

Practical – 9 Cursor Handling

Cursor

Implicit Cursor

- Whenever DML operations such as INSERT, UPDATE, and DELETE are processed in the database, implicit cursors are generated automatically and used

Explicit Cursor

To handle a result set inside a [stored procedure](#), you use a cursor. A cursor allows you to [iterate](#) a set of rows returned by a query and process each row individually.

- **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.

Step 1 Declare Cursor

DECLARE cursor_name **CURSOR FOR**
Select statement;

The cursor declaration must come after any [variable](#) declaration. If you declare a cursor before the variable declarations, MySQL will issue an error. Additionally, a cursor must always associate with a SELECT statement.

2. Open Cursor

After declaring the cursor the next step is to open the cursor using open statement.

Open cursor_name;

3. Fetch Cursor

FETCH [NEXT [FROM]] cursor_name **INTO** variable_list;

4. Close Cursor

The final step is to close the cursor.

1. **Close** cursor_name;

- Declare handler

The [DECLARE ... HANDLER](#) statement specifies a handler that deals with one or more conditions. If one of these conditions occurs, the specified *statement* executes. *statement* can be a simple statement such as SET *var_name* = *value*.

if there is no data found at the end of the set, you must use a NOT FOUND handler to handle this condition with the MySQL cursor.

-
- Continue : Execution of the current program continues.

e.g DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
The done is a variable to indicate that the cursor has reached the end of the result set

When working with MySQL cursor, you must also declare a NOT FOUND handler to manage the situation when the cursor cannot find any row.

E.g Create a procedure which show the author id and authername of book passed into arguments

step 1 :DELIMITER //

Step 2:

```
CREATE PROCEDURE proccursor(in bid int)
```

```
BEGIN
```

```
    DECLARE done INT DEFAULT 0;
```

```
    DECLARE aid int;
```

```
    DECLARE aname VARCHAR(20);
```

```
    DECLARE cur CURSOR FOR select author3.authid ,authname from
                             author3,auth_book where author3.authid=auth_book.authid and
                             auth_book.bookid=bid;
```

```
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
```

```
    OPEN cur;
```

```
label: LOOP
```

```
    FETCH cur INTO aid,aname;
```

```
        IF done = 1 THEN
```

```
            LEAVE label;
```

```
        END IF;
```

```
    select aid , aname;
```

```
    END LOOP;
```

```
    CLOSE cur;
```

```
END//
```

Step 3 ; DELIMITER ;

Step 4 :

```
call proccursor(3);
```

```
+-----+-----+
```

```
| aid | aname |
```

```
+-----+-----+
```

```
| 1 | aaa |
```

```
+-----+-----+
```

```
1 row in set (0.09 sec)
```

```
+-----+-----+
```

```
| aid | aname |
```

```
+-----+-----+
```

```
| 2 | bbb |
```

+-----+-----+

Exercise

1	Create procedure using cursor which calculate bonus 10% of all employee which have salary greater than 2000.
2	Create proceddure to calculate total marks of given subject id. Hint :Use cursor, use stud_sub for calculating total marks
3	Create procedure to update the salary increase salary by 300 of all given department into the procedure parameter Use cursor
4	create procedure which pass the orderid as parameter and find the total quantity order form sales_order_detail, total of quantity order should be stored in OUT parameter. Use cursor
5	Create procedure which calculate total price of order number 019001. Hint : use any technique of subquery/nested query for creating cursor. Product_master and sales_order_detail