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/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:-

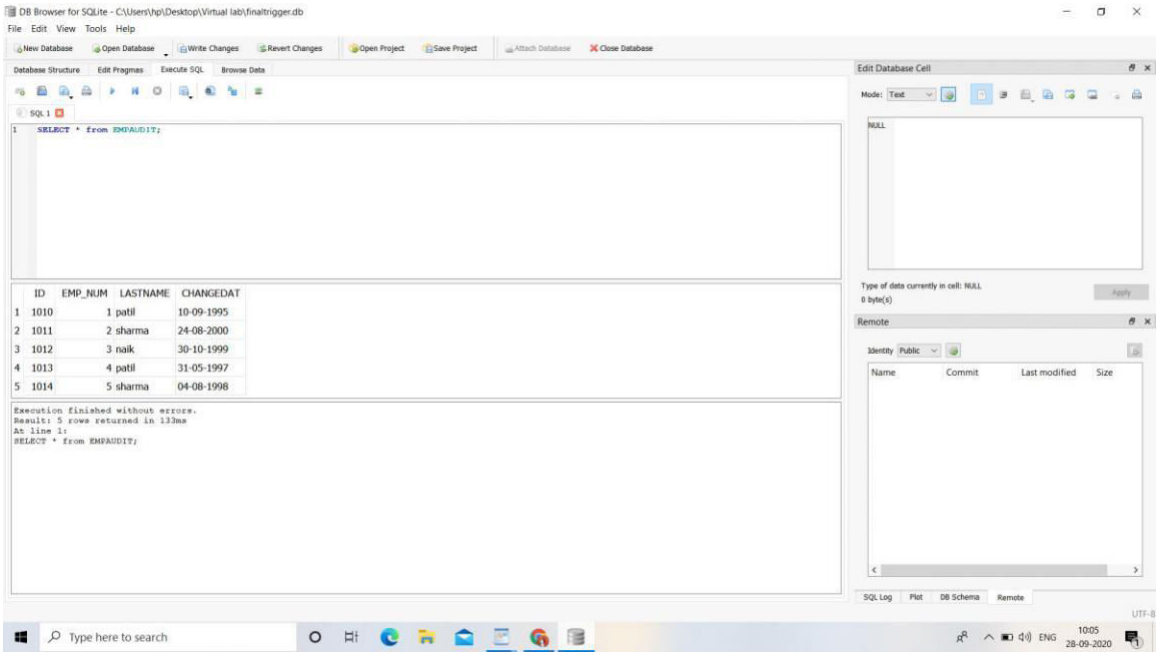
We have studied and executed different types of database triggers.

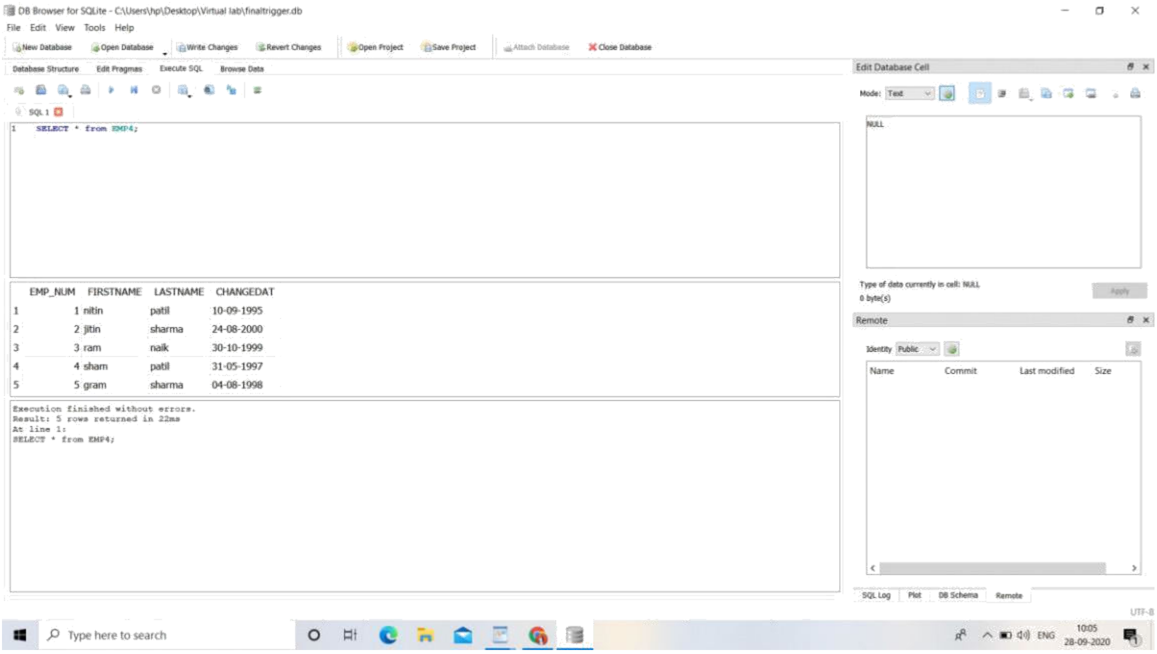
Code & Output: -

```
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
);
```

