

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;

```
8  Select product_id,
9      product_name,
10         unit_price,
11         units_in_stock
12 From products
13 Where units_in_stock < 10
14 order by unit_price;
15
16
17 Create View vw_updatable_products As
18 Select product_id,
19         product_name,
20         unit_price,
21         units_in_stock
22 From products;
23
24 Update vw_updatable_products
25 Set unit_price = unit_price * 1.1
26 Where units_in_stock < 10;
```

```
29 Select *
30 From vw_updatable_products
31 Where units_in_stock < 10
32 order by unit_price;
33
34 Select product_id,
35         product_name,
36         unit_price,
37         units_in_stock
38 From products
39 Where units_in_stock < 10
40 order by unit_price;
```

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	20	Thymian Roast Beef	49.4	0
Total rows: 12		Query complete 00:00:00.144		

2. Transaction:

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

```
49 SELECT * FROM products WHERE category_id = 1 order by unit_price;
50
51 Begin;
52 Update products
53 Set unit_price = unit_price * 1.10
54 Where category_id = 1;
55
56 SELECT * FROM products WHERE category_id = 1 order by unit_price;
57
58 Rollback;
59
60 SELECT * FROM products WHERE category_id = 1 order by unit_price;
61
62 Commit;
```

Data Output

Messages

Notifications

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

```

79  Create View vw_Employee_details As
80  Select
81      e.Employee_id,
82      CONCAT(e.first_name, ' ', e.last_name) As Employee_full_name,
83      e.Title,
84      t.Territory_id,
85      t.territory_description,
86      r.region_description
87  From employees e
88  Join employee_territories et On e.employee_id = et.employee_id
89  Join territories t on et.territory_id = t.territory_id
90  Join region r on t.region_id = r.region_id;
91
92  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	32250	Colony	Southern
Total rows: 49		Query complete 00:00:00.095				

4. Create a recursive CTE based on Employee Hierarchy

```
99  With Recursive cte_employee_hierarchy
100  AS(
101  Select
102      employee_id,
103      first_name,
104      last_name,
105      reports_to,
106      0 As Level
107  From employees e
108  Where reports_to Is Null
109
110  Union All
111
112  Select
113      e.employee_id,
114      e.first_name,
115      e.last_name,
116      e.reports_to,
117      eh.level+1
118  From employees e
119  Join cte_employee_hierarchy eh
120  On eh.employee_id = e.reports_to
121  )
122
123  Select Level, employee_id, concat(first_name, ' ', last_name) as employee_name
124  From cte_employee_hierarchy
125  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 00:00:00.076