NAME:AVINASH BHIWALE ROLL NO.:T1851050 PRN NO.:71901284J DIV:A

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.
- {INS**ERT [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:-

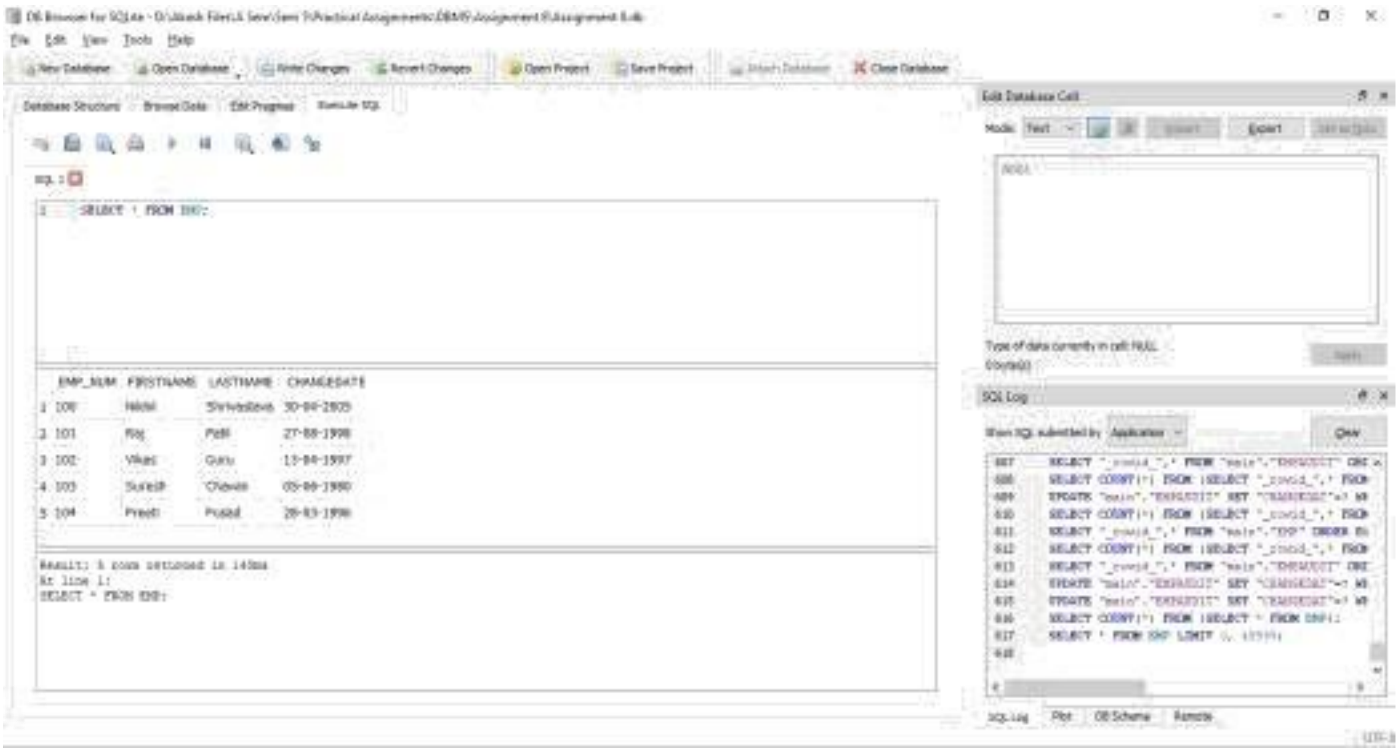
We have studied and executed different types of database triggers.

Output:

1. **CREATE TABLE:**

   a) EMP:

```
CREATE TABLE EMP(
    EMP_NUM INT(5) AUTOINCREMENT,
    FIRSTNAME VARCHAR(20),
    LASTNAME VARCHAR (20),
    CHANGEDATE DATE
);
```



```
SELECT * FROM EMP;
```

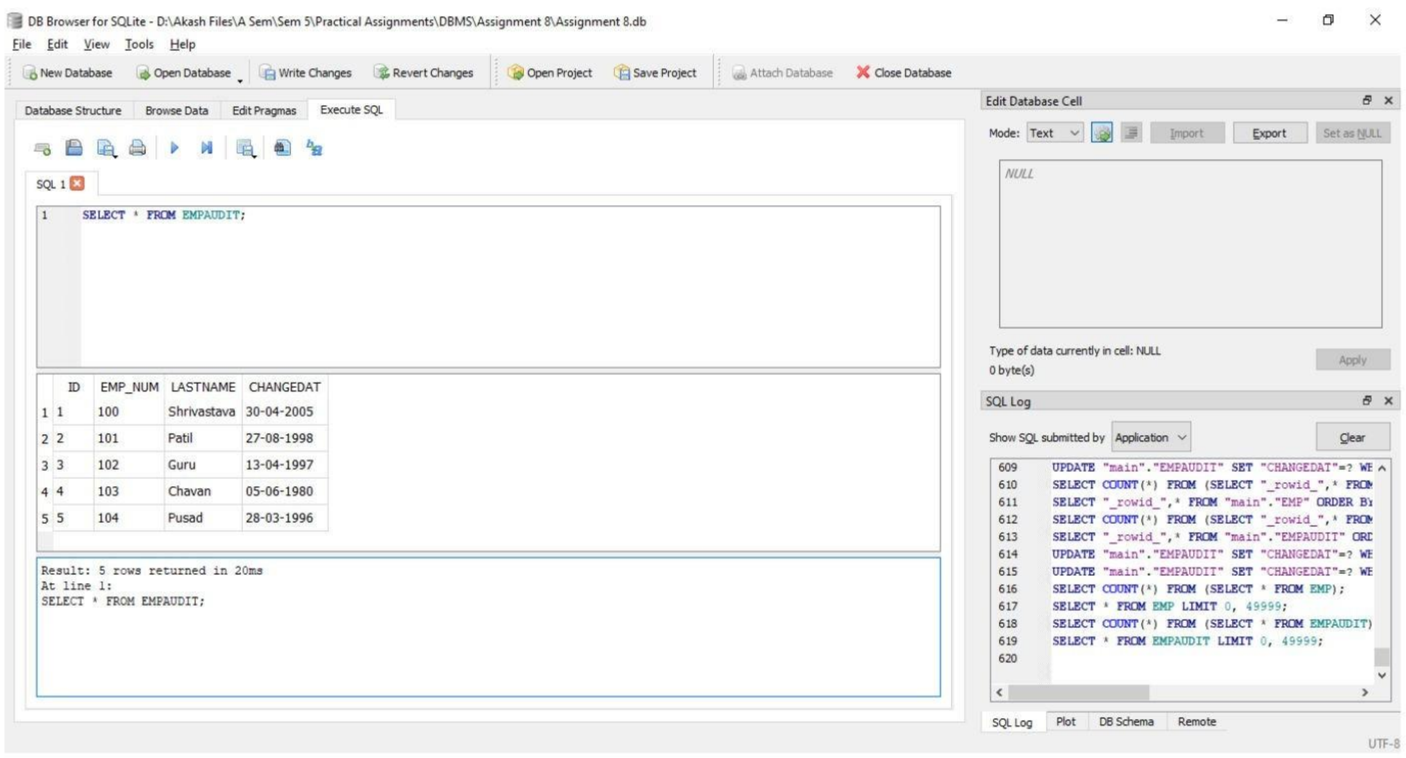
```
   EMP_NUM  FIRSTNAME  LASTNAME    CHANGEDATE
1  100      Nikhil     Shrivastava 30-04-2005
2  101      Raj        Patil       27-08-1998
3  102      Vikas      Guru        13-04-1997
4  103      Suresh     Chavan      05-09-1980
5  104      Preeti     Prasad      28-03-1996
```

```
Result: 5 rows returned in 143ms
At line 1:
SELECT * FROM EMP;
```

```
SELECT "_rowid_",* FROM "main"."EMPAUDIT" ORI
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
UPDATE "main"."EMPAUDIT" SET "CHANGEDAT"=? WH
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
SELECT "_rowid_",* FROM "main"."EMP" ORDER BY
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
SELECT "_rowid_",* FROM "main"."EMPAUDIT" ORI
UPDATE "main"."EMPAUDIT" SET "CHANGEDAT"=? WH
UPDATE "main"."EMPAUDIT" SET "CHANGEDAT"=? WH
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
```

