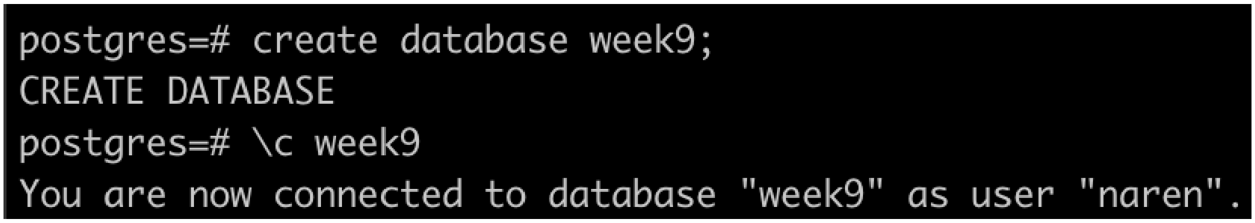
DBMS Week 9-10
A Narendiran
PES1UG19CS001

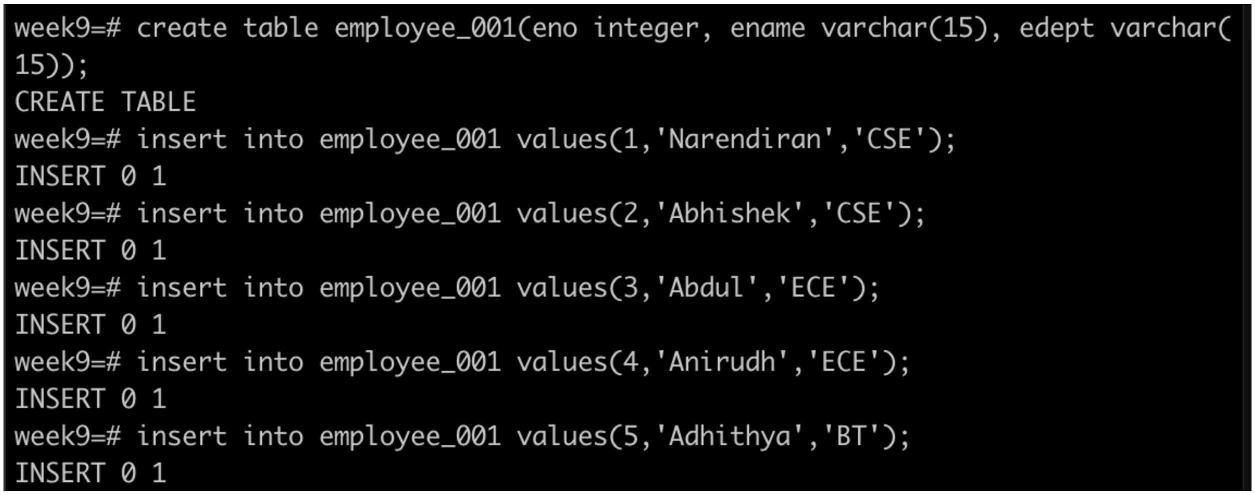
Creating Database



```
postgres=# create database week9;
CREATE DATABASE
postgres=# \c week9
You are now connected to database "week9" as user "naren".
```

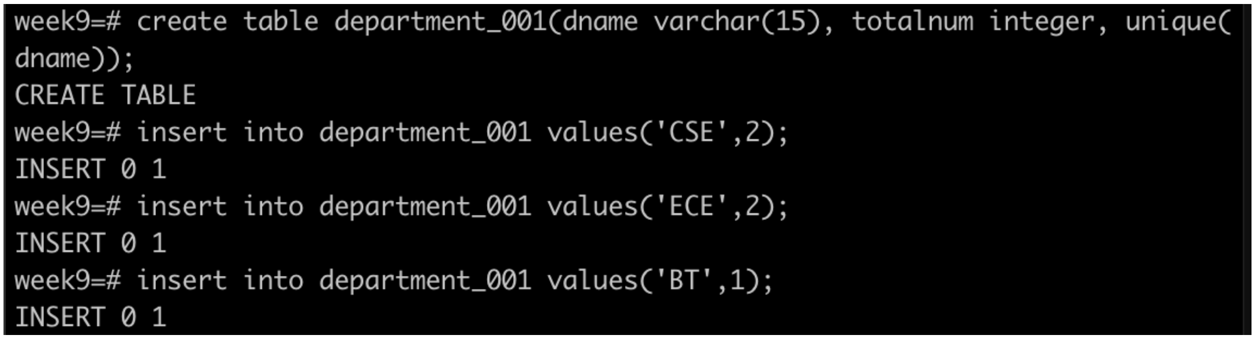
1. Create an employee table which contains employee details and the department he works for. Create another table department consisting of dname and number of employees. Write triggers to increment or decrement the number of employees in a department table when the record in the employee table is inserted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

