Roll No. :- COMPTEB1449
Name :- Paarth Kothari
PRN :- 72018337F

Assignment No. 7

Title :- Write PL/SQL block to implement Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers) on Library table.

Problem Statement :- Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers). Write a database trigger on Library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library_Audit table. Frame the problem statement for writing Database Triggers of all types, in-line with above statement. The problem statement should clearly state the requirements.

Objectives :-

1. To learn and understand PL/SQL in Oracle.
2. To learn and understand triggers.

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, or UPDATE)
- A database definition (DDL) statement (CREATE, ALTER, or DROP).
- 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 values automatically
- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables
- Imposing security authorizations
- Preventing invalid transactions

Types of PL/SQL Triggers

- There are two types of triggers based on the which level it is triggered,

  1) **Row level :-** An event is triggered for each row upated, inserted or deleted.

  2) **Statement level trigger** :- An event is triggered for each sql statement executed.

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 a view.
- {INSERT [OR] | UPDATE [OR] | DELETE} − This specifies the DML operation.
- [OF col_name] − This specifies the column name that will be updated.

- [ON table_name] − This specifies the name of the table associated with the trigger.
- [REFERENCING OLD AS o NEW AS n] − This allows you to refer new and old values for various DML statements, such as INSERT, UPDATE, and DELETE.
- [FOR EACH ROW] − This specifies a rowlevel 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 level trigger.
- WHEN (condition) − This provides a condition for rows for which the trig fire. This clause is valid only for row.

Triggering a Trigger :-

Let us perform some DML operations on the CUSTOMERS table. statement, which will create a new record in the table −

```
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (7, 'Kriti', 22, 'HP', 7500.00 );
```

When a record is created in the CUSTOMERS table, the above create trigger, **display_salary_changes** will be fired and it will display result −

```
Old salary:
New salary: 7500
Salary difference:
```

Trigger for Library updation and deletion :

CREATE OR REPLACE TRIGGER BOOKS_AUDIT

BEFORE DELETE OR UPDATE ON library
REFERENCING OLD AS OLD NEW AS NEW
FOR EACH ROW
BEGIN

INSERT INTO library_audit
VALUES
(:old.id,
:old.book,
:old.author,
sysdate);
END;

Outcome:- After completing this assignment, students will be able to learn about Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers) on Library table.

Code & Output :-

```
mysql> use info;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> create table borrower2(roll_no int,name varchar(20),date_of_issue
date,book varchar2);
Query OK, 0 rows affected (0.44 sec)
mysql>insert into borrower2 values('1','nick','2018-06-
10','wings_of_fire','avaliable','APJ');
Query OK, 1 row affected (0.07 sec)
mysql> insert into borrower2 values('2','mira','2018-05-
11','leaves_life','not_avaliable','borwarkar');
Query OK, 1 row affected (0.05 sec)
```

```
mysql> insert into borrower2 values('3','rina','2018-02-
12','unusal','avaliable','johar');
Query OK, 1 row affected (0.04 sec)
mysql> insert into borrower2 values('4','harsha','2018-06-
20','skylimit','avaliable','ingale');
Query OK, 1 row affected (0.05 sec)
mysql> insert into borrower2 values('5','tej','2018-04-
20','highway','not_avaliable','klm');
Query OK, 1 row affected (0.05 sec)
mysql> select *from borrower1;
+---------+--------+---------------+---------------
+---------------+-----------+
| roll_no | name
| author
|
| date_of_issue | book_name
| status
+---------+--------+---------------+---------------
+---------------+-----------+
|
| APJ
1 | nick
|
| 2018-06-10
| wings_of_fire | avaliable
|
2 | mira
| 2018-05-11
not_avaliable | borwarkar | | leaves_life |
|
3 | rina
| johar
| | unusal | avaliable
|
4 | harsha | 2018-06-20
| ingale
| | skylimit | avaliable
|
5 | tej
| 2018-04-20
```

```
not_avaliable | klm
| | highway |
| 2018-02-12
+---------+--------+---------------+---------------
+---------------+-----------+
5 rows in set (0.00 sec)
//INSERT TRIGGER
mysql> delimiter //
mysql> create trigger library after insert on borrower1 for
each row
-> begin
-> insert into audit1
values(new.roll_no,new.name,new.date_of_issue,new.book_name,ne
w.status,new.author,current_timestamp);
-> end;
-> //Query OK, 0 rows affected (0.10 sec)
mysql> insert into borrower1 values('6','xyz','2018-09-
06','aaa','avaliable','xxx');
-> //
Query OK, 1 row affected (0.07 sec)
mysql> select * from borrower1;
-> //
+---------+--------+---------------+---------------
+---------------+-----------+
| roll_no | name
| author
|
| date_of_issue | book_name
| status
+---------+--------+---------------+---------------
+---------------+-----------+
|
| APJ
1 | nick
|
| 2018-06-10
| wings_of_fire | avaliable
|
2 | mira
| 2018-05-11
```

```
not_avaliable | borwarkar | | leaves_life |
|
3 | rina
| johar
| | unusal | avaliable
|
4 | harsha | 2018-06-20
| ingale
| | skylimit | avaliable
|
5 | tej
| 2018-04-20
not_avaliable | klm
| | highway |
|
| xxx | aaa | avaliable
6 | xyz
|
| 2018-02-12
| 2018-09-06
+---------+--------+---------------+---------------
+---------------+-----------+
6 rows in set (0.00 sec)
mysql> select * from audit1;
-> //
+---------+------+---------------+-----------+-----------
+--------+---------------------+| roll_no | name | date_of_issue |
book_name | status
author | ts
|
|
+---------+------+---------------+-----------+-----------
+--------+---------------------+
|
6 | xyz | 2018-09-06
| 2018-08-29 15:46:13 |
| aaa
| avaliable | xxx
+---------+------+---------------+-----------+-----------
+--------+---------------------+
```

```
1 row in set (0.00 sec)
// UPDATE TRIGGER
mysql> delimiter //
mysql> create trigger library1 after update on borrower1 for
each row
->
begin
->
insert into audit1
values(new.roll_no,new.name,new.date_of_issue,new.book_name,ne
w.status,new.author,current_timestamp);
-> end;
-> //
Query OK, 0 rows affected (0.08 sec)
mysql> update borrower1 set roll_no='8',book_name='leaf' where name='xyz';
-> //
Query OK, 1 row affected (0.04 sec)
Rows matched: 1
Changed: 1
Warnings: 0
mysql> select *from borrower1;
-> //
+---------+--------+---------------+---------------
+---------------+-----------+
| roll_no | name
| author
|
| date_of_issue | book_name
| status+---------+--------+---------------+---------------
+---------------+-----------+
|
| APJ
1 | nick
|
| 2018-06-10
| wings_of_fire | avaliable
|
2 | mira
| 2018-05-11
not_avaliable | borwarkar | | leaves_life |
```

```
|
3 | rina
| johar
| | unusal | avaliable
|
4 | harsha | 2018-06-20
| ingale
| | skylimit | avaliable
|
5 | tej
| 2018-04-20
not_avaliable | klm
| | highway |
|
| xxx | leaf | avaliable
8 | xyz
|
| 2018-02-12
| 2018-09-06
+---------+--------+---------------+---------------
+---------------+-----------+
6 rows in set (0.00 sec)
```

Conclusion: Thus we have successfully implemented trigger to keep track of update and performed on library table.