate;
    INSERT INTO backup VALUES(lectureID, lectureTitle, lectureAuthor, lectureDate);
    IF done = 1 THEN LEAVE label;
    END IF;
    END LOOP;
    CLOSE cur;
  END&&
DELIMITER ;
```

Check the output

- You can call the above procedure as shown below –

```
mysql> CALL ExampleProc;
```

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

```
mysql> select * from backup;
```

Thank you

@mitu_skillologies @mITuSkillologies @mitu_group @mitu-skillologies @MITUSkillologies

kaggle

@mituskillologies

Web Resources

https://mitu.co.in

http://tusharkute.com

@mituskillologies

contact@mitu.co.in

tushar@tusharkute.com