

1. Get USA and UK Customer's List and their Contact Information. (Table: Customers)

```
SELECT CustomerID, ContactName, Phone, Fax, Country, Region FROM Customers WHERE Country IN ('USA', 'UK')
```

The screenshot shows the Azure Data Studio interface. The top navigation bar includes File, Edit, View, Window, Help, and a status bar showing the date and time. The left sidebar has sections for CONNECTIONS, SERVERS, and AZURE. The main area displays a query window with the following content:

```
SQLQuery_2.sql - disconnected
Users > ashminbandari > Downloads > Training > DAY2_SQL (extract.me) > SQLQuery_2.sql
Run Cancel Connect Change Connection Select Database
1 SELECT CustomerID, ContactName, Phone, Fax, Country, Region FROM Customers WHERE Country IN ('USA', 'UK')
```

The results pane shows a table with 16 rows of customer data. The columns are: CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, and PostalCode. The data includes various companies like Around the Horn, B's Beverages, and Lonesome Pine Restaurant, located in cities like London, Eugene, Elgin, and Walla Walla, with regions ranging from WA1 1 to WY1 6.

CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	
1	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	NULL	WA1 1
2	BSBEV	B's Beverages	Victoria Ashworth	Sales Representative	Fauntleroy Circus	London	NULL	EC2 5
3	CONSH	Consolidated Holdings	Elizabeth Brown	Sales Representative	Berkeley Gardens 12 Brew...	London	NULL	WX1 6
4	EASTC	Eastern Connection	Ann Devon	Sales Agent	35 King George	London	NULL	WX3 6
5	GREAL	Great Lakes Food Market	Howard Snyder	Marketing Manager	2732 Baker Blvd.	Eugene	OR	97403
6	HUNGC	Hungry Coyote Import Store	Yoshi Latimer	Sales Representative	City Center Plaza 516 Mai...	Elgin	OR	97827
7	ISLAT	Island Trading	Helen Bennett	Marketing Manager	Garden House Crowther Way	Cowes	Isle of Wight	P031
8	LAZYK	Lazy K Kountry Store	John Steel	Marketing Manager	12 Orchestra Terrace	Walla Walla	WA	99362
9	LETSS	Let's Stop N Shop	Jaime Yorres	Owner	87 Polk St. Suite 5	San Francisco	CA	94117
10	LONEP	Lonesome Pine Restaurant	Fran Wilson	Sales Manager	89 Chiaroscuro Rd.	Portland	OR	97219
11	NORTS	North/South	Simon Crowther	Sales Associate	South House 300 Queensbri...	London	NULL	SW7 1
12	OLDWO	Old World Delicatessen	Rene Phillips	Sales Representative	2743 Bering St.	Anchorage	AK	99508
13	RATTG	Rattlesnake Canyon Grocery	Paula Wilson	Assistant Sales Represent...	2817 Milton Dr.	Albuquerque	NM	87110
14	SAVEA	Save-a-lot Markets	Jose Pavarotti	Sales Representative	187 Suffolk Ln.	Boise	ID	83720
15	SEVES	Seven Seas Imports	Hari Kumar	Sales Manager	90 Wadhurst Rd.	London	NULL	OX15
16	SPLIR	Split Rail Beer & Ale	Art Braunschweiger	Sales Manager	P.O. Box 555	Lander	WY	82520

At the bottom, there are various system icons and a status bar indicating the number of selected rows (105), spaces used (4), and the execution time (00:00:08).

2. Get Customer List who are from USA or from SP Region of Brazil. (Table: Customers)

```
SELECT CustomerID, ContactName, Country, Region FROM Customers  
WHERE Country = 'USA' OR Region = 'SP'
```

The screenshot shows a SQL query execution interface. At the top, there are tabs for 'Results' and 'Messages'. The 'Results' tab is selected, displaying a table of customer data. The table has columns: CustomerID, CompanyName, ContactName, and ContactTitle. The data consists of 30 rows, each representing a different customer. The 'Messages' tab is also visible at the top.

