

## UE19CS304 – DBMS LABORATORY

SRN: PES1UG19CS545

Name: TUSHAR Y S

### Week 9-10: 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.

- create.sql file containing functions and triggers.

```
create database tusharys_cs545;
\c tusharys_cs545
create table department(
    Dname varchar(15),
    dnumber int not null,
    primary key (Dname)
);

create table employee(
    emp_id int not null,
    Fname varchar(10) not null,
    lname varchar(10) not null,
    Dname varchar(15) not null,
    primary key(emp_id),
    foreign key (Dname ) references department(Dname)
);
```

```

CREATE FUNCTION insert_emp()
returns trigger as $$
BEGIN
    UPDATE department
    SET dnumber= dnumber+1
    where Dname= NEW.Dname;
    return NEW;
END;
$$
LANGUAGE 'plpgsql';

CREATE FUNCTION delete_emp()
returns trigger as $$
BEGIN
    UPDATE department
    SET dnumber= dnumber-1
    where Dname= OLD.Dname;
    return OLD;
END;
$$
LANGUAGE 'plpgsql';

CREATE TRIGGER insertEMPLOYEE
AFTER INSERT ON employee
for each ROW
EXECUTE PROCEDURE insert_emp();

CREATE TRIGGER deleteEMPLOYEE
BEFORE DELETE ON employee
for each ROW
EXECUTE PROCEDURE delete_emp();

```

- insert.sql file containing insert statements.

```

C: > Users > LENOVO > Desktop > 1 > insert.sql
1  \c tusharys_cs545
2  insert into department values ('CSE',0);
3  insert into department values ('ECE',0);
4  insert into department values ('BIO',0);
5

```

```

C:\Program Files\PostgreSQL\13\bin>psql -U postgres -f C:\Users\LENOVO\Desktop\create.sql
Password for user postgres:
CREATE DATABASE
You are now connected to database "tusharys_cs545" as user "postgres".
CREATE TABLE
CREATE TABLE
CREATE FUNCTION
CREATE FUNCTION
CREATE TRIGGER
CREATE TRIGGER

C:\Program Files\PostgreSQL\13\bin>psql -U postgres -f C:\Users\LENOVO\Desktop\insert.sql
Password for user postgres:
You are now connected to database "tusharys_cs545" as user "postgres".
INSERT 0 1
INSERT 0 1
INSERT 0 1

C:\Program Files\PostgreSQL\13\bin>

```

- All fields are initially set to 0 since no record is inserted initially.

```

postgres=# \c tusharys_cs545;
You are now connected to database "tusharys_cs545" as user "postgres".
tusharys_cs545=# select * from department;
 dname | dnumber
-----+-----
 CSE   |      0
 ECE   |      0
 BIO   |      0
(3 rows)

tusharys_cs545=# select * from employee;
 emp_id | fname | lname | dname
-----+-----+-----+-----
(0 rows)

tusharys_cs545=#

```

- Triggers to increment number of employees in department table when the record in employee table is inserted.

Adding an employee to CSE department:

```
insert into EMPLOYEE values(1,'Virat','Kohli','CSE');
```

```
tusharys_cs545=# insert into EMPLOYEE values(1,'Virat','Kohli','CSE');
INSERT 0 1
tusharys_cs545=# select * from employee;
 emp_id | fname | lname | dname
-----+-----+-----+-----
       1 | Virat | Kohli | CSE
(1 row)

tusharys_cs545=# select * from department;
 dname | dnumber
-----+-----
 ECE   |       0
 BIO   |       0
 CSE   |       1
(3 rows)

tusharys_cs545=#
```

Adding an employee to ECE department:

```
tusharys_cs545=# insert into EMPLOYEE values(2,'MS','Dhoni','ECE');
INSERT 0 1
tusharys_cs545=# select * from department;
 dname | dnumber
-----+-----
 BIO   |       0
 CSE   |       1
 ECE   |       1
(3 rows)

tusharys_cs545=# select * from employee;
 emp_id | fname | lname | dname
-----+-----+-----+-----
       1 | Virat | Kohli | CSE
       2 | MS    | Dhoni | ECE
(2 rows)

tusharys_cs545=#
```

Adding an employee to BIO department:

```
tusharys_cs545=# insert into EMPLOYEE values(3,'KL','Rahul','BIO');
INSERT 0 1
tusharys_cs545=# select * from department;
 dname | dnumber
-----+-----
 CSE   |       1
 ECE   |       1
 BIO   |       1
(3 rows)

tusharys_cs545=#
```

Final data in department table:

```
tusharys_cs545=# insert into EMPLOYEE values(4,'ABD','Villears','CSE');
INSERT 0 1
tusharys_cs545=# insert into EMPLOYEE values(5,'Faf','Duplessis','ECE');
INSERT 0 1
tusharys_cs545=# insert into EMPLOYEE values(6,'S','Iyer','BIO');
INSERT 0 1
tusharys_cs545=# select * from department;
  dname | dnumber
-----+-----
  CSE   |       2
  ECE   |       2
  BIO   |       2
(3 rows)

tusharys_cs545=#
```

- Triggers to decrement number of employees in department table when the record in employee table is deleted.

