Valentine
PAGE NO.:
DATE:

Unit - 3 working with Cursor

Cursor :-

- A cursor allows you to iterate a set of rows returned by a query and process each row accordingly.
- A cursor is an object that provides programmatic access to the result set returned by your SELECT statement.

Declaration.

syntax

Declare cursor_name Cursor for
select_statement;

- Declaring a cursor before declaring our variables generates error.

ex

Declare S1 Cursor for select customer_name, conduct_surname, contact_firstname from Customers;

Valentine

PAGE NO.:

DATE:

Cursor statements:

- Open.
- FETCH
- CLOSE

- **Open**

- Initializes the result set for the cursor. We must open a cursor before fetching any rows from that cursor.

Syntax:

```
Open cursor_name;
```

ex.

```
Open s1;
```

- **Close**

- Deactivates the cursor and releases the memory associated with that Cursor.

Syntax

```
Close cursor_name;
```

ex

```
Close s1;
```

- **Fetch :**

- Retrieves the next row from the cursor and moves the cursor "pointer" to the following row in the result set.

- The variable list must contain one of a compatible data type for each column return by select statement.

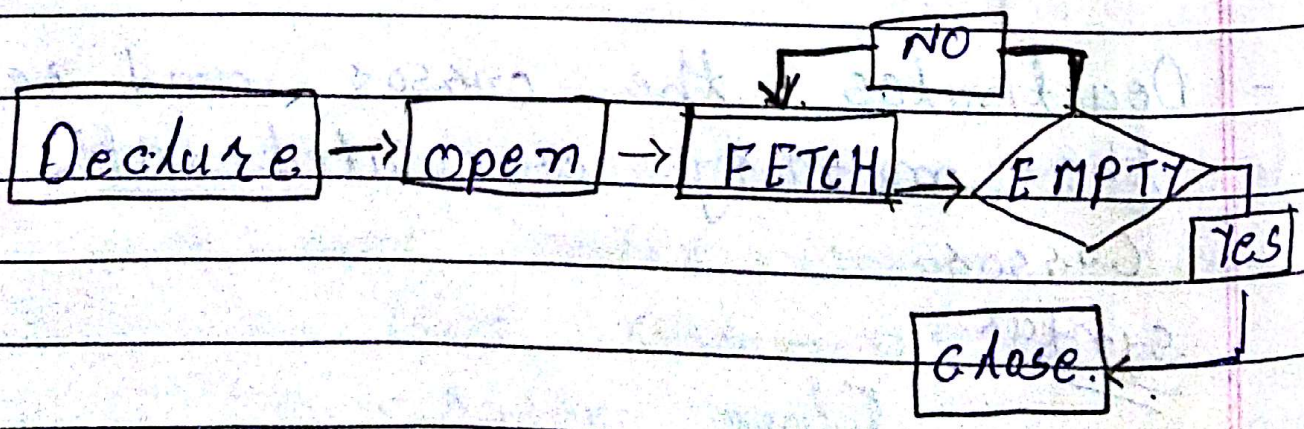
Syntax:

Fetch cursor_name INTO variable list

ex.

Fetch C1 id into a,b,c;

→ **Working of cursor :**



Example

DATE:

```
Create pro---- getD().
begin
declare s,p int;
dell n varchar(20);

declare c cursor for select name,
salary from emp;
declare exit handler for not
found set p=1;
set p=0;
open c;
repeat
  fetch c into n,s;
  select s,n;
until (p=1)
end repeat;
close c;
end //
```

DROP PROCEDURE If Exists. Procedure - Mar

Valentine

PAGE NO.:

DATE:

Q// Write a procedure witch gives bonus to all the empoly based on Condistion.

salary between 10 - 20 1000 bonus
" " 20 - 30 2000 "
30 - 40 3000 "
40 - 50 4000 "

⇒ Create procedure getDI()
begin
declare s, r, i int;

declare b cursor for select id, salary from emp;
declare Exit handler for not found
set r=1;
set r=0;
open b;
repeat
Fetch b into i, s;
if (s>10000 && s<20000) then
set s = s+1000;
else if (s>20000 && s<30000) then
set s = s+2000;

a<>1 = ≠

```
elseif (s>30000 && s<40000) then
    set s = s+3000;
elseif (s>40000 && s<50000) then
    set s = s+4000;
end if;

Update emp set salary = s, where id = i;
Until (p=1)
end repeat;
Close b;
end //
```

Properties of Cursor :-

- **Asensitive :-**

  Cursor can not create copy of its result table. In MySQL cursor just traverse on every row of the table or result set.

- **Read Only :-**

  - Not updatable.

- Non-scrollable:

  - Only fetch rows in the order determined by the select statem
  - Can not fetch rows in the reversed order.
  - Can not skip rows.

⇒ Advantages of using Mysql Cursor:

- No need to write business logic after fetching data from your code logic.

- Gives you better flexibility for operating even on the single column of the row. Manipulation become always easy.

- Saves you from the simplex join structure.

- Easy to maintain the business logic at your application at one place.

DATE:

★ Disadvantages of using Mysql Cursor

- slow down your stored procedure or function performance in cause of large record set in cursor.
- Debugging of your business logic become tough.
- Hard to manage.
- Need to consider locking of the database.

Ex

```
Create procedure xyz()
begin
declare i,sal,a int;
declare n,city,desig,email,cno varchar(20);
declare c cursor for select * from emp;
declare continue handler for not
found set a=1;
set a=0;
open c;
```