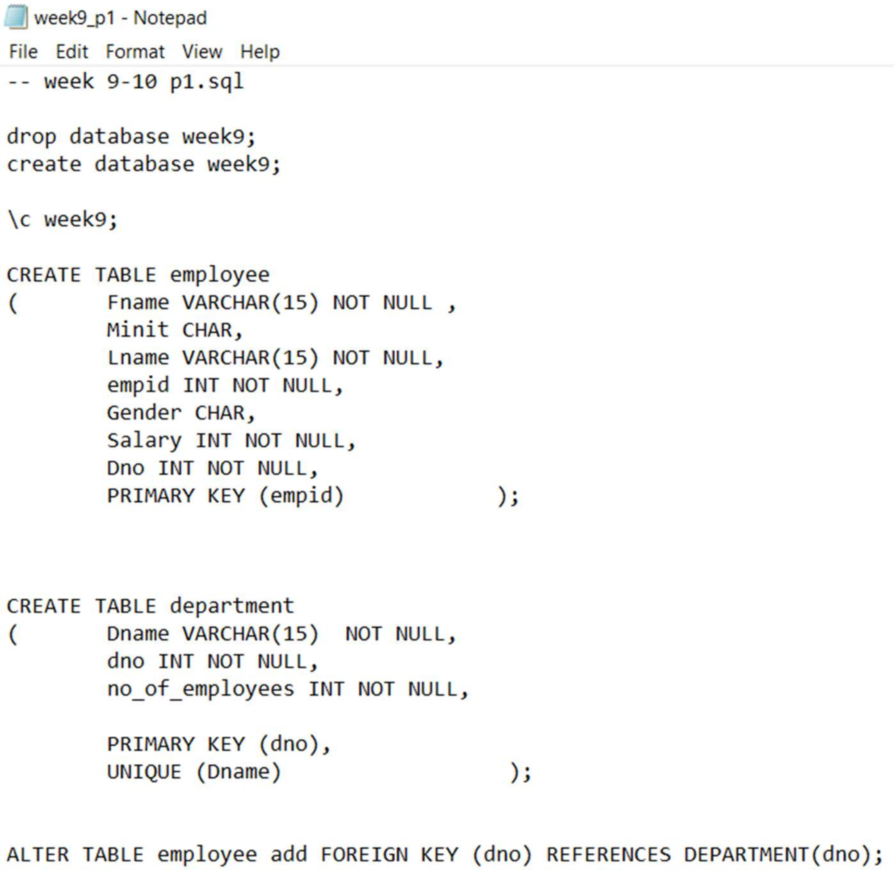
WEEK 9-10

Name – B.Pravena SRN – PES2UG19CS076
Section – 5B

SQL – Creating Triggers and Functions

1) Create an employee table which contains employee details and the department he works for. Create another table department consisting of dname and number of employees. Write triggers to increment or decrement the number of employees in a department table when the record in the employee table is inserted or deleted respectively.

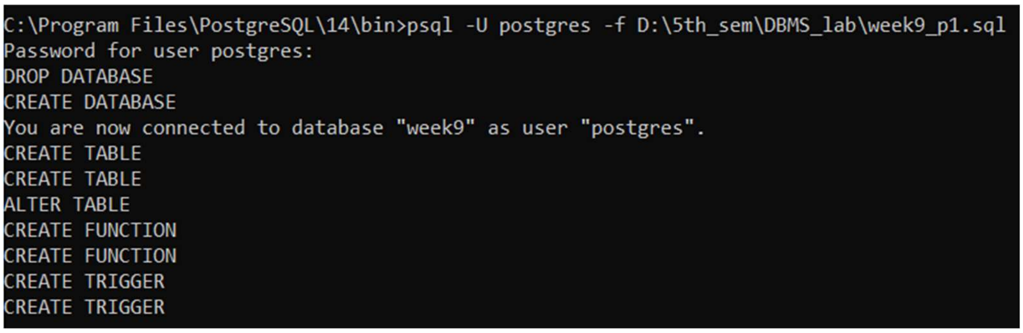
CODE -:



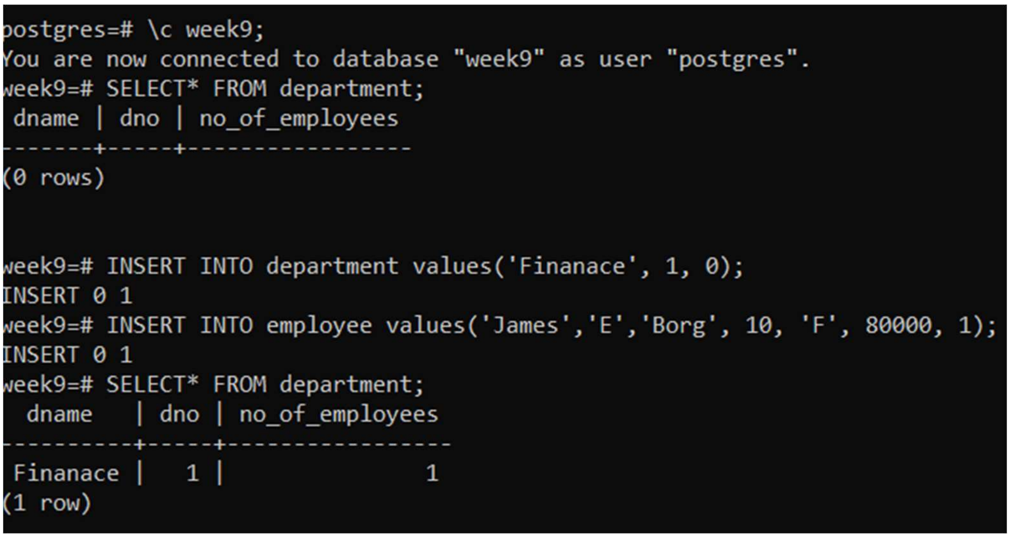
```
-- week 9-10 p1.sql

drop database week9;
create database week9;

\c week9;

CREATE TABLE employee
(       Fname VARCHAR(15) NOT NULL ,
        Minit CHAR,
        Lname VARCHAR(15) NOT NULL,
        empid INT NOT NULL,
        Gender CHAR,
        Salary INT NOT NULL,
        Dno INT NOT NULL,
        PRIMARY KEY (empid)            );



CREATE TABLE department
(       Dname VARCHAR(15)  NOT NULL,
        dno INT NOT NULL,
        no_of_employees INT NOT NULL,

        PRIMARY KEY (dno),
        UNIQUE (Dname)                  );


ALTER TABLE employee add FOREIGN KEY (dno) REFERENCES DEPARTMENT(dno);
```

```
CREATE FUNCTION ins_emp()
        RETURNS trigger as $$
        BEGIN
                UPDATE department
                SET no_of_employees = no_of_employees + 1
                WHERE dno = NEW.dno;
                RETURN NEW;
        END;
        $$
        LANGUAGE 'plpgsql';

        CREATE FUNCTION del_emp()
        RETURNS trigger as $$
        BEGIN
                UPDATE department
                SET no_of_employees = no_of_employees - 1
                WHERE dno = OLD.dno;
                RETURN OLD;
        END;
        $$
        LANGUAGE 'plpgsql';

-- Creating triggers

CREATE TRIGGER insertEmployee
        AFTER INSERT ON employee
        FOR EACH ROW
        EXECUTE PROCEDURE ins_emp();

CREATE TRIGGER deleteEmployee
        BEFORE DELETE ON employee
        FOR EACH ROW
        EXECUTE PROCEDURE del_emp();
```

OUTPUT -:



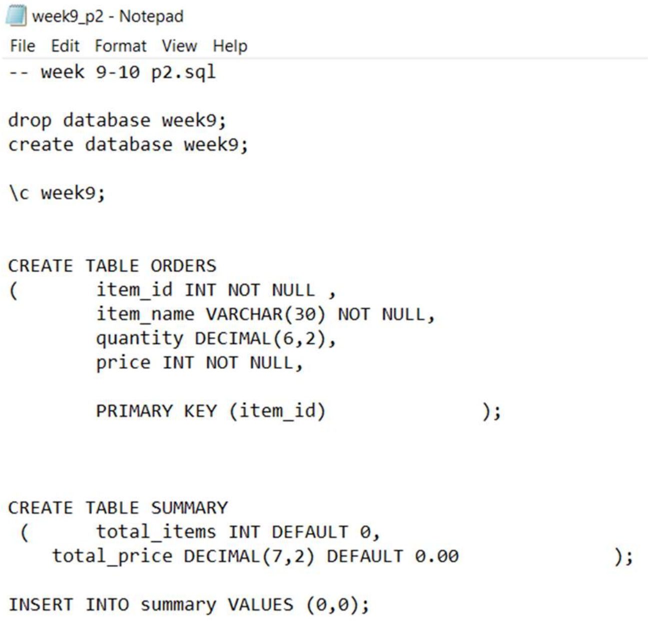
```
C:\Program Files\PostgreSQL\14\bin>psql -U postgres -f D:\5th_sem\DBMS_lab\week9_p1.sql
Password for user postgres:
DROP DATABASE
CREATE DATABASE
You are now connected to database "week9" as user "postgres".
CREATE TABLE
CREATE TABLE
ALTER TABLE
CREATE FUNCTION
CREATE FUNCTION
CREATE TRIGGER
CREATE TRIGGER
```