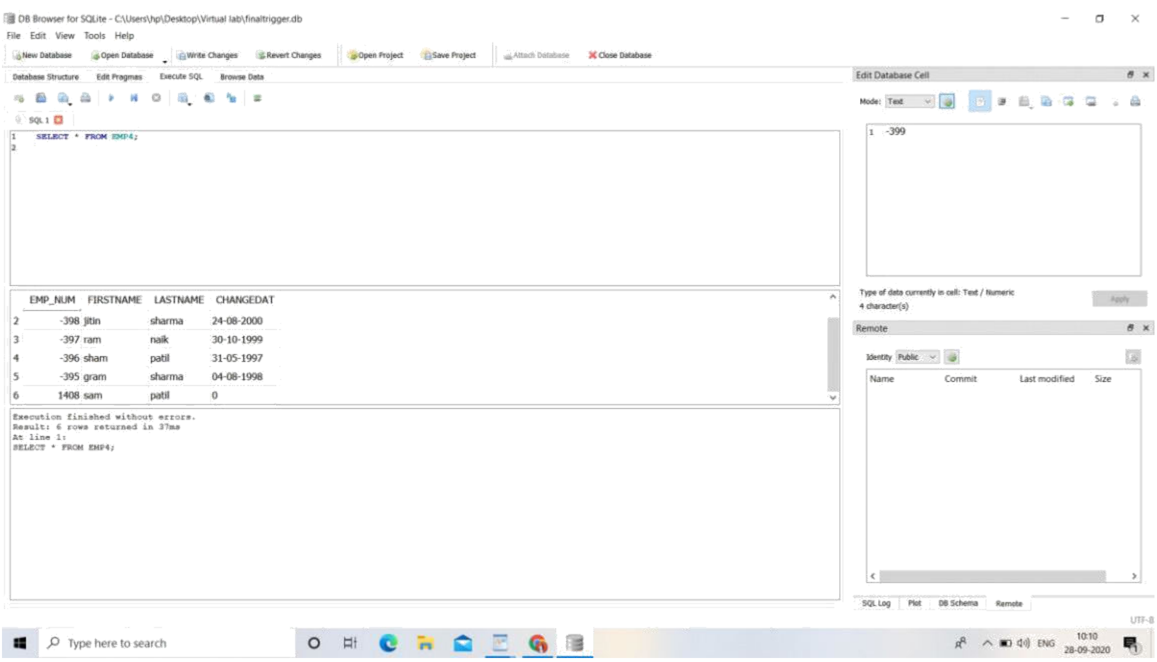




BEFORE INSERT→
CREATE TRIGGER inst_trigger BEFORE INSERT ON EMP4
FOR EACH ROW
BEGIN
UPDATE EMP4 SET EMP_NUM=EMP_NUM-400 ;
end;

