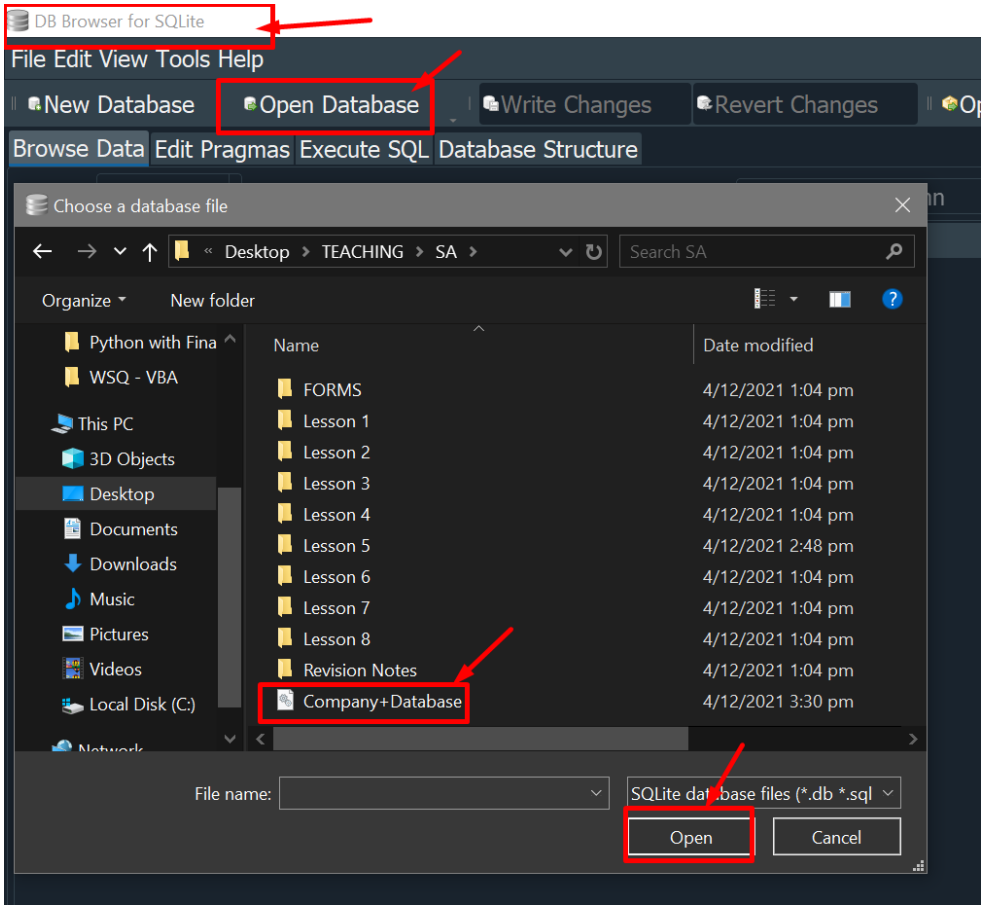
LEARNING DB BROWSER

DR. ALVIN ANG

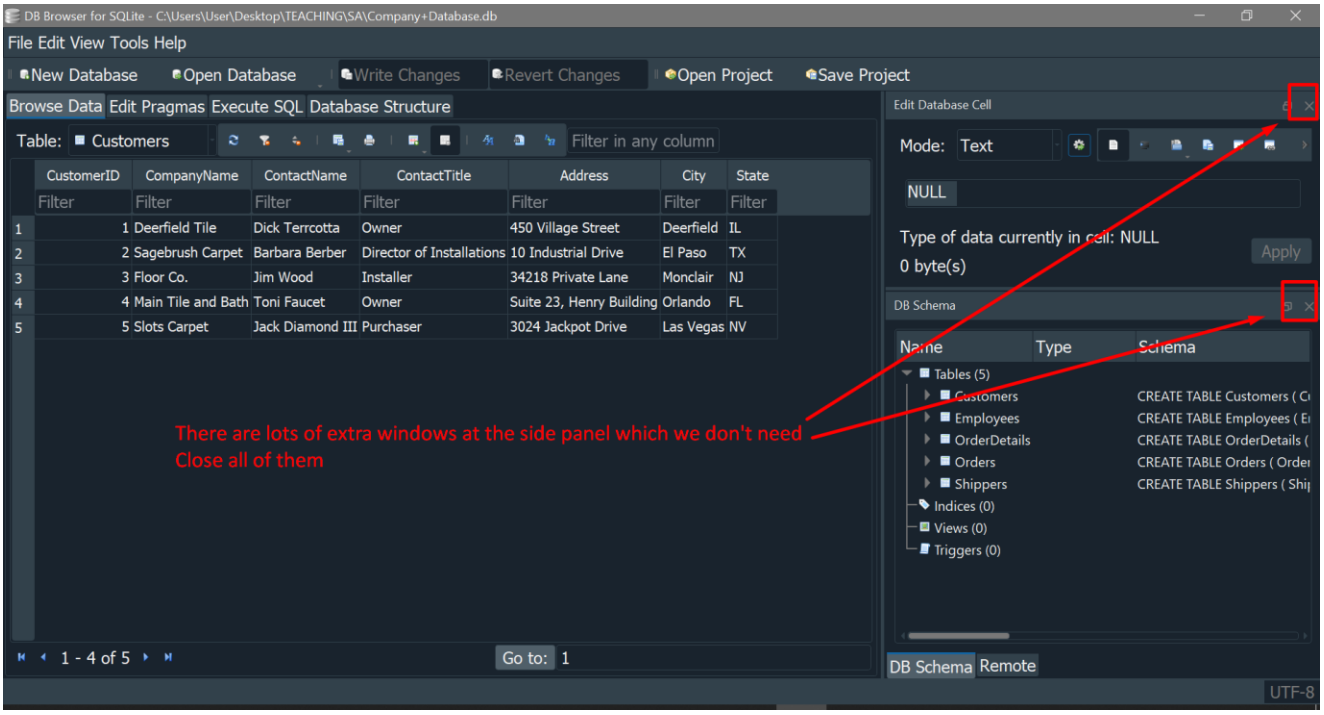
I. USING DB BROWSER

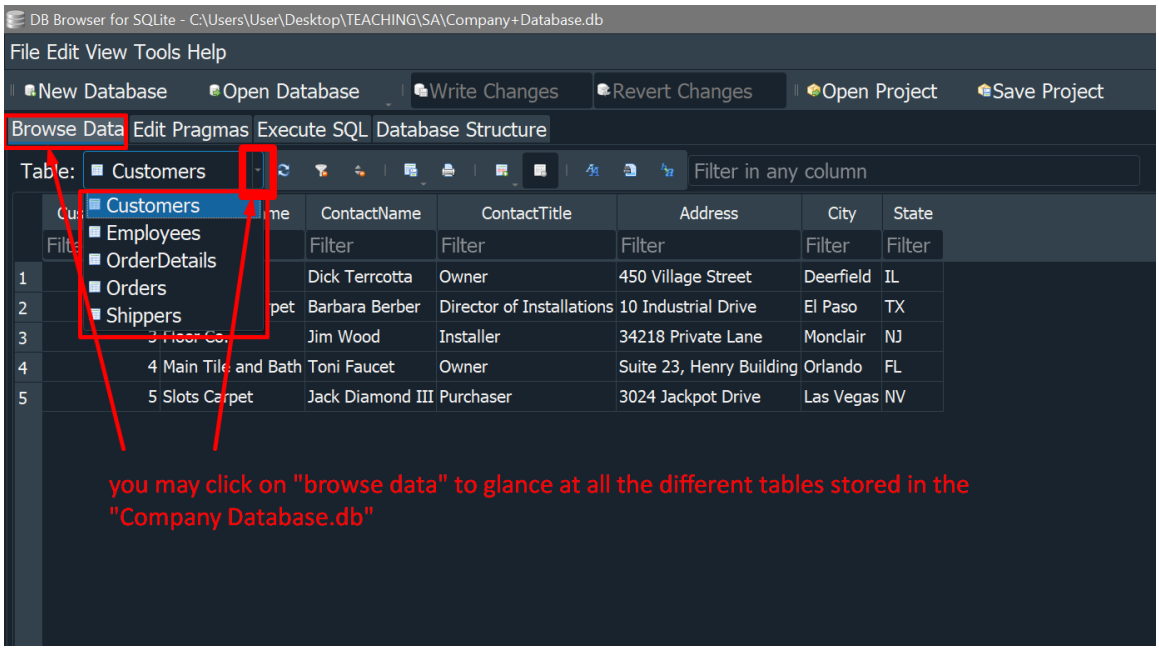
A. INSTALLATION

- https://sqlitebrowser.org/
- https://www.alvinang.sg/s/Company-Database.db
- Use the Sqlite browser software to open the .db file



