SQL Assignment 8

PostgreSQL

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, discontinued

FROM products

WHERE discontinued = 0

WITH CHECK OPTION

UPDATE vw_updatable_products SET unit_price = unit_price * 1.1 WHERE
units_in_stock < 10;
```

OUTPUT:

Data Output Messages Notifications

	product_id smallint	product_name character varying (40)	unit_price real	units_in_stock smallint	discontinued integer
1	3	Aniseed Syrup	10	13	0
2	4	Chef Anton's Cajun Seasoning	22	53	0
3	6	Grandma's Boysenberry Spread	25	120	0
4	7	Uncle Bob's Organic Dried Pears	30	15	0
5	8	Northwoods Cranberry Sauce	40	6	0
6	10	Ikura	31	31	0
7	11	Queso Cabrales	21	22	0
8	12	Queso Manchego La Pastora	38	86	0

Data Output Messages Notifications

	product_id smallint	product_name character varying (40)	unit_price real	units_in_stock smallint	discontinued integer
59	77	Original Frankfurter grüne Soße	13	32	0
60	8	Northwoods Cranberry Sauce	44	6	0
61	21	Sir Rodney's Scones	11	3	0
62	31	Gorgonzola Telino	13.75	0	0
63	32	Mascarpone Fabioli	35.2	9	0
64	45	Rogede sild	10.45	5	0
65	66	Louisiana Hot Spiced Okra	18.7	4	0
66	68	Scottish Longbreads	13.75	6	0
67	74	Longlife Tofu	11	4	0

Total rows: 67 Query complete 00:00:00.120

2. Transaction: Update the product price for products by 10% in category id=1

Try COMMIT and ROLLBACK and observe what happens.

QUERY:

SELECT * FROM products WHERE category_id = 1;

```
BEGIN;

UPDATE products

SET unit_price = unit_price * 1.10

WHERE category_id = 1;

ROLLBACK;

COMMIT;
```

OUTPUT:

Data Output Messages Notifications

Showing rows: 1 to 12

	product_id [PK] smallint	product_name character varying (40)	supplier_id smallint	category_id smallint	quantity_per_unit character varying (20)	unit_price real	units_in_stock smallint	units_on_order smallint	reorder_level smallint	discontinued integer
1	1	Chai	8	1	10 boxes x 30 bags	18	39	0	10	1
2	2	Chang	1	1	24 - 12 oz bottles	19	17	40	25	1
3	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.5	20	0	0	1
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	14	111	0	15	0
5	35	Steeleye Stout	16	1	24 - 12 oz bottles	18	20	0	15	0
6	38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5	17	0	15	0
7	39	Chartreuse verte	18	1	750 cc per bottle	18	69	0	5	0
8	43	Ipoh Coffee	20	1	16 - 500 g tins	46	17	10	25	0
9	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14	52	0	10	0

Total rows: 12 Query complete 00:00:00.140

Data Output Messages Notifications

Showing rows: 1 to 12

	product_id [PK] smallint	product_name character varying (40)	supplier_id smallint	category_id smallint	quantity_per_unit character varying (20)	unit_price real	units_in_stock smallint	units_on_order smallint	reorder_level smallint	discontinued integer
1	1	Chai	8	1	10 boxes x 30 bags	19.8	39	0	10	1
2	2	Chang	1	1	24 - 12 oz bottles	20.9	17	40	25	1
3	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.95	20	0	0	1
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	15.4	111	0	15	0
5	35	Steeleye Stout	16	1	24 - 12 oz bottles	19.8	20	0	15	0
6	38	Côte de Blaye	18	1	12 - 75 cl bottles	289.85	17	0	15	0
7	39	Chartreuse verte	18	1	750 cc per bottle	19.8	69	0	5	0
8	43	Ipoh Coffee	20	1	16 - 500 g tins	50.6	17	10	25	0
9	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	15.4	52	0	10	0

Total rows: 12 Query complete 00:00:00.102

Data Output Messages Notifications

Showing rows: 1 to 12

	product_id [PK] smallint	product_name character varying (40)	supplier_id smallint	category_id smallint	quantity_per_unit character varying (20)	unit_price real	units_in_stock smallint	units_on_order smallint	reorder_level smallint	discontinued integer
1	1	Chai	8	1	10 boxes x 30 bags	19.8	39	0	10	1
2	2	Chang	1	1	24 - 12 oz bottles	20.9	17	40	25	1
3	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.95	20	0	0	1
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	15.4	111	0	15	0
5	35	Steeleye Stout	16	1	24 - 12 oz bottles	19.8	20	0	15	0
6	38	Côte de Blaye	18	1	12 - 75 cl bottles	289.85	17	0	15	0
7	39	Chartreuse verte	18	1	750 cc per bottle	19.8	69	0	5	0
8	43	Ipoh Coffee	20	1	16 - 500 g tins	50.6	17	10	25	0
9	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	15.4	52	0	10	0

Total rows: 12 Query complete 00:00:00.102

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_territory AS

SELECT e.employee_id,

        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_territory;

OUTPUT:

Data Output Messages Notifications

	employee_id smallint	employee_full_name text	title character varying (30)	territory_id character varying (20)	territory_description character varying (60)	region_description character varying (60)
1	1	Nancy Davolio	Sales Representative	06897	Wilton	Eastern
2	1	Nancy Davolio	Sales Representative	19713	Neward	Eastern
3	2	Andrew Fuller	Vice President, Sales	01581	Westboro	Eastern
4	2	Andrew Fuller	Vice President, Sales	01730	Bedford	Eastern
5	2	Andrew Fuller	Vice President, Sales	01833	Georgetow	Eastern
6	2	Andrew Fuller	Vice President, Sales	02116	Boston	Eastern
7	2	Andrew Fuller	Vice President, Sales	02139	Cambridge	Eastern
8	2	Andrew Fuller	Vice President, Sales	02184	Braintree	Eastern
9	2	Andrew Fuller	Vice President, Sales	40222	Louisville	Eastern

Total rows: 49 Query complete 00:00:00.117

4. Create a recursive CTE based on Employee Hierarchy

QUERY:

```
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


SELECT 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,

e.first_name || ' ' || e.last_name AS employee_full_name

FROM cte_employee_hierarchy

ORDER BY level, employee_id;
```

OUTPUT:

Data Output Messages Notifications

	level integer	employee_id smallint	employee_full_name text
1	0	2	Andrew Fuller
2	1	1	Nancy Davolio
3	1	3	Janet Leverling
4	1	4	Margaret Peacock
5	1	5	Steven Buchanan
6	1	8	Laura Callahan
7	2	6	Michael Suyama
8	2	7	Robert King
9	2	9	Anne Dodsworth

Total rows: 9 Query complete 00:00:00.145