PB_05_Kushagra Suryawanshi Batch B1.

DBMS : Cursors and Triggers. (LAB 06)

Aim: Write PLSQL Cursor and triggers for given problem statements

Objective: To study PLSQL Cursors and Triggers.

Theory:

PLSQL Triggers : A trigger is a statement that is executed automatically by the system as a side effect of a modification to the database i.e when changes are made to a table. A trigger can be defined to be invoked either before or after the data is changed by INSERT, UPDATE, DELETE.

Syntax : create trigger name time event.
on table name
for each row
Begin
...
END

PLSQL Cursor : A cursor allows us to iterate a set of rows returned by a query and process each row accordingly

Syntax: To declare cursor - DECLARE cursor_name Cursor
FOR SELECT statement;

To open cursor - OPEN cursor_name
To fetch data pointed by cursor in a variable
Fetch name INTO variables list;
To close the cursor : CLOSE cursor_name ;

INPUT : Database
OUTPUT : Data as per request
PLATFORM : My SQL

Conclusion: Thus, we have learned PLSQL database programming

FAQs.

A1. Application of triggers :
1. Generating some derived column values automatically.
2. Enforcing referential integrity.
3. Event logging and storing information on table access.
4. Auditing
5. Synchronous replication of tables.
6. Imposing security authorizations
7. Preventing invalid transactions.

A2. The major function of a cursor is to retrive data, one row at a time, from a result set, unlike the SQL commands which operate on all the rows in the result set at one time. The help update records row by row.

PB_05_Kushagra Suryawanshi

DBMS LAB 6

Enter password: **********

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 10

Server version: 8.0.21 MySQL Community Server - GPL

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> create database LAB6;
Query OK, 1 row affected (0.34 sec)

mysql> use LAB6;
Database changed
mysql> CREATE TABLE student_fees_detail(name VARCHAR(63), total_fees_deposited DECIMAL(10,
2), till_date DATE);
Query OK, 0 rows affected (2.69 sec)

mysql> SELECT name, total_fees_depositedFROM student_fees_detailWHERE name LIKE 'aj_%';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to
your MySQL server version for the right syntax to use near 'name LIKE 'aj_%'' at line 1
mysql> insert into student_fees_detail value('ajay', 100000.00, '2021-07-07');
Query OK, 1 row affected (0.32 sec)
```

```
mysql> insert into student_fees_detail value('ajinkya', 120000.00, '2021-08-07');
Query OK, 1 row affected (0.18 sec)

mysql> insert into student_fees_detail value('kush', 120000.00, '2021-07-15');
Query OK, 1 row affected (0.25 sec)

mysql> SELECT name, total_fees_deposited FROM student_fees_detail WHERE name LIKE 'aj_%';
+---------+---------------------+
| name    | total_fees_deposited |
+---------+---------------------+
| ajay    |          100000.00 |
| ajinkya |          120000.00 |
+---------+---------------------+
2 rows in set (0.00 sec)

mysql> SELECT name, SUM(total_fees_deposited) FROM student_fees_detail WHERE name LIKE
'aj_%' GROUP BY name;
+---------+--------------------------+
| name    | SUM(total_fees_deposited) |
+---------+--------------------------+
| ajay    |              100000.00 |
| ajinkya |              120000.00 |
+---------+--------------------------+
2 rows in set (0.01 sec)

mysql> CREATE TABLE demo(stud_fees DECIMAL(10, 2));
Query OK, 0 rows affected (1.91 sec)