B. PREPARATION



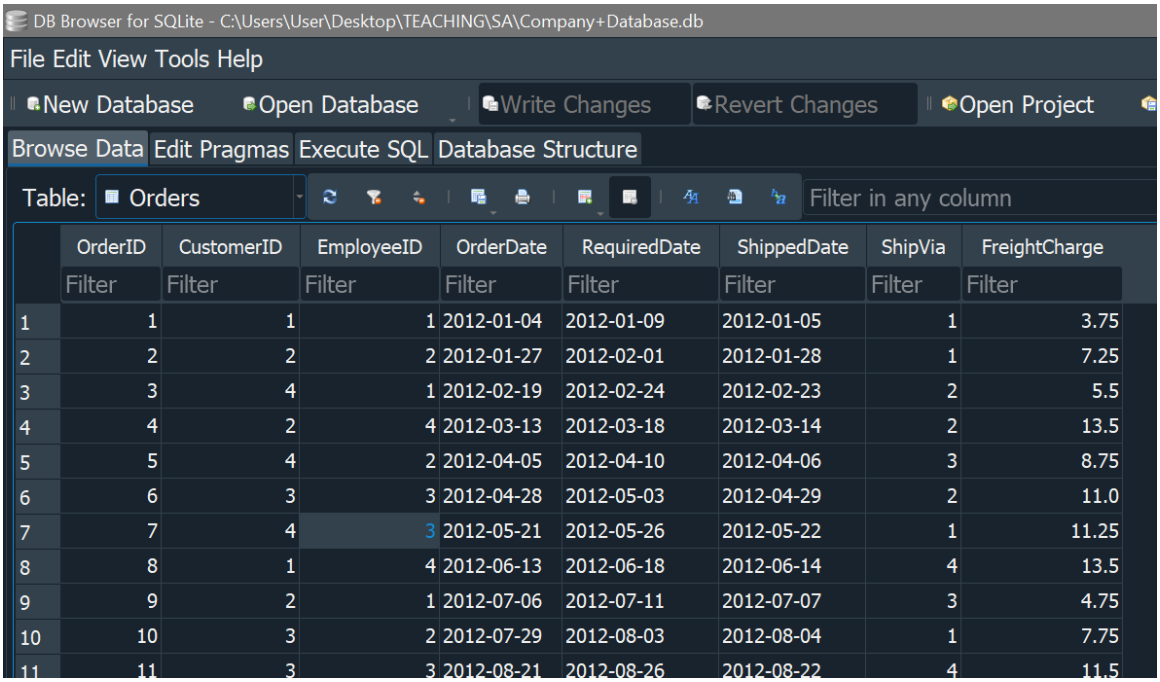


C. SELECT

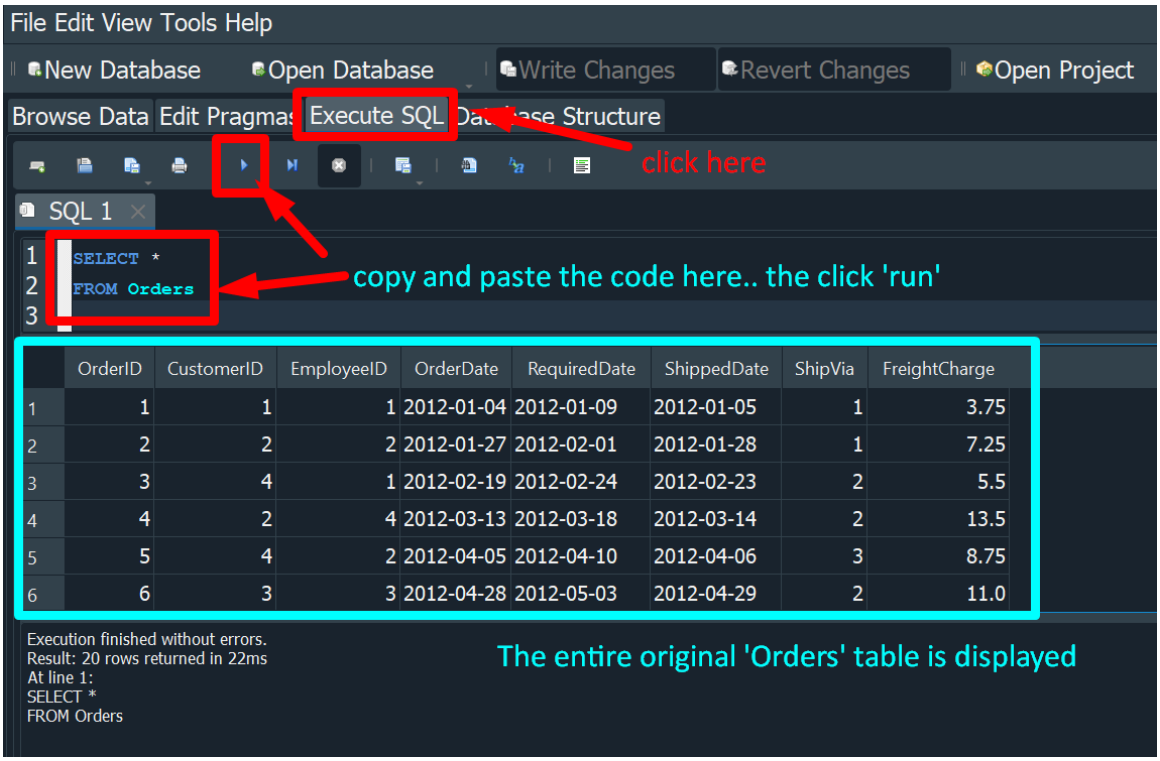
1. CODE USED

```
SELECT *

FROM Orders
```

2. ORIGINAL 'ORDERS' TABLE



	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	FreightCharge
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	1	2012-01-04	2012-01-09	2012-01-05	1	3.75
2	2	2	2	2012-01-27	2012-02-01	2012-01-28	1	7.25
3	3	4	1	2012-02-19	2012-02-24	2012-02-23	2	5.5
4	4	2	4	2012-03-13	2012-03-18	2012-03-14	2	13.5
5	5	4	2	2012-04-05	2012-04-10	2012-04-06	3	8.75
6	6	3	3	2012-04-28	2012-05-03	2012-04-29	2	11.0
7	7	4	3	2012-05-21	2012-05-26	2012-05-22	1	11.25
8	8	1	4	2012-06-13	2012-06-18	2012-06-14	4	13.5
9	9	2	1	2012-07-06	2012-07-11	2012-07-07	3	4.75
10	10	3	2	2012-07-29	2012-08-03	2012-08-04	1	7.75
11	11	3	3	2012-08-21	2012-08-26	2012-08-22	4	11.5



3. EXPLANATION

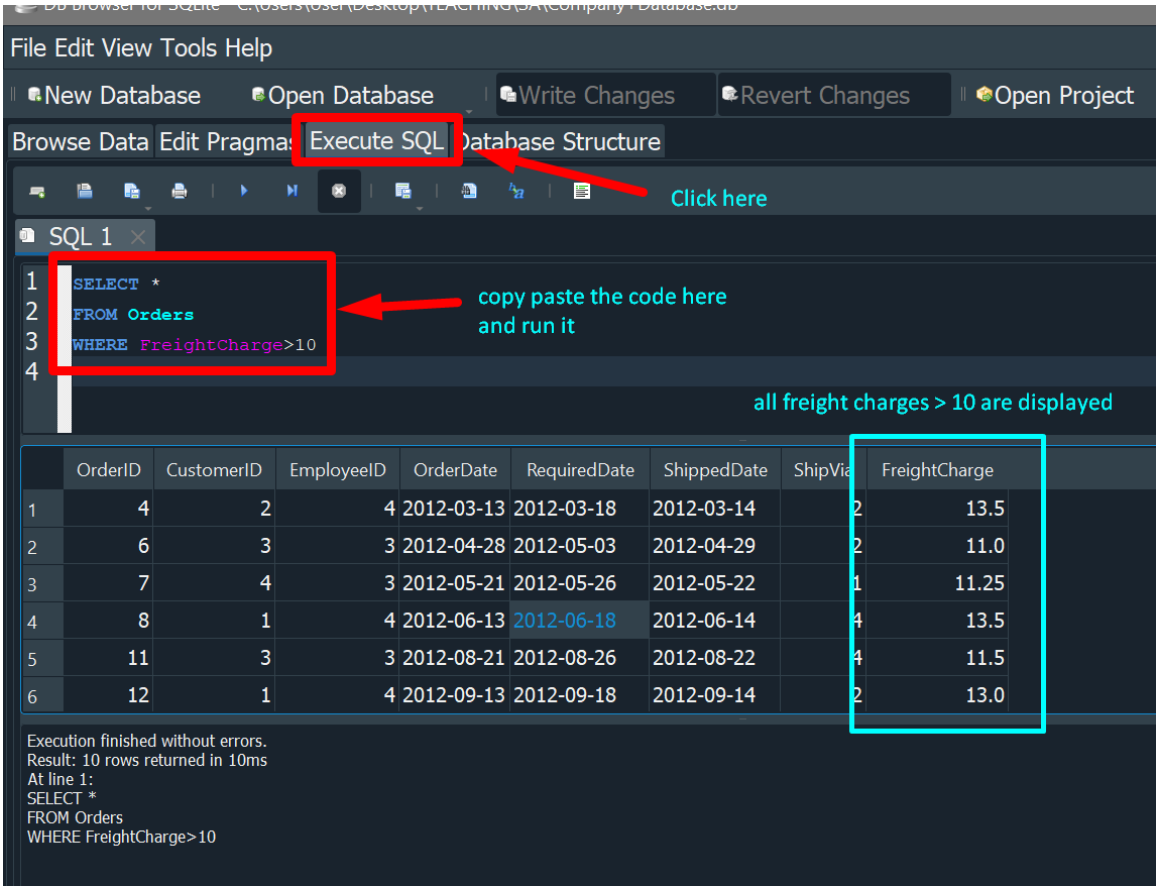
- Select * means Select all columns
- From the Orders Table
- All the columns in Orders Table is taken out for display only.
- The original database is left untouched – nothing has been changed within the tables.

D. WHERE

1. ORDERS WITH FREIGHT CHARGE MORE THAN $10

a) Code Used

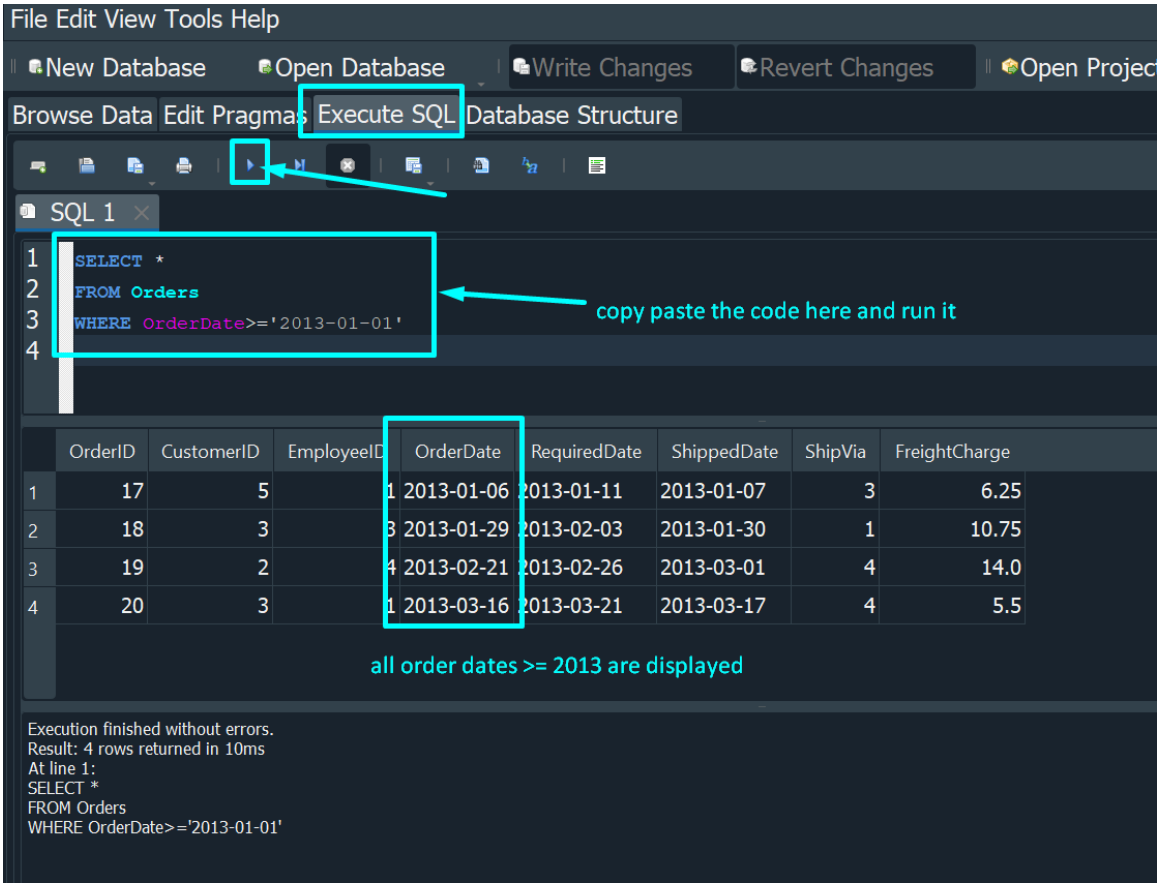
```
SELECT *

FROM Orders

WHERE FreightCharge>10
```



