

**NAME: Atharva Chavan**  
**TE-A IT**  
**ROLL NO: T1851010**  
**PRN: 71901316L**

**Group B: SQL & PL/SQL**  
**Assignment No. 8**

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

**Objective:**

- To study and implement PL/SQL 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

## 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 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 level trigger.
- WHEN (condition) – This provides a condition for rows for which the trigger would fire. This clause is valid only for row-level triggers.

### **Conclusion:-**

We have studied and executed different types of database triggers.

### Code & Output: -

CREATE TABLE EMP4

```
(  
EMP_NUM INT(5),  
FIRSTNAME VARCHAR(20),  
LASTNAME VARCHAR (20),  
CHANGEDAT DATE);
```

```
CREATE TABLE "EMPAUDIT" (  
    "ID"    INTEGER,  
    "EMP_NUM" INTEGER,  
    "LASTNAME" TEXT,  
    "CHANGEDAT" DATE  
);
```

DB Browser for SQLite - C:\Users\hp\Desktop\Virtual lab\finaltrigger.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Edit Pragma Execute SQL Browse Data

SQL 1

```
1 SELECT * FROM EMPAUDIT;
```

ID	EMP_NUM	LASTNAME	CHANGEDAT
1	1010	1 patil	10-09-1995
2	1011	2 sharma	24-08-2000
3	1012	3 naik	30-10-1999
4	1013	4 patil	31-05-1997
5	1014	5 sharma	04-08-1998

Execution finished without errors.  
Result: 5 rows returned in 133ms  
At line 1:  
SELECT \* FROM EMPAUDIT;

Execution finished without errors.  
Result: 5 rows returned in 133ms  
At line 1:  
SELECT \* FROM EMPAUDIT;

Edit Database Cell

Mode: Text

NULL

Type of data currently in cell: NULL  
0 byte(s)

Remote

Identity: Public

Name	Commit	Last modified	Size
------	--------	---------------	------

SQL Log Plot DB Schema Remote

UTF-8

1005  
28-09-2020



-----BEFORE UPDATE-----  
-----

[illegible]

The screenshot shows the DB Browser for SQLite application. The main window has a menu bar (File, Edit, View, Tools, Help) and a toolbar. The 'Database Structure' tab is active, showing a tree view of the database schema. The 'SQL' tab is also visible, showing a query: `select * from emp4;`. The 'Results' tab shows the output of the query as a table with 8 rows and 4 columns: EMP\_NUM, FIRSTNAME, LASTNAME, and CHANGEDAT. The results are: 4 -596 sham patil 31-05-1997, 5 -595 gram sharma 04-08-1998, 6 1208 sam patil 0, 7 1218 Ram patil -2003, 8 1418 DEV patil -1979. The status bar at the bottom indicates 'Execution finished without errors. Results: 8 rows returned in 21ms. At line 1: select \* from emp4;'. On the right side, there is a 'Database Cell' panel showing the 'Type of data currently in cell: NULL' and '0 byte(s)'. Below this, the 'Remote' tab is active, showing a table with columns: Name, Commit, Last modified, and Size. The table is currently empty.

EMP_NUM	FIRSTNAME	LASTNAME	CHANGEDAT
4	-596 sham	patil	31-05-1997
5	-595 gram	sharma	04-08-1998
6	1208 sam	patil	0
7	1218 Ram	patil	-2003
8	1418 DEV	patil	-1979

Execution finished without errors.  
Results: 8 rows returned in 21ms  
At line 1:  
select \* from emp4;

Results of the last executed statements.  
You may want to collapse this panel and use the SQL Log dock with User selection instead.

Name	Commit	Last modified	Size
------	--------	---------------	------

.....

