

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;

Create view vw_updatable_products As

Select product_id,

product_name,

unit_price,

units_in_stock

From products ;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑

🗄

⬇

📈

SQL

Showing rows: 1 to 12

✎

Page No: 1

of 1

⏪

⏴

⏵

⏩

	product_id [PK] smallint ✎	product_name character varying (40) ✎	unit_price real ✎	units_in_stock smallint ✎
1	45	Rogede sild	9.5	5
2	74	Longlife Tofu	10	4
3	21	Sir Rodney's Scones	10	3
4	68	Scottish Longbreads	12.5	6
5	31	Gorgonzola Telino	12.5	0
6	66	Louisiana Hot Spiced Okra	17	4
7	5	Chef Anton's Gumbo Mix	21.35	0
8	32	Mascarpone Fabioli	32	9
9	53	Perth Pasties	32.8	0
10	17	Alice Mutton	39	0
11	8	Northwoods Cranberry Sauce	40	6
12	29	Thüringer Rostbratwurst	123.79	0

Create View vw_updatable_products As

Select product_id,

product_name,
unit_price,
units_in_stock
From products;

Data Output Messages Notifications				
<div> <div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗑️</div> <div>📥</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div> </div>				
Showing rows: 1 to 77 ✎ Page No: 1 of 1 <div> ⏪ ⏴ ⏵ ⏩ </div>				
	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;

Data Output	Messages	Notifications
UPDATE 12		
Query returned successfully in 72 msec.		

Check in the view table for update

Select *

From vw_updatable_products

Where units_in_stock < 10

order by unit_price;

Data Output

Messages

Notifications

</

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;

Data Output Messages Notifications				
<div> <div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑</div> <div>🗄</div> <div>⬇</div> <div>📈</div> <div>SQL</div> </div> <div>Showing rows: 1 to 12 <div>✎</div> Page No: 1 of 1 <div>⏪</div> <div>⏴</div> <div>⏵</div> <div>⏩</div></div> </div>				
	product_id [PK] smallint <div>✎</div>	product_name character varying (40) <div>✎</div>	unit_price real <div>✎</div>	units_in_stock smallint <div>✎</div>
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 WHERE category_id = 1 order by unit_price;

Data Output Messages Notifications						
<div> <div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗑️</div> <div>📥</div> <div>📥</div> <div>📈</div> <div>SQL</div> </div> <div> Showing rows: 1 to 12 <div>✎</div> Page No: 1 of 1 <div>⏪</div> <div>⏴</div> <div>⏵</div> <div>⏩</div> </div> </div>						
	product_id [PK] smallint	product_name character varying (40)	supplier_id smallint	category_id smallint	quantity_per_unit character varying (20)	unit_price real
1	24	Guaraná Fantástica	10	1	12 - 355 ml cans	4.5
2	75	Rhönbräu Klosterbier	12	1	24 - 0.5 l bottles	7.75
3	67	Laughing Lumberjack Lager	16	1	24 - 12 oz bottles	14
4	34	Sasquatch Ale	16	1	24 - 12 oz bottles	14
5	70	Outback Lager	7	1	24 - 355 ml bottles	15
6	76	Lakkalikööri	23	1	500 ml	18
7	35	Steeleye Stout	16	1	24 - 12 oz bottles	18
8	39	Chartreuse verte	18	1	750 cc per bottle	18
9	1	Chai	8	1	10 boxes x 30 bags	18
10	2	Chang	1	1	24 - 12 oz bottles	19
11	43	Ipoh Coffee	20	1	16 - 500 g tins	46
12	38	Côte de Blaye	18	1	12 - 75 cl bottles	263.5

Begin;

Update products

Set unit_price = unit_price * 1.10

Where category_id = 1;

Data Output Messages Notifications
UPDATE 12
Query returned successfully in 97 msec.

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

Query Query History

53

54

55

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

Data Output Messages Notifications


SQL













Showing rows: 1 to 12

P

	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

Rollback;

 northwind1/postgres@Local

 No limit 

Query

Query History

55

56

57

Rollback;

Data Output

Messages

Notifications

ROLLBACK

Query returned successfully in 143 msec.

Check Transaction Rollback:

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

57
58 `SELECT * FROM products WHERE category_id = 1 order by unit_price;`
59

Data Output
Messages
Notifications

Showing rows: 1 to 12
Page 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

Commit;

60 `Commit;`
61
62

Data Output
Messages
Notifications

COMMIT

Query returned successfully in 65 msec.

60 `Commit;`
61
62

Data Output
Messages
Notifications

Query returned successfully in 65 msec.

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	Query History
62	
63	/* 3.Create a regular view which will have the details below (Need to do joins):
64	Employee_id, Employee_full_name, Title, Territory_id, territory_description, region_description*/
65	
66	Create View vw_Employee_details As
67	Select
68	e.Employee_id,
69	CONCAT(e.first_name, ' ', e.last_name) As Employee_full_name,
70	e.Title,
71	t.Territory_id,
72	t.territory_description,
73	r.region_description
74	From employees e
75	Join employee_territories et On e.employee_id = et.employee_id
76	Join territories t on et.territory_id = t.territory_id
77	Join region r on t.region_id = r.region_id;
78	
79	

Data Output	Messages	Notifications
CREATE VIEW		
Query returned successfully in 75 msec.		

```
select * from vw_Employee_details;
```


Query

Query History

79

80

81

select * from vw_Employee_details;

Data Output

Messages

Notifications

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

```

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;

```

Query	Query History
84	▼ With Recursive cte_employee_hierarchy
85	AS(
86	Select
87	employee_id,
88	first_name,
89	last_name,
90	reports_to,
91	0 As Level
92	From employees e
93	Where reports_to Is Null
94	Union All
95	Select
96	e.employee_id,
97	e.first_name,
98	e.last_name,
99	e.reports_to,
100	eh.level+1
101	From employees e
102	Join cte_employee_hierarchy eh
103	On eh.employee_id = e.reports_to
104)
105	
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	



SQL

	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