2. ORDERS ON AND AFTER THE 1ST JAN 2013

a) Code Used

```
SELECT *

FROM Orders

WHERE OrderDate>='2013-01-01'
```

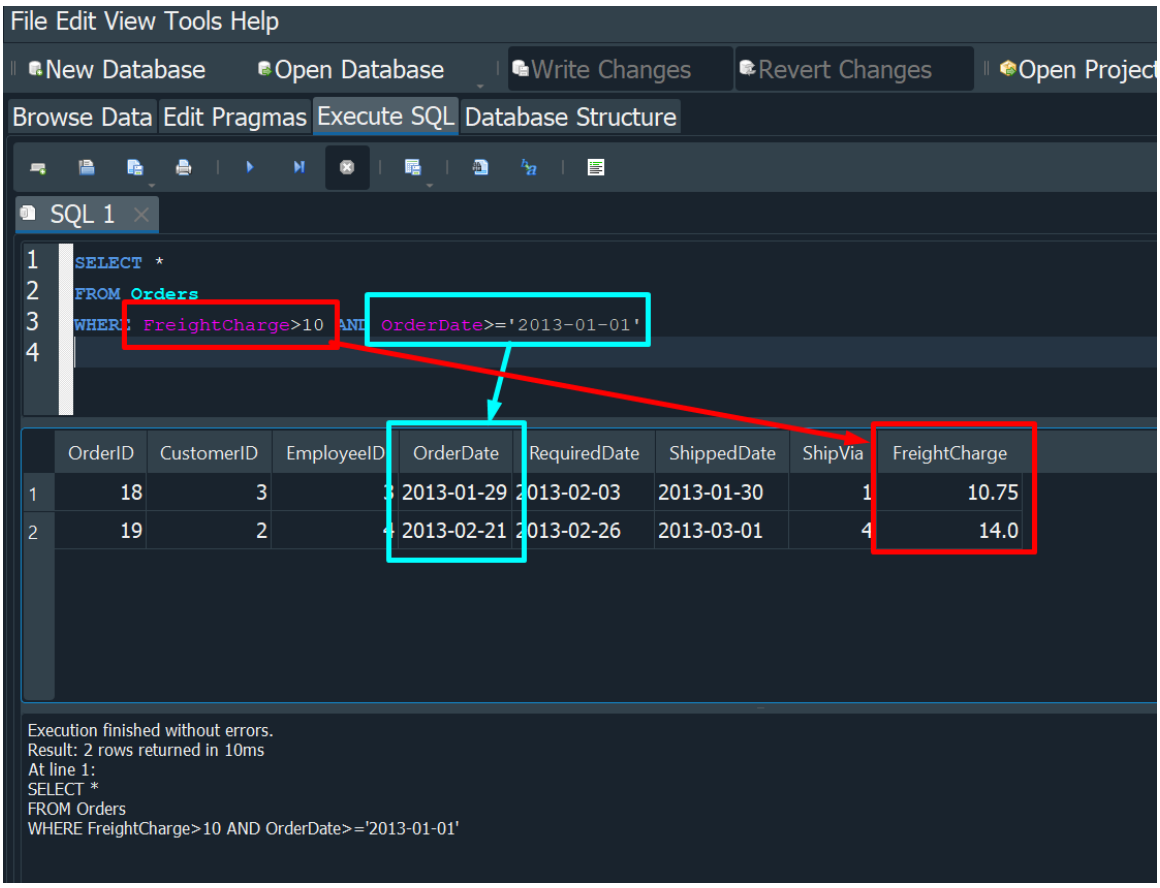


E. AND

1. ORDERS WITH FREIGHT CHARGE MORE THAN $10 AND ON AND AFTER THE 1ST JAN 2013

a) Code Used

```
SELECT *

FROM Orders

WHERE FreightCharge>10 AND OrderDate>='2013-01-01'
```

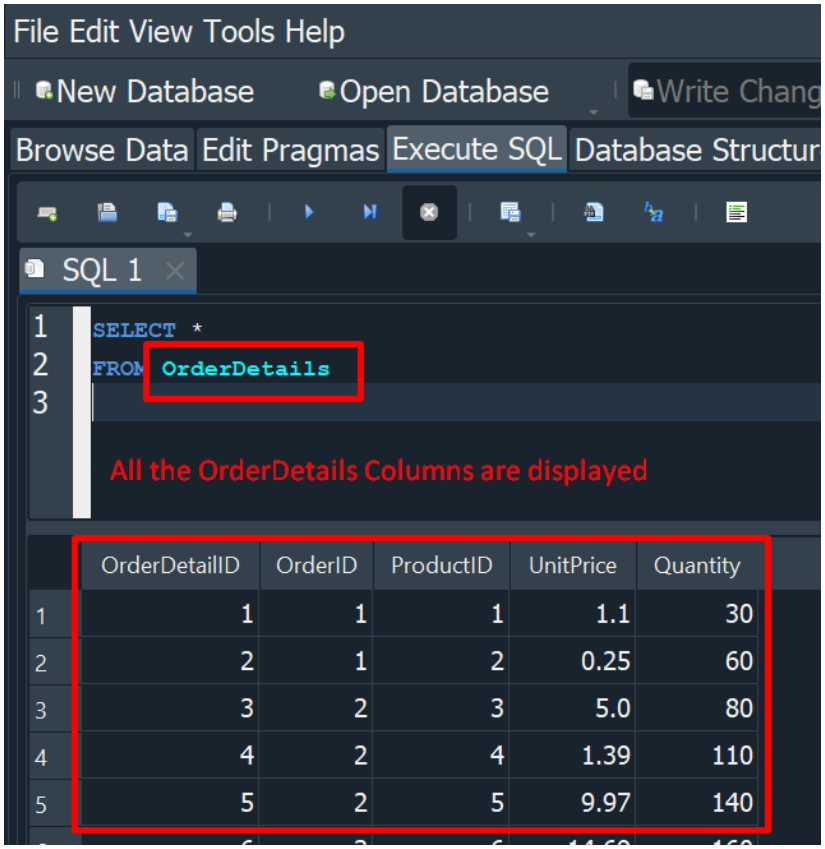


F. OR

1. ORDER DETAILS TABLE

a) Code Used

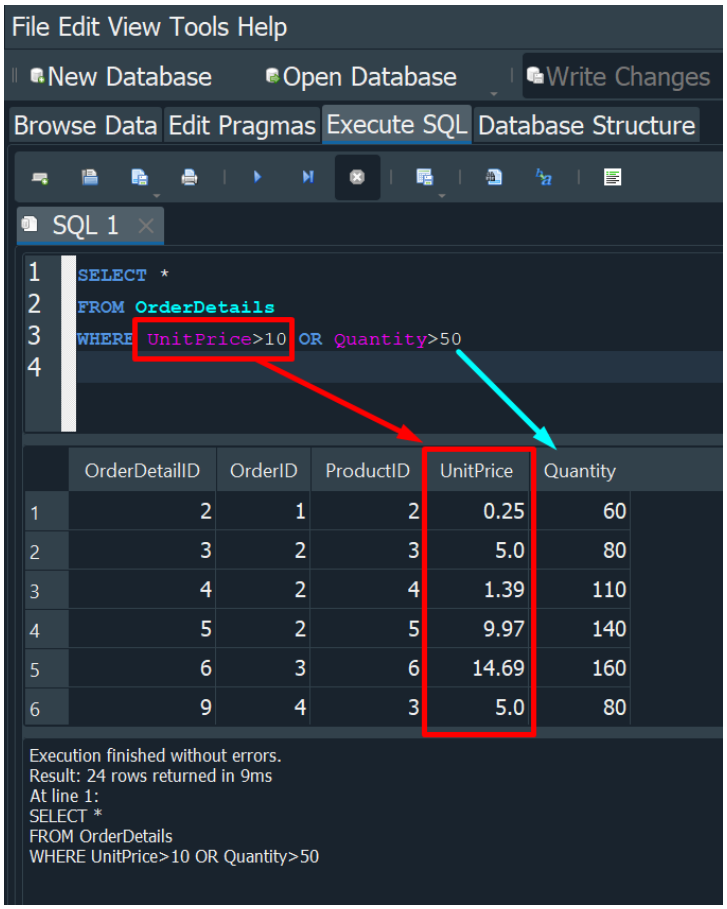
```
SELECT *

FROM OrderDetails
```



2. ORDER DETAILS WITH UNIT PRICE MORE THAN 10 OR QUANTITY MORE THAN 50

a) Code Used

```
SELECT *

FROM OrderDetails

WHERE UnitPrice>10 OR Quantity>50
```



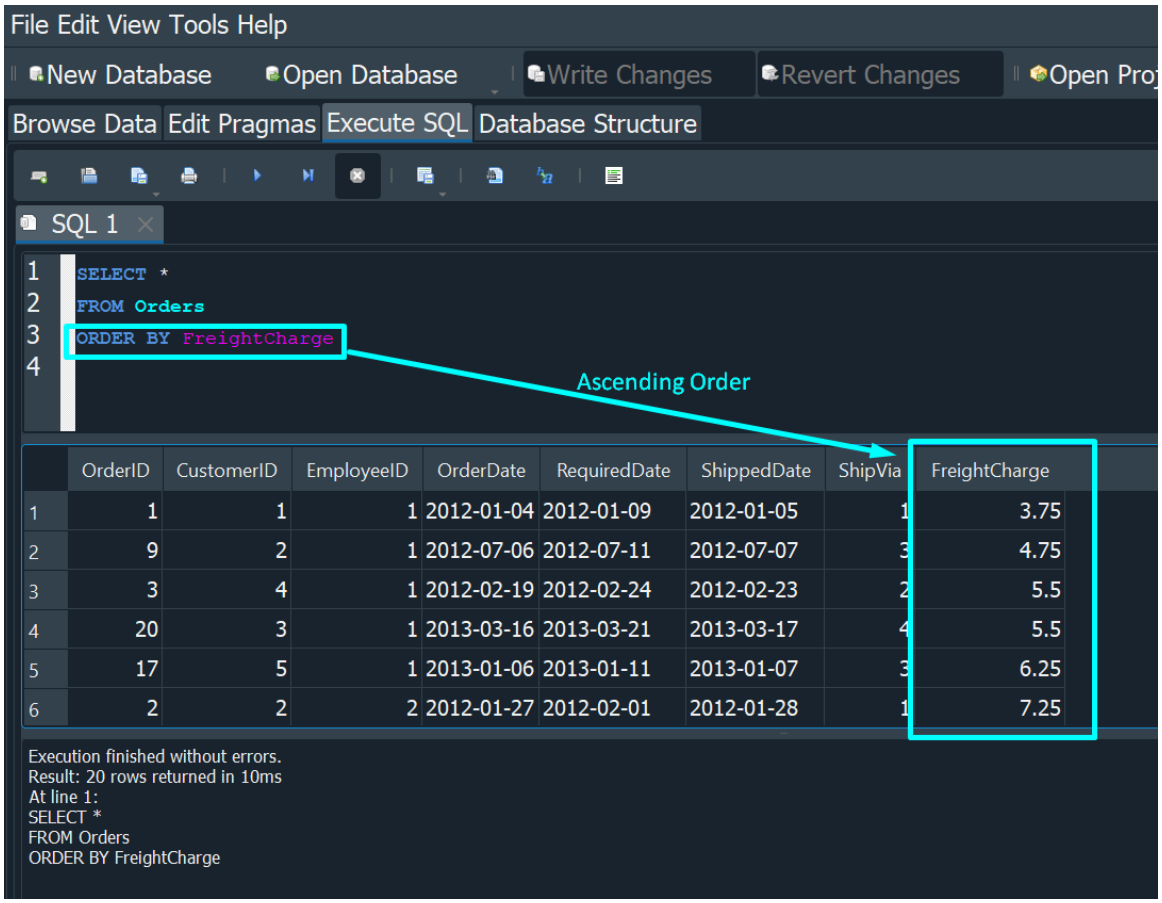
- Notice that because its OR, even if UnitPrice < 10 will be displayed as long as Qty > 50