[illegible]

DB Browser for SQLite - C:\Users\jpr\Desktop\Virtual lab\finaltrigger2.r2

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Edit Progress Execute SQL Browse Data

SQL 1 Edit Trigger not\_trigger

```
1 SELECT * FROM EMP4;
```

EMP_NUM	FIRSTNAME	LASTNAME	CHANGEDAT
3	-597 ram	naik	30-10-1999
4	-596 sham	patil	31-05-1997
5	-595 gram	sharma	04-08-1998
6	1208 sam	patil	0
7	1218 Ram	patil	-2003

Execution finished without errors.  
Results: 7 rows returned in 30ms  
At line 1:  
SELECT \* FROM EMP4;

Results of the last executed statements.  
You may want to collapse this panel and use the SQL Log dock with User selection instead.

Edit Database Cell

Mode: Text

NULL

Type of data currently in cell: NULL  
0 byte(s)

Remote

Identity: Public

Name	Commit	Last modified	Size
------	--------	---------------	------

SQL Log Plot DB Schema Remote

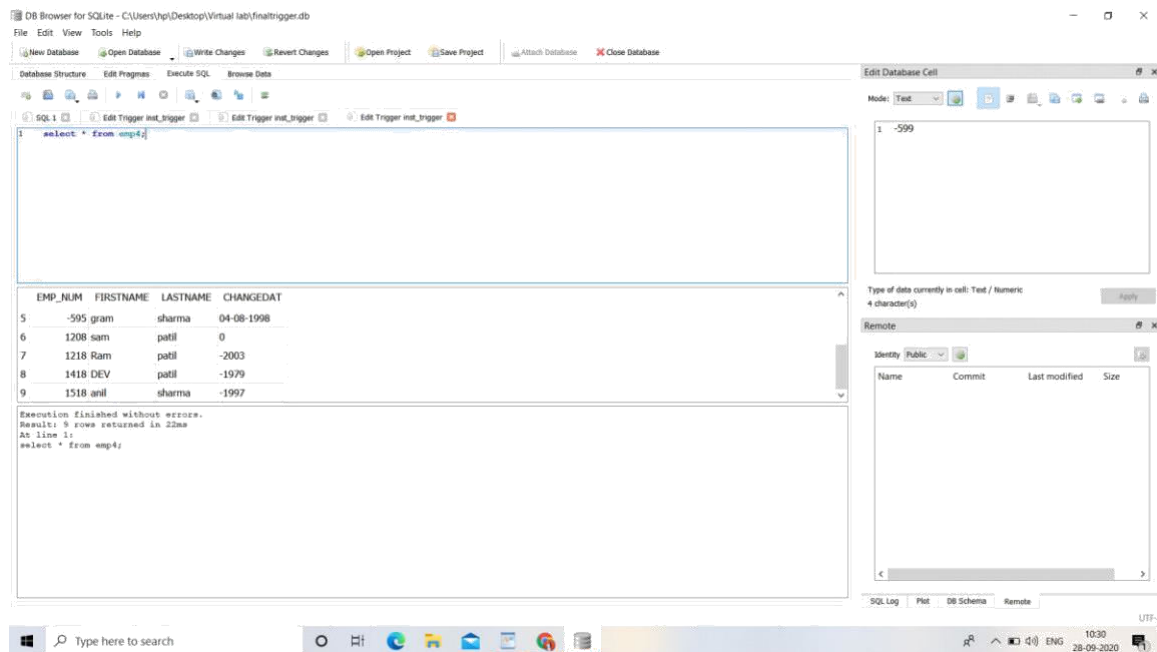
UTF-8

-----AFTER UPDATE-----

```
DROP TRIGGER "main"."inst_trigger";
CREATE TRIGGER inst_trigger after UPDATE ON EMP4
FOR EACH ROW
BEGIN
UPDATE EMP4 SET EMP_NUM=EMP_NUM-200 ;
end
```

[illegible]

```
INSERT INTO EMP4 VALUES(1518,'anil','sharma','09-08-1998');
```



-----Delete Trigger-----

**S**

DROP TRIGGER INST\_TRIGGER;