Day 8 Assignment

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;

Step1: Create a view table

```
Create View vw_updatable_products As
Select product_id,
        product_name,
        unit_price,
        units_in_stock
From products;
```

```
10  /* 1.Create view vw_updatable_products.
11  Try updating the view with the below query and see
12  Update query:
13  UPDATE updatable_products
14  SET unit_price = unit_price * 1.1
15  WHERE units_in_stock < 10;
16  */
17
18  Create View vw_updatable_products As
19  Select product_id,
20          product_name,
21          unit_price,
22          units_in_stock
23  From products;
24
```

Data Output | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 52 msec.
```

Step 2: Update the view table created

Update vw_updatable_products
Set unit_price = unit_price * 1.1
Where units_in_stock < 10;

```
18  Create View vw_updatable_products As
19  Select product_id,
20         product_name,
21         unit_price,
22         units_in_stock
23  From products;
24
25  Update vw_updatable_products
26  Set unit_price = unit_price * 1.1
27  Where units_in_stock < 10;
28
```

Data Output | Messages | Notifications

```
UPDATE 12

Query returned successfully in 55 msec.
```

Step 3: Check in the view table for update

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

```
34  Select *
35  From vw_updatable_products
36  Where units_in_stock < 10
37  order by unit_price;
```

Data Output | Messages | Notifications

	product_id smallint	product_name character varying (40)	unit_price real	units_in_stock smallint
1	45	Rogede sild	10.45	5
2	74	Longlife Tofu	11	4
3	21	Sir Rodney's Scones	11	3
4	68	Scottish Longbreads	13.75	6
5	31	Gorgonzola Telino	13.75	0
6	66	Louisiana Hot Spiced Okra	18.7	4
7	5	Chef Anton's Gumbo Mix	23.485	0
8	32	Mascarpone Fabioli	35.2	9
9	53	Perth Pasties	36.079998	0
10	17	Alice Mutton	42.9	0
11	8	Northwoods Cranberry Sauce	44	6
12	29	Thüringer Rostbratwurst	136.169	0

Total rows: 12 | Query complete 00:00:00.091

Step 4: Check in the products table also for update

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

```
39  Select product_id,
40  product_name,
41          unit_price,
42          units_in_stock
43  From products
44  Where units_in_stock < 10
45  order by unit_price;
46
47
```

Data Output | Messages | Notifications

	product_id [PK] smallint	product_name character varying (40)	unit_price real	units_in_stock smallint
1	45	Rogede sild	10.45	5
2	74	Longlife Tofu	11	4
3	21	Sir Rodney's Scones	11	3
4	68	Scottish Longbreads	13.75	6
5	31	Gorgonzola Telino	13.75	0
6	66	Louisiana Hot Spiced Okra	18.7	4
7	5	Chef Anton's Gumbo Mix	23.485	0
8	32	Mascarpone Fabioli	35.2	9
9	53	Perth Pasties	36.079998	0
10	17	Alice Mutton	42.9	0
11	8	Northwoods Cranberry Sauce	44	6
12	29	Thüringer Rostbratwurst	136.169	0

Total rows: 12 | Query complete 00:00:00.104

The products also got updated

2. Transaction: Update the product price for products by 10% in category id=1
Try COMMIT and ROLLBACK and observe what happens.

Step1: SELECT * FROM products WHERE category_id = 1 order by unit_price;

```
29  /* 2.Transaction:
30  Update the product price for products by 10% in category id=1
31  Try COMMIT and ROLLBACK and observe what happens.*/
32  -- select * from products;
33  SELECT * FROM products WHERE category_id = 1 order by unit_price;
34
```

Data Output | Messages | Notifications

	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	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.5	20	0	0	1
2	75	Rhönbräu Klosterbier	12	1	24 - 0.5 l bottles	7.75	125	0	25	0
3	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14	52	0	10	0
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	14	111	0	15	0
5	70	Outback Lager	7	1	24 - 355 ml bottles	15	15	10	30	0
6	76	Lakkalikööri	23	1	500 ml	18	57	0	20	0
7	35	Steeleye Stout	16	1	24 - 12 oz bottles	18	20	0	15	0
8	39	Chartreuse verte	18	1	750 cc per bottle	18	69	0	5	0
9	1	Chai	8	1	10 boxes x 30 bags	18	39	0	10	1
10	2	Chang	1	1	24 - 12 oz bottles	19	17	40	25	1
11	43	Ipoh Coffee	20	1	16 - 500 g tins	46	17	10	25	0
12	38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5	17	0	15	0

Total rows: 12 | Query complete 00:00:00.067

Step 2: **Begin;**
Update products
Set unit_price = unit_price * 1.10
Where category_id = 1;

