**Day 8 SQL Bootcamp Assignment**

**1. Create view vw\_updatable\_products (use same query whatever I used in the training)**

**Try updating view with below query and see if the product table also gets updated.**

**Update query:**

**UPDATE updatable\_products SET unit\_price = unit\_price \* 1.1 WHERE units\_in\_stock < 10;**

**QUERY**

**CREATE VIEW vw\_updatable\_products AS**

**SELECT product\_id, product\_name, unit\_price, units\_in\_stock**

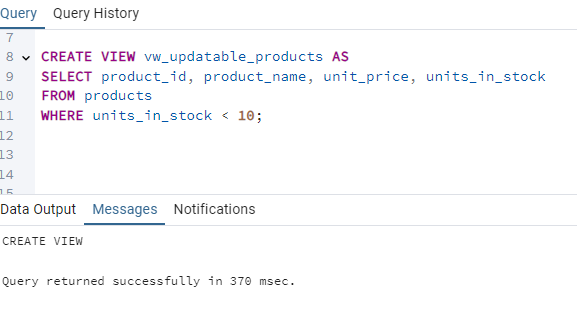
**FROM products**

**WHERE units\_in\_stock < 10;**

**UPDATE products**

**SET unit\_price = unit\_price \* 1.1**

**WHERE units\_in\_stock < 10;**

****

**2. Transaction:**

**Update the product price for products by 10% in category id=1**

**Try COMMIT and ROLLBACK and observe what happens.**

**QUERY**

**BEGIN TRANSACTION;**

**UPDATE products**

**SET unit\_price = unit\_price \* 1.1**

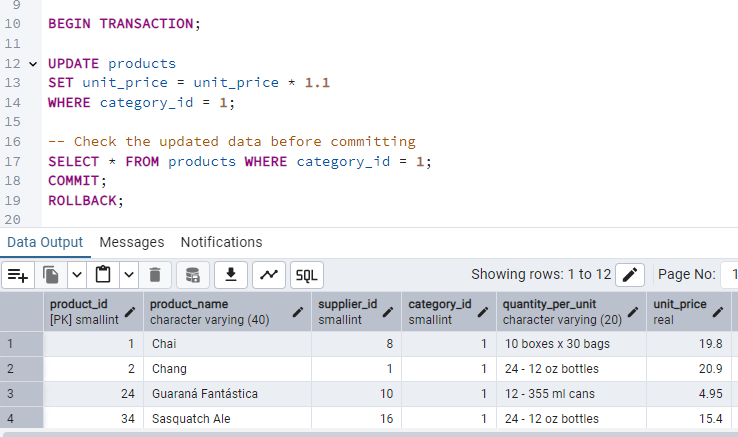
**WHERE category\_id = 1;**

**-- Check the updated data before committing**

**SELECT \* FROM products WHERE category\_id = 1;**

**COMMIT;**

**ROLLBACK;**

****

**3. Create a regular view which will have below details (Need to do joins):**

**Employee\_id,**

**Employee\_full\_name,**

**Title,**

**Territory\_id,**

**territory\_description,**

**region\_description**

**QUERY**

**CREATE VIEW vw\_employee\_territories AS**

**SELECT e.employee\_id,**

**CONCAT(e.first\_name, ' ', e.last\_name) AS employee\_full\_name,**

**e.title,**

**t.territory\_id,**

**t.territory\_description,**

**r.region\_description**

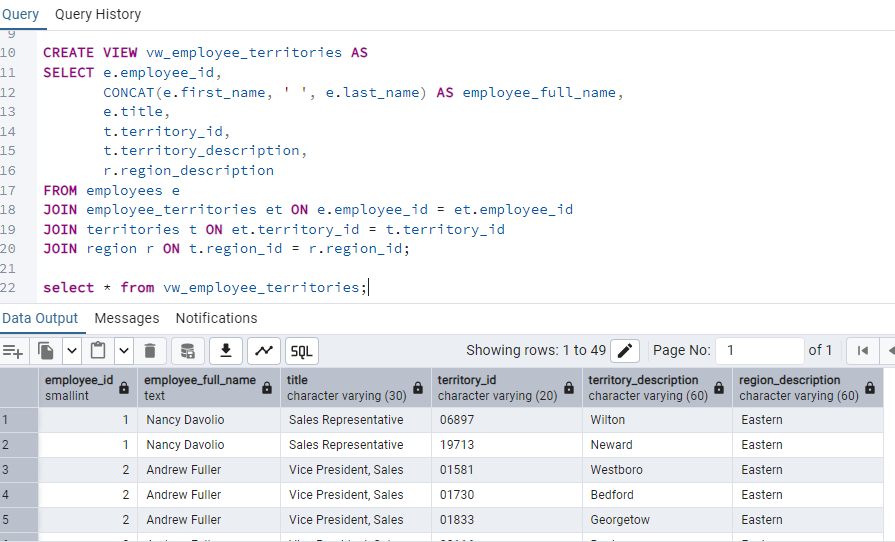
**FROM employees e**

**JOIN employee\_territories et ON e.employee\_id = et.employee\_id**

**JOIN territories t ON et.territory\_id = t.territory\_id**

**JOIN region r ON t.region\_id = r.region\_id;**

**select \* from vw\_employee\_territories;**



**4.**  **Create a recursive CTE based on Employee Hierarchy**

**QUERY**

**with recursive cte\_employees\_hierarchy as**

**( select employee\_id,**

**first\_name || '' || last\_name as employee\_name,**

**reports\_to,**

**1 as level**

**from employees**

**where reports\_to is NULL**

**UNION ALL**

**select e.employee\_id,**

**e.first\_name || '' || e.last\_name as employee\_name,**

**e.reports\_to,eh.level +1**

**from employees e**

**INNER JOIN cte\_employees\_hierarchy eh ON e.reports\_to = eh.employee\_id**

**)**

**SELECT \* FROM cte\_employees\_hierarchy**

**ORDER BY level, reports\_to, employee\_id;**