CustomerID	CompanyName	ContactName	ContactTitle
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner
AROUT	Around the Horn	Thomas Hardy	Sales Representative
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative
BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager
BOLID	Bólido Comidas preparadas	Martín Sommer	Owner
BSBEV	B's Beverages	Victoria Ashworth	Sales Representative
CHOPS	Chop-suey Chinese	Yang Wang	Owner
DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Picker
ERNSH	Ernst Handel	Roland Mendel	Sales Representative
FRANK	Frankenversand	Peter Franken	Marketing Manager
FRANR	France restauration	Carine Schmitt	Marketing Manager
FRANS	Franchi S.p.A.	Paolo Accorti	Sales Representative
GREAL	Great Lakes Food Market	Howard Snyder	Marketing Manager
GROSR	GROSELLA-Restaurante	Manuel Pereira	Owner
ISLAT	Island Trading	Helen Bennett	Marketing Manager
KOENE	Königlich Essen	Philip Cramer	Sales Representative
MEREP	Mère Paillard	Jean Fresnière	Marketing Manager
OCEAN	Océano Atlántico Ltda.	Yvonne Moncada	Sales Representative
OLDWO	Old World Delicatessen	Rene Phillips	Sales Representative
OTTIK	Ottiles Käseladen	Henriette Pfalzheim...	Owner
PRINI	Princesa Isabel Vinhos	Isabel de Castro	Sales Representative
SPEC'D	Spécialités du monde	Dominique Perrier	Marketing Manager
SPLIR	Split Rail Beer & Ale	Art Braunschweiger	Sales Representative
THEBI	The Big Cheese	Liz Nixon	Marketing Manager
THECR	The Cracker Box	Liu Wong	Marketing Manager
TRADH	Tradição Hipermercados	Anabela Domingues	Sales Representative

3. Get all the CustomerID and OrderID for order placed in third quarter of 1997. (Table: Orders)

```
SELECT CustomerID, OrderID FROM Orders WHERE YEAR(OrderDate) = 1997 AND Month(OrderDate) >= 7 AND Month(OrderDate) <= 9
```

```
SELECT CustomerID, OrderID FROM Orders WHERE YEAR(OrderDate) = 1997 AND Month(OrderDate) >= 7 AND Month(OrderDate) <= 9
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer displays a connection to 'database-1.c0lnneqcgsql.us-east-2.rds.amazonaws.com' with several saved queries listed. The main area shows the results of a query:

```
SELECT CustomerID, OrderID FROM Orders WHERE YEAR(OrderDate) = 1997 AND Month(OrderDate) >= 7 AND Month(OrderDate) <= 9
```

The results grid contains the following data:

	CustomerID	OrderID
1	WELLI	10585
2	REGGC	10586
3	QUEDA	10587
4	QUICK	10588
5	GREAL	10589
6	MEREP	10590
7	VAFFE	10591
8	LEHMS	10592
9	LEHMS	10593
10	OLDWO	10594
11	ERNSH	10595
12	WHITE	10596
13	PICCO	10597
14	RATTC	10598
15	BSBEV	10599
16	HUNG	10600
17	HILAA	10601
18	VAFFE	10602
19	SAVEA	10603
20	FURIB	10604
21	MEREP	10605
22	TRADH	10606
23	SAVEA	10607
24	TOMSP	10608
25	DUNON	10609
26	LAMAI	10610
27	WOLZA	10611
28	SAVEA	10612

Below the results, the SQL query is displayed again, followed by the output message: '10 rows affected'.

4. Get the Customer who's Company Name starts with either A, B, C, D, E, F, or G and ends with N or E. (Table: Customers)