Creating employee table and inserting values



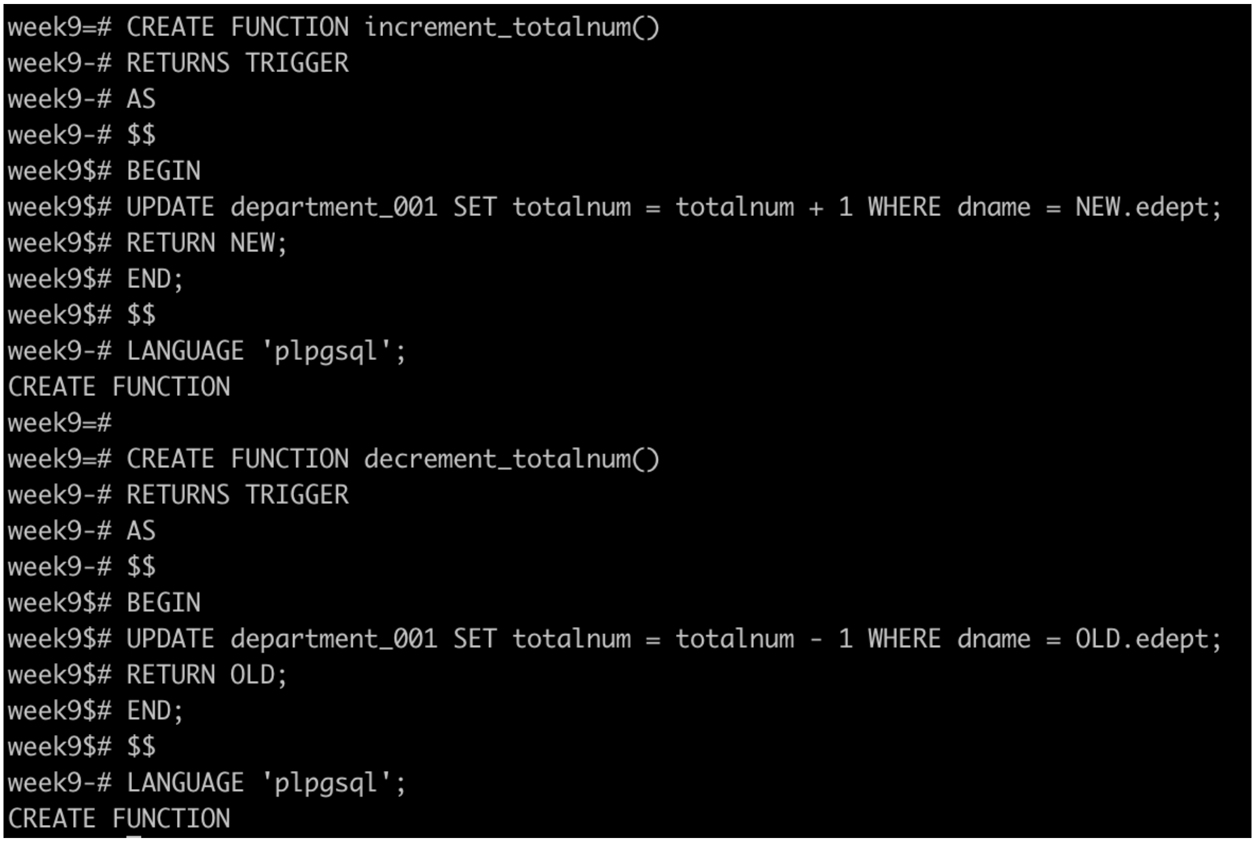
```
week9=# create table employee_001(eno integer, ename varchar(15), edept varchar(
15));
CREATE TABLE
week9=# insert into employee_001 values(1,'Narendiran','CSE');
INSERT 0 1
week9=# insert into employee_001 values(2,'Abhishek','CSE');
INSERT 0 1
week9=# insert into employee_001 values(3,'Abdul','ECE');
INSERT 0 1
week9=# insert into employee_001 values(4,'Anirudh','ECE');
INSERT 0 1
week9=# insert into employee_001 values(5,'Adhithya','BT');
INSERT 0 1
```

