SQL – Creating Triggers and Functions

WEEK 9-10

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.**

**Ans.**

**SQL for funcs and triggers:**

--function to increment emp count on a new hire

create or replace function new\_hire\_f()

returns trigger as $example\_table$

BEGIN

update dept

set count\_emp=count\_emp+1

where new.dno=dept.dnumber;

return new;

end;

$example\_table$ language plpgsql;

--Trigger for new hire

create trigger new\_hire

after insert

on emp

for each row

execute procedure new\_hire\_f();

-- function to decrement count\_emp

create or replace function new\_fire\_f()

returns trigger as $example\_table$

BEGIN

update dept

set count\_emp=count\_emp-1

where old.dno=dept.dnumber;

return old;

end;

$example\_table$ language plpgsql;'

--Trigger to deceremnt count once employee entry is deleted

create trigger new\_fire

before delete

on emp

for each row

execute procedure new\_fire\_f();

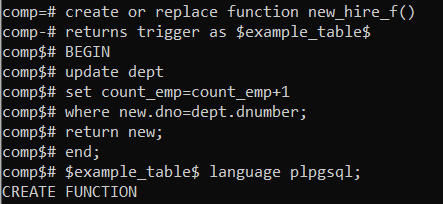
Creating db, creating tables:



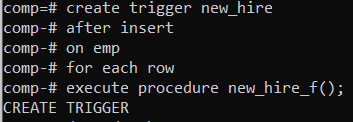




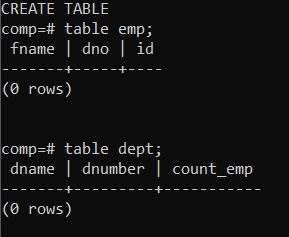
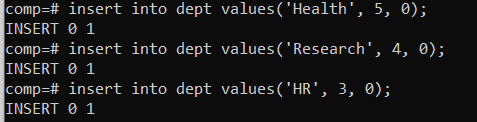
Creating function:

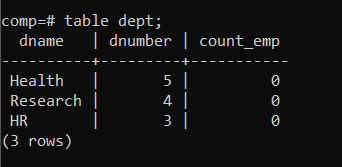


Creating the trigger:

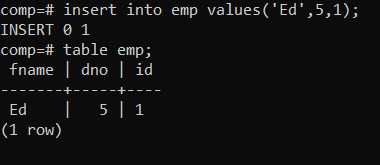
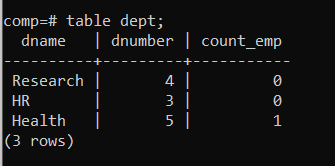


Inserting records into dept table:

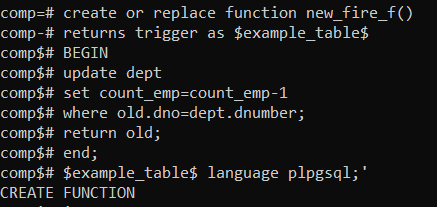
 



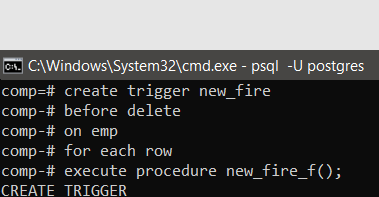
Inserting Values into emp table [increment]:

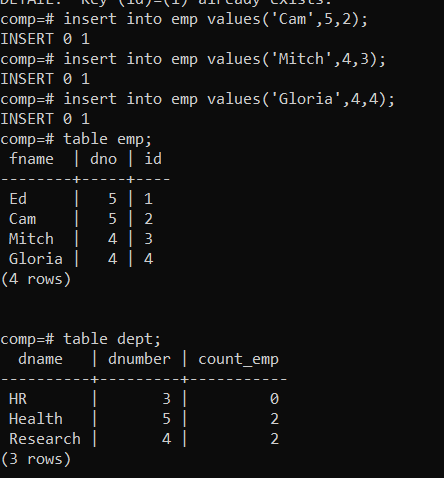
Creating a function for delete:



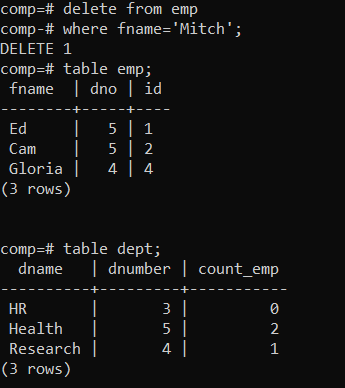
Creating trigger:



Inserting more records:



Deleting a record to check if the trigger works:



It works!