```
29  /* 2.Transaction:
30  Update the product price for products by 10% in category id=1
31  Try COMMIT and ROLLBACK and observe what happens.*/
32  -- select * from products;
33  SELECT * FROM products WHERE category_id = 1 order by unit_price;
34
35  Begin;
36  Update products
37  Set unit_price = unit_price * 1.10
38  Where category_id = 1;
39
```

Data Output | Messages | Notifications

```
UPDATE 12

Query returned successfully in 81 msec.
```

Step 3: SELECT * FROM products WHERE category_id = 1 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;
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;

Commit;

Rollback;
```

Data Output | Messages | Notifications

	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	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.95	20	0	0	1
2	75	Rhönbräu Klosterbier	12	1	24 - 0.5 l bottles	8.525	125	0	25	0
3	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	15.4	52	0	10	0
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	15.4	111	0	15	0
5	70	Outback Lager	7	1	24 - 355 ml bottles	16.5	15	10	30	0
6	76	Lakkalikööri	23	1	500 ml	19.8	57	0	20	0
7	35	Steeleye Stout	16	1	24 - 12 oz bottles	19.8	20	0	15	0
8	39	Chartreuse verte	18	1	750 cc per bottle	19.8	69	0	5	0
9	1	Chai	8	1	10 boxes x 30 bags	19.8	39	0	10	1
10	2	Chang	1	1	24 - 12 oz bottles	20.9	17	40	25	1
11	43	Ipoh Coffee	20	1	16 - 500 g tins	50.6	17	10	25	0
12	38	Côte de Blaye	18	1	12 - 75 cl bottles	289.85	17	0	15	0

Total rows: 12 | Query complete 00:00:00.074

We can observe the updated prices, but the change is not permanent yet.

Step 4: Rollback;

```
Rollback;



/* 3.Create a regular view which wil
Employee_id,
Employee_full_name,
Title,
Territory_id,
territory_description,
```

Data Output | Messages | Notifications

```
ROLLBACK

Query returned successfully in 78 msec.
```

Check again : SELECT * FROM products WHERE category_id = 1;

```
44  Rollback;
45
46  SELECT * FROM products WHERE category_id = 1 order by unit_price;
47
```

Data Output | Messages | Notifications

	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	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.5	20	0	0	1
2	75	Rhönbräu Klosterbier	12	1	24 - 0.5 l bottles	7.75	125	0	25	0
3	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14	52	0	10	0
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	14	111	0	15	0
5	70	Outback Lager	7	1	24 - 355 ml bottles	15	15	10	30	0
6	76	Lakkalikööri	23	1	500 ml	18	57	0	20	0
7	35	Steeleye Stout	16	1	24 - 12 oz bottles	18	20	0	15	0
8	39	Chartreuse verte	18	1	750 cc per bottle	18	69	0	5	0
9	1	Chai	8	1	10 boxes x 30 bags	18	39	0	10	1
10	2	Chang	1	1	24 - 12 oz bottles	19	17	40	25	1
11	43	Ipoh Coffee	20	1	16 - 500 g tins	46	17	10	25	0
12	38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5	17	0	15	0

Total rows: 12 | Query complete 00:00:00.086

We can see the original prices, since the transaction was rolled back.

Step 5: Commit; Commit makes the updated prices permanent in the database.

```
46  Commit;
47
```

Data Output | Messages | Notifications

```
COMMIT

Query returned successfully in 49 msec.
```

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

Employee_id,Employee_full_name,Title,Territory_id,territory_description,region_description

```
/* 3.Create a regular view which will have the details below (Need to do joins):
Employee_id,
Employee_full_name,
Title,
Territory_id,
territory_description,
region_description*/

-- select * from employees;
-- select * from territories;
-- select * from employee_territories;
--Drop View vw_Employee_details;

-- CONCAT(first_name, ' ' , last_name) As
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;


select * from vw_Employee_details;
```

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
10	3	Janet Leverling	Sales Representative	30346	Atlanta	Southern
11	3	Janet Leverling	Sales Representative	31406	Savannah	Southern
12	3	Janet Leverling	Sales Representative	32859	Orlando	Southern
13	3	Janet Leverling	Sales Representative	33607	Tampa	Southern

Total rows: 49 | Query complete 00:00:00.267

4. Create a recursive CTE based on Employee Hierarchy

```
/* 4.Create a recursive CTE based on Employee Hierarchy*/
--Select employee_id, first_name, reports_to From employees;

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, concat(first_name, ' ' , last_name) as employee_name
From cte_employee_hierarchy
Order By level, employee_id;
```

Data Output | Messages | Notifications

	level integer	employee_id smallint	employee_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 00:00:00.070