```
SELECT * FROM Customers WHERE CompanyName LIKE '[abcdefg] %[ne]'
```

The screenshot shows the SSMS interface with the following details:

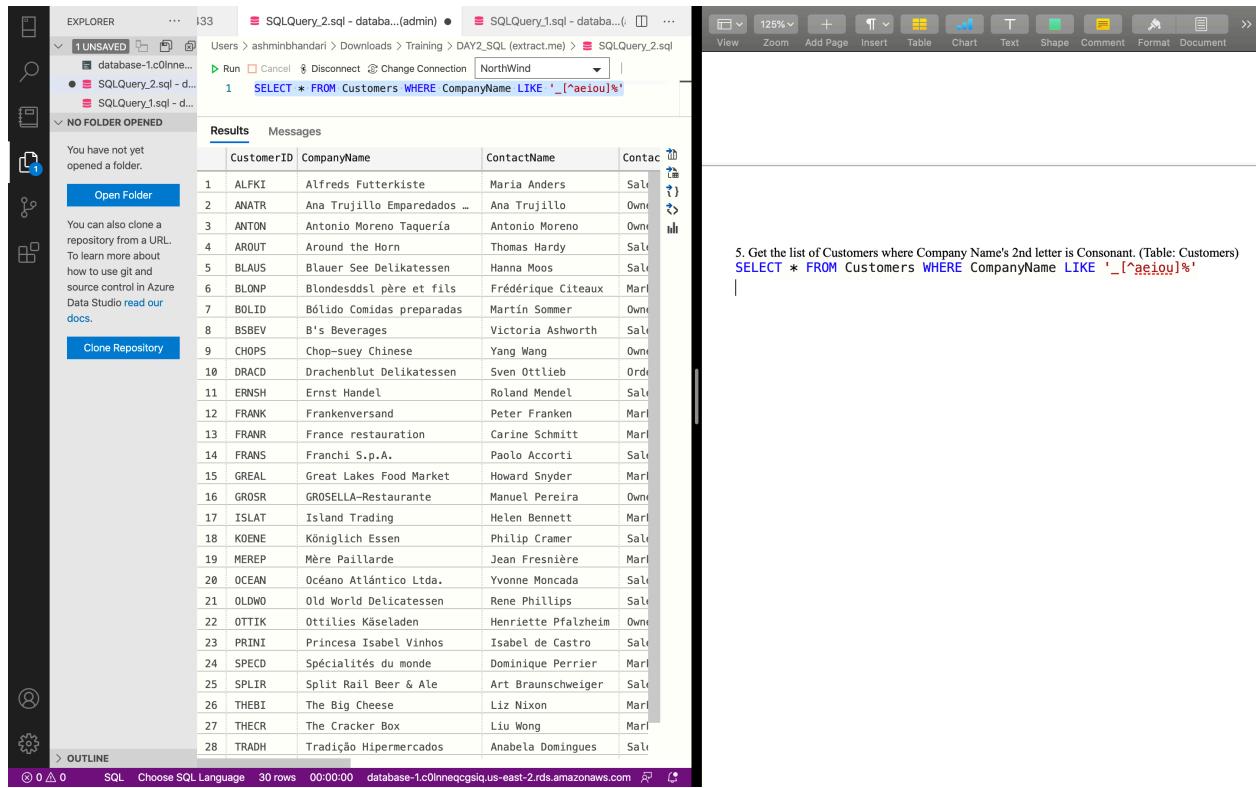
- EXPLORER** pane: Shows 1 UNSAVED file named "SQLQuery_2.sql".
- SQL Query Editor**:
 - Connection: database-1.c0lnneqcgsql.us-east-2.rds.amazonaws.com
 - Query: `SELECT * FROM Customers WHERE CompanyName LIKE '[abcdefg] %[ne]'`
 - Results pane:
 - Shows 8 rows of customer data from the Northwind database.
 - Columns: CustomerID, CompanyName, ContactName, ContactTitle.
 - Data:

CustomerID	CompanyName	ContactName	ContactTitle
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Represer
AROUT	Around the Horn	Thomas Hardy	Sales Represer
BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Represer
CHOPS	Chop-suey Chinese	Yang Wang	Owner
DRACD	Drachenblut Delikatessen	Sven Ottlieb	Order Administ
EASTC	Eastern Connection	Ann Devon	Sales Agent
FRANR	France restaurant	Carina Schmitt	Marketing Manag
GROSR	GROSELLA-Restaurante	Manuel Pereira	Owner
- Messages** pane: No messages displayed.
- Outline** pane: Shows the structure of the query.
- Bottom Status Bar**: Shows the connection status, number of rows (8 rows), and execution time (00:00:00).

4. Get the Customer who's Company Name starts with either A, B, C, D, E, F, or G and ends with N or E. (Table: Customers)