mysql> CREATE TRIGGER fee
    -> AFTER UPDATE
    -> ON student_fees_detail
```

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO demo VALUES (OLD.total_fees_deposited); END;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 6

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'END' at line 1

mysql> delimiter //

mysql> CREATE TRIGGER fee

-> AFTER UPDATE

-> ON student_fees_detail

-> FOR EACH ROW

-> BEGIN

-> INSERT INTO demo VALUES (OLD.total_fees_deposited);

-> END;

-> //

Query OK, 0 rows affected (0.18 sec)

mysql> CREATE TABLE employee(emp_id BIGINT PRIMARY KEY, emp_name VARCHAR(63), age INT, address VARCHAR(127), salary DECIMAL(10, 2))//

Query OK, 0 rows affected (2.42 sec)

mysql> INSERT INTO employeeVALUES (55, 'ABC', 23, 'gyhjgd', 45000)//

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '55, 'ABC', 23, 'gyhjgd', 45000)' at line 1

mysql> INSERT INTO employee VALUES(55, 'ABC', 23, 'gyhjgd', 45000)//

Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO employee VALUES (56, 'LMN', 54, 'sffrd', 48000);

-> //

Query OK, 1 row affected (0.15 sec)

```
mysql> INSERT INTO employee VALUES (57, 'DEF', 41, 'sdv', 55000);
   -> //
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO employee VALUES (58, 'GHI', 65, 'sdcfcssf', 35000);
   -> //
Query OK, 1 row affected (0.14 sec)

mysql> INSERT INTO employee VALUES (59, 'JKL', 53, 'dyua', 25000);
   -> //
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO employee VALUES (60, 'XYZ', 57, 'addr', 25000);
   -> //
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO employee VALUES (61, 'NEW', 35, 'addrr', 20000);
   -> //
Query OK, 1 row affected (0.12 sec)

mysql> CREATE PROCEDURE emp()
   -> BEGIN
   ->     DECLARE done INT DEFAULT 0;
   ->
   ->     DECLARE _emp_id BIGINT;
   ->     DECLARE _emp_name VARCHAR(63);
   ->     DECLARE _age INT;
   ->     DECLARE _address VARCHAR(127);
   ->     DECLARE _salary DECIMAL(10, 2);
   ->
   ->     DECLARE cur CURSOR FOR SELECT * FROM employee;
```

```
    ->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->     OPEN cur;
    ->
    ->     lbl :
    ->     LOOP
    ->         FETCH cur INTO _emp_id, _emp_name, _age, _address, _salary;
    ->
    ->         IF done = 1 THEN
    ->             LEAVE lbl;
    ->         END IF;
    ->
    ->         IF _age > 50 THEN
    ->             UPDATE employee SET salary = salary + 5000 WHERE emp_id = _emp_id;
    ->         END IF;
    ->
    ->     END LOOP;
    ->     CLOSE cur;
    -> END;
    -> //
Query OK, 0 rows affected (0.26 sec)

mysql> CALL emp()//
Query OK, 0 rows affected (0.42 sec)

mysql> select * from employee//
+--------+----------+------+----------+----------+
| emp_id | emp_name | age  | address  | salary   |
+--------+----------+------+----------+----------+
|     55 | ABC      |   23 | gyhjgd   | 45000.00 |
|     56 | LMN      |   54 | sffrd    | 53000.00 |
```

```
|    57 | DEF     |   41 | sdv      | 55000.00 |
|    58 | GHI     |   65 | sdcfcssf | 40000.00 |
|    59 | JKL     |   53 | dyua     | 30000.00 |
|    60 | XYZ     |   57 | addr     | 30000.00 |
|    61 | NEW     |   35 | addrr    | 20000.00 |
+--------+----------+------+----------+----------+
7 rows in set (0.00 sec)

mysql> select * from DEMO//
Empty set (0.11 sec)

mysql> select * from student_fees_detail//
+---------+---------------------+------------+
| name    | total_fees_deposited | till_date  |
+---------+---------------------+------------+
| ajay    |           100000.00 | 2021-07-07 |
| ajinkya |           120000.00 | 2021-08-07 |
| kush    |           120000.00 | 2021-07-15 |
+---------+---------------------+------------+
3 rows in set (0.07 sec)

mysql> insert into table student_fees_detail value('ben', 50000.00, '2021-08-12')//
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'table student_fees_detail value('ben', 50000.00, '2021-08-12')' at line 1
mysql> insert into table student_fees_detail values('ben', 50000.00, '2021-08-12')//
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'table student_fees_detail values('ben', 50000.00, '2021-08-12')' at line 1
mysql> insert into student_fees_detail values('ben', 50000.00, '2021-08-12')//
Query OK, 1 row affected (0.10 sec)
```