^^

INSERT INTO EMP

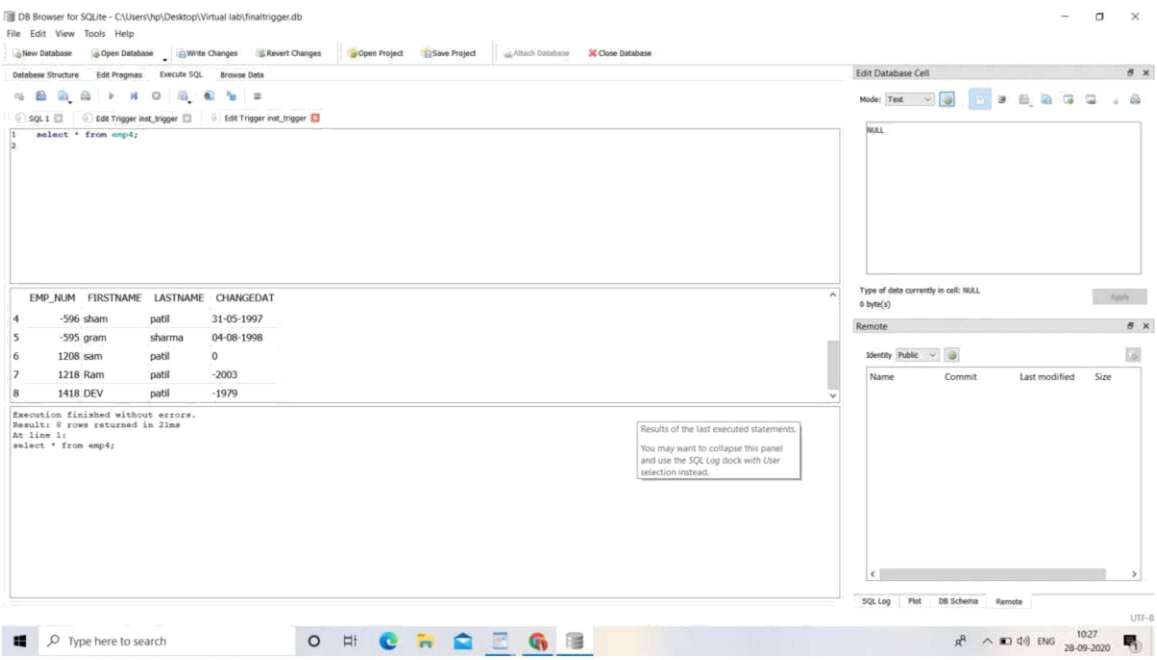


INSERT INTO EMP4 VALUES(1408,'sam','patil',0000-00-00);
-----------------------------BEFORE UPDATE--

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

^^

INSERT INTO EMP4 VALUES(1418,'DEV','patil',27-08-1998);