```
SELECT * FROM Customers WHERE CompanyName LIKE '[abcdefg] %[ne]'
```

5. Get the list of Customers where Company Name's 2nd letter is Consonant. (Table: Customers)
SELECT * FROM Customers WHERE CompanyName LIKE '_[aeiou]%'



The screenshot shows the SSMS interface with the following details:

- EXPLORER** pane: Shows 1 UNSAVED file named SQLQuery_2.sql.
- SQL Query Editor**:
 - Connection: ashminbandari > Downloads > Training > DAY2_SQL (extract.me) > SQLQuery_2.sql
 - Query:

```
1 SELECT * FROM Customers WHERE CompanyName LIKE '_[aeiou]%'
```
 - Results pane: Displays the results of the query as a table. The table has columns: CustomerID, CompanyName, ContactName, and ContactTitle. The data shows 28 rows of customer information from the NorthWind database, such as ALFKI, ANATR, ANTON, AROUT, BLAUS, BOLID, BSBEV, CHOPS, DRACD, ERNSH, FRANK, FRANR, FRANS, GREAL, GROSR, ISLAT, KOENE, MEREP, OCEAN, OLDWO, OTTIK, PRINI, SPECD, SPLIR, THEBI, THECR, and TRADH.
- Messages** pane: Shows the message: "5. Get the list of Customers where Company Name's 2nd letter is Consonant. (Table: Customers) SELECT * FROM Customers WHERE CompanyName LIKE '_[aeiou]%'".
- Bottom Status Bar**: Shows 0 rows affected, SQL chosen as the language, 30 rows selected, 00:00:00 duration, and the connection string database-1.c0lnneqcg5iq.us-east-2.rds.amazonaws.com.

7. We acquire new Shipper and it does not ship to PO Box. Generate the list of Customers which has PO Box address. (Table: Customers)

