Day 8

/\*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; \*/

-- CREATE VIEW

CREATE VIEW vw\_updatable\_products AS

SELECT

product\_id,

product\_name,

unit\_price,

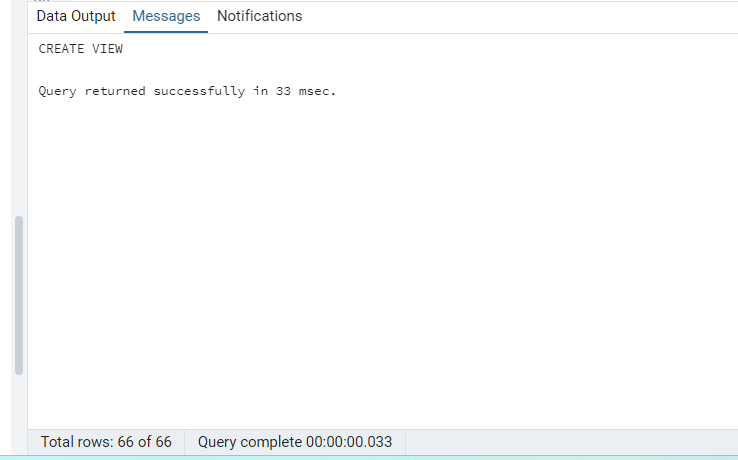
units\_in\_stock,

discontinued

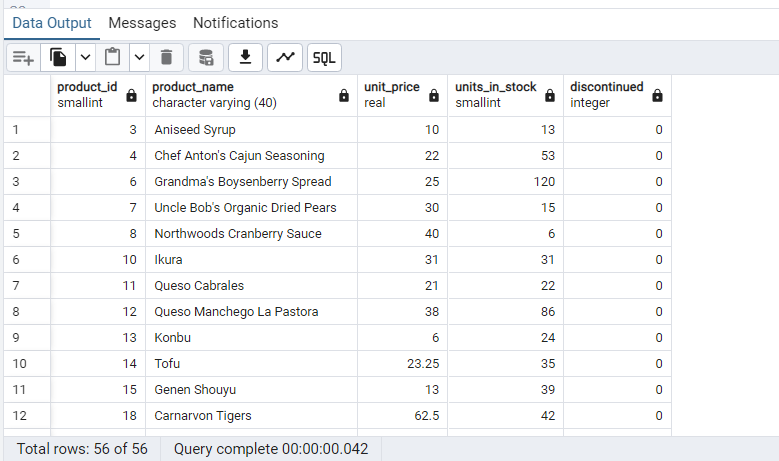
FROM products

where discontinued = 0

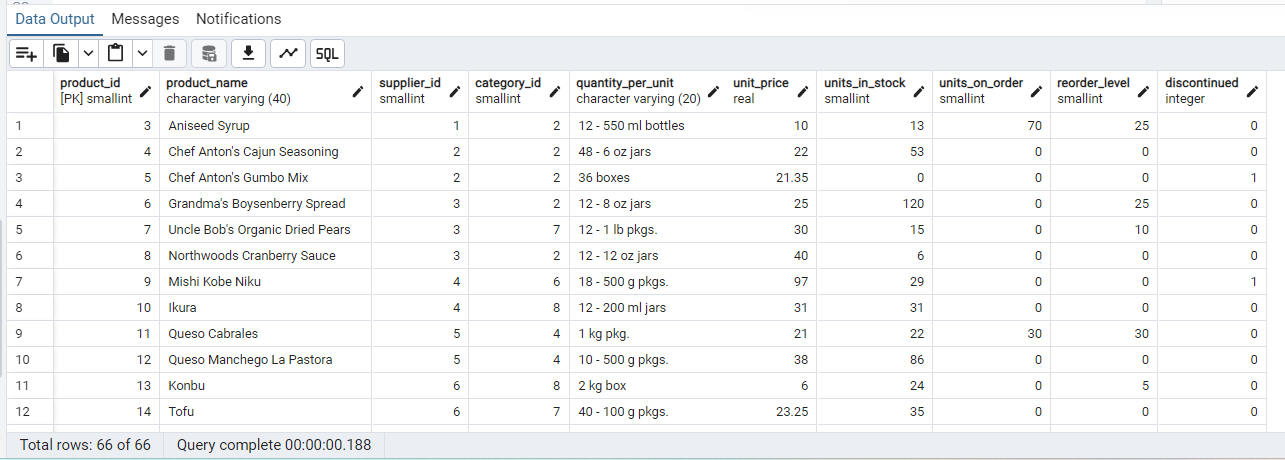
WITH CHECK OPTION;



select \* from vw\_updatable\_products;



select \* from products;



