PES UNIVERSITY, RR CAMPUS

DBMS LAB WEEK: 9-10

Name: Ananya Uppal

SRN: PES1UG19CS058

Section: 'A'

TASK1

1. Create database and create table schema

```
drop database company;
create database company;

\c company

CREATE TABLE Employee
 (  Fname VARCHAR(15) NOT NULL ,
    Minit CHAR,
    Lname VARCHAR(15) NOT NULL,
    ID INT NOT NULL ,
    Dept VARCHAR(20) NOT NULL ,
    PRIMARY KEY (ID)
);

CREATE TABLE DEPARTMENT
 (  Dname VARCHAR(15)  NOT NULL,
    Dnumber INT NOT NULL ,
    PRIMARY KEY (Dname)
);

```

```
postgres=# \i 'C:/Users/anany/Desktop/3rd Year College/DBMS Lab/Week9-10/companyddl.sql'
DROP DATABASE
CREATE DATABASE
You are now connected to database "company" as user "postgres".
CREATE TABLE
CREATE TABLE
company=#
```

2. Insert Data in the Database Tables

```
\c company

INSERT into DEPARTMENT values('Research', 4);
INSERT into DEPARTMENT values('Administration', 2);
INSERT into DEPARTMENT values('Headquarters', 2);

INSERT into EMPLOYEE values ('James','E','Borg', '888665555','Research');
INSERT into EMPLOYEE values ('John','B','Smith','123456789','Administration');
INSERT into EMPLOYEE values ('Franklin','T','Wong','333445555','Research');
INSERT into EMPLOYEE values ('Alicia','J','Zelaya','999887777','Headquarters');
INSERT into EMPLOYEE values ('Jennifer','S','Wallace','987654321','Research');
INSERT into EMPLOYEE values ('Ramesh','K','Narayan','666884444','Administration');
INSERT into EMPLOYEE values ('Joyce','A','English','453453453','Research');
INSERT into EMPLOYEE values ('Ahmed','V','Jabbar','987987987','Headquarters');
```

```
company=# \i 'C:/Users/anany/Desktop/3rd Year College/DBMS Lab/Week9-10/company_insert.sql'
You are now connected to database "company" as user "postgres".
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

3. Run command to trigger updates

```
-- Question 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.

CREATE FUNCTION INCREASE() returns trigger
    language plpgsql
    as
    $$
    begin
    update DEPARTMENT SET DNUMBER = DNUMBER + 1
    WHERE Dname = NEW.Dname;
        return NEW;
    end;
    $$;

CREATE TRIGGER increment
    AFTER INSERT
    ON EMPLOYEE
    FOR EACH ROW
    EXECUTE PROCEDURE INCREASE();

-- commands to test
SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
INSERT into EMPLOYEE values ('Ananya','M','Uppal', '888661234','Research');
SELECT * FROM DEPARTMENT;
SELECT * FROM EMPLOYEE;
```

```
company=# \i 'C:/Users/anany/Desktop/3rd Year College/DBMS Lab/Week9-10/commands.sql'
CREATE FUNCTION
CREATE TRIGGER
     dname       | dnumber
-----------------+---------
 Research        |       4
 Administration  |       2
 Headquarters    |       2
(3 rows)


  fname   | minit |  lname   |    id     |     dname
----------+-------+----------+-----------+----------------
 James    | E     | Borg     | 888665555 | Research
 John     | B     | Smith    | 123456789 | Administration
 Franklin | T     | Wong     | 333445555 | Research
 Alicia   | J     | Zelaya   | 999887777 | Headquarters
 Jennifer | S     | Wallace  | 987654321 | Research
 Ramesh   | K     | Narayan  | 666884444 | Administration
 Joyce    | A     | English  | 453453453 | Research
 Ahmed    | V     | Jabbar   | 987987987 | Headquarters
(8 rows)


INSERT 0 1
     dname       | dnumber
-----------------+---------
 Administration  |       2
 Headquarters    |       2
 Research        |       5
(3 rows)


  fname   | minit |  lname   |    id     |     dname
----------+-------+----------+-----------+----------------
 James    | E     | Borg     | 888665555 | Research
 John     | B     | Smith    | 123456789 | Administration
 Franklin | T     | Wong     | 333445555 | Research
 Alicia   | J     | Zelaya   | 999887777 | Headquarters
 Jennifer | S     | Wallace  | 987654321 | Research
 Ramesh   | K     | Narayan  | 666884444 | Administration
 Joyce    | A     | English  | 453453453 | Research
 Ahmed    | V     | Jabbar   | 987987987 | Headquarters
 Ananya   | M     | Uppal    | 888661234 | Research
```

TASK2

1. Create Database and Tables.