Creating department table and inserting values



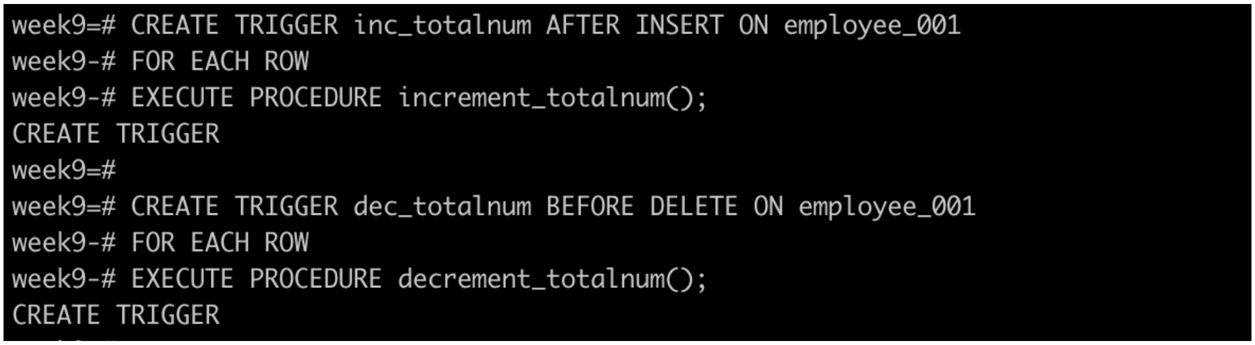
```
week9=# create table department_001(dname varchar(15), totalnum integer, unique(
dname));
CREATE TABLE
week9=# insert into department_001 values('CSE',2);
INSERT 0 1
week9=# insert into department_001 values('ECE',2);
INSERT 0 1
week9=# insert into department_001 values('BT',1);
INSERT 0 1
```

