

Name-Achyut Jagini

SRN-PES2UG19CS013

WEEK 9-10 SQL – Creating Triggers and Functions

Write the SQL Triggers and functions for the following using Postgres sql.

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.

```
postgres=# \c company
You are now connected to database "company" as user "postgres".
company=# CREATE TABLE EMP_DETAILS(
company(# Fname VARCHAR(15) NOT NULL,
company(# Minit CHAR,
company(# Lname VARCHAR(15) NOT NULL,
company(# Ssn CHAR(9) UNIQUE NOT NULL,
company(# Dname VARCHAR(15) NOT NULL
company(# );
CREATE TABLE
```

```
company=# CREATE TABLE DEPT(
company(# Dept_name VARCHAR(15) NOT NULL,
company(# no_of_employees INT NOT NULL
company(# );
CREATE TABLE
```

```

company=# CREATE OR REPLACE FUNCTION increment_dept_table()
company-# RETURNS TRIGGER AS
company-# $$
company$# BEGIN
company$#     IF (NEW.Dname NOT IN (SELECT Dept_name FROM DEPT)) THEN
company$#     IF (NEW.Dname IN (SELECT Dname FROM DEPARTMENT)) THEN
company$#         INSERT INTO DEPT VALUES(NEW.Dname,1);
company$#     ELSE
company$#         NULL;
company$#     END IF;
company$# ELSE
company$#     UPDATE DEPT SET no_of_employees=no_of_employees + 1 WHERE Dept_name
company$# =NEW.Dname;
company$#     END IF;
company$#     RETURN NEW;
company$# END;
company$# $$
company-# LANGUAGE 'plpgsql';
CREATE FUNCTION

```

```

company=# CREATE OR REPLACE FUNCTION decrement_dept_table()
company-# RETURNS TRIGGER AS
company-# $$
company$# BEGIN
company$# IF (OLD.Ssn IN (SELECT Ssn FROM EMP_DETAILS)) THEN
company$#     UPDATE DEPT SET no_of_employees=no_of_employees-1 WHERE Dept_name
company$# =(SELECT Dname FROM EMP_DETAILS WHERE Ssn=OLD.Ssn);
company$# ELSE
company$#     UPDATE DEPT SET no_of_employees=no_of_employees-(SELECT COUNT(no_of_employees) FROM DEPT WHERE Dept_name=OLD.Dname);
company$#     END IF;
company$#     RETURN OLD;
company$# END;
company$# $$
company-# LANGUAGE 'plpgsql';
CREATE FUNCTION

```

```

company=# CREATE TRIGGER increment_dept_table
company-# BEFORE INSERT
company-# ON EMP_DETAILS
company-# FOR EACH ROW
company-# EXECUTE PROCEDURE increment_dept_table();
CREATE TRIGGER

```

```

CREATE TRIGGER
company=# CREATE TRIGGER decre_dept_table
company-# BEFORE DELETE
company-# ON EMP_DETAILS
company-# FOR EACH ROW
company-# EXECUTE PROCEDURE decrement_dept_table();
CREATE TRIGGER

```

```

company=#
company=# \echo 'a.Inserting into the table\n'
a.Inserting into the table

company=# SELECT * FROM EMP_DETAILS;
   fname |   init | lname |   ssn |   dname
-----+-----+-----+-----+-----
(0 rows)

company=# SELECT * FROM DEPT;
   dept_name | no_of_employees
-----+-----
(0 rows)

company=# INSERT INTO EMP_DETAILS VALUES('Alcott','N','Kimberley','988234013','Research');
INSERT 0 1
company=# SELECT * FROM DEPT;
   dept_name | no_of_employees
-----+-----
  Research   |                1
(1 row)

company=# INSERT INTO EMP_DETAILS VALUES('ibrahim','N','Mahmood','988234012','Administration');
INSERT 0 1

```

```

company=# SELECT * FROM DEPT;
   dept_name | no_of_employees
-----+-----
  Research   |                0
Administration |                0
(2 rows)

company=# DELETE FROM EMP_DETAILS WHERE Ssn='988234013';
DELETE 0
company=#
company=#
company=# SELECT * FROM DEPT;
   dept_name | no_of_employees
-----+-----
  Research   |                0
Administration |                0
(2 rows)

company=# DELETE FROM EMP_DETAILS WHERE Dname='Administration';
DELETE 0
company=# SELECT * FROM DEPT;+
   dept_name | no_of_employees
-----+-----
  Research   |                0
Administration |                0
(2 rows)

```

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.

```
company=# CREATE TABLE ORDER_ITEM(  
company(# OI_ID INT NOT NULL,  
company(# name VARCHAR(250) NOT NULL,  
company(# Quantity INT NOT NULL,  
company(# Unit_price INT NOT NULL  
company(# );  
CREATE TABLE
```