Day 8 Assignment SQL

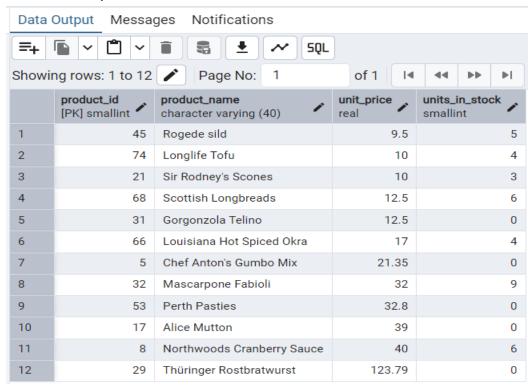
1. Create view vw_updatable_products

Try updating the view with the 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;

From products;



Create View vw_updatable_products As

Select product_id,

product_name,
unit_price,
units_in_stock

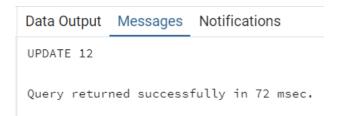
From products;

Data Output Messages Notifications				
= +		1		
Showii	ng rows: 1 to 77	Page No: 1	of 1	
	product_id [PK] smallint	product_name character varying (40)	unit_price real	units_in_stock smallint
1	1	Chai	18	39
2	2	Chang	19	17
3	3	Aniseed Syrup	10	13
4	4	Chef Anton's Cajun Seasoning	22	53
5	5	Chef Anton's Gumbo Mix	21.35	0
6	6	Grandma's Boysenberry Spread	25	120
7	7	Uncle Bob's Organic Dried Pears	30	15
8	8	Northwoods Cranberry Sauce	40	6
9	9	Mishi Kobe Niku	97	29
10	10	Ikura	31	31
11	11	Queso Cabrales	21	22
12	12	Queso Manchego La Pastora	38	86
13	13	Konbu	6	24
14	14	Tofu	23.25	35
15	15	Genen Shouyu	13	39
16	16	Pavlova	17.45	29
17	17	Alice Mutton	39	0
18	18	Carnarvon Tigers	62.5	42
19	19	Teatime Chocolate Biscuits	9.2	25
20	20	Sir Rodney's Marmalade	81	40

-- Update the view to increase the unit price by 10% for products with low stock

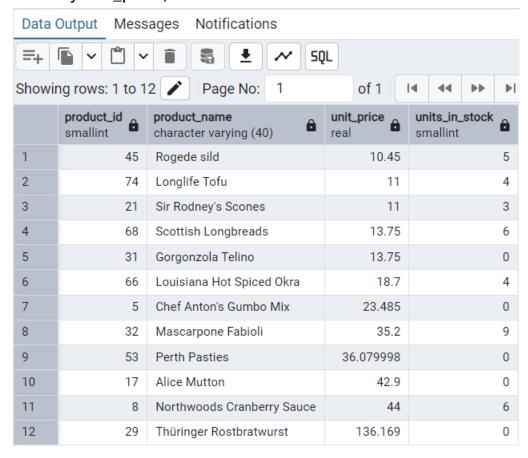
Update vw_updatable_products

Set unit_price = unit_price * 1.1 Where units_in_stock < 10;



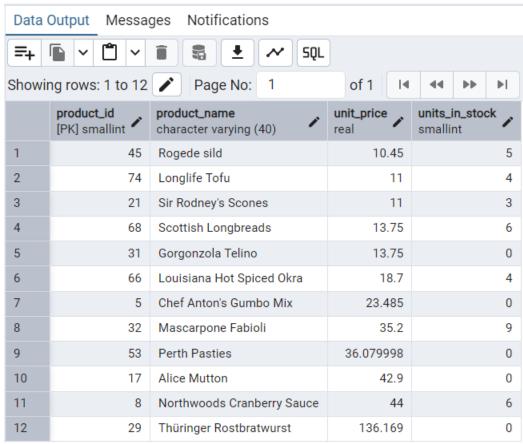
Check in the view table for update

Select *
From vw_updatable_products
Where units_in_stock < 10
order by unit_price;



PRODUCTS TABLE FOR UPDATE

Select product_id,
product_name,
unit_price,
units_in_stock
From products
Where units_in_stock < 10
order by unit_price;

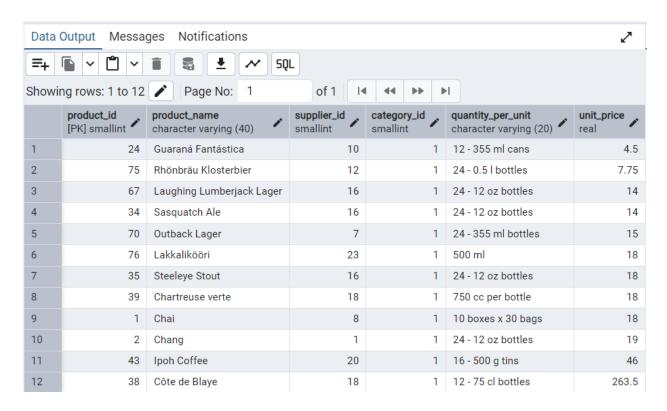


2. Transaction:

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

Try COMMIT and ROLLBACK and observe what happens.

