Name – Aditya Somani Roll No. – T1851061 PRN No. : 71901204L

Assignment No. 8

Aim: Write and execute suitable database triggers .Consider row level and statement level triggers.

Objective:

- To study and implement PL/SQLtriggers.

Theory :

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events.

- A **database manipulation (DML)** statement (DELETE, INSERT, orUPDATE)
- A **database definition (DDL)** statement (CREATE, ALTER, orDROP).
- A**database operation**(SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

Triggers can be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers

Triggers can be written for the following purposes −

- Generating some derived column valuesautomatically
- Enforcing referentialintegrity
- Event logging and storing information on tableaccess
- Auditing
- Synchronous replication oftables
- Imposing securityauthorizations
- Preventing invalidtransactions

Creating Triggers

The syntax for creating a trigger is −

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
```

```
[FOR EACH ROW]
WHEN (condition)
DECLARE
  Declaration-statements
BEGIN
  Executable-statements
EXCEPTION
  Exception-handling-statements
END;
```

Where,

- CREATE [OR REPLACE] TRIGGER trigger_name − Creates or replaces an existing trigger with the*trigger_name*.
- {BEFORE | AFTER | INSTEAD OF} − This specifies when the trigger will be executed. The INSTEAD OF clause is used for creating trigger on aview.
- {INSERT [OR] | UPDATE [OR] | DELETE} − This specifies the DMLoperation.
- [OF col_name] − This specifies the column name that will beupdated.
- [ON table_name] − This specifies the name of the table associated with thetrigger.
- [REFERENCING OLD AS o NEW AS n] − This allows you to refer new and old values for various DML statements, such as INSERT, UPDATE, andDELETE.
- [FOR EACH ROW] − This specifies a row-level trigger, i.e., the trigger will be executed for each row being affected. Otherwise the trigger will execute just once when the SQL statement is executed, which is called a table leveltrigger.
- WHEN (condition) − This provides a condition for rows for which the trigger would fire. This clause is valid only for row-leveltriggers.

Conclusion:-

We have studied and executed different types of database triggers.

OUTPUT –

1) Create Trigger

```
mysql> DELIMITER $
mysql> CREATE TRIGGER UPDATEADSTATUS BEFORE INSERT ON EMPLOYEE
 -> FOR EACH ROW
 -> BEGIN
 -> IF NEW.AGE>=18 THEN SET NEW.ADULT_STATUS=1;
 -> ELSE SET NEW.ADULT_STATUS=0;
 -> END IF;
 -> END;
 -> $
Query OK, 0 rows affected (0.06 sec)
mysql> INSERT INTO EMPLOYEE VALUES(1, 12, 90)$
Query OK, 1 row affected (0.04 sec)
mysql> SELECT * FROM EMPLOYEE$
+----+------+--------------+
| ID | AGE | ADULT_STATUS |
+----+------+--------------+
| 1 | 12 | 0 |
+----+------+--------------+
1 row in set (0.00 sec)
mysql> INSERT INTO EMPLOYEE(ID, AGE) VALUES(2, 19)$
Query OK, 1 row affected (0.04 sec)
mysql> SELECT * FROM EMPLOYEE$
+----+------+--------------+
| ID | AGE | ADULT_STATUS |
+----+------+--------------+
| 1 | 12 | 0 |
| 2 | 19 | 1 |
+----+------+--------------+
2 rows in set (0.00 sec)
```

2) Trigger on update

```
mysql> create trigger t before update on employee1 FOR EACH ROW BEGIN INSERT INTO
employees_audit SET action = 'update', employeeNumber = OLD.empno, firstname =
OLD.empname,
changedat = NOW(); END$
Query OK, 0 rows affected (0.07 sec)
mysql> SELECT * FROM employee1$
+-------+---------+----------------+-------+------------+--------+-------+
| empno | empname | job | mngid | hiredate | salary | depno |
+-------+---------+----------------+-------+------------+--------+-------+
| 1 | ashish | assitant | 23 | 2016-03-03 | 20000 | 10 |
| 2 | sachin | assitant | 22 | 2015-03-03 | 25000 | 10 |
| 3 | rahul | developer | 24 | 2014-04-03 | 26000 | 20 |
```