```
mysql> select * from student_fees_detail//
+---------+----------------------+------------+
| name    | total_fees_deposited | till_date  |
+---------+----------------------+------------+
| ajay    |          100000.00 | 2021-07-07 |
| ajinkya |          120000.00 | 2021-08-07 |
| kush    |          120000.00 | 2021-07-15 |
| ben     |           50000.00 | 2021-08-12 |
+---------+----------------------+------------+
4 rows in set (0.00 sec)

mysql> select * from DEMO//
Empty set (0.00 sec)

mysql> desc DEMO//
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| stud_fees | decimal(10,2) | YES  |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
1 row in set (0.08 sec)

mysql> update student_fees_detail set total_fees_deposited = total_fees_deposited - 20000.00
where name = 'kush'//
Query OK, 1 row affected (0.12 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from DEMO//
+-----------+
| stud_fees |
+-----------+
```

| 120000.00 |

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

1 row in set (0.00 sec)

mysql> select * from student_fees_detail//

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

| name | total_fees_deposited | till_date |

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

| ajay | 100000.00 | 2021-07-07 |

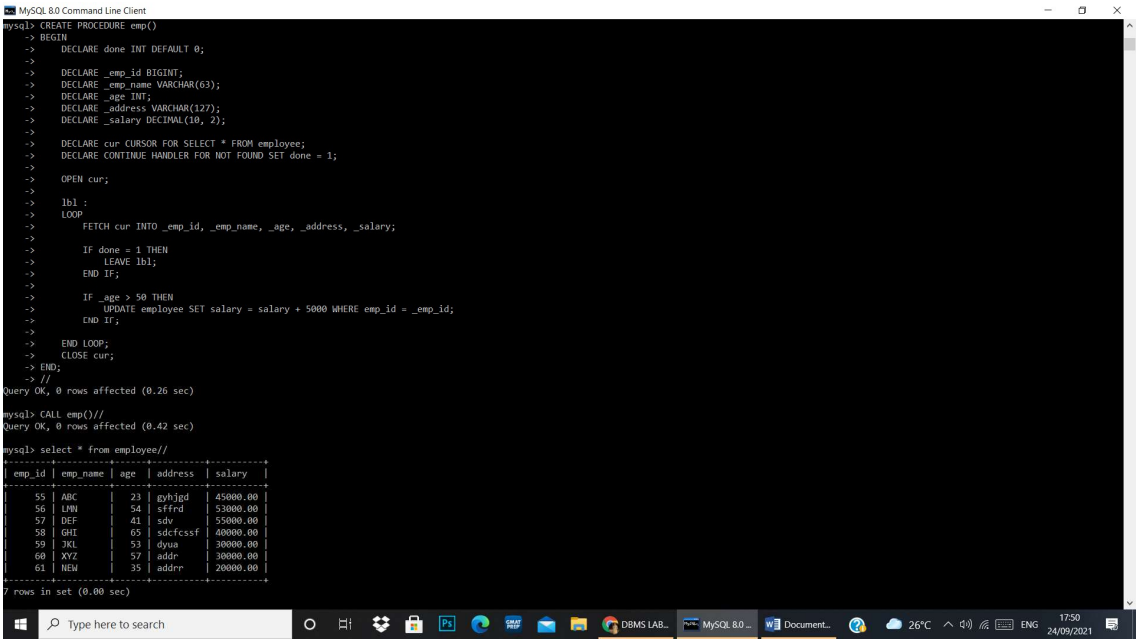
| ajinkya | 120000.00 | 2021-08-07 |

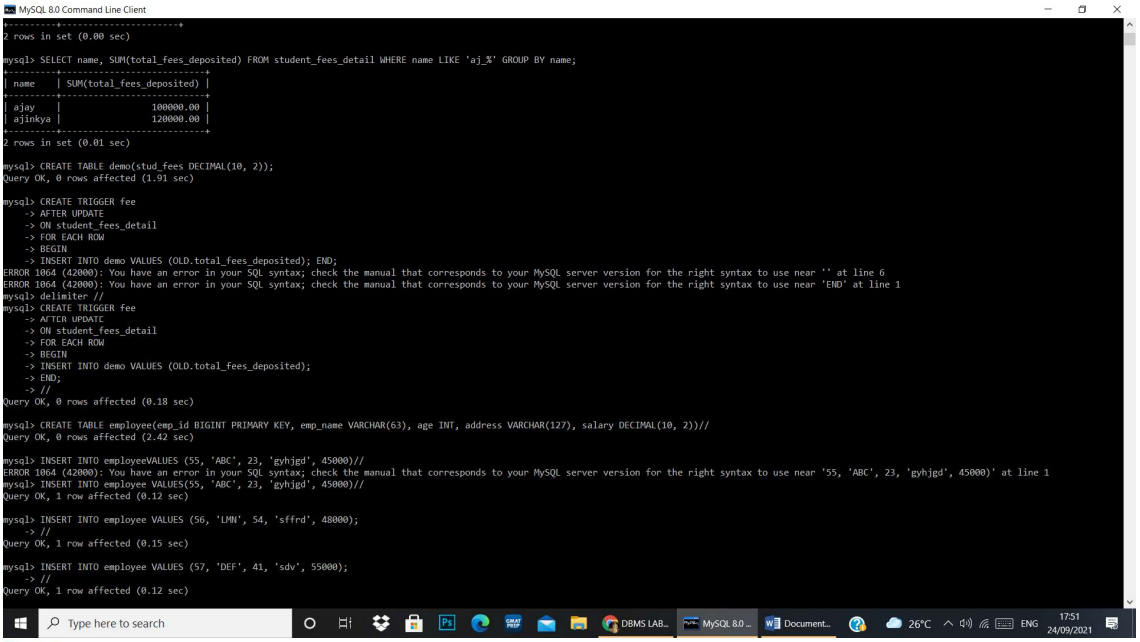
| kush | 100000.00 | 2021-07-15 |