   b) EMPAUDIT:

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


DB Browser for SQLite - D:\Akash Files\A Sem\Sem 5\Practical Assignments\DBMS\Assignment 8\Assignment 8.db

```
SELECT * FROM EMPAUDIT;
```

	ID	EMP_NUM	LASTNAME	CHANGEDAT
1	1	100	Shrivastava	30-04-2005
2	2	101	Patil	27-08-1998
3	3	102	Guru	13-04-1997
4	4	103	Chavan	05-06-1980
5	5	104	Pusad	28-03-1996

```
Result: 5 rows returned in 20ms
At line 1:
SELECT * FROM EMPAUDIT;
```

Edit Database Cell — Mode: Text — NULL

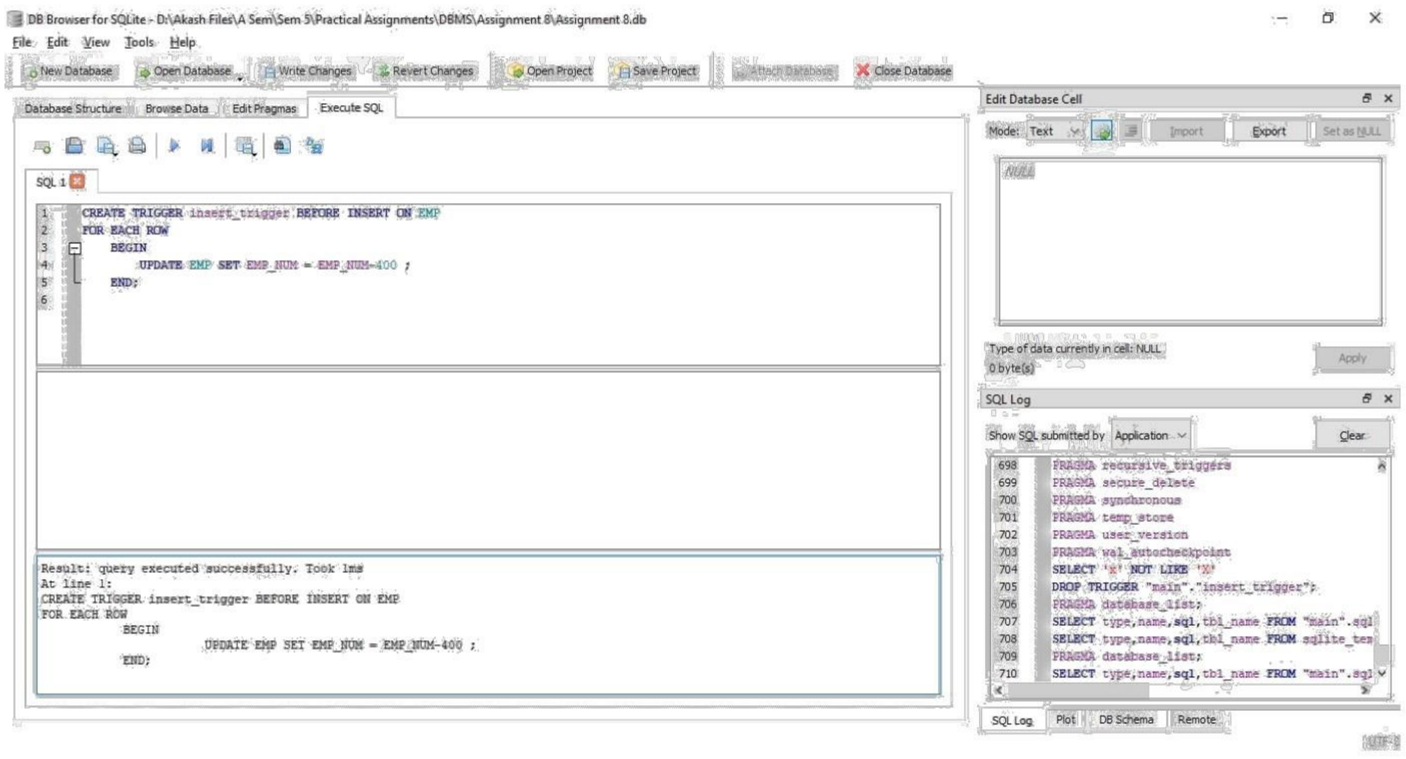
Type of data currently in cell: NULL
0 byte(s)