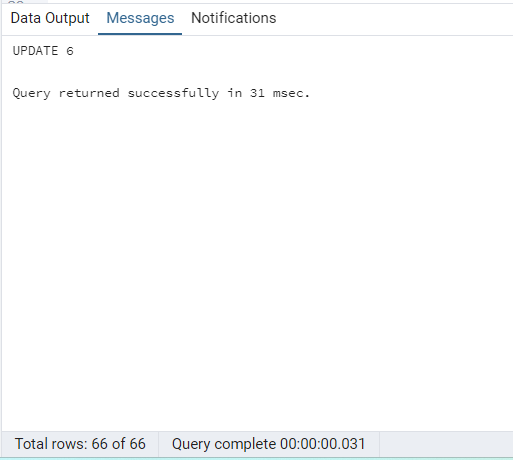
-- After the UPDATE product\_id 8 unit\_price will be increased by 10%

--UPDATE VIEW

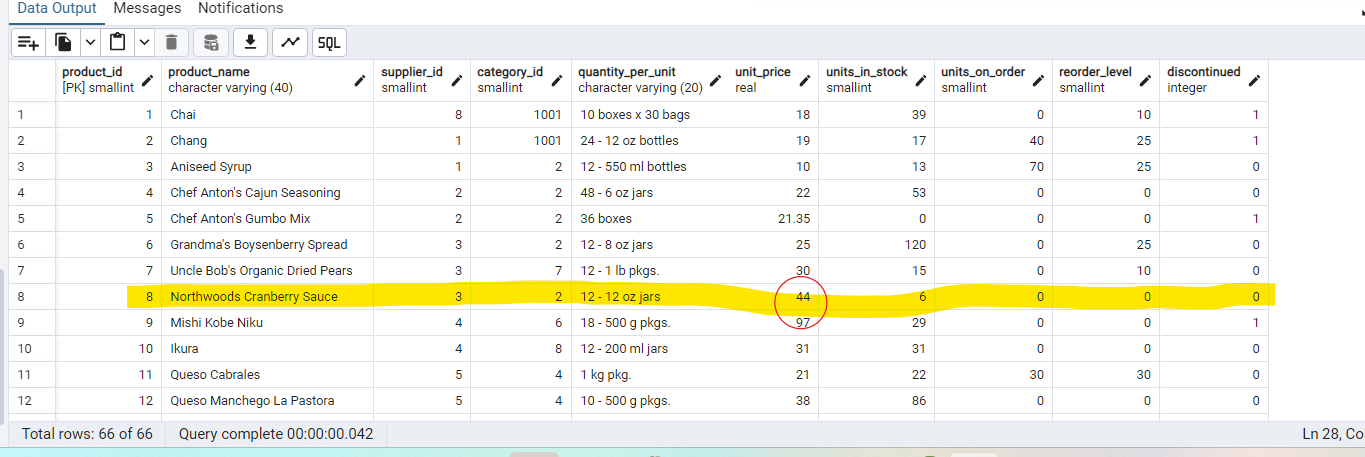
UPDATE vw\_updatable\_products

SET unit\_price = unit\_price \* 1.1

WHERE units\_in\_stock < 10;



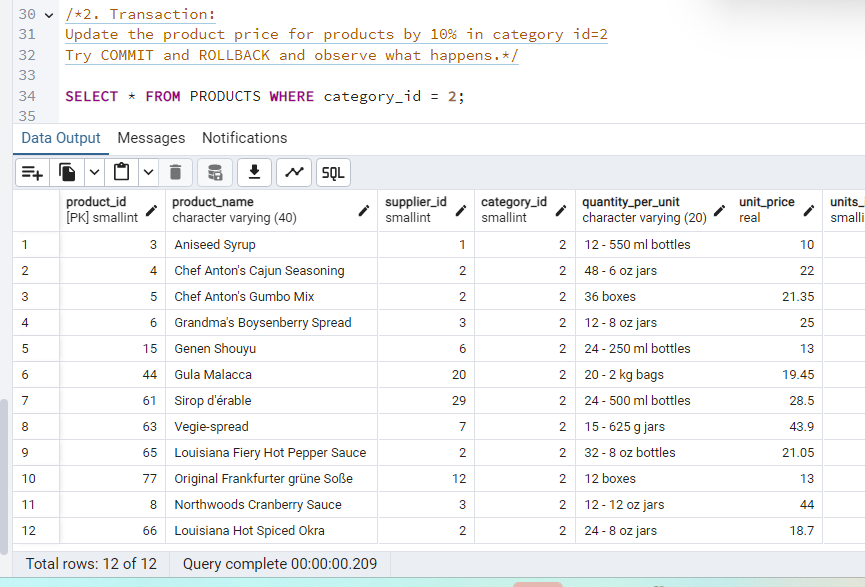
select \* from products order by product\_id;



/\*2. Transaction:

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

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

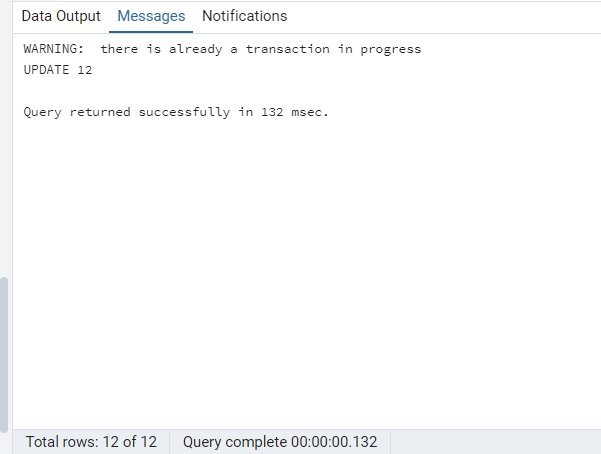


BEGIN;

