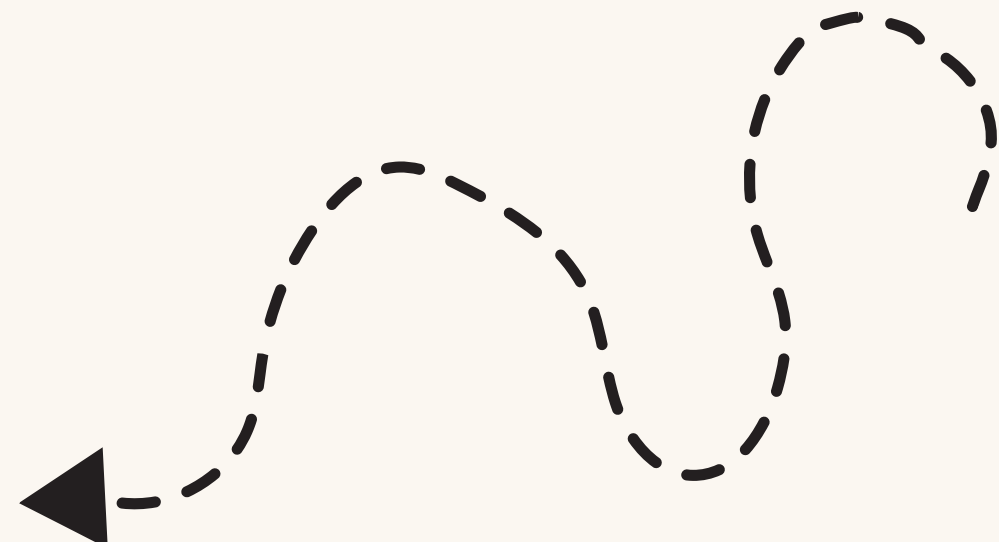


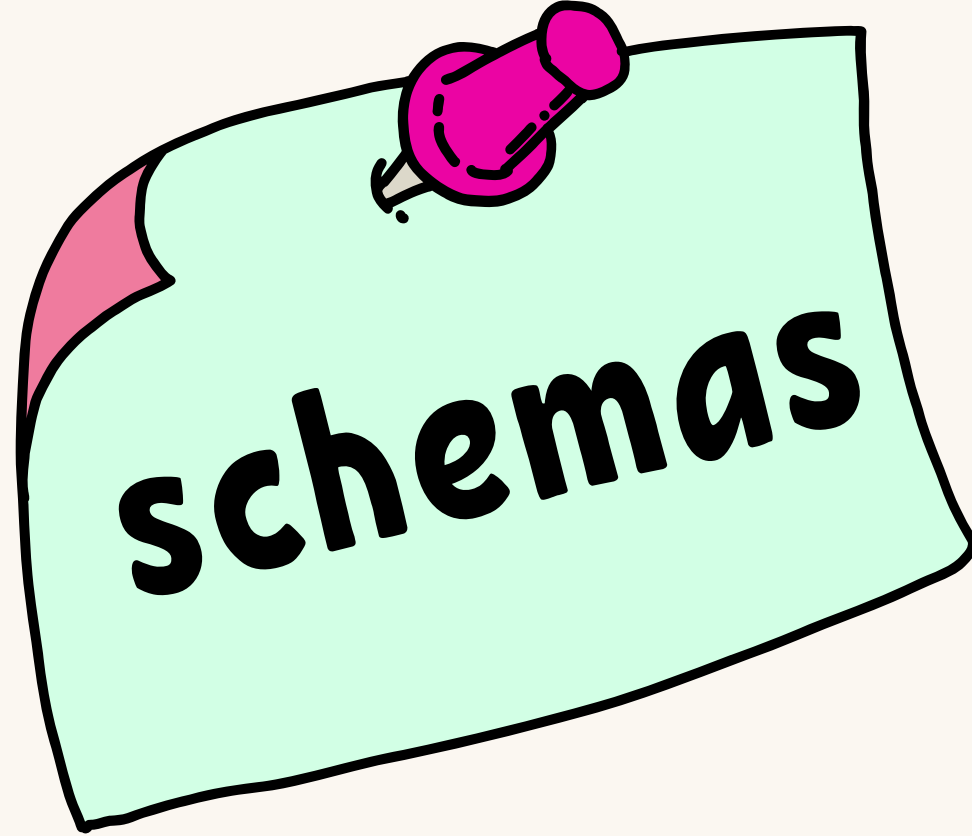
By Kareem Shaaban



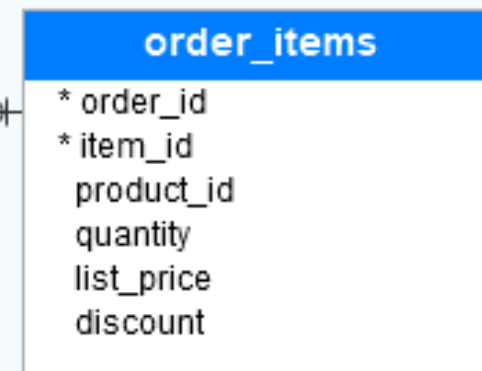
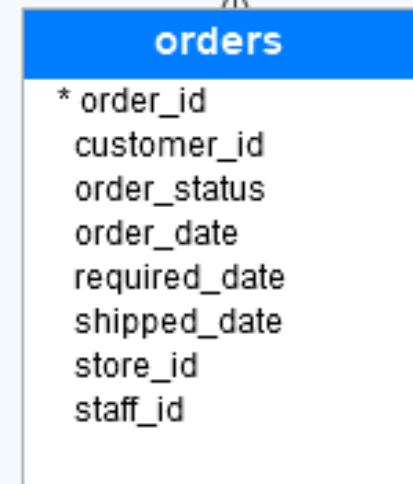
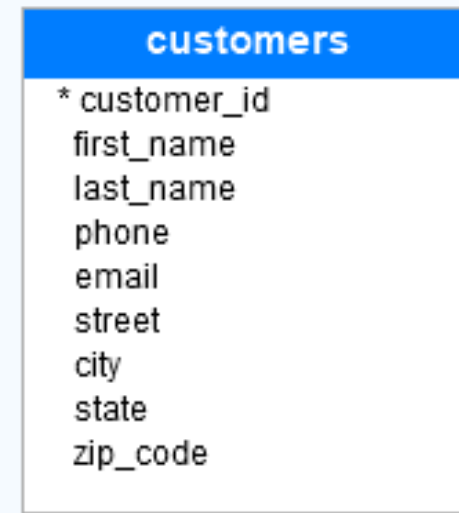
SQL

SECOND WEEK ASSIGNMENT WITH TDI

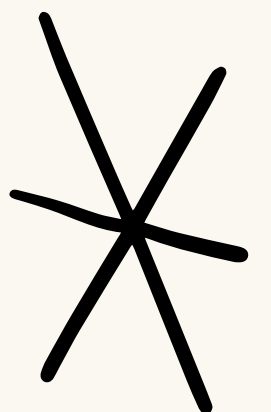
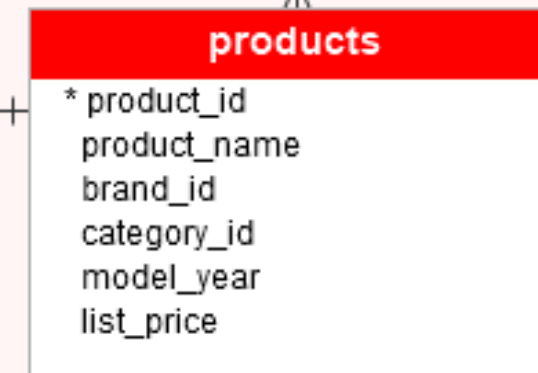




Sales



Production



1 Write a query to select the top 10 records from the orders table.

Query

```
Select Top (10) * From sales.orders;
```

Output

	order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id
1	1	259	4	2016-01-01	2016-01-03	2016-01-03	1	2
2	2	1212	4	2016-01-01	2016-01-04	2016-01-03	2	6
3	3	523	4	2016-01-02	2016-01-05	2016-01-03	2	7
4	4	175	4	2016-01-03	2016-01-04	2016-01-05	1	3
5	5	1324	4	2016-01-03	2016-01-06	2016-01-06	2	6
6	6	94	4	2016-01-04	2016-01-07	2016-01-05	2	6
7	7	324	4	2016-01-04	2016-01-07	2016-01-05	2	6
8	8	1204	4	2016-01-04	2016-01-05	2016-01-05	2	7
9	9	60	4	2016-01-05	2016-01-08	2016-01-08	1	2
10	10	442	4	2016-01-05	2016-01-06	2016-01-06	2	6

MS SQL server view

2 Write a query to find out where the order date is in 2016, and the State is New York.

Query

```
SELECT * FROM sales.orders AS o INNER JOIN sales.stores AS s ON o.store_id = s.store_id
WHERE order_date BETWEEN '2016-01-01' AND '2016-12-31'
AND s.state = 'NY';
```

Output

Results		Messages														
	order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id	store_id	store_name	phone	email	street	city	state	zip_code