SQL Log — Show SQL submitted by: Application

```
UPDATE "main"."EMPAUDIT" SET "CHANGEDAT"=? WE
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
SELECT "_rowid_",* FROM "main"."EMP" ORDER By
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
SELECT "_rowid_",* FROM "main"."EMPAUDIT" ORD
UPDATE "main"."EMPAUDIT" SET "CHANGEDAT"=? WE
UPDATE "main"."EMPAUDIT" SET "CHANGEDAT"=? WE
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
SELECT COUNT(*) FROM (SELECT * FROM EMPAUDIT)
SELECT * FROM EMPAUDIT LIMIT 0, 49999;
```

2. CREATE TRIGGER (BEFORE INSERTING NEW VALUE IN TABLE):

```
CREATE TRIGGER insert_trigger BEFORE INSERT ON
EMP FOR EACH ROW
        BEGIN
                UPDATE EMP SET EMP_NUM = EMP_NUM-400 ;
        END;
```


DB Browser for SQLite - D:\Akash Files\A Sem\Sem 5\Practical Assignments\DBMS\Assignment 8\Assignment 8.db

```
CREATE TRIGGER insert_trigger BEFORE INSERT ON EMP
FOR EACH ROW
    BEGIN
        UPDATE EMP SET EMP_NUM = EMP_NUM-400 ;
    END;
```

```
Result: query executed successfully. Took 1ms
At line 1:
CREATE TRIGGER insert_trigger BEFORE INSERT ON EMP
FOR EACH ROW
          BEGIN
                    UPDATE EMP SET EMP_NUM = EMP_NUM-400 ;
          END;
```

Edit Database Cell — Mode: Text — NULL

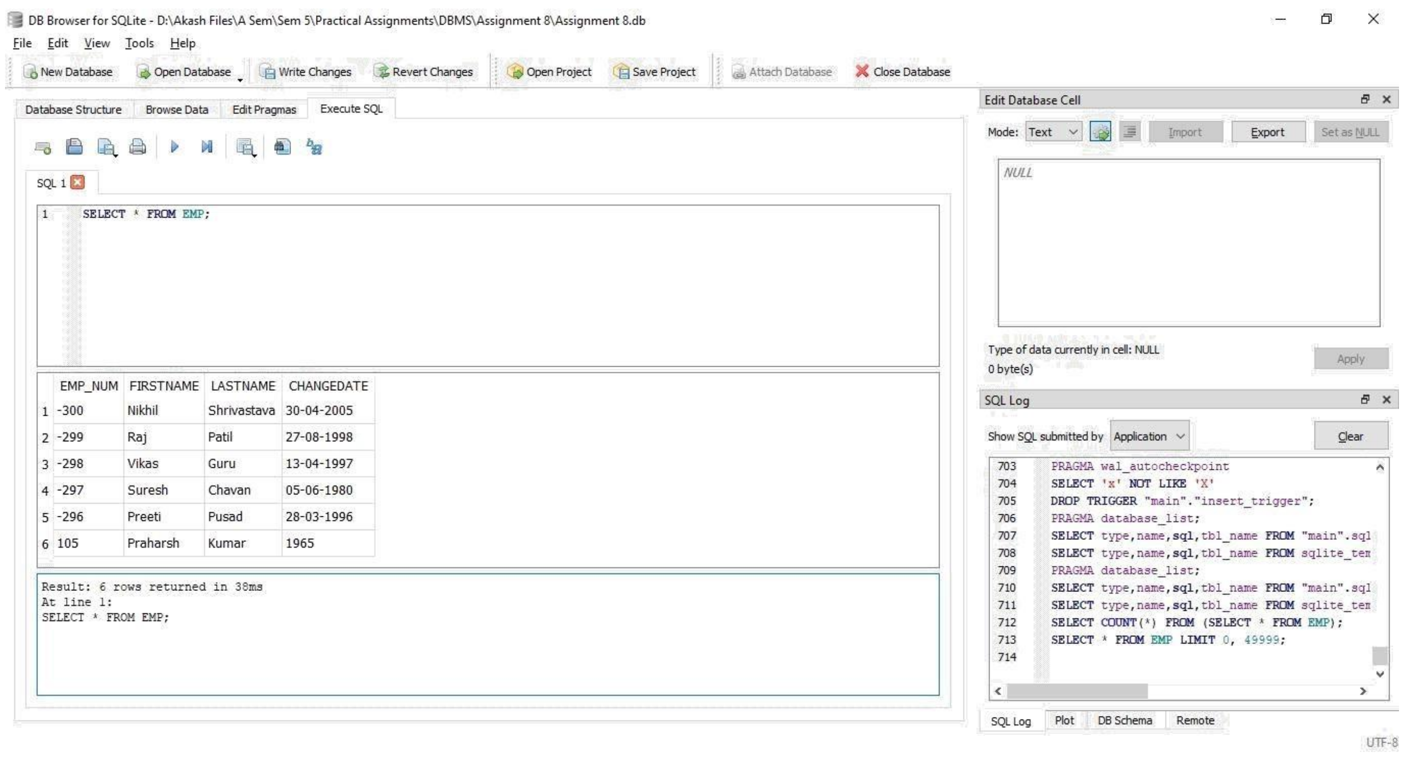
Type of data currently in cell: NULL
0 byte(s)

SQL Log — Show SQL submitted by: Application

```
PRAGMA recursive_triggers
PRAGMA secure_delete
PRAGMA synchronous
PRAGMA temp_store
PRAGMA user_version
PRAGMA wal_autocheckpoint
SELECT 'x' NOT LIKE 'X'
DROP TRIGGER "main"."insert_trigger";
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
```

3. INSERT VALUE IN EMP TABLE:

INSERT INTO EMP VALUES(105,'Praharsh','Kumar',1995-10-20);



```
SELECT * FROM EMP;
```

	EMP_NUM	FIRSTNAME	LASTNAME	CHANGEDATE
1	-300	Nikhil	Shrivastava	30-04-2005
2	-299	Raj	Patil	27-08-1998
3	-298	Vikas	Guru	13-04-1997
4	-297	Suresh	Chavan	05-06-1980
5	-296	Preeti	Pusad	28-03-1996
6	105	Praharsh	Kumar	1965