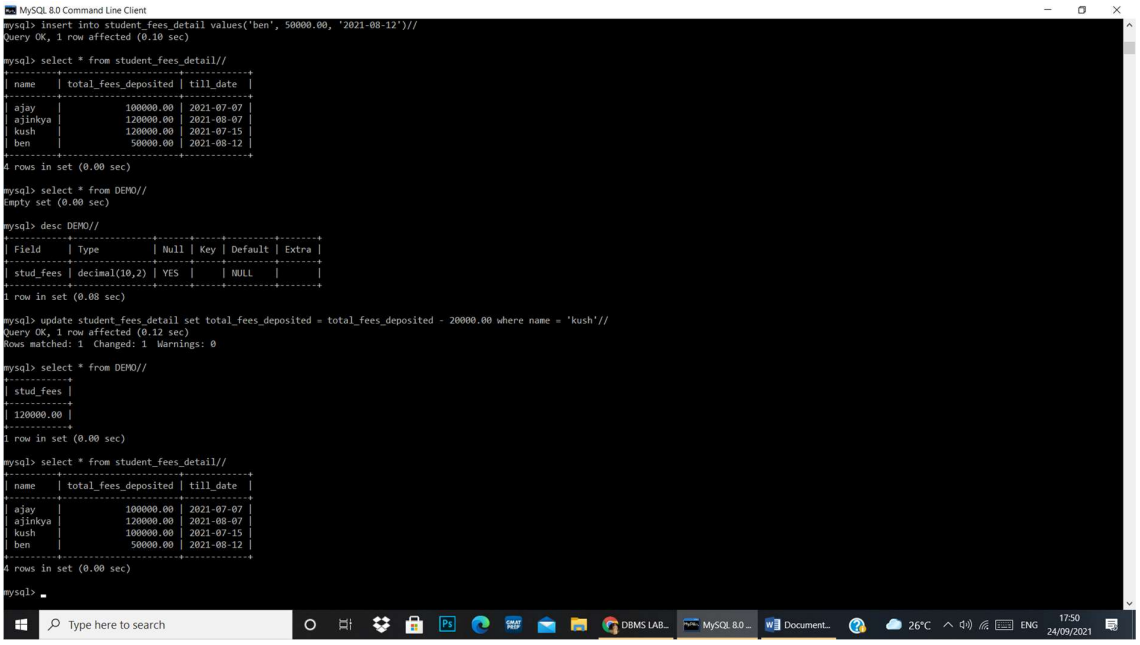
| ben | 50000.00 | 2021-08-12 |

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

4 rows in set (0.00 sec)




MySQL 8.0 Command Line Client
2 rows in set (0.00 sec)
mysql> SELECT name, SUM(total_fees_deposited) FROM student_fees_detail WHERE name LIKE 'aj_%' GROUP BY name;
name	SUM(total_fees_deposited)
ajay	100000.00
ajinkya	120000.00
2 rows in set (0.01 sec)
mysql> CREATE TABLE demo(stud_fees DECIMAL(10, 2));
Query OK, 0 rows affected (1.91 sec)
mysql> CREATE TRIGGER fee
-> AFTER UPDATE
-> ON student_fees_detail
-> FOR EACH ROW
-> BEGIN
-> INSERT INTO demo VALUES (OLD.total_fees_deposited); END;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 6
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'END' at line 1
mysql> delimiter //
mysql> CREATE TRIGGER fee
-> AFTER UPDATE
-> ON student_fees_detail
-> FOR EACH ROW
-> BEGIN
-> INSERT INTO demo VALUES (OLD.total_fees_deposited);
-> END;
-> //
Query OK, 0 rows affected (0.18 sec)
mysql> CREATE TABLE employee(emp_id BIGINT PRIMARY KEY, emp_name VARCHAR(63), age INT, address VARCHAR(127), salary DECIMAL(10, 2))//
Query OK, 0 rows affected (2.42 sec)
mysql> INSERT INTO employeeVALUES (55, 'ABC', 23, 'gyhjgd', 45000)//
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '55, 'ABC', 23, 'gyhjgd', 45000)' at line 1
mysql> INSERT INTO employee VALUES(55, 'ABC', 23, 'gyhjgd', 45000)//