G. ORDER BY

1. ORDER BY FREIGHT CHARGE

a) Code Used

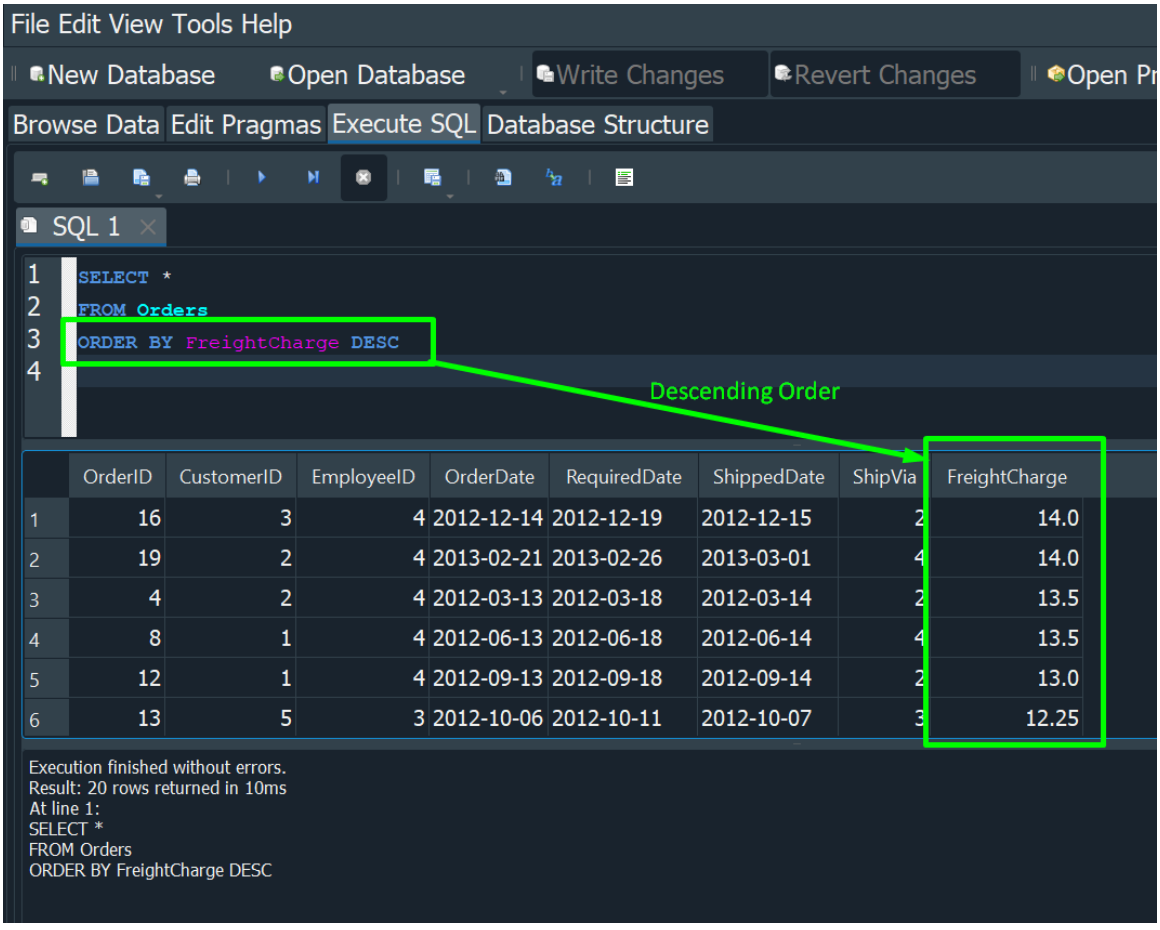
```
SELECT *

FROM Orders

ORDER BY FreightCharge
```



2. ORDER BY FREIGHT CHARGE WITH HIGHEST COST FIRST

a) Code Used

```
SELECT *

FROM Orders

ORDER BY FreightCharge DESC
```

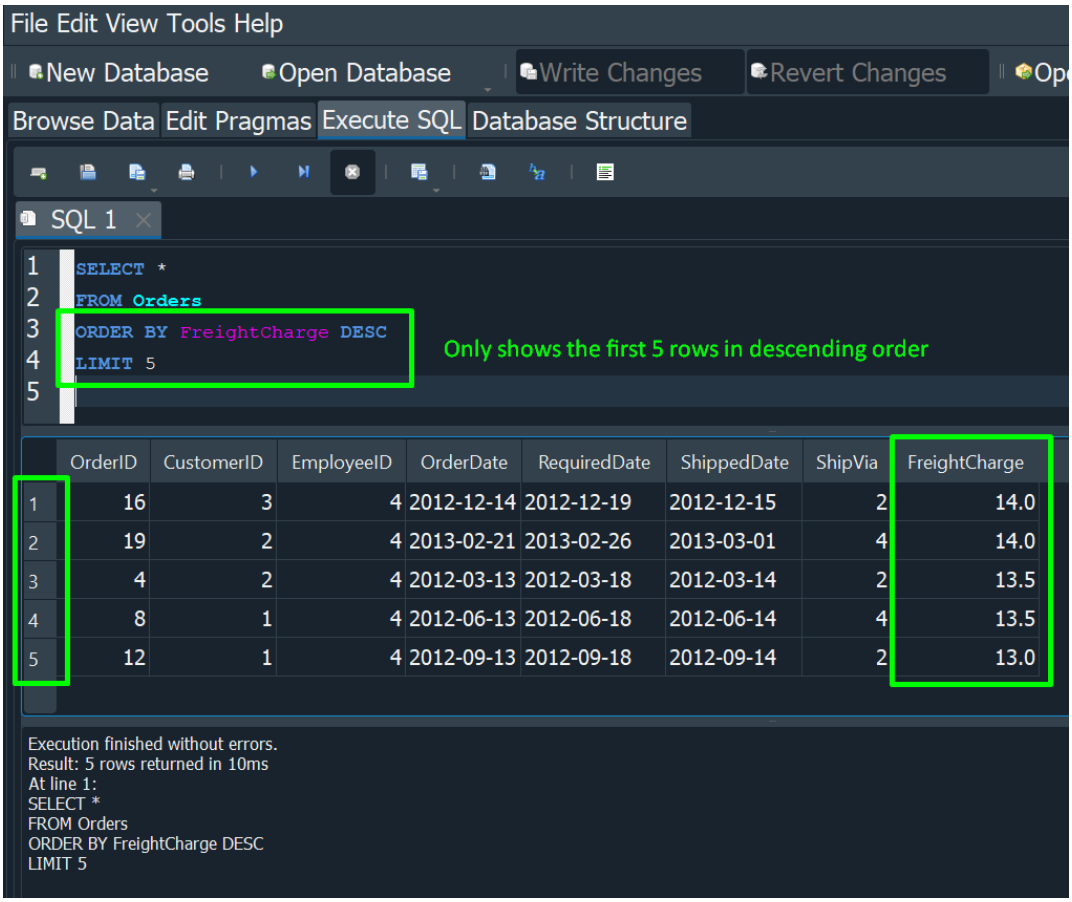


H. TOP/LIMIT

1. SELECT THE TOP 5 HIGHEST FREIGHT CHARGE ORDERS

a) Code Used

```
SELECT *

FROM Orders

ORDER BY FreightCharge DESC

LIMIT 5
```

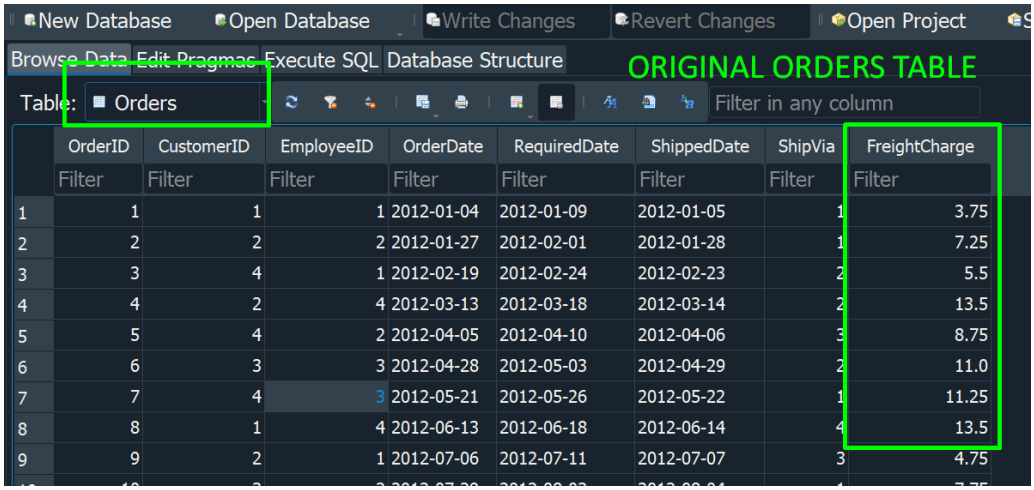


I. AVG

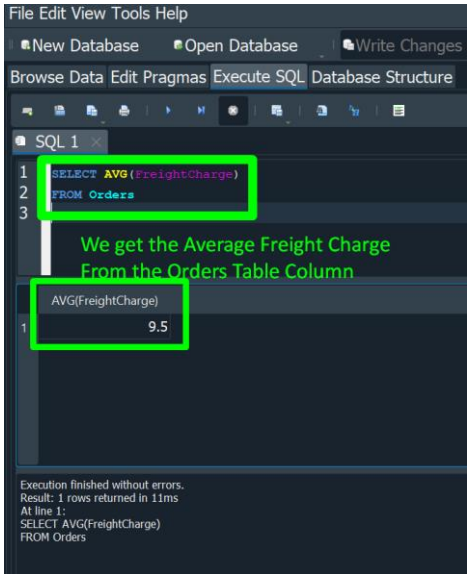
1. AVERAGE FREIGHT CHARGE

a) Code Used

```
SELECT AVG(FreightCharge)

FROM Orders
```



	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	FreightCharge
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	1	2012-01-04	2012-01-09	2012-01-05	1	3.75
2	2	2	2	2012-01-27	2012-02-01	2012-01-28	1	7.25
3	3	4	1	2012-02-19	2012-02-24	2012-02-23	2	5.5
4	4	2	4	2012-03-13	2012-03-18	2012-03-14	2	13.5
5	5	4	2	2012-04-05	2012-04-10	2012-04-06	3	8.75
6	6	3	3	2012-04-28	2012-05-03	2012-04-29	2	11.0
7	7	4	3	2012-05-21	2012-05-26	2012-05-22	1	11.25
8	8	1	4	2012-06-13	2012-06-18	2012-06-14	4	13.5
9	9	2	1	2012-07-06	2012-07-11	2012-07-07	3	4.75