```
SELECT * FROM Customers WHERE Address LIKE '%P.O. Box%'
```

The screenshot shows the Microsoft Data Studio interface. On the left is the Explorer pane with a tree view of saved queries and a note about cloning from a URL. The main area contains a query editor with the SQL command: `SELECT * FROM Customers WHERE Address LIKE '%P.O. Box%'`. Below the editor is a results grid showing one customer record:

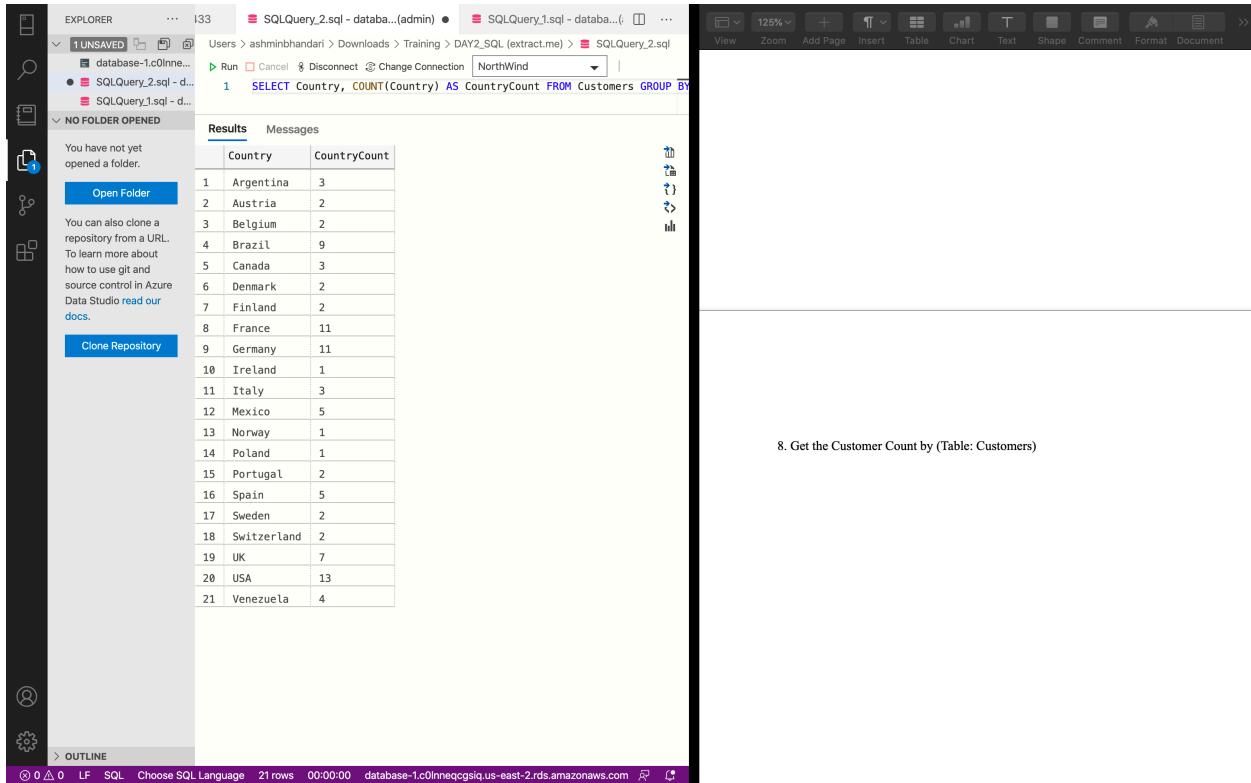
CustomerID	CompanyName	ContactName	ContactTitle
1	SPLIR	Split Rail Beer & Ale	Art Braunschweiger

At the bottom, the status bar indicates 1 row was returned in 0:00:01.

8. Get the Customer Count by (Table: Customers)

i. Country

```
SELECT Country, COUNT(Country) AS CountryCount FROM Customers  
GROUP BY Country
```



The screenshot shows the SSMS interface with the following details:

- EXPLORER** pane: Shows 1 UNSAVED file named "SQLQuery_2.sql".
- RESULTS** pane: Displays the query results in a table format.
- Messages** pane: Shows a message: "You have not yet opened a folder." with buttons "Open Folder" and "Clone Repository".
- Toolbars**: Standard SSMS toolbar with View, Zoom, Add Page, Insert, Table, Chart, Text, Shape, Comment, Format, Document.
- Status Bar**: Shows the query language as SQL, the number of rows as 21, and the database connection information as "database-1c0lnneqcgsiq.us-east-2.rds.amazonaws.com".

Country	CountryCount
Argentina	3
Austria	2
Belgium	2
Brazil	9
Canada	3
Denmark	2
Finland	2
France	11
Germany	11
Ireland	1
Italy	3
Mexico	5
Norway	1
Poland	1
Portugal	2
Spain	5
Sweden	2
Switzerland	2
UK	7
USA	13
Venezuela	4

8. Get the Customer Count by (Table: Customers)

ii. Country, Region

```
SELECT Country, Region, COUNT(*) AS CountryAndRegionCount FROM Customers GROUP BY Country, Region HAVING Region IS NOT NULL
```

The screenshot shows the Microsoft Data Studio interface. On the left is the Explorer sidebar with a single unsaved file named 'SQLQuery_2.sql'. The main area displays the results of a SQL query:

```
SELECT Country, Region, COUNT(*) AS CountryAndRegionCount FROM Customers GROUP BY Country, Region HAVING Region IS NOT NULL
```

The results table has three columns: Country, Region, and CountryAndRegionCount. The data is as follows:

Country	Region	CountryAndRegionCount
USA	AK	1
Canada	BC	2
USA	CA	1
Ireland	Co. Cork	1
Venezuela	DF	1
USA	ID	1
UK	Isle of Wight	1
Venezuela