Creating functions for increment and decrement



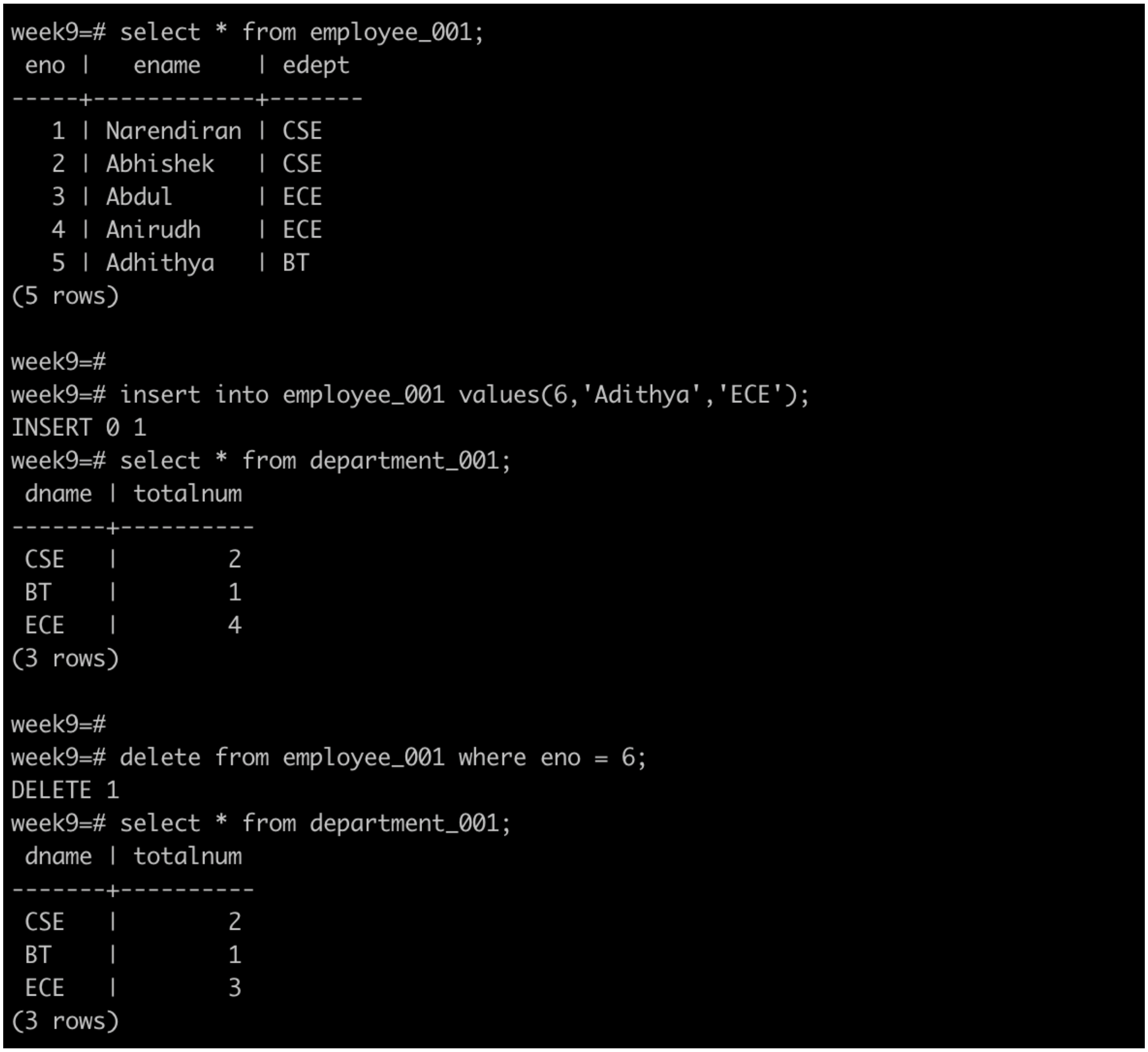
```
week9=# CREATE FUNCTION increment_totalnum()
week9-# RETURNS TRIGGER
week9-# AS
week9-# $$
week9$# BEGIN
week9$# UPDATE department_001 SET totalnum = totalnum + 1 WHERE dname = NEW.edept;
week9$# RETURN NEW;
week9$# END;
week9$# $$
week9-# LANGUAGE 'plpgsql';
CREATE FUNCTION
week9=#
week9=# CREATE FUNCTION decrement_totalnum()
week9-# RETURNS TRIGGER
week9-# AS
week9-# $$
week9$# BEGIN
week9$# UPDATE department_001 SET totalnum = totalnum - 1 WHERE dname = OLD.edept;
week9$# RETURN OLD;
week9$# END;
week9$# $$
week9-# LANGUAGE 'plpgsql';
CREATE FUNCTION
```

Creating Triggers for insert and delete



```
week9=# CREATE TRIGGER inc_totalnum AFTER INSERT ON employee_001
week9-# FOR EACH ROW
week9-# EXECUTE PROCEDURE increment_totalnum();
CREATE TRIGGER
week9=#
week9=# CREATE TRIGGER dec_totalnum BEFORE DELETE ON employee_001
week9-# FOR EACH ROW
week9-# EXECUTE PROCEDURE decrement_totalnum();
CREATE TRIGGER
```

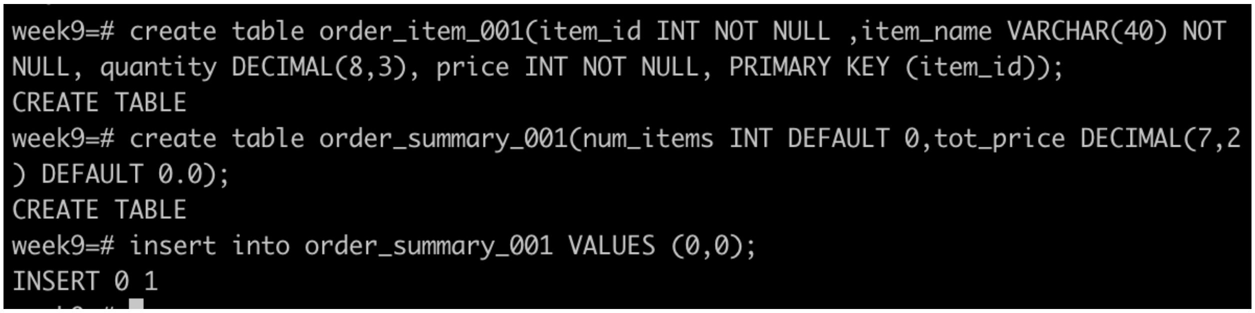
Inserting and deleting values



```
week9=# select * from employee_001;
 eno |   ename    | edept
-----+------------+-------
   1 | Narendiran | CSE
   2 | Abhishek   | CSE
   3 | Abdul      | ECE
   4 | Anirudh    | ECE
   5 | Adhithya   | BT
(5 rows)

week9=#
week9=# insert into employee_001 values(6,'Adithya','ECE');
INSERT 0 1
week9=# select * from department_001;
 dname | totalnum
-------+----------
 CSE   |        2
 BT    |        1
 ECE   |        4
(3 rows)

week9=#
week9=# delete from employee_001 where eno = 6;
DELETE 1
week9=# select * from department_001;
 dname | totalnum
-------+----------
 CSE   |        2
 BT    |        1
 ECE   |        3
(3 rows)
```