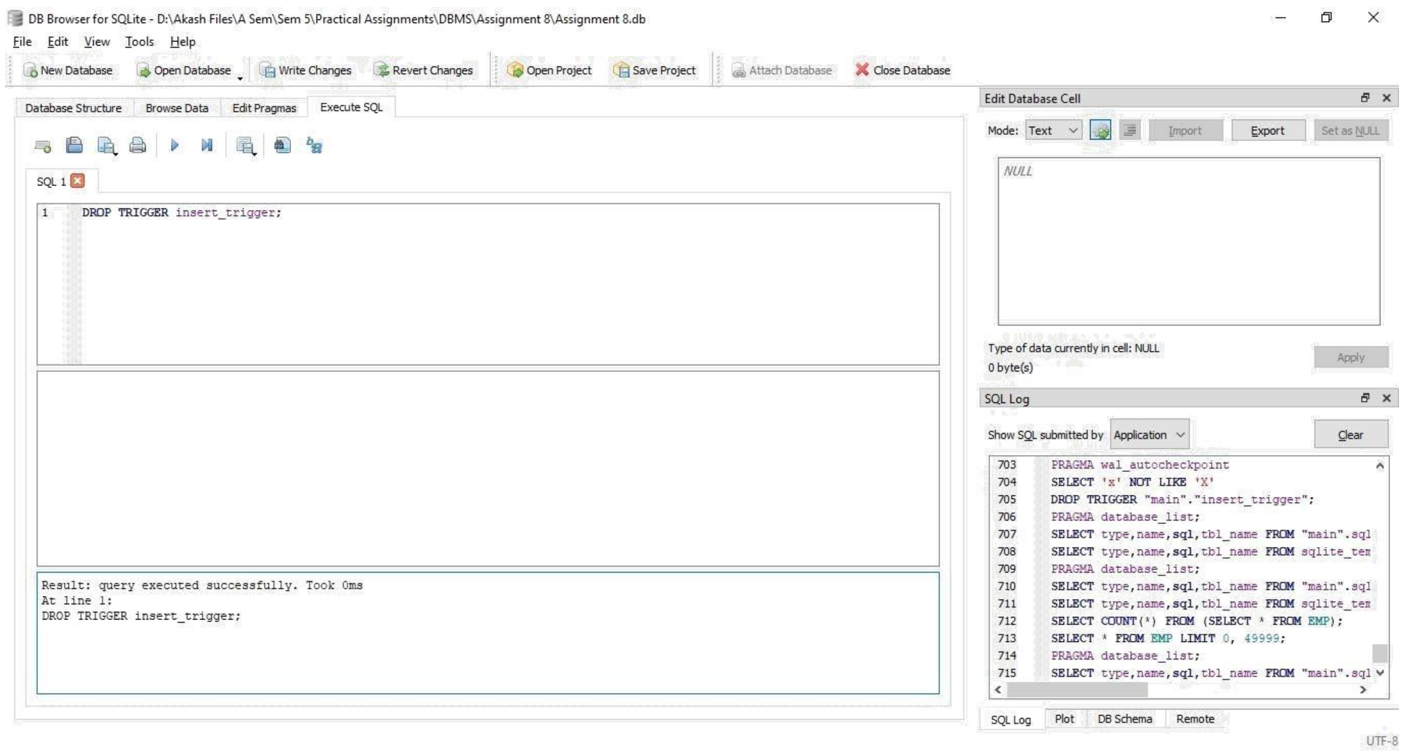
```
Result: 6 rows returned in 38ms
At line 1:
SELECT * FROM EMP;
```

```
PRAGMA wal_autocheckpoint
SELECT 'x' NOT LIKE 'X'
DROP TRIGGER "main"."insert_trigger";
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
```

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

4. DROP TRIGGER:

DROP TRIGGER insert_trigger;



```
DROP TRIGGER insert_trigger;
```

```
Result: query executed successfully. Took 0ms
At line 1:
DROP TRIGGER insert_trigger;
```

```
PRAGMA wal_autocheckpoint
SELECT 'x' NOT LIKE 'X'
DROP TRIGGER "main"."insert_trigger";
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
```

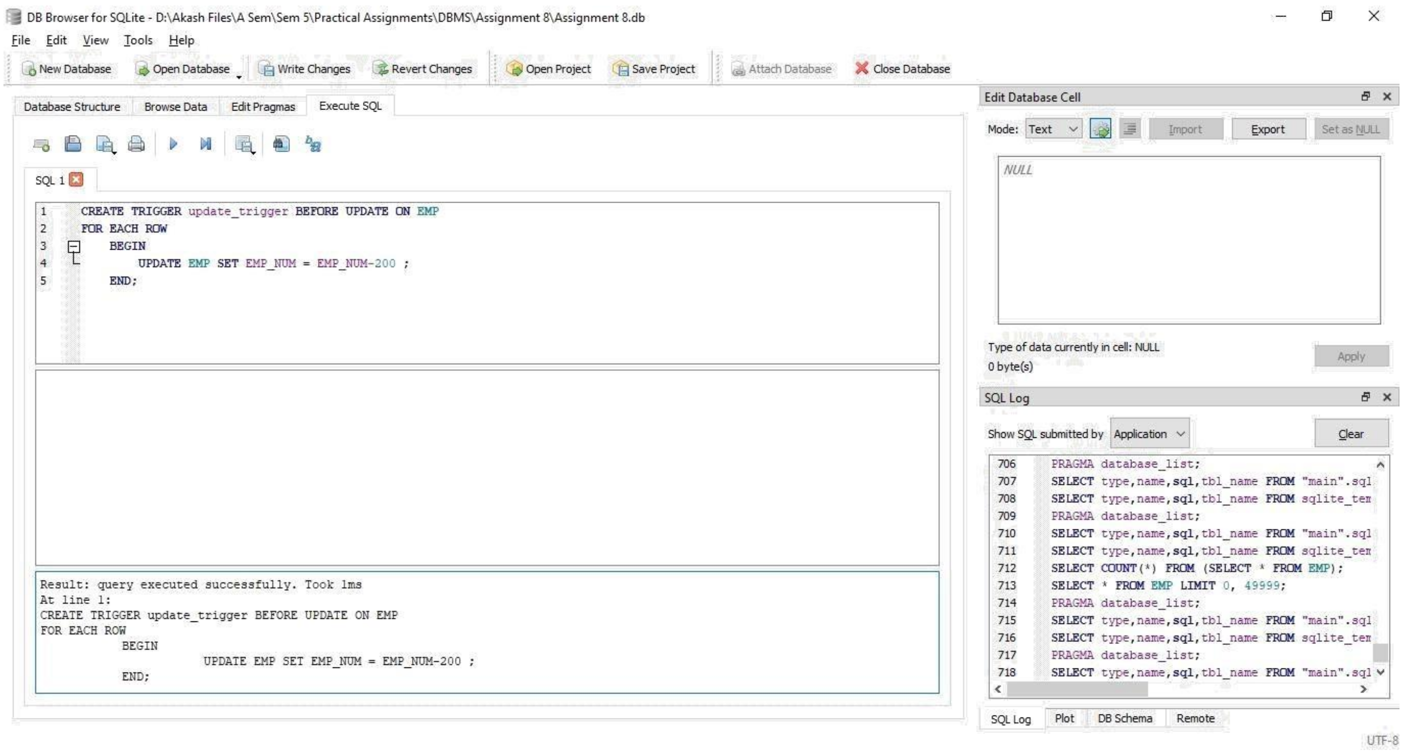
Type of data currently in cell: NULL
0 byte(s)