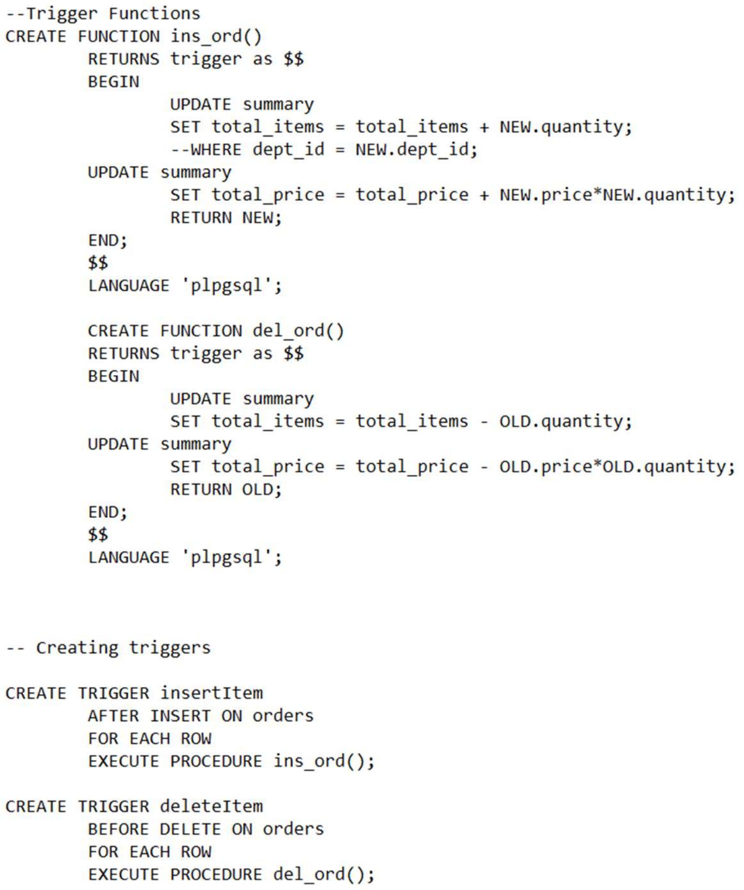
```
postgres=# \c week9;
You are now connected to database "week9" as user "postgres".
week9=# SELECT* FROM department;
 dname | dno | no_of_employees
-------+-----+-----------------
(0 rows)


week9=# INSERT INTO department values('Finanace', 1, 0);
INSERT 0 1
week9=# INSERT INTO employee values('James','E','Borg', 10, 'F', 80000, 1);
INSERT 0 1
week9=# SELECT* FROM department;
  dname   | dno | no_of_employees
----------+-----+-----------------
 Finanace |   1 |               1
(1 row)
```

2) Create an order_item table which contains details like name, quantity and unit price of every item purchased. Create an order summary table that contains number of items and total price. Create triggers to update entry in order summary whenever an item is inserted or deleted in the order item table.

CODE -:

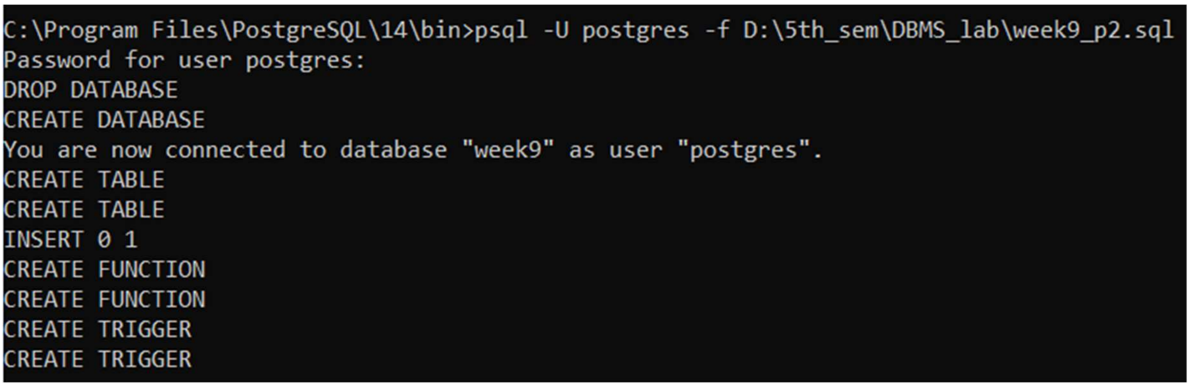


```
-- week 9-10 p2.sql

drop database week9;
create database week9;

\c week9;


CREATE TABLE ORDERS
(       item_id INT NOT NULL ,
        item_name VARCHAR(30) NOT NULL,
        quantity DECIMAL(6,2),
        price INT NOT NULL,

        PRIMARY KEY (item_id)              );



CREATE TABLE SUMMARY
 (      total_items INT DEFAULT 0,
    total_price DECIMAL(7,2) DEFAULT 0.00              );

INSERT INTO summary VALUES (0,0);
```