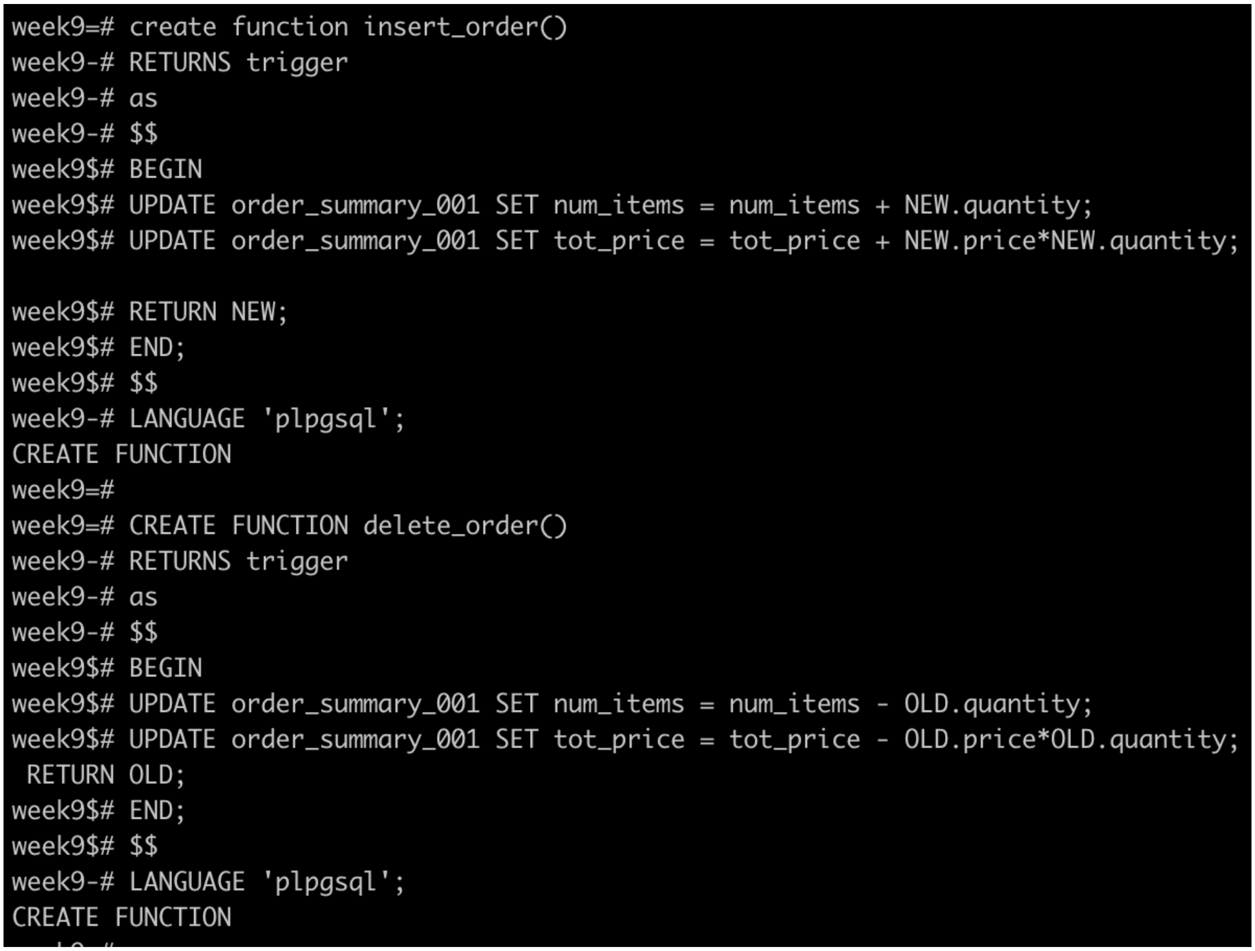
2.Create an order_item table which contains details like name, quantity and unit price of every item purchased. Create an order summary table that contains number of items and total price. Create triggers to update entry in order summary whenever an item is inserted or deleted in the order item table.

