

YAMUNA DAY8 ASSIGNMENTS:

/* 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;

*/

Select product_id,

product_name,

unit_price,

units_in_stock

From products

Where units_in_stock < 10

order by unit_price;

Create View vw_updatable_products As

Select product_id,

product_name,

unit_price,

units_in_stock

From products;

```

25  Create View vw_updatable_products As
26  Select product_id,
27         product_name,
28         unit_price,
29         units_in_stock
30  From products;
31
32

```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 37 msec.

Update vw_updatable_products

Set unit_price = unit_price * 1.1

Where units_in_stock < 10;

```

33  Update vw_updatable_products
34  Set unit_price = unit_price * 1.1
35  Where units_in_stock < 10;
36
37

```

Data Output Messages Notifications

UPDATE 12

Query returned successfully in 35 msec.

Select *

From vw_updatable_products

Where units_in_stock < 10

order by unit_price;

```
39  Select *
40  From vw_updatable_products
41  Where units_in_stock < 10
42  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

Select product_id,

product_name,

unit_price,

units_in_stock

From products

Where units_in_stock < 10

order by unit_price;

```
45 Select product_id,  
46 product_name,  
47 unit_price,  
48 units_in_stock  
49 From products  
50 Where units_in_stock < 10  
51 order by unit_price;  
52  
53
```

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

/* 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;
```

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

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

61 -- select * from products;]

62

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

Data Output

Messages

Notifications

≡

SQL

Showing rows: 1 to 12

Page No: 1

	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

Begin;

Update products

Set unit_price = unit_price * 1.10

Where category_id = 1;

```

65 Begin;
66 Update products
67 Set unit_price = unit_price * 1.10
68 Where category_id = 1;
69

```

Data Output Messages Notifications

UPDATE 12

Query returned successfully in 50 msec.

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

```

70 SELECT * FROM products WHERE category_id = 1 order by unit_price;
71

```

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	Ippoh 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

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

Rollback;

73

[Data Output](#) [Messages](#) [Notifications](#)

ROLLBACK

Query returned successfully in 74 msec.

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

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

Data Output Messages Notifications

Showing rows: 1 to 12 Page No: 1 of 1

	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

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

Commit;

Commit makes the updated price permanent in the data base.

/* 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;
```

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;


```

92
93 Create View vw_Employee_details As
94 Select
95     e.Employee_id,
96     CONCAT(e.first_name, ' ', e.last_name) As Employee_full_name,
97     e.Title,
98     t.Territory_id,
99     t.territory_description,
100     r.region_description
101 From employees e
102 Join employee_territories et On e.employee_id = et.employee_id
103 Join territories t on et.territory_id = t.territory_id
104 Join region r on t.region_id = r.region_id;
105
106
107 select * from vw_Employee_details;
108

```

Data Output Messages Notifications

Notifications							Showing rc
	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	

/* 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;

4. Create a recursive CTE based on Employee Hierarchy

```
34  /* 4.Create a recursive CTE based on Employee Hierarchy*/
35  --Select employee_id, first_name, reports_to From employees;
36
37  With Recursive cte_employee_hierarchy
38  AS(
39  Select
40      employee_id,
41      first_name,
42      last_name,
43      reports_to,
44      0 As Level
45  From employees e
46  Where reports_to Is Null
47
48  Union All
49
50  Select
51      e.employee_id,
52      e.first_name,
53      e.last_name,
54      e.reports_to,
55      eh.level+1
56  From employees e
57  Join cte_employee_hierarchy eh
58  On eh.employee_id = e.reports_to
59  )
60
61  Select Level, employee_id, concat(first_name, ' ', last_name) as employee_name
62  From cte_employee_hierarchy
63  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