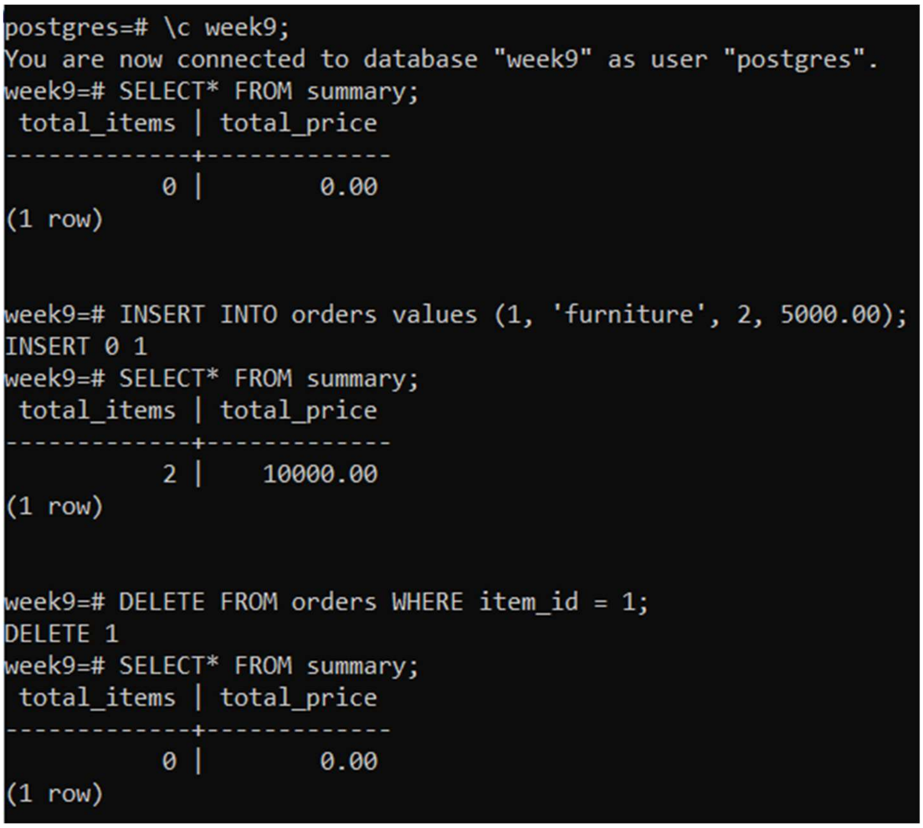


```
--Trigger Functions
CREATE FUNCTION ins_ord()
        RETURNS trigger as $$
        BEGIN
                UPDATE summary
                SET total_items = total_items + NEW.quantity;
                --WHERE dept_id = NEW.dept_id;
        UPDATE summary
                SET total_price = total_price + NEW.price*NEW.quantity;
                RETURN NEW;
        END;
        $$
        LANGUAGE 'plpgsql';

        CREATE FUNCTION del_ord()
        RETURNS trigger as $$
        BEGIN
                UPDATE summary
                SET total_items = total_items - OLD.quantity;
        UPDATE summary
                SET total_price = total_price - OLD.price*OLD.quantity;
                RETURN OLD;
        END;
        $$
        LANGUAGE 'plpgsql';


-- Creating triggers

CREATE TRIGGER insertItem
        AFTER INSERT ON orders
        FOR EACH ROW
        EXECUTE PROCEDURE ins_ord();

CREATE TRIGGER deleteItem
        BEFORE DELETE ON orders
        FOR EACH ROW
        EXECUTE PROCEDURE del_ord();
```

OUTPUT -:



```
C:\Program Files\PostgreSQL\14\bin>psql -U postgres -f D:\5th_sem\DBMS_lab\week9_p2.sql
Password for user postgres:
DROP DATABASE
CREATE DATABASE
You are now connected to database "week9" as user "postgres".
CREATE TABLE
CREATE TABLE
INSERT 0 1
CREATE FUNCTION
CREATE FUNCTION
CREATE TRIGGER
CREATE TRIGGER
```



```
postgres=# \c week9;
You are now connected to database "week9" as user "postgres".
week9=# SELECT* FROM summary;
 total_items | total_price
-------------+-------------
           0 |        0.00
(1 row)


week9=# INSERT INTO orders values (1, 'furniture', 2, 5000.00);
INSERT 0 1
week9=# SELECT* FROM summary;
 total_items | total_price
-------------+-------------
           2 |    10000.00
(1 row)


week9=# DELETE FROM orders WHERE item_id = 1;
DELETE 1
week9=# SELECT* FROM summary;
 total_items | total_price
-------------+-------------
           0 |        0.00
(1 row)
```