```sql
drop database orders;
create database orders;

\c orders


CREATE TABLE ORDER_ITEM
(
    NAME VARCHAR(30) NOT NULL,
    QUANTITY INT ,
    ITEM_PRICE INT NOT NULL,
    PRIMARY KEY (NAME)
);

CREATE TABLE SUMMARY
(
    ITEMS INT DEFAULT 0,
    TOTAL_PRICE INT DEFAULT 0
);

```

```
postgres=# \i 'C:/Users/anany/Desktop/3rd Year College/DBMS Lab/Week9-10/task2_ddl.sql'
DROP DATABASE
CREATE DATABASE
You are now connected to database "orders" as user "postgres".
CREATE TABLE
CREATE TABLE
orders=#
```

2. Insert Data in the Tables.

```sql
\c orders;

INSERT INTO ORDER_ITEM values('Pen' , 20 , 10);
INSERT INTO ORDER_ITEM values('Notebook' , 50 , 100);
INSERT INTO ORDER_ITEM values('Pencil' , 20 , 5);

INSERT INTO SUMMARY values(0, 0);

```

```
orders=# \i 'C:/Users/anany/Desktop/3rd Year College/DBMS Lab/Week9-10/task2_data.sql'
You are now connected to database "orders" as user "postgres".
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
orders=#
```

3. Run commands to trigger updates.

```
CREATE FUNCTION UPD() RETURNS TRIGGER
    language 'plpgsql'
    as
    $$
    begin
        UPDATE SUMMARY
        SET ITEMS = ITEMS + NEW.QUANTITY;
        UPDATE SUMMARY
        SET TOTAL_PRICE = TOTAL_PRICE + NEW.ITEM_PRICE*NEW.QUANTITY;
        RETURN NEW;
    end;
    $$;

CREATE FUNCTION DEL() RETURNS TRIGGER
    language 'plpgsql'
    as
    $$
    begin
    UPDATE SUMMARY
    SET ITEMS = ITEMS - OLD.QUANTITY;
    UPDATE SUMMARY
    SET TOTAL_PRICE = TOTAL_PRICE - OLD.ITEM_PRICE*OLD.QUANTITY;
    RETURN OLD;
    end;
    $$;
```

```sql
CREATE TRIGGER UPDATE
    AFTER INSERT
    ON ORDER_ITEM
    FOR EACH ROW
    EXECUTE PROCEDURE UPD();

CREATE TRIGGER DELETE
    AFTER DELETE
    ON ORDER_ITEM
    FOR EACH ROW
    EXECUTE PROCEDURE DEL();

SELECT * FROM ORDER_ITEM;
SELECT * FROM SUMMARY;

INSERT INTO ORDER_ITEM values('Stickers' , 100 , 5);

SELECT * FROM ORDER_ITEM;
SELECT * FROM SUMMARY;
```

```
orders=# \i 'C:/Users/anany/Desktop/3rd Year College/DBMS Lab/Week9-10/commands_task2.sql'
CREATE FUNCTION
CREATE FUNCTION
CREATE TRIGGER
CREATE TRIGGER
   name    | quantity | item_price
-----------+----------+------------
 Pen       |       20 |         10
 Notebook  |       50 |        100
 Pencil    |       20 |          5
(3 rows)


 items | total_price
-------+-------------
     0 |           0
(1 row)


INSERT 0 1
   name    | quantity | item_price
-----------+----------+------------
 Pen       |       20 |         10
 Notebook  |       50 |        100
 Pencil    |       20 |          5
 Stickers  |      100 |          5
(4 rows)


 items | total_price
-------+-------------
   100 |         500
(1 row)
```