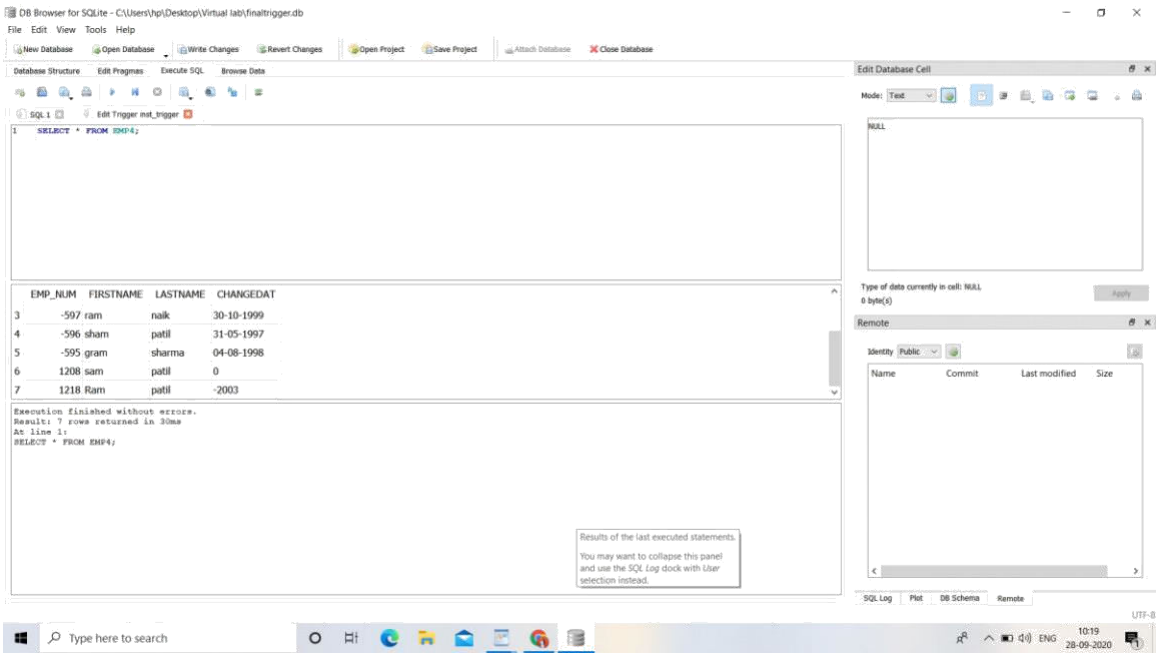


----------AFTER INSERT----------

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

^^

```
INSERT INTO EMP4 VALUES(1418,'Ram','patil',09-08-1998);
```

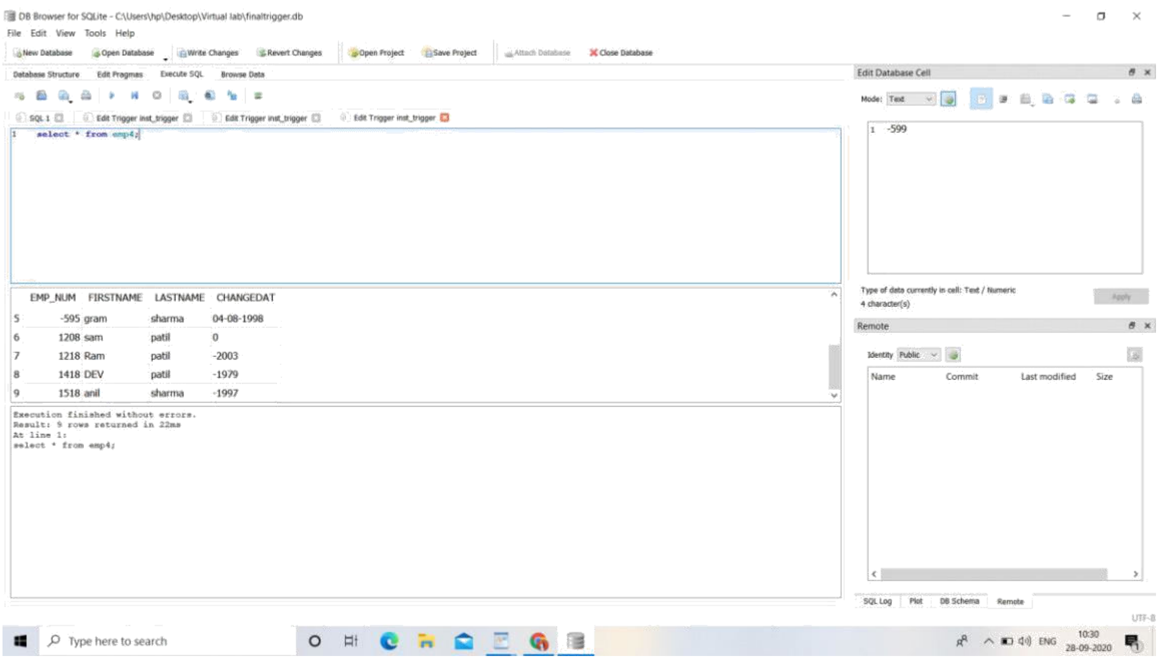


-------------------------------------AFTER UPDATE--

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

^^^

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



---------------------------------Delete Trigger---
s

DROP TRIGGER INST_TRIGGER;