5. UPDATE TRIGGER:

a) BEFORE UPDATE:

CREATE TRIGGER update_trigger BEFORE UPDATE ON EMP
FOR EACH ROW

```
BEGIN
        UPDATE EMP SET EMP_NUM = EMP_NUM-200 ;
```



```
CREATE TRIGGER update_trigger BEFORE UPDATE ON EMP
FOR EACH ROW
    BEGIN
        UPDATE EMP SET EMP_NUM = EMP_NUM-200 ;
    END;
```

```
Result: query executed successfully. Took 1ms
At line 1:
CREATE TRIGGER update_trigger BEFORE UPDATE ON EMP
FOR EACH ROW
            BEGIN
                        UPDATE EMP SET EMP_NUM = EMP_NUM-200 ;
            END;
```

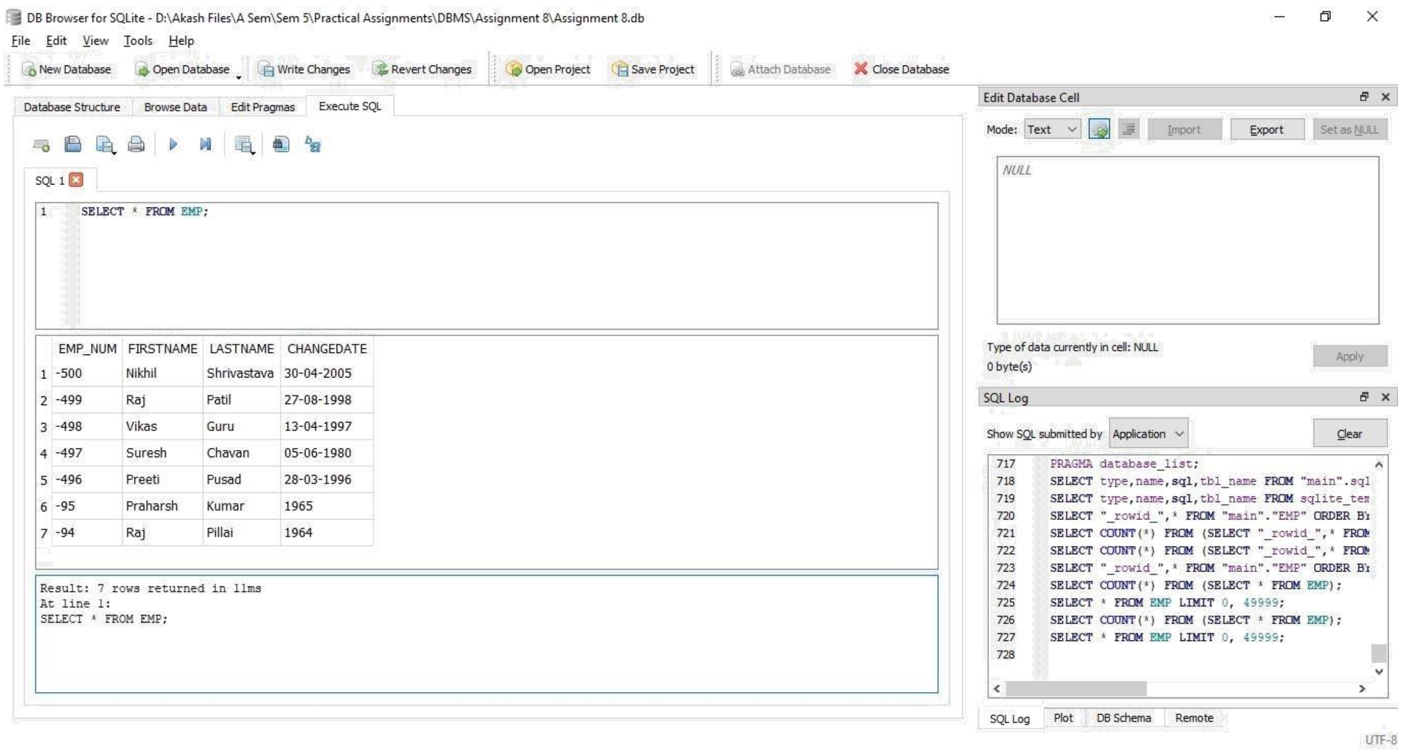
Type of data currently in cell: NULL
0 byte(s)

Show SQL submitted by Application

```
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
```

```
END;
```

```
UPDATE EMP
SET FIRSTNAME = 'Raj'
WHERE EMP_NUM = 106;
```



```
SELECT * FROM EMP;
```

	EMP_NUM	FIRSTNAME	LASTNAME	CHANGEDATE
1	-500	Nikhil	Shrivastava	30-04-2005
2	-499	Raj	Patil	27-08-1998
3	-498	Vikas	Guru	13-04-1997
4	-497	Suresh	Chavan	05-06-1980
5	-496	Preeti	Pusad	28-03-1996
6	-95	Praharsh	Kumar	1965
7	-94	Raj	Pillai	1964

```
Result: 7 rows returned in 11ms
At line 1:
SELECT * FROM EMP;
```