Query OK, 1 row affected (0.12 sec)
mysql> INSERT INTO employee VALUES (56, 'LMN', 54, 'sffrd', 48000);
-> //
Query OK, 1 row affected (0.15 sec)
mysql> INSERT INTO employee VALUES (57, 'DEF', 41, 'sdv', 55000);
-> //
Query OK, 1 row affected (0.12 sec)
Type here to search
DBMS LAB...
MySQL 8.0 ...
Document...
26°C
ENG
17:51
24/09/2021


MySQL 8.0 Command Line Client
mysql> insert into student_fees_detail values('ben', 50000.00, '2021-08-12')//
Query OK, 1 row affected (0.10 sec)
mysql> select * from student_fees_detail//
name	total_fees_deposited	till_date
ajay	100000.00	2021-07-07
ajinkya	120000.00	2021-08-07
kush	120000.00	2021-07-15
ben	50000.00	2021-08-12
4 rows in set (0.00 sec)
mysql> select * from DEMO//
Empty set (0.00 sec)
mysql> desc DEMO//
Field	Type	Null	Key	Default	Extra
stud_fees	decimal(10,2)	YES		NULL	
1 row in set (0.08 sec)
mysql> update student_fees_detail set total_fees_deposited = total_fees_deposited - 20000.00 where name = 'kush'//
Query OK, 1 row affected (0.12 sec)
Rows matched: 1 Changed: 1 Warnings: 0
mysql> select * from DEMO//
stud_fees
120000.00
1 row in set (0.00 sec)
mysql> select * from student_fees_detail//
name	total_fees_deposited	till_date
ajay	100000.00	2021-07-07
ajinkya	120000.00	2021-08-07
kush	100000.00	2021-07-15
ben	50000.00	2021-08-12
4 rows in set (0.00 sec)
mysql>
Type here to search
DBMS LAB...
MySQL 8.0 ...
Document...
26°C
ENG
17:50
24/09/2021