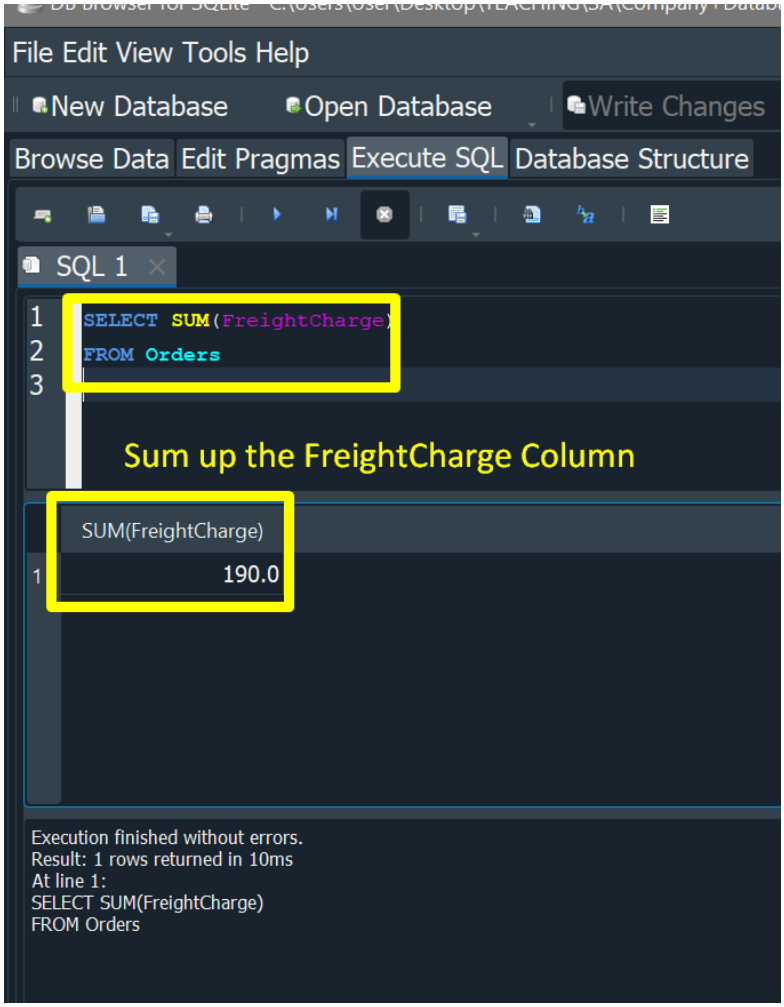


J. SUM

1. TOTAL FREIGHT CHARGE

a) Code Used

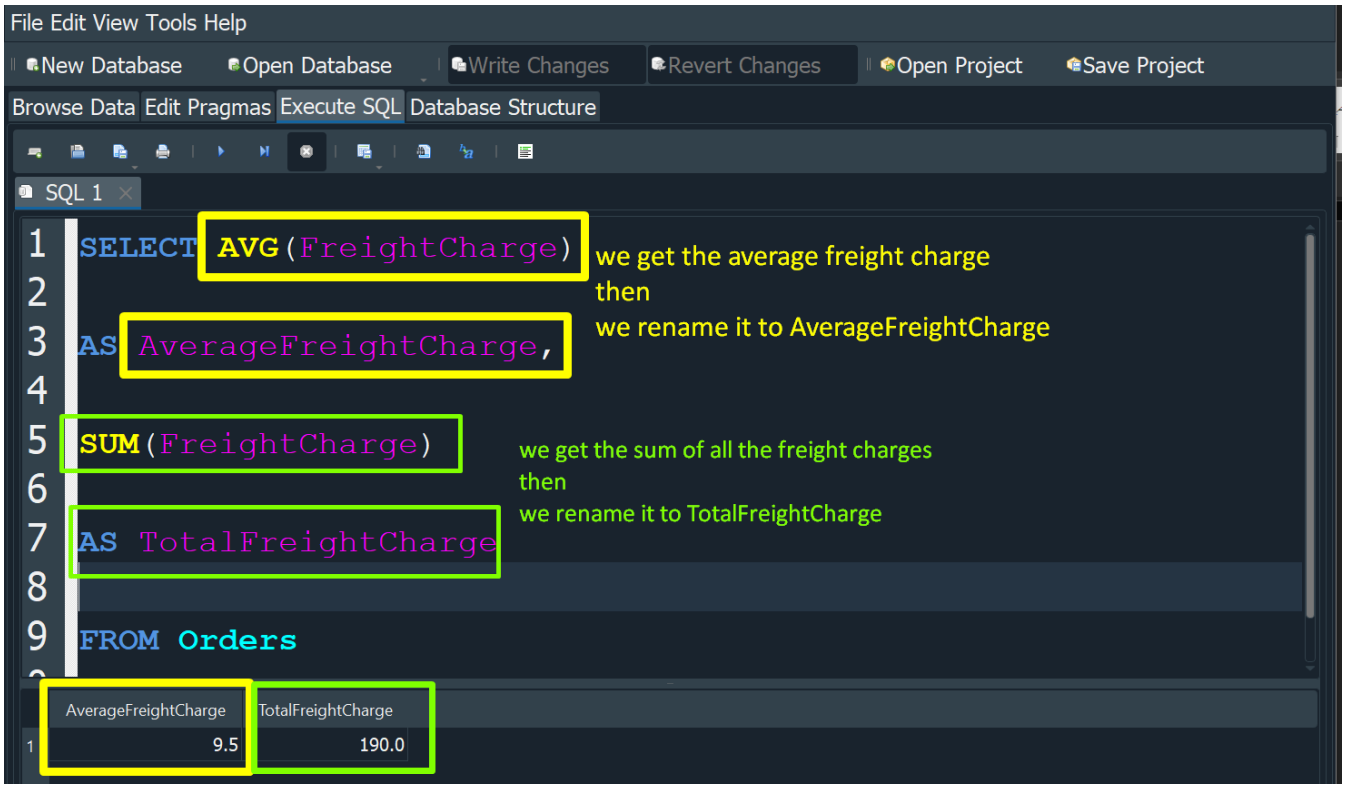
```
SELECT SUM(FreightCharge)

FROM Orders
```



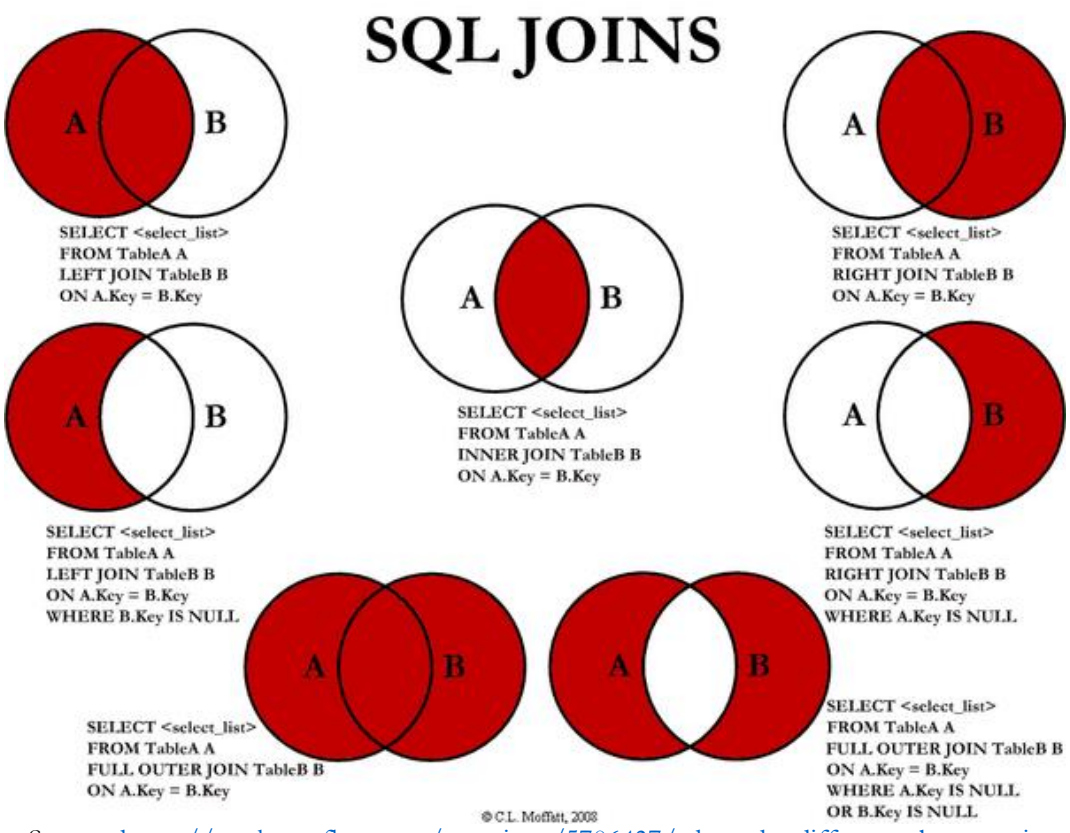
K. AS (RENAME)

a) Code Used

```
SELECT AVG(FreightCharge)

AS AverageFreightCharge,

SUM(FreightCharge)

AS TotalFreightCharge

FROM Orders
```



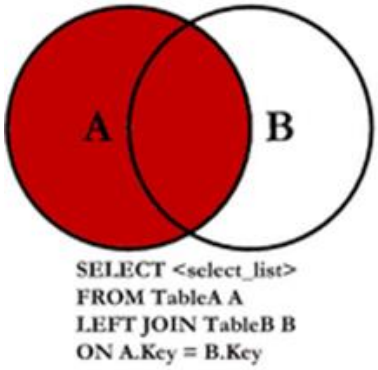
L. JOINS



Source: https://stackoverflow.com/questions/5706437/whats-the-difference-between-inner-join-left-join-right-join-and-full-join

- INNER JOIN: returns rows when there is a match in both tables.
- LEFT JOIN: returns all rows from the left table, even if there are no matches in the right table.
- RIGHT JOIN: returns all rows from the right table, even if there are no matches in the left table.
- FULL JOIN: combines the results of both left and right outer joins. The joined table will contain all records from both the tables and fill in NULLs for missing matches on either side.
- SELF JOIN: joins a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.

1. LEFT JOIN



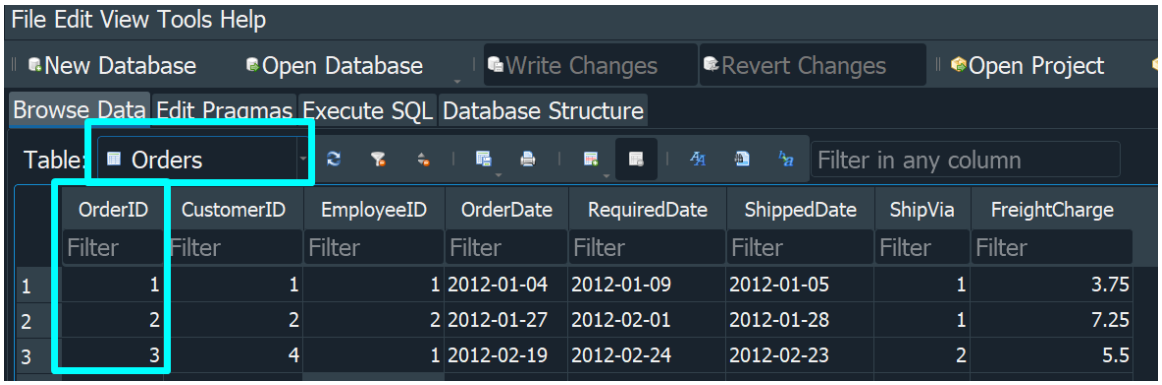
a) Code Used

```
SELECT Orders.OrderID, Customers.ContactName

FROM Orders

LEFT JOIN Customers

ON Orders.CustomerID=Customers.CustomerID
```

- Table A: Orders Table
    - We want the *OrderID column* from the 'Orders Table'
- Table B: Customers Table
    - We want the *ContactName column* from the 'Customers Table'
- Left Join:
    - Entire *OrderID column* from Table A (the 'Orders Table') will be displayed
- Key: CustomerID
    - Both Table A and Table B share the same Key (CustomerID column) → we need to match this column to join both tables.

b) Original Orders Table (Table A)

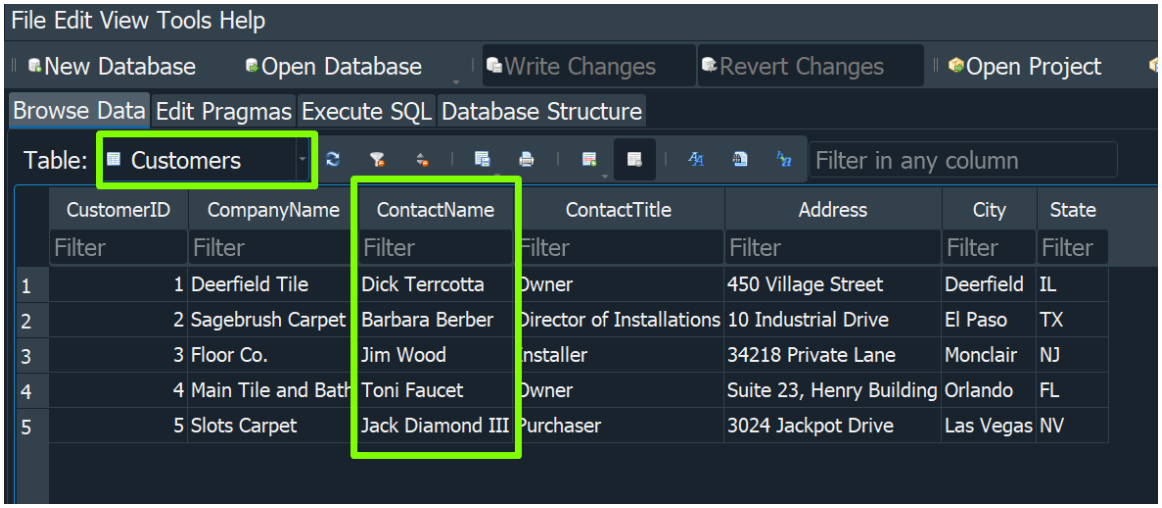
- We select the Orders → OrderID column.