```
| 4 | sankalp | project head | 21 | 2013-04-03 | 36000 | 30 |
| 5 | krishna | manging partner | 25 | 2011-04-03 | 66000 | 60 |
+-------+---------+----------------+-------+------------+--------+-------+
5 rows in set (0.00 sec)
mysql> update employee1 set empname='manjeet' where empno=1;
 -> $
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT * FROM employee1$
+-------+---------+----------------+-------+------------+--------+-------+
| empno | empname | job | mngid | hiredate | salary | depno |
+-------+---------+----------------+-------+------------+--------+-------+
| 1 | manjeet | assitant | 23 | 2016-03-03 | 20000 | 10 |
| 2 | sachin | assitant | 22 | 2015-03-03 | 25000 | 10 |
| 3 | rahul | developer | 24 | 2014-04-03 | 26000 | 20 |
| 4 | sankalp | project head | 21 | 2013-04-03 | 36000 | 30 |
| 5 | krishna | manging partner | 25 | 2011-04-03 | 66000 | 60 |
+-------+---------+----------------+-------+------------+--------+-------+
5 rows in set (0.00 sec)
mysql> select * from employees_audit$
+----+---------------+-----------+---------------------+--------+
| id | employeeNumber | firstname | changedat | action |
+----+---------------+-----------+---------------------+--------+
| 1 | 1 | ashish | 2019-08-27 12:26:44 | update |
+----+---------------+-----------+---------------------+--------+
1 row in set (0.00 sec
```

3) After insert on table

```
mysql> CREATE TABLE ITEM(ITEM_ID INT PRIMARY KEY, ITEM_DESCRIPTION
VARCHAR(20), QOH INT,
PRICE FLOAT, CATEGORY VARCHAR(20))$
Query OK, 0 rows affected (0.19 sec)
mysql> CREATE TABLE SALES(SID INT PRIMARY KEY, ITEM_ID INT, Q_SOLD INT,
PRICE FLOAT, TOTAL
INT)$
Query OK, 0 rows affected (0.19 sec)
mysql> insert into ITEM values(2,'batman',100,100,'toy')$
Query OK, 1 row affected (0.04 sec)
mysql> insert into ITEM values(3,'superman',100,100,'toy');$
Query OK, 1 row affected (0.04 sec)
mysql> insert into ITEM values(4,'rice',100,100,'food');$
Query OK, 1 row affected (0.04 sec)
mysql> insert into ITEM values(5,'dettol',100,100,'health');$
Query OK, 1 row affected (0.04 sec)
mysql> SELECT * FROM ITEM$
+---------+-----------------+------+-------+----------+
| ITEM_ID | ITEM_DESCRIPTION | QOH | PRICE | CATEGORY |
+---------+-----------------+------+-------+----------+
| 1 | car | 100 | 100 | toy |
```

```
| 2 | batman | 100 | 100 | toy |
| 3 | superman | 100 | 100 | toy |
| 4 | rice | 100 | 100 | food |
| 5 | dettol | 100 | 100 | health |
+---------+-----------------+------+-------+----------+
5 rows in set (0.00 sec)
mysql> CREATE TRIGGER ITEMTRIG AFTER INSERT ON SALES FOR EACH ROW
BEGIN UPDATE ITEM SET QOH
= QOH - NEW.Q_SOLD; END;$
Query OK, 0 rows affected (0.06 sec)
mysql> CREATE TRIGGER ITEMTRIG AFTER INSERT ON SALES FOR EACH ROW
BEGIN UPDATE ITEM SET QOH
= QOH - NEW.Q_SOLD WHERE ITEM_ID = NEW.ITEM_ID; END;$
Query OK, 0 rows affected (0.06 sec)
mysql> INSERT INTO SALES VALUES(2, 1, 10, 100, 100)$
Query OK, 1 row affected (0.06 sec)
mysql> SELECT * FROM ITEM$
+---------+-----------------+------+-------+----------+
| ITEM_ID | ITEM_DESCRIPTION | QOH | PRICE | CATEGORY |
+---------+-----------------+------+-------+----------+
| 1 | car | 80 | 100 | toy |
| 2 | batman | 90 | 100 | toy |
| 3 | superman | 90 | 100 | toy |
| 4 | rice | 90 | 100 | food |
| 5 | dettol | 90 | 100 | health |
+---------+-----------------+------+-------+----------+
5 rows in set (0.00 sec)
```

4) After insert on table