1	2	1212	4	2016-01-01	2016-01-04	2016-01-03	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
2	3	523	4	2016-01-02	2016-01-05	2016-01-03	2	7	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
3	5	1324	4	2016-01-03	2016-01-06	2016-01-06	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
4	6	94	4	2016-01-04	2016-01-07	2016-01-05	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
5	7	324	4	2016-01-04	2016-01-07	2016-01-05	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
6	8	1204	4	2016-01-04	2016-01-05	2016-01-05	2	7	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
7	10	442	4	2016-01-05	2016-01-06	2016-01-06	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
8	11	1326	4	2016-01-05	2016-01-08	2016-01-07	2	7	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
9	13	873	4	2016-01-08	2016-01-11	2016-01-11	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
10	15	450	4	2016-01-09	2016-01-10	2016-01-12	2	7	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
11	21	1250	4	2016-01-15	2016-01-16	2016-01-18	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
12	24	636	4	2016-01-18	2016-01-20	2016-01-19	2	7	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
13	25	657	4	2016-01-18	2016-01-21	2016-01-21	2	6	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
14	26	1280	4	2016-01-18	2016-01-21	2016-01-19	2	7	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432
15	27	57	4	2016-01-19	2016-01-21	2016-01-20	2	7	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut Lane	Baldwin	NY	11432