SELECT * FROM products WHERE category_id = 1 order by unit_price;

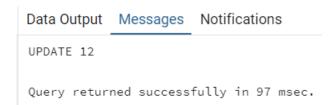


Begin;

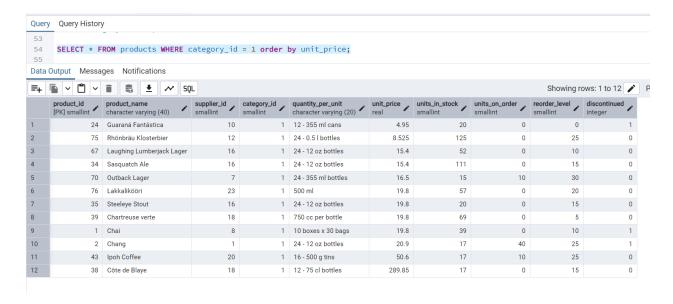
Update products

Set unit_price = unit_price * 1.10

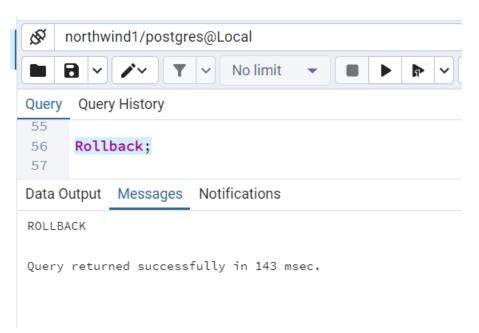
Where category_id = 1;



SELECT * FROM products WHERE category_id = 1 order by unit_price;



Rollback;



Check Transaction Rollback:

SELECT * FROM products WHERE category_id = 1 order by unit_price;



Commit;



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

Create View vw_Employee_details 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

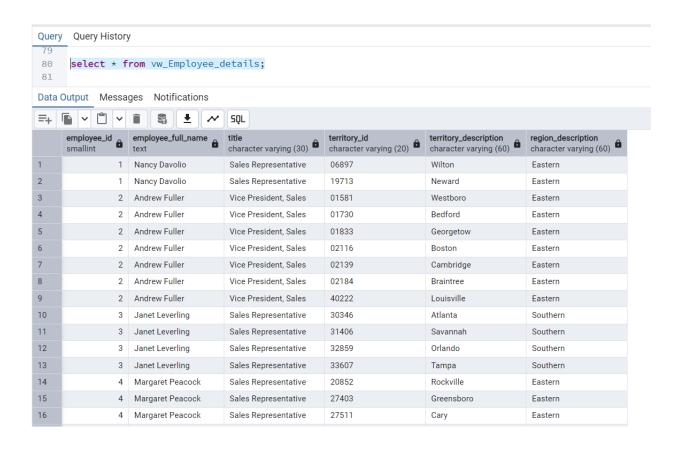
Join region r on t.region_id = r.region_id;

Query History

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



select * from vw_Employee_details;



4. Create a recursive CTE based on Employee Hierarchy

```
With Recursive cte_employee_hierarchy
AS(
Select
employee_id,
first_name,
last_name,
reports_to,
0 As Level
From employees e
Where reports_to Is Null
```

Union All

```
e.employee_id,
e.first_name,
e.last_name,
e.reports_to,
eh.level+1

From employees e

Join cte_employee_hierarchy eh

On eh.employee_id = e.reports_to
)

Select Level, employee_id, concat(first_name, '', last_name) as employee_name

From cte_employee_hierarchy

Order By level, employee_id;
```

```
Query Query History
84 v With Recursive cte_employee_hierarchy
86 Select
87 employee_id,
        first_name,
88
89
        last_name,
      reports_to,
90
91
        0 As Level
92 From employees e
93
    Where reports_to Is Null
    Union All
        e.employee_id,
97
        e.first_name,
98
        e.last_name,
99
        e.reports_to,
100
        eh.level+1
101 From employees e
Join cte_employee_hierarchy eh
103
     On eh.employee_id = e.reports_to
104
106 Select Level, employee_id, concat(first_name, ' ' , last_name) as employee_name
107 From cte_employee_hierarchy
108 Order By level, employee_id;
109
```

Data Output Messages Notifications =+ SQL employee_name level employee_id integer 🔒 smallint text 1 0 Andrew Fuller 2 2 1 1 Nancy Davolio 3 1 3 Janet Leverling Margaret Peacock 4 1 5 1 5 Steven Buchanan 6 1 Laura Callahan 8 7 2 6 Michael Suyama 8 2 7 Robert King 9 2 9 Anne Dodsworth