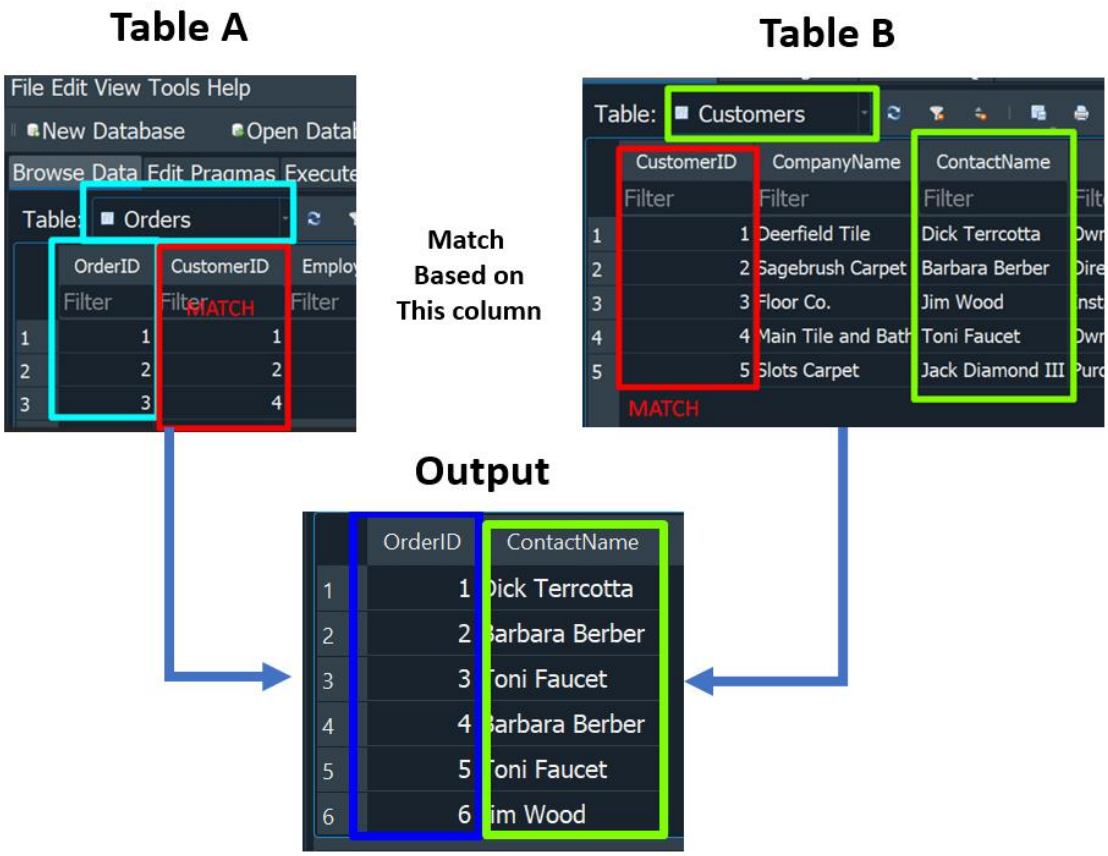
	OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	FreightCharge
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	1	2012-01-04	2012-01-09	2012-01-05	1	3.75
2	2	2	2	2012-01-27	2012-02-01	2012-01-28	1	7.25
3	3	4	1	2012-02-19	2012-02-24	2012-02-23	2	5.5

c) Original Customers Table (Table B)

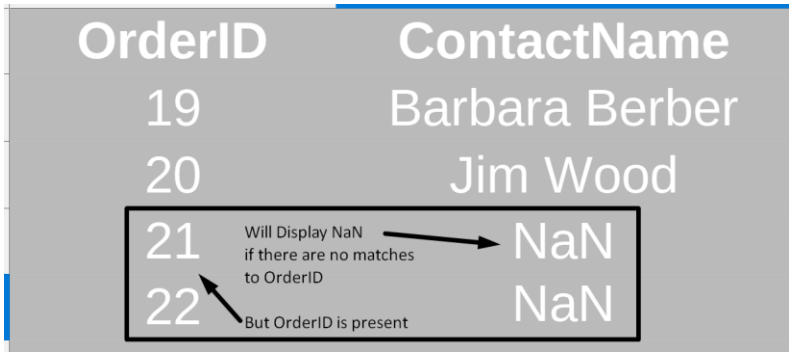
- We select the Customers → Contact Name column



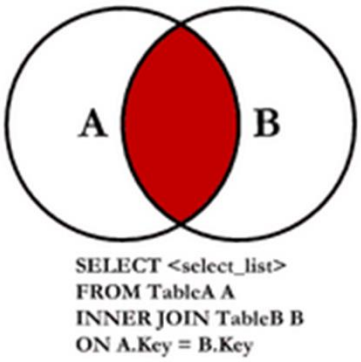
	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	State
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Deerfield Tile	Dick Terrcotta	Owner	450 Village Street	Deerfield	IL
2	2	Sagebrush Carpet	Barbara Berber	Director of Installations	10 Industrial Drive	El Paso	TX
3	3	Floor Co.	Jim Wood	Installer	34218 Private Lane	Monclair	NJ
4	4	Main Tile and Bath	Toni Faucet	Owner	Suite 23, Henry Building	Orlando	FL
5	5	Slots Carpet	Jack Diamond III	Purchaser	3024 Jackpot Drive	Las Vegas	NV



- Note that the CustomerID column will not be displayed.
- Note that the ENTIRE OrderID column will be displayed even though there might not be any ContactNames linked to it.
- For example….



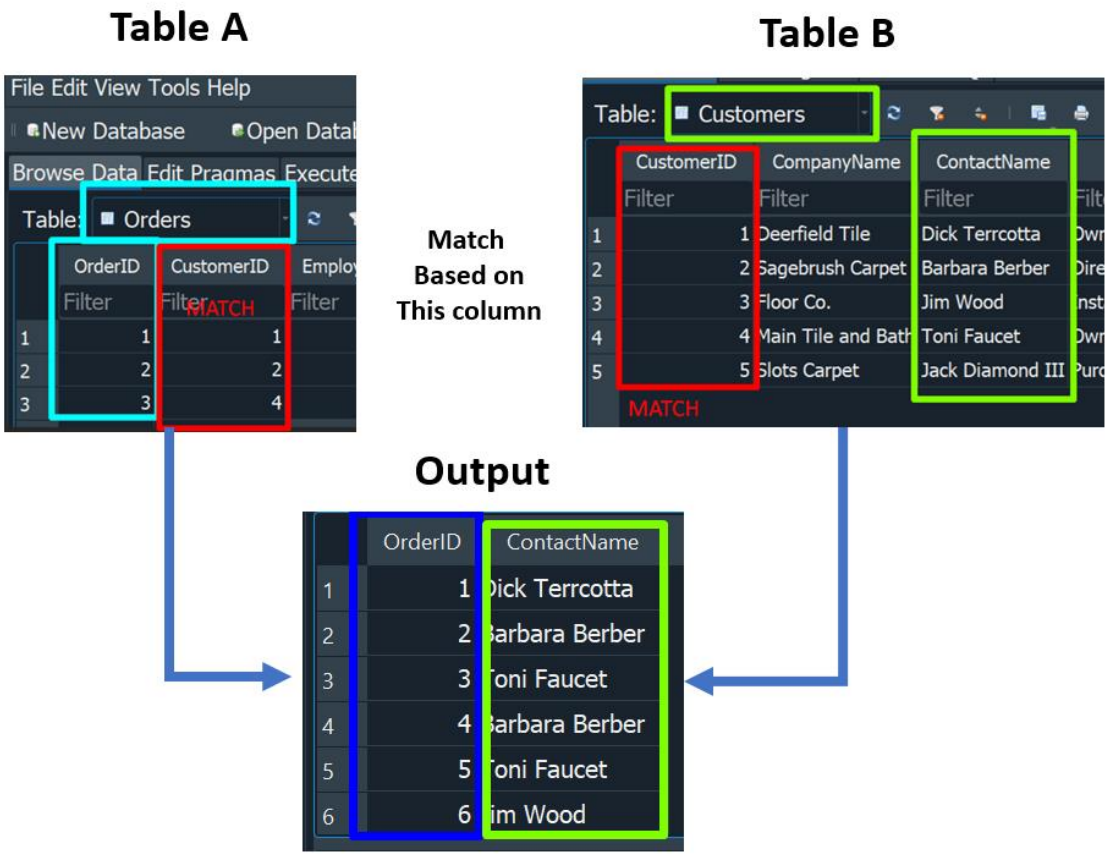
2. INNER JOIN



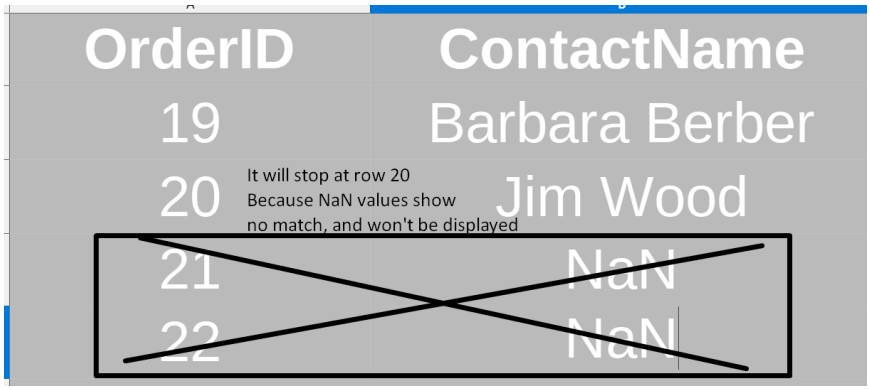
a) Code Used

```
SELECT Orders.OrderID , Customers.ContactName

FROM Orders

INNER JOIN Customers

ON Orders.CustomerID =Customers.CustomerID
```

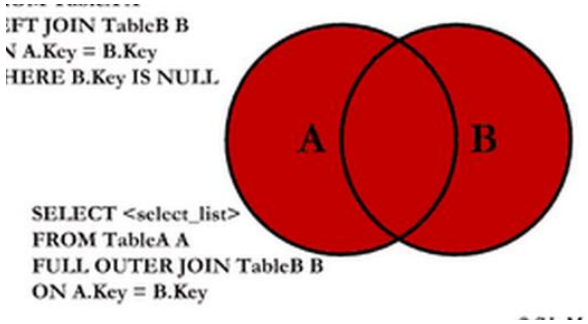
- Table A: Orders Table
    - We want the *OrderID column* from the 'Orders Table'
- Table B: Customers Table
    - We want the *ContactName column* from the 'Customers Table'
- Inner Join:
    - Only Matching Values from CustomerID column will be displayed
- Key: CustomerID
    - Both Table A and Table B share the same Key (CustomerID column) → we need to match this column to join both tables.



- Note that the CustomerID column will not be displayed.
- Note that only MATCHING values of the CustomerID column will be displayed.
- For example….