MS SQL server view

3 Write a query to find all sales where the stocks are lower than 10 pieces.

Query

```
SELECT * FROM production.stocks AS s INNER JOIN production.products AS p  
ON s.product_id = p.product_id  
WHERE s.quantity < 10;
```

Output

Results		Messages							
	store_id	product_id	quantity	product_id	product_name	brand_id	category_id	model_year	list_price
1	1	2	5	2	Ritchey Timberwolf Frameset - 2016	5	6	2016	749.99
2	1	3	6	3	Surly Wednesday Frameset - 2016	8	6	2016	999.99
3	3	3	0	3	Surly Wednesday Frameset - 2016	8	6	2016	999.99
4	2	4	2	4	Trek Fuel EX 8 29 - 2016	9	6	2016	2899.99
5	2	5	1	5	Heller Shagamaw Frame - 2016	3	6	2016	1320.99
6	3	5	3	5	Heller Shagamaw Frame - 2016	3	6	2016	1320.99
7	1	6	0	6	Surly Ice Cream Truck Frameset - 2016	8	6	2016	469.99
8	1	7	8	7	Trek Slash 8 27.5 - 2016	9	6	2016	3999.99
9	2	7	8	7	Trek Slash 8 27.5 - 2016	9	6	2016	3999.99
10	2	8	1	8	Trek Remedy 29 Carbon Frameset - 2016	9	6	2016	1799.99
11	1	8	0	8	Trek Remedy 29 Carbon Frameset - 2016	9	6	2016	1799.99
12	1	11	8	11	Surly Straggler 650b - 2016	8	4	2016	1680.99
13	2	12	2	12	Electra Townie Original 21D - 2016	1	3	2016	549.99
14	2	13	1	13	Electra Cruiser 1 (24-Inch) - 2016	1	3	2016	269.99
15	1	14	8	14	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	1	3	2016	269.99

MS SQL server view

4 Write a query to find the count of all orders grouped by states order by the count descending.

Query

```
SELECT s.state , COUNT(DISTINCT o.order_id) as Total_Orders
FROM sales.orders AS o INNER JOIN sales.stores AS s ON o.store_id = s.store_id
GROUP BY s.state
order by Total_Orders Desc ;
```

Output

Results		Messages
	state	Total_Orders
1	NY	1093
2	CA	348
3	TX	174

MS SQL server view

5 Write a query to find out the total orders for each customer that has ordered more than 2 orders in total.

Query

```
Select customer_id , count(order_id) as total_orders from sales.orders  
group by customer_id  
having count(order_id) > 2;
```

Output

	customer_id	total_orders
1	1	3
2	2	3
3	3	3
4	4	3
5	5	3
6	6	3
7	7	3
8	8	3
9	9	3
10	10	3
11	11	3
12	12	3
13	13	3
14	14	3
15	15	3
16	16	3
17	17	3

MS SQL server view

6 Write a query to find out the total revenue, the avg unit price, minimum unite price and the maximum unit price.

Query

```
Select sum(Quantity * list_price *(1- discount)) as total_revenue ,  
min(list_price) as minimum_unite_price ,  
max(list_price) as maximum_unite_price  
from sales.order_items
```

Output

Results Messages			
	total_revenue	minimum_unite_price	maximum_unite_price
1	7689116.5576	89.99	11999.99

MS SQL server view

7 Write a query to find the total revenue each year.

Query

```
select YEAR(order_date) as "Year" , sum(Quantity * list_price * (1-discount)) as total_revenue
from sales.order_items as oi join sales.orders as o on oi.order_id = o.order_id
group by YEAR(order_date);
```

Output

	Results	Messages
	Year	total_revenue
1	2016	2427378.5276
2	2017	3447208.2425
3	2018	1814529.7875

MS SQL server view

8

Write a query to select Productid, count the number of sales, and calculate the total revenue for each product. Only include products that have more than 20 sales and order the results by total revenue in descending order.

Query

```
select product_id , count(product_id) as num_of_sales,  
sum(Quantity * list_price * (1-discount)) as total_revenue  
from sales.order_items  
group by product_id  
having count(product_id) > 20;
```

Output

	product_id	num_of_sales	total_revenue
1	23	100	41011.6329
2	92	26	7183.7579
3	29	24	30939.6906
4	9	101	389248.7025
5	15	91	65061.5724
6	3	86	112288.8771
7	52	26	27690.0439
8	95	31	12785.5738
9	72	22	18289.7050
10	78	27	22899.9666
11	66	24	8011.6008
12	32	22	11937.7460
13	26	90	72424.7929
14	12	104	75772.1223
15	106	27	17036.6214

MS SQL server view



DONE :)

Presented By Kareem Shaaban