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

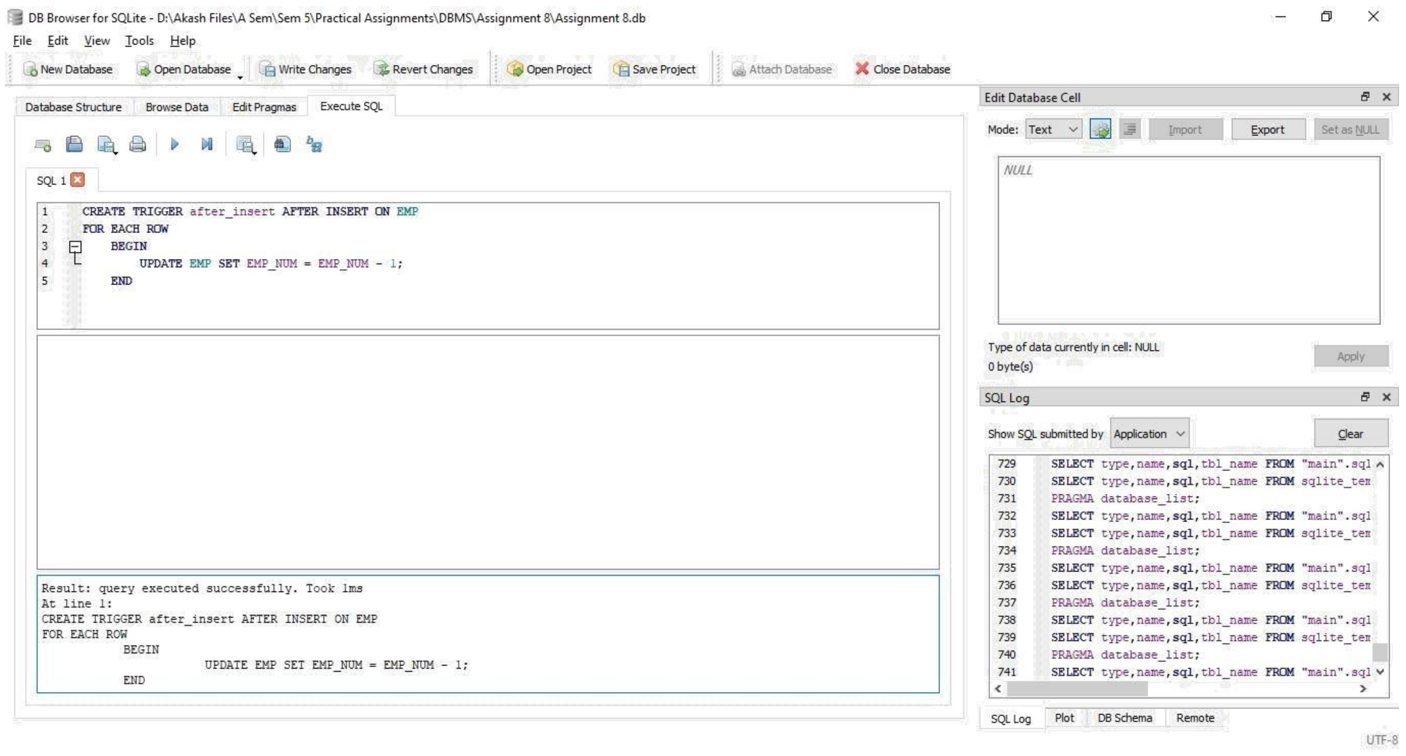
Show SQL submitted by Application

```
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
SELECT "_rowid_",* FROM "main"."EMP" ORDER BY
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
SELECT "_rowid_",* FROM "main"."EMP" ORDER BY
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
```

```
DROP TRIGGER update_trigger;
```

6. AFTER:

```
CREATE TRIGGER after_insert AFTER INSERT ON
EMP FOR EACH ROW
    BEGIN
          UPDATE EMP SET EMP_NUM = EMP_NUM - 1;
```



DB Browser for SQLite - D:\Akash Files\A Sem\Sem 5\Practical Assignments\DBMS\Assignment 8\Assignment 8.db

```
CREATE TRIGGER after_insert AFTER INSERT ON EMP
FOR EACH ROW
    BEGIN
        UPDATE EMP SET EMP_NUM = EMP_NUM - 1;
    END
```

```
Result: query executed successfully. Took 1ms
At line 1:
CREATE TRIGGER after_insert AFTER INSERT ON EMP
FOR EACH ROW
           BEGIN
                      UPDATE EMP SET EMP_NUM = EMP_NUM - 1;
           END
```