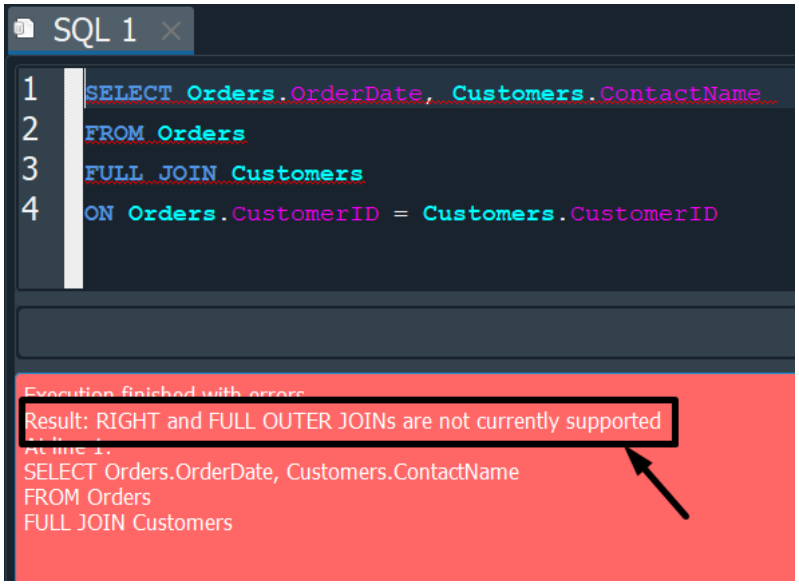


3. FULL JOIN



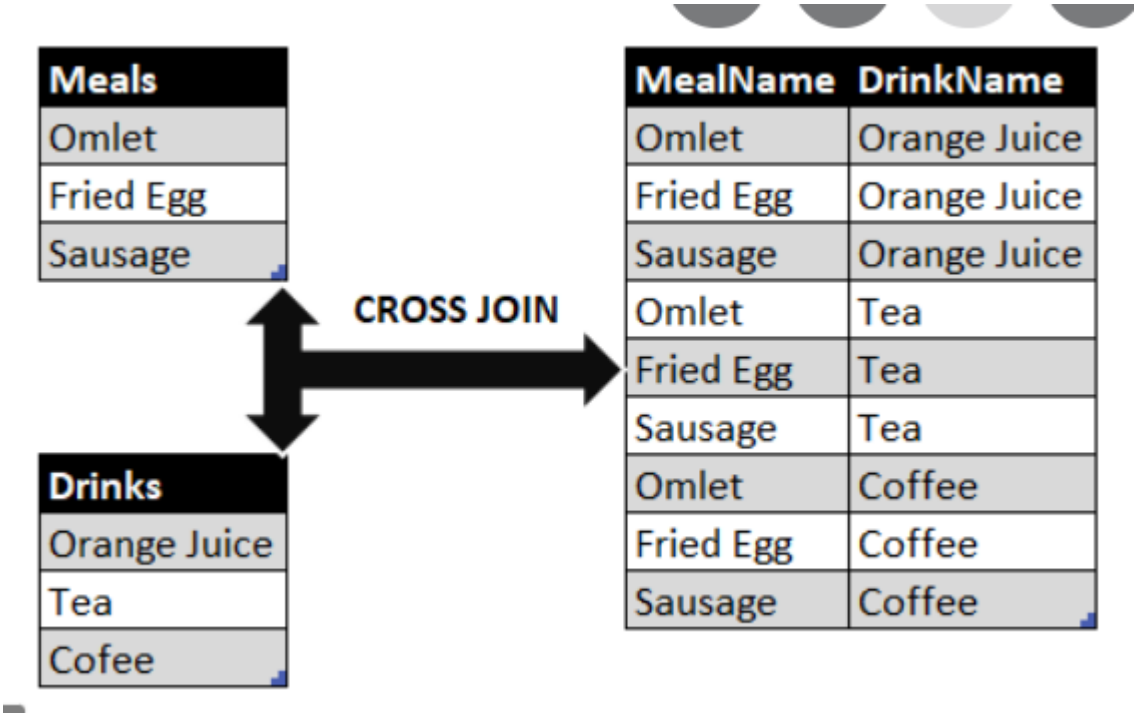
a) Code Used

```
SELECT Orders.OrderDate, Customers.ContactName
FROM Orders
FULL JOIN Customers
ON Orders.CustomerID = Customers.CustomerID
```



Note that currently, DB for SQLite does not support RIGHT and FULL JOINS.

4. CROSS JOIN



This is how Cross Join Works

- It creates a combination of every kind from both tables
- You see that Orange Juice is combined with every Meal
- You see that Tea is combined with every Meal.
- And so forth…

a) Lists all Employees

SELECT FirstName AS EmployeeName

FROM Employees

Table: Employees

	EmployeeID	LastName	FirstName	Title	Address	HireDate
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	White	James	Account Manager		2011-04-03
2	2	Lee	Patty	Account Manager	NULL	2008-09-15
3	3	Smith	Robert	Account Manager	NULL	2004-06-28
4	4	Baker	Lisa	Account Manager	NULL	2010-11-20

SQL 1

```
1  SELECT FirstName AS EmployeeName
2  FROM Employees
3
```

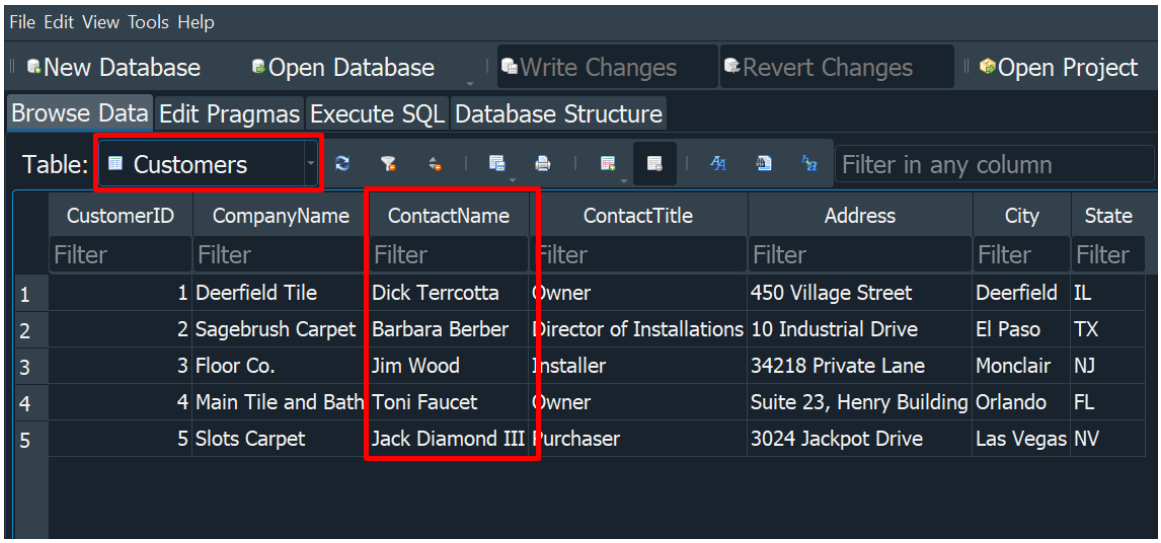
	EmployeeName
1	James
2	Patty
3	Robert
4	Lisa

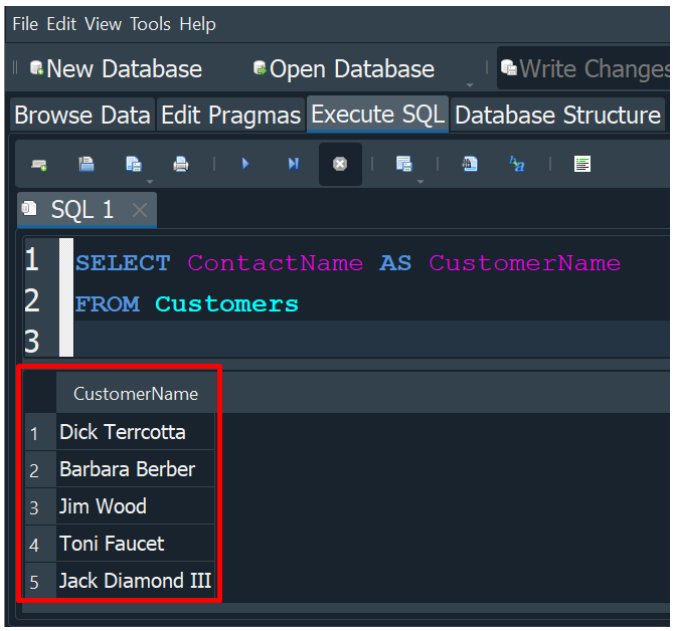
Execution finished without errors.
Result: 4 rows returned in 11ms
At line 1:
SELECT FirstName AS EmployeeName
FROM Employees

b) Lists all Customers

SELECT ContactName AS CustomerName

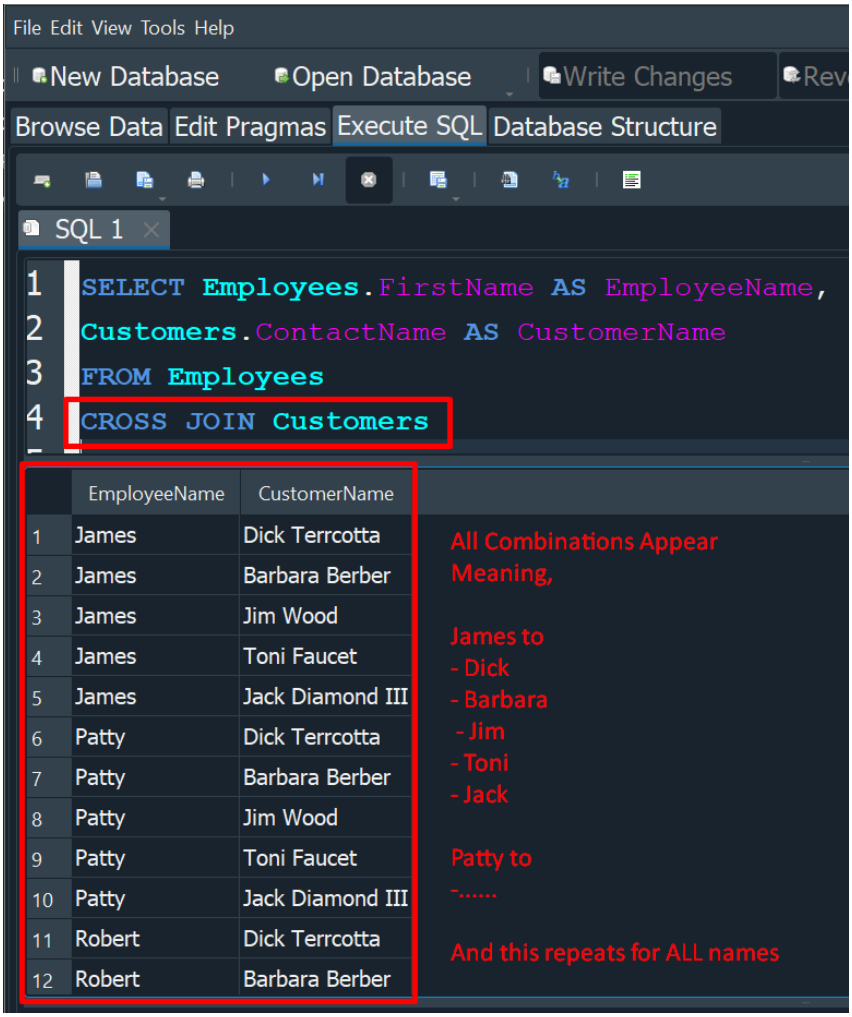
FROM Customers





c) Lists all combinations of Employees and Customers

```
SELECT Employees.FirstName AS EmployeeName,

Customers.ContactName AS CustomerName

FROM Employees

CROSS JOIN Customers
```



