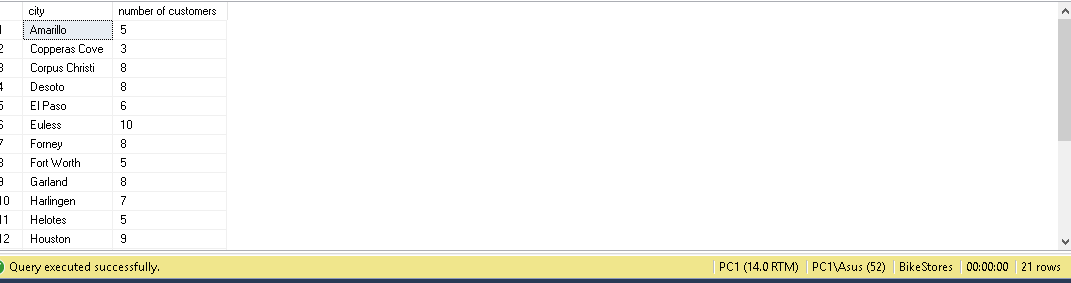
-----------BASICS----------

SQL is very easy to learn. ENJOY!!!

----1. All the cities in the Texas and the numbers of customers in each city.---



**Solution:**

select distinct city, count(customer\_id) as 'number of customers'

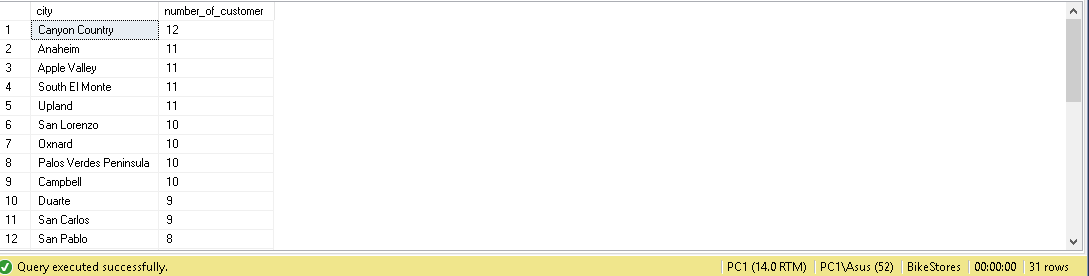
from sales.customers

where state = 'TX'

group by city

order by city;

----2. All the cities in the California which has more than 5 customer, by showing the cities which have more customers first.---



**Solution:**

select distinct city, count(customer\_id) as number\_of\_customers

from sales.customers

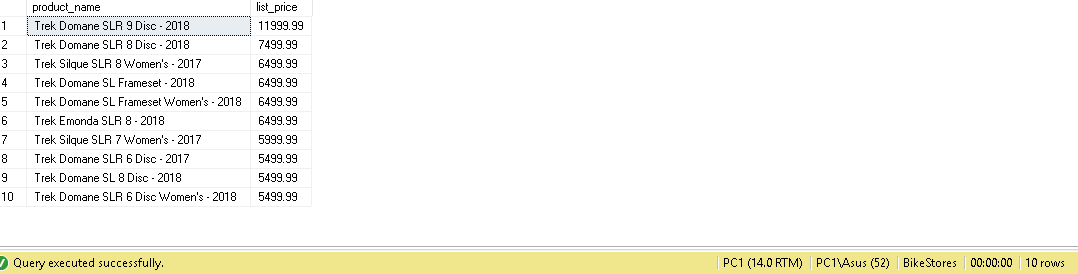
where state = 'CA'

group by city

having count(customer\_id) > 5

order by number\_of\_customers desc;

-----3. The top 10 most expensive products------



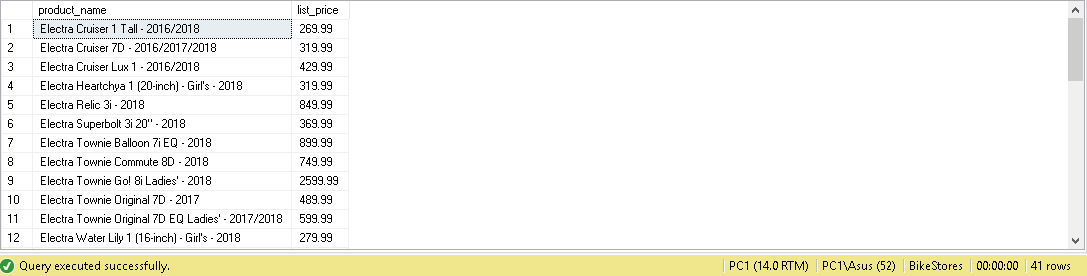
Solution:

select top 10 product\_name, list\_price

from production.products

order by list\_price desc;