Creating Table and inserting value



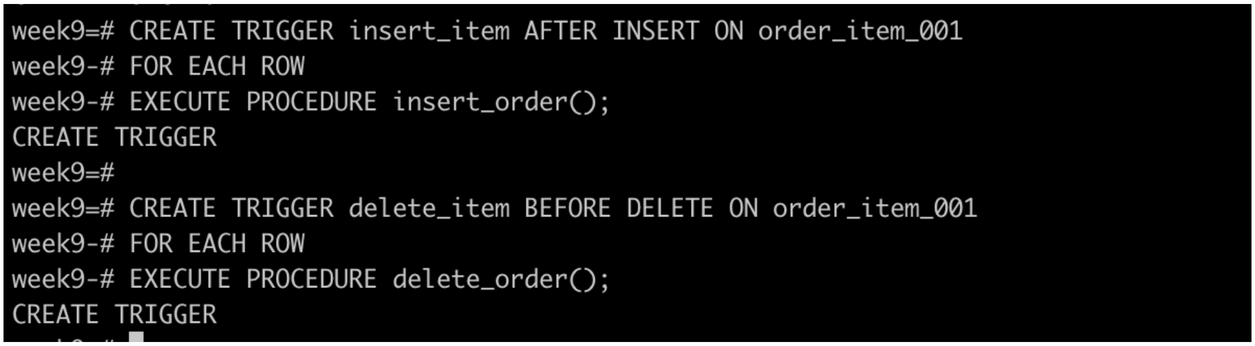
```
week9=# create table order_item_001(item_id INT NOT NULL ,item_name VARCHAR(40) NOT
NULL, quantity DECIMAL(8,3), price INT NOT NULL, PRIMARY KEY (item_id));
CREATE TABLE
week9=# create table order_summary_001(num_items INT DEFAULT 0,tot_price DECIMAL(7,2
) DEFAULT 0.0);
CREATE TABLE
week9=# insert into order_summary_001 VALUES (0,0);
INSERT 0 1
```