Deleting an employee with employee id =5:

```
tusharys_cs545=# delete from employee where emp_id=5;
DELETE 1
tusharys_cs545=# select * from department;
  dname | dnumber
-----+-----
  CSE   |       2
  BIO   |       2
  ECE   |       1
(3 rows)

tusharys_cs545=#
```

Deleting an employee with employee id =1:

```
tusharys_cs545=# delete from employee where emp_id=1;
DELETE 1
tusharys_cs545=# select * from department;
  dname | dnumber
-----+-----
  BIO   |        2
  ECE   |        1
  CSE   |        1
(3 rows)

tusharys_cs545=#
```

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.

- File that contains functions and triggers to update entry in order summary table when an item is inserted or deleted in order item table.

```
1  create database tusharys2_cs545;
2  \c tusharys2_cs545
3
4  CREATE TABLE ORDERS
5  (
6    item_id INT NOT NULL ,
7    item_name VARCHAR(30) NOT NULL,
8    quantity DECIMAL(7,2),
9    price INT NOT NULL,
10
11    PRIMARY KEY (item_id)
12  );
13
14  CREATE TABLE SUMMARY
15  (
16    total_items INT DEFAULT 0,
17    total_price DECIMAL(7,2) DEFAULT 0.00
18  );
19  INSERT INTO summary VALUES (0,0);
20
21
22  CREATE FUNCTION insert_order()
23  RETURNS trigger as $$
24  BEGIN
25      UPDATE summary
26      SET total_items = total_items + NEW.quantity;
27
28      UPDATE summary
29      SET total_price = total_price + NEW.price*NEW.quantity;
30      RETURN NEW;
31  END;
32  $$
33  LANGUAGE 'plpgsql';
34
```

```

35 CREATE FUNCTION delete_order()
36 RETURNS trigger as $$
37 BEGIN
38     UPDATE summary
39     SET total_items = total_items - OLD.quantity;
40     UPDATE summary
41     SET total_price = total_price - OLD.price*OLD.quantity;
42     RETURN OLD;
43 END;
44 $$
45 LANGUAGE 'plpgsql';
46
47
48
49 CREATE TRIGGER insertItem
50 AFTER INSERT ON orders
51 FOR EACH ROW
52 EXECUTE PROCEDURE insert_order();
53
54 CREATE TRIGGER deleteItem
55 BEFORE DELETE ON orders
56 FOR EACH ROW
57 EXECUTE PROCEDURE delete_order();

```

```

C:\Program Files\PostgreSQL\13\bin>psql -U postgres -f C:\Users\LENOVO\Desktop\create_ord.sql
Password for user postgres:
CREATE DATABASE
You are now connected to database "tusharys2_cs545" as user "postgres".
CREATE TABLE
CREATE TABLE
INSERT 0 1
CREATE FUNCTION
CREATE FUNCTION
CREATE TRIGGER
CREATE TRIGGER

C:\Program Files\PostgreSQL\13\bin>

```

```

tusharys_cs545=# \c tusharys2_cs545
You are now connected to database "tusharys2_cs545" as user "postgres".
tusharys2_cs545=# \d
      List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | orders    | table | postgres
public | summary   | table | postgres
(2 rows)

```

- Initially orders and summary table doesn't contain anything.

```
tusharys2_cs545=# select * from orders;
 item_id | item_name | quantity | price
-----+-----+-----+-----
(0 rows)

tusharys2_cs545=# select * from summary;
total_items | total_price
-----+-----
          0 |         0.00
(1 row)
```

- There is Update (because of trigger) in summary table when an item is inserted in orders table:

```
tusharys2_cs545=# insert into orders values(1,'Milk',10,20);
INSERT 0 1
tusharys2_cs545=# select * from summary;
total_items | total_price
-----+-----
         10 |       200.00
(1 row)
```

- Similarly when an item is deleted from orders table there is update in summary table:

```
tusharys2_cs545=# delete from orders where item_id=1;
DELETE 1
tusharys2_cs545=# select * from summary;
total_items | total_price
-----+-----
          0 |         0.00
(1 row)
```

```
tusharys2_cs545=#
```