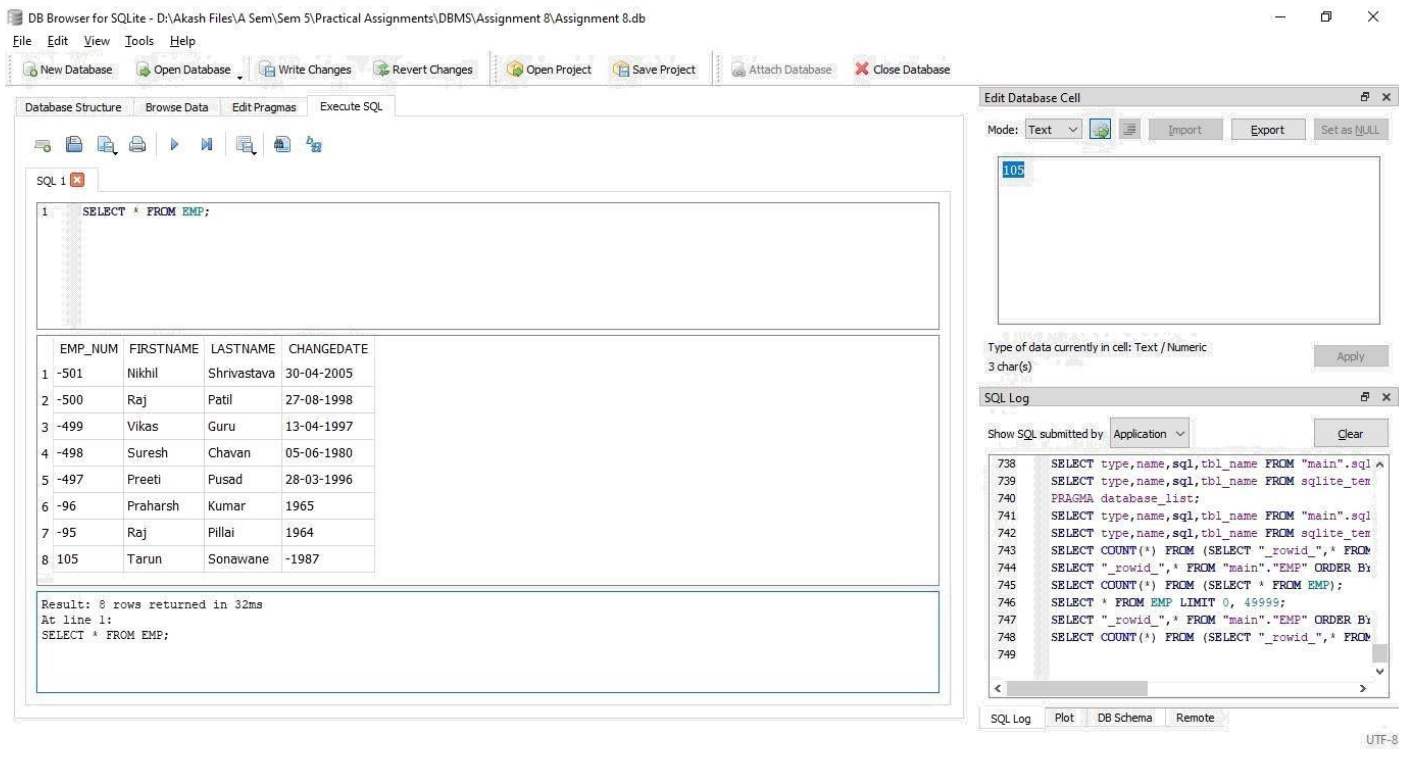
NULL

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

```
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
```

```
    END
```

```
INSERT INTO EMP VALUES(106,'Tarun','Sonawane',19-08-1998);
```



DB Browser for SQLite - D:\Akash Files\A Sem\Sem 5\Practical Assignments\DBMS\Assignment 8\Assignment 8.db

```
SELECT * FROM EMP;
```

	EMP_NUM	FIRSTNAME	LASTNAME	CHANGEDATE
1	-501	Nikhil	Shrivastava	30-04-2005
2	-500	Raj	Patil	27-08-1998
3	-499	Vikas	Guru	13-04-1997
4	-498	Suresh	Chavan	05-06-1980
5	-497	Preeti	Pusad	28-03-1996
6	-96	Praharsh	Kumar	1965
7	-95	Raj	Pillai	1964
8	105	Tarun	Sonawane	-1987

```
Result: 8 rows returned in 32ms
At line 1:
SELECT * FROM EMP;
```

105

Type of data currently in cell: Text / Numeric
3 char(s)

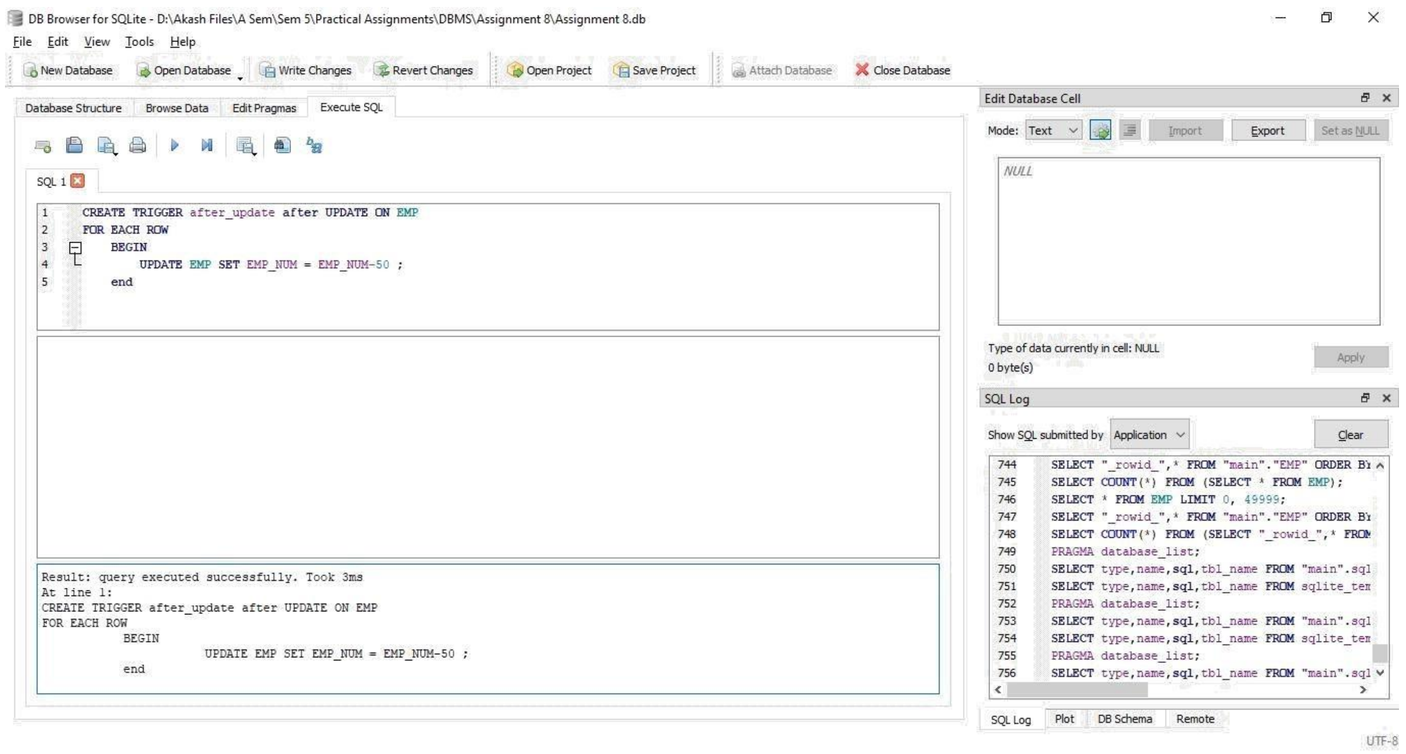
```
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
SELECT "_rowid_",* FROM "main"."EMP" ORDER BY
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
SELECT "_rowid_",* FROM "main"."EMP" ORDER BY
SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
```

DROP TRIGGER after_insert;

7. AFTER UPDATE:

```
CREATE TRIGGER after_update after UPDATE ON EMP
FOR EACH ROW
    BEGIN
        UPDATE EMP SET EMP_NUM = EMP_NUM-50 ;
    END
```