```
mysql> create table product(prod_id int primary key,price int,quantity int, total_cost int)$
Query OK, 0 rows affected (0.18 sec)
mysql> create trigger t2 before insert on product for each row begin set
new.total_cost=new.price*new.quantity; end;$
Query OK, 0 rows affected (0.06 sec)
mysql> insert into product(prod_id,price,quantity) values
(1,200,10),(2,50,25),(3,80,10),(4,10,100)$
Query OK, 4 rows affected (0.03 sec)
Records: 4 Duplicates: 0 Warnings: 0
mysql> select * from product$
+---------+-------+----------+------------+
| prod_id | price | quantity | total_cost |
+---------+-------+----------+------------+
| 1 | 200 | 10 | 2000 |
| 2 | 50 | 25 | 1250 |
| 3 | 80 | 10 | 800 |
| 4 | 10 | 100 | 1000 |
+---------+-------+----------+------------+
4 rows in set (0.00 sec)
```

5) Show triggers

```
mysql> show triggers$
+------------------------+--------+----------+-----------------------------------------
-----------------------------------------------------------------------------------
+--------+----------------------+-----------------------------------------------------
-----------------------------------------------------------------------------+-------
---------+--------------------+--------------------+-------------------+
| Trigger | Event | Table | Statement
| Timing | Created | sql_mode
| Definer | character_set_client | collation_connection | Database Collation |
+------------------------+--------+----------+-----------------------------------------
-----------------------------------------------------------------------------------
+--------+----------------------+-----------------------------------------------------
-----------------------------------------------------------------------------+-------
---------+--------------------+--------------------+-------------------+
| UPDATEADSTATUS | INSERT | EMPLOYEE | BEGIN
IF NEW.AGE>=18 THEN SET NEW.ADULT_STATUS=1;
ELSE SET NEW.ADULT_STATUS=0;
END IF;
END | BEFORE | 2019-08-27 12:17:14.63 |
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_
DATE,ERROR_FOR_DIVISION_BY_Z
ERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION | root@localhost | utf8 |
utf8_general_ci | latin1_swedish_ci |
| before_employee1_update | UPDATE | employee1 | BEGIN INSERT INTO employees_audit
SET action
= 'update', employeeNumber = OLD.empno, firstname = OLD.empname, changedat = NOW();
END |
BEFORE | 2019-08-27 12:26:01.18 |
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_
DATE,ERROR_FOR_DIVISION_BY_Z
ERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION | root@localhost | utf8 |
utf8_general_ci | latin1_swedish_ci |
| t2 | INSERT | product | begin set new.total_cost=new.price*new.quantity;
end | BEFORE
| 2019-08-27 12:31:05.84 |
ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_
DATE,ERROR_FOR_DIVISION_BY_Z
ERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION | root@localhost | utf8 |
utf8_general_ci | latin1_swedish_ci |
+------------------------+--------+----------+-----------------------------------------
-----------------------------------------------------------------------------------
+--------+----------------------+-----------------------------------------------------
-----------------------------------------------------------------------------+-------
---------+--------------------+--------------------+-------------------+
3 rows in set (0.00 sec)
```

6) Show custom errors on insert

```
mysql> create trigger t3 before insert on employee1 for each row begin if new.salary<10000
then signal sqlstate '20000' set message_text='error'; end if; end;$
Query OK, 0 rows affected (0.06 sec)
mysql> insert into employee1 values(6,'manjeets','ass',25,'2018/02/02',5000,10)$
ERROR 1644 (20000): error
```

7) Trigger on delete.

```
mysql> create trigger T4 before delete on employee1 for each row begin if old.salary<20000
then signal sqlstate '20000' set message_text='error cannot delete'; end if; end;$
Query OK, 0 rows affected (0.07 sec)
mysql> delete from employee1 where salary=15000$
ERROR 1644 (20000): error cannot delete
```