UPDATE products

SET unit\_price = unit\_price \* 1.10

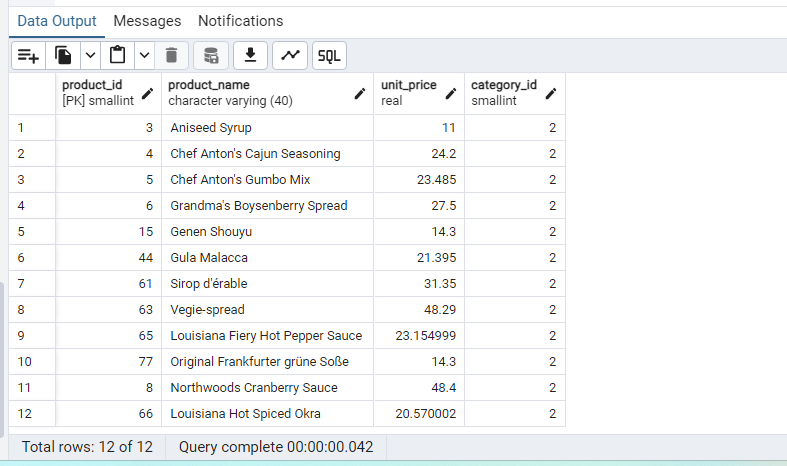
WHERE category\_id = 2;



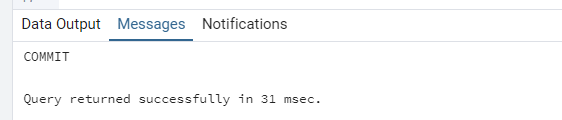
SELECT product\_id, product\_name, unit\_price, category\_id

FROM products

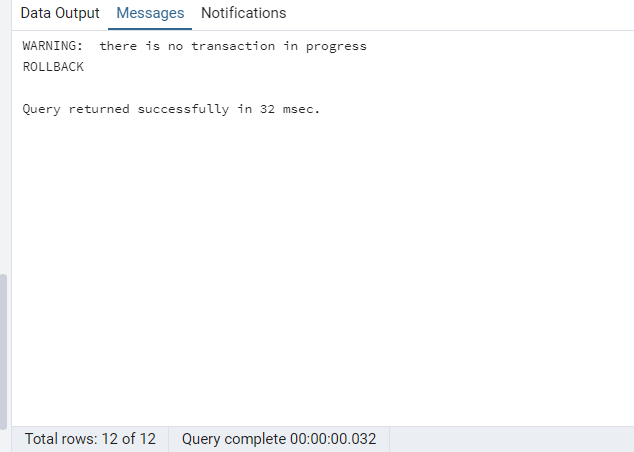
WHERE category\_id = 2;



COMMIT;



ROLLBACK;



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

Employee\_id,

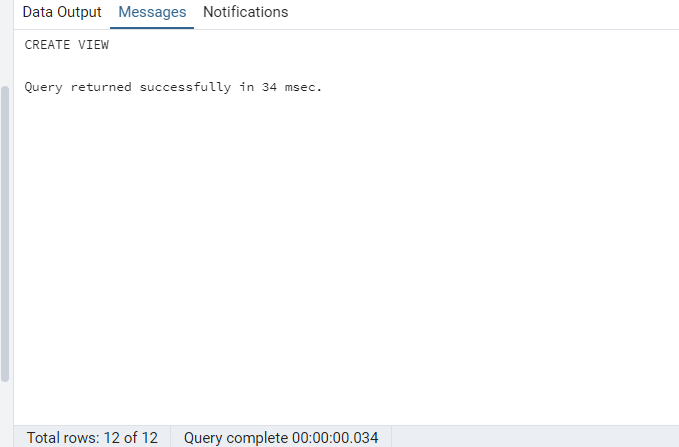
Employee\_full\_name,

Title,

Territory\_id,

territory\_description,

region\_description



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

WITH RECURSIVE employee\_hierarchy AS (

--Top-level managers

SELECT

employee\_id,

first\_name,

last\_name,

title,

reports\_to,

1 AS level,

CONCAT(first\_name, ' ', last\_name) AS hierarchy

FROM employees

WHERE reports\_to IS NULL

UNION ALL

-- Recursive member: Employees reporting to others

SELECT

e.employee\_id,

e.first\_name,

e.last\_name,

e.title,

e.reports\_to,

eh.level + 1,

CONCAT(eh.hierarchy, ' > ', e.first\_name, ' ', e.last\_name)

FROM employees e

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

)

SELECT \*

FROM employee\_hierarchy

ORDER BY hierarchy;