Creating Functions



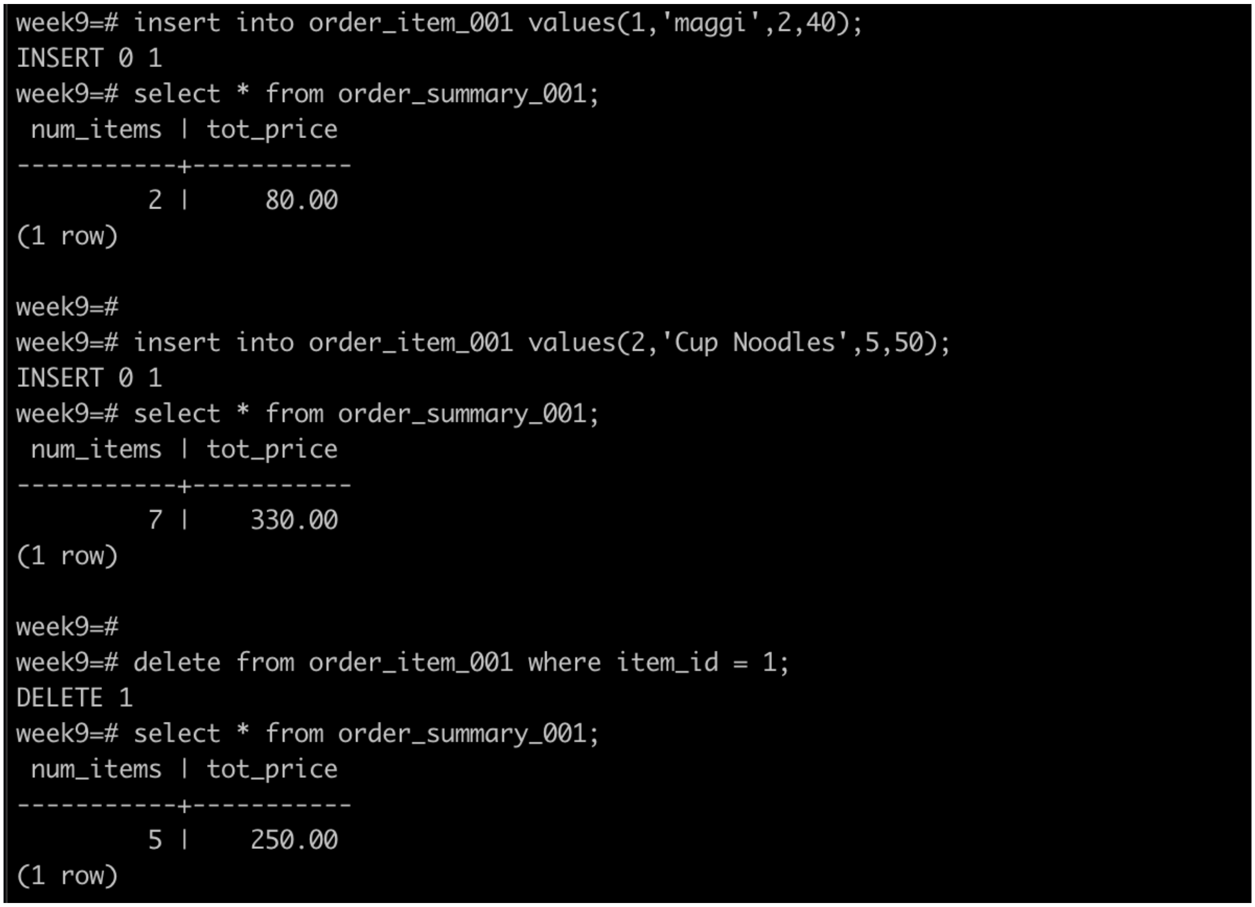
```
week9=# create function insert_order()
week9-# RETURNS trigger
week9-# as
week9-# $$
week9$# BEGIN
week9$# UPDATE order_summary_001 SET num_items = num_items + NEW.quantity;
week9$# UPDATE order_summary_001 SET tot_price = tot_price + NEW.price*NEW.quantity;

week9$# RETURN NEW;
week9$# END;
week9$# $$
week9-# LANGUAGE 'plpgsql';
CREATE FUNCTION
week9=#
week9=# CREATE FUNCTION delete_order()
week9-# RETURNS trigger
week9-# as
week9-# $$
week9$# BEGIN
week9$# UPDATE order_summary_001 SET num_items = num_items - OLD.quantity;
week9$# UPDATE order_summary_001 SET tot_price = tot_price - OLD.price*OLD.quantity;
 RETURN OLD;
week9$# END;
week9$# $$
week9-# LANGUAGE 'plpgsql';
CREATE FUNCTION
```

Creating Triggers



```
week9=# CREATE TRIGGER insert_item AFTER INSERT ON order_item_001
week9-# FOR EACH ROW
week9-# EXECUTE PROCEDURE insert_order();
CREATE TRIGGER
week9=#
week9=# CREATE TRIGGER delete_item BEFORE DELETE ON order_item_001
week9-# FOR EACH ROW
week9-# EXECUTE PROCEDURE delete_order();
CREATE TRIGGER
```

Inserting and Deleting Values



```
week9=# insert into order_item_001 values(1,'maggi',2,40);
INSERT 0 1
week9=# select * from order_summary_001;
 num_items | tot_price
-----------+-----------
         2 |     80.00
(1 row)

week9=#
week9=# insert into order_item_001 values(2,'Cup Noodles',5,50);
INSERT 0 1
week9=# select * from order_summary_001;
 num_items | tot_price
-----------+-----------
         7 |    330.00
(1 row)

week9=#
week9=# delete from order_item_001 where item_id = 1;
DELETE 1
week9=# select * from order_summary_001;
 num_items | tot_price
-----------+-----------
         5 |    250.00
(1 row)
```