M. ARITHMETIC OPERATORS

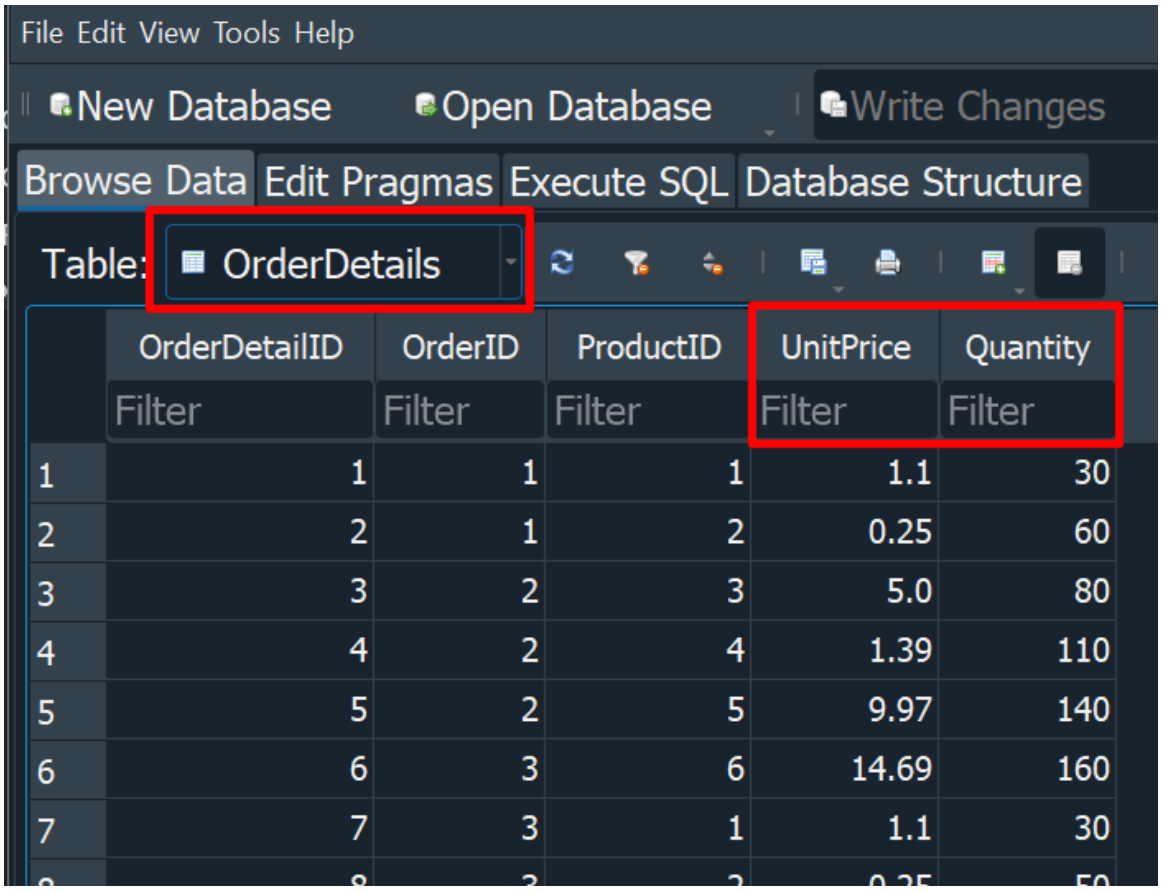
1. TOTAL PRICE OF EACH ITEM IN THE "ORDERDETAILS" TABLE

a) Code Used

```
SELECT *, UnitPrice*Quantity AS Price

FROM OrderDetails
```

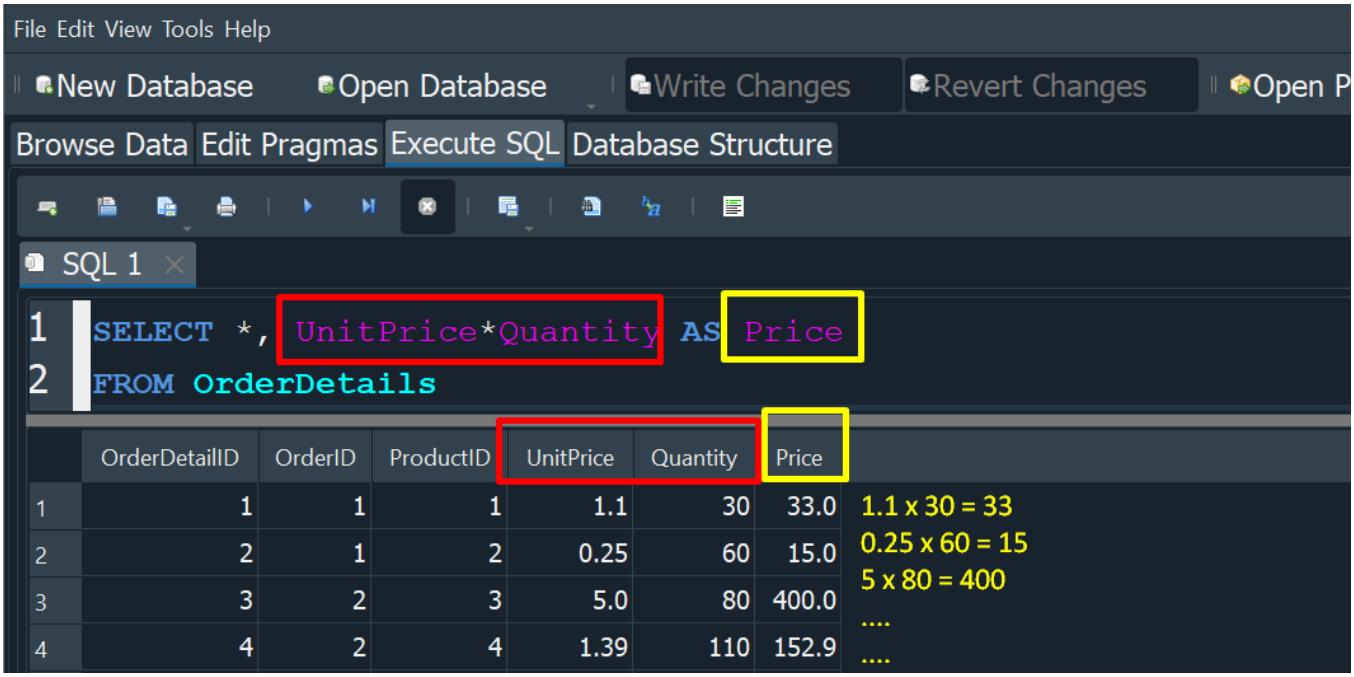
b) Original Table

- This is the original OrderDetails Table.



	OrderDetailID	OrderID	ProductID	UnitPrice	Quantity
	Filter	Filter	Filter	Filter	Filter
1	1	1	1	1.1	30
2	2	1	2	0.25	60
3	3	2	3	5.0	80
4	4	2	4	1.39	110
5	5	2	5	9.97	140
6	6	3	6	14.69	160
7	7	3	1	1.1	30

c) Output

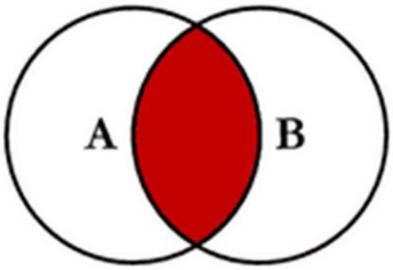


N. GROUP BY + ORDER BY

1. COUNT THE NUMBER OF ORDERS EACH CUSTOMER HAS MADE

a) Code Used

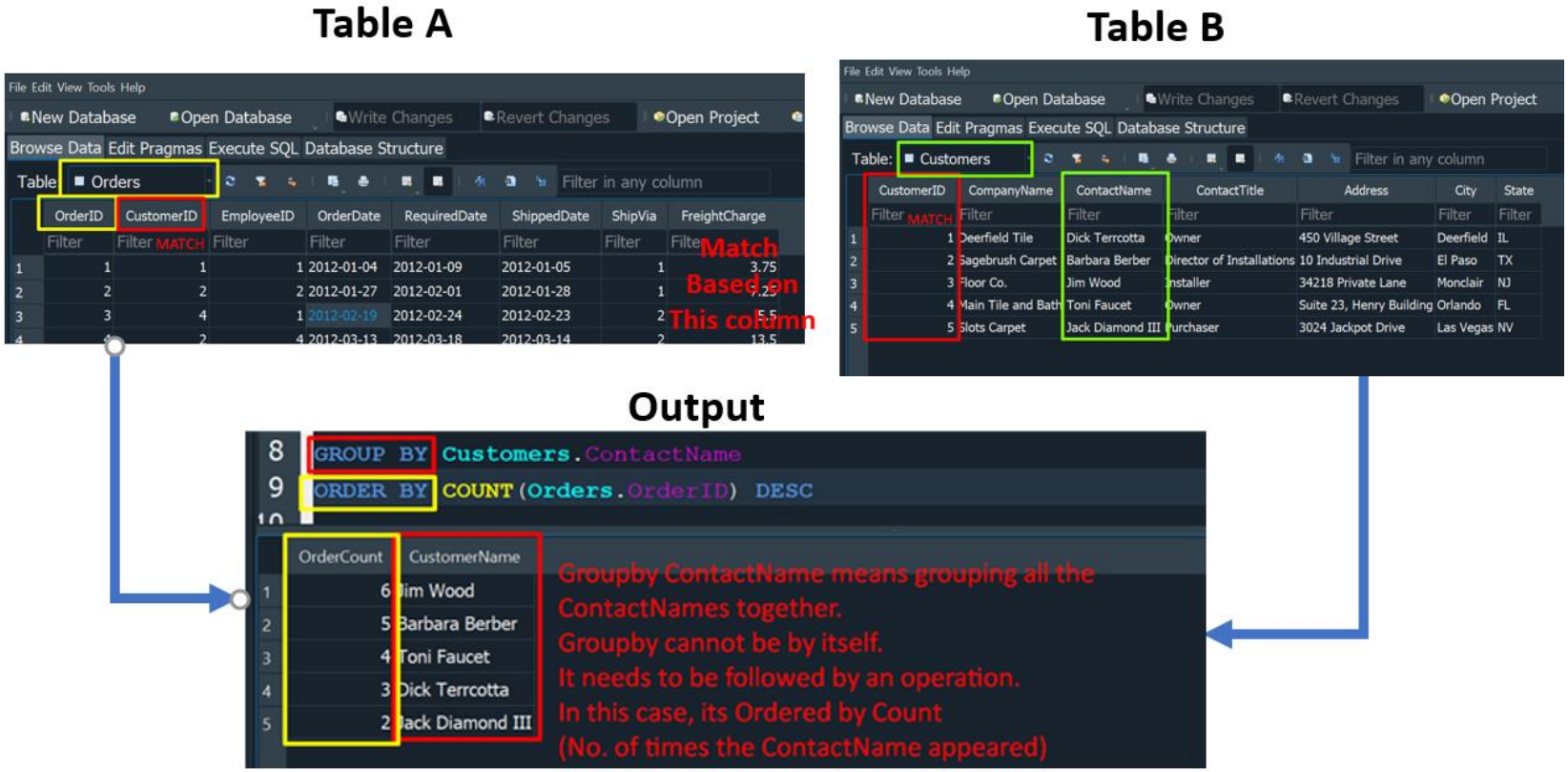
```
SELECT

COUNT(Orders.OrderID) AS OrderCount ,

Customers.ContactName AS CustomerName,


FROM Orders

INNER JOIN Customers ON Customers.CustomerID=Orders.CustomerID


GROUP BY Customers.ContactName

ORDER BY COUNT(Orders.OrderID) DESC
```




```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

- Table A: Orders Table
    - We want the *OrderID column* from the 'Orders Table'
- Table B: Customers Table
    - We want the *ContactName column* from the 'Customers Table'
- Inner Join:
    - Only Matching Values from CustomerID column will be displayed
- Key: CustomerID
    - Both Table A and Table B share the same Key (CustomerID column) → we need to match this column to join both tables.
- Groupby: ContactName
    - We group all the same 'ContactNames' together
- Orderby: Count of OrderID
    - Groupby cannot be used by itself, it needs to be followed by an operation.
    - We count the number of times each 'ContactName' appeared (based on tallying its CustomerID) in the 'Orders' Table; and put it in Descending Order.

b) Output



ABOUT DR ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He was previously a Professor, Scientist and Financial Consultant. Currently, he owns multiple startups and is a Personal/Business Advisor.

More about him at www.AlvinAng.sg