INSERT INTO EMP VALUES (100, 'Viraj', 'Kate', 25-03-2001);



```
CREATE TRIGGER after_update after UPDATE ON EMP
FOR EACH ROW
    BEGIN
        UPDATE EMP SET EMP_NUM = EMP_NUM-50 ;
    end
```

```
Result: query executed successfully. Took 3ms
At line 1:
CREATE TRIGGER after_update after UPDATE ON EMP
FOR EACH ROW
          BEGIN
                    UPDATE EMP SET EMP_NUM = EMP_NUM-50 ;
          end
```

Mode: Text

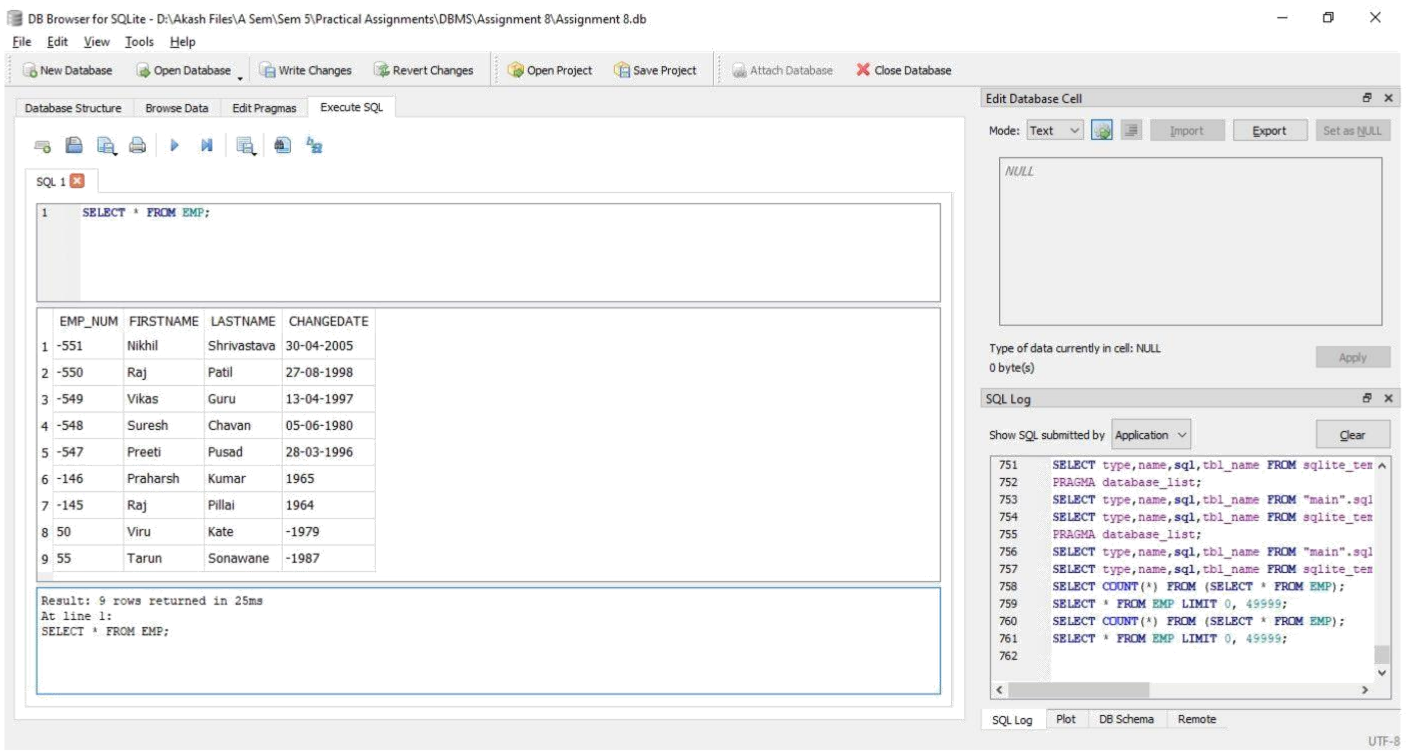
NULL

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

Show SQL submitted by: Application

```
744   SELECT "_rowid_",* FROM "main"."EMP" ORDER B
745   SELECT COUNT(*) FROM (SELECT * FROM EMP);
746   SELECT * FROM EMP LIMIT 0, 49999;
747   SELECT "_rowid_",* FROM "main"."EMP" ORDER B
748   SELECT COUNT(*) FROM (SELECT "_rowid_",* FROM
749   PRAGMA database_list;
750   SELECT type,name,sql,tbl_name FROM "main".sql
751   SELECT type,name,sql,tbl_name FROM sqlite_tem
752   PRAGMA database_list;
753   SELECT type,name,sql,tbl_name FROM "main".sql
754   SELECT type,name,sql,tbl_name FROM sqlite_tem
755   PRAGMA database_list;
756   SELECT type,name,sql,tbl_name FROM "main".sql
```

UPDATE EMP
SET FIRSTNAME = 'Viru'
WHERE EMP_NUM = 100;


DB Browser for SQLite - D:\Akash Files\A Sem\Sem 5\Practical Assignments\DBMS\Assignment 8\Assignment 8.db
File Edit View Tools Help
New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database
Database Structure Browse Data Edit Pragmas Execute SQL
SQL 1
Edit Database Cell
Mode: Text Import Export Set as NULL
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
SQL Log
Show SQL submitted by Application Clear
SQL Log Plot DB Schema Remote
UTF-8

```
SELECT * FROM EMP;
```

	EMP_NUM	FIRSTNAME	LASTNAME	CHANGEDATE
1	-551	Nikhil	Shrivastava	30-04-2005
2	-550	Raj	Patil	27-08-1998
3	-549	Vikas	Guru	13-04-1997
4	-548	Suresh	Chavan	05-06-1980
5	-547	Preeti	Pusad	28-03-1996
6	-146	Praharsh	Kumar	1965
7	-145	Raj	Pillai	1964
8	50	Viru	Kate	-1979
9	55	Tarun	Sonawane	-1987

```
Result: 9 rows returned in 25ms
At line 1:
SELECT * FROM EMP;
```

```
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main".sql
SELECT type,name,sql,tbl_name FROM sqlite_tem
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
SELECT COUNT(*) FROM (SELECT * FROM EMP);
SELECT * FROM EMP LIMIT 0, 49999;
```

DROP TRIGGER after_update;