-----4. Product name and list price of the products whic are located in the store id 2 and the quantity is greater than 25------



Solution:

select

A.product\_name,

A.list\_price

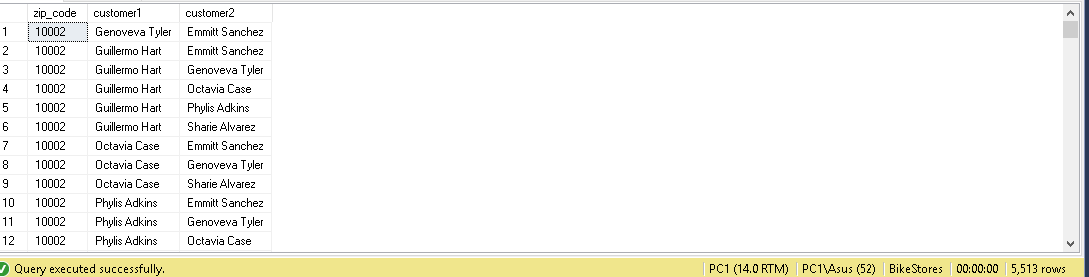
from production.products as A, production.stocks as B

where a.product\_id = B.product\_id

and B.store\_id = 2

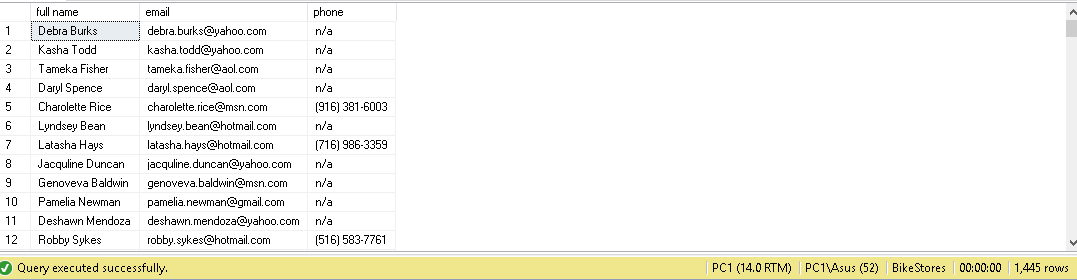
and B.quantity > 25;

-----5. Find the customers who locate in the same zip code------



Solution:

-----6. Return first name, last name, e-mail and phone number of the customers------

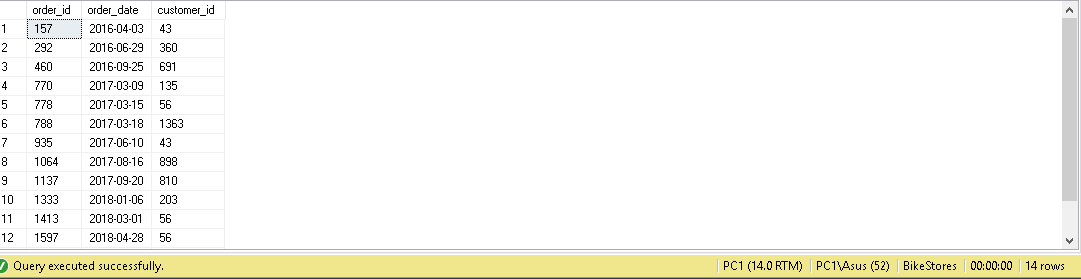


Solution:

select concat(first\_name, ' ', last\_name) as 'full name', email, phone

from sales.customers;

-----7. Find the sales order of the customers who lives in Houston order by order date------



Solution:

select

B.order\_id,

B.order\_date,

A.customer\_id

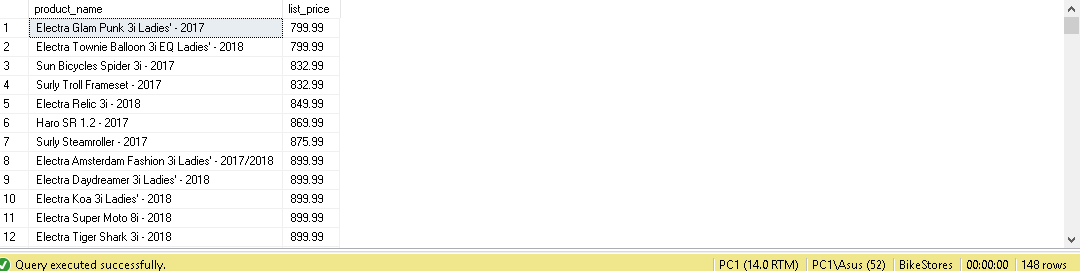
from sales.customers as A, sales.orders as B

where A.customer\_id = B.customer\_id

and A.city = 'Houston'

order by B.order\_date

-----8. Find the products whose list price is greater than the average list price of all products with the Electra or Heller ------



Solution:

select

distinct A.product\_name,

A.list\_price

from production.products as A, production.brands as B

where A.brand\_id = B.brand\_id

and A.list\_price > (

select

avg(A.list\_price)

from production.products as A, production.brands as B

where A.brand\_id = B.brand\_id

and B.brand\_name = 'Heller'

)

or A.list\_price > (

select

avg(A.list\_price)

from production.products as A, production.brands as B

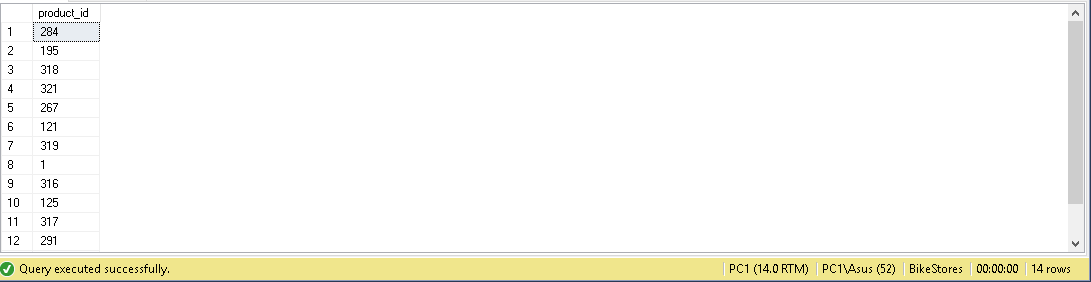
where A.brand\_id = B.brand\_id

and B.brand\_name = 'Electra'

)

order by A.list\_price;

-----9. Find the products that have no sales ------



Solution:

select

A.product\_id,

B.order\_id

from production.products as A

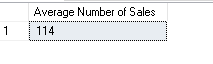
left join sales.order\_items as B

on A.product\_id = B.product\_id

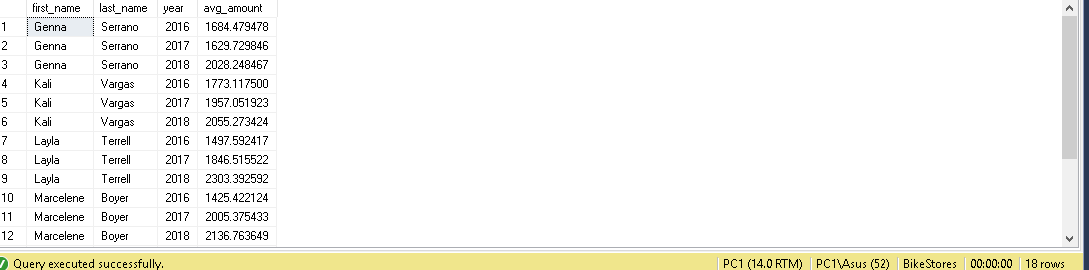
where B.order\_id is null;

**IF YOU WANT TO IMPROVE YOUR SKILLS IN SQL TRY THESE 🡪**

---- 10. Return the average number of sales orders in 2017 sales----



----11. By using view get the sales by staffs and years using the AVG() aggregate function:



Note: Finding nth largest, highest number…

select min(list\_price)

from (select top 7 list\_price

from production.products

order by list\_price desc)

as temp