**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.**

Ans:

--Creating function for insert

create or replace function add\_item\_f()

returns trigger as $example\_table$

BEGIN

update order\_summary

set tot\_price=tot\_price+new.qty\*new.unit\_price,

no\_item=no\_item+new.qty;

return new;

end;

$example\_table$ language plpgsql;

--trigger for insert

create trigger new\_item\_added

after insert

on order\_item

for each row

execute procedure add\_item\_f();

--function on delete

create or replace function remove\_item\_f()

returns trigger as $example\_table$

BEGIN

update order\_summary

set tot\_price=tot\_price-old.qty\*old.unit\_price,

no\_item=no\_item-old.qty;

return old;

end;

$example\_table$ language plpgsql;

--trigger on delete

create trigger item\_removed

before delete

on order\_item

for each row

execute procedure remove\_item\_f();

--creating function on update

create or replace function update\_qty\_f()

returns trigger as $example\_table$

BEGIN

update order\_summary

set tot\_price=tot\_price+((new.qty-old.qty)\*old.unit\_price),

no\_item=no\_item-old.qty+new.qty;

return new;

end;

$example\_table$ language plpgsql;

-- Creating Trigger on update

create trigger qty\_changed

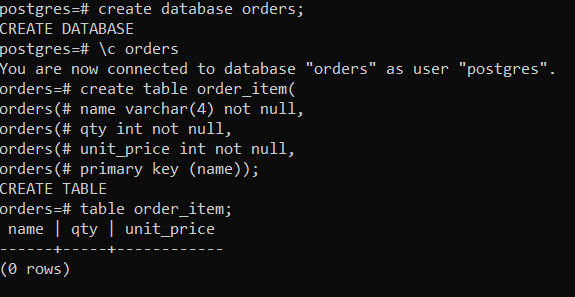
after update

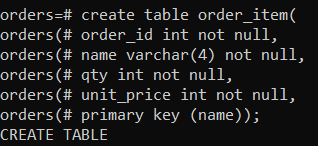
on order\_item

for each row

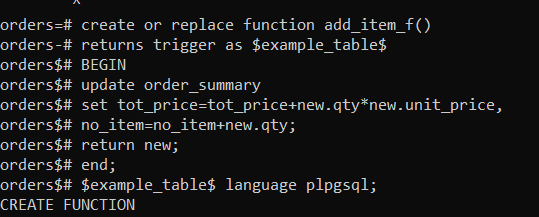
execute procedure remove\_item\_f();

creating db and tables:

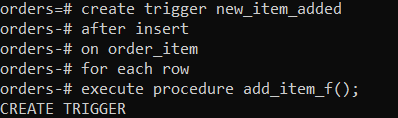




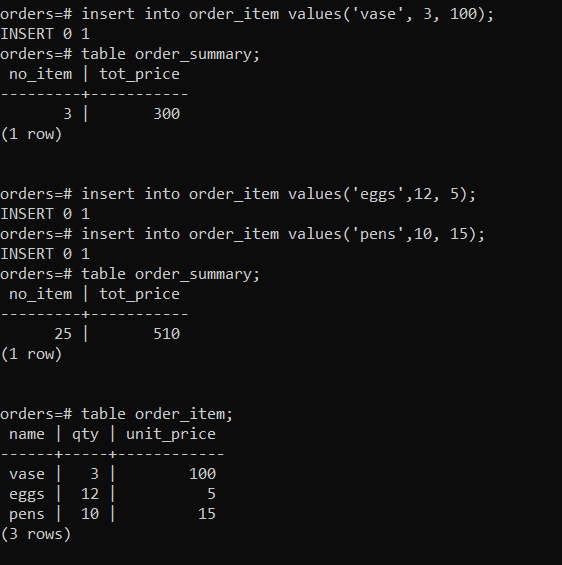
Creating function for insert:



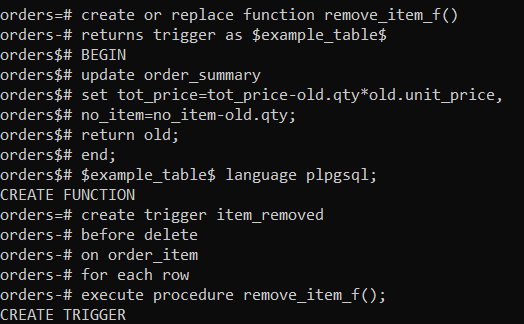
Create trigger for insert:



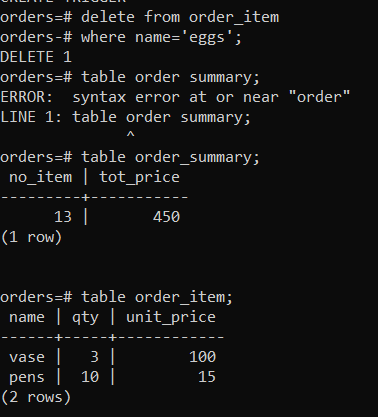
Insert trigger in action:



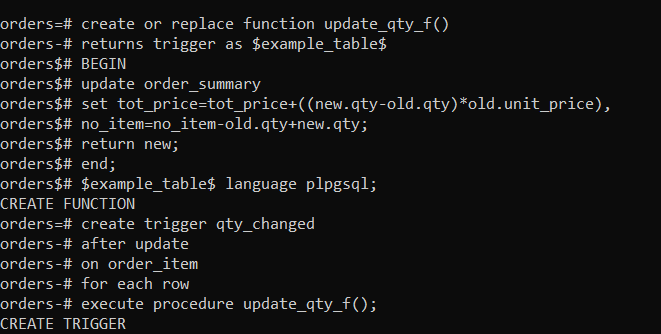
Creating function and trigger for delete:



Delete trigger in action:



Creating a function and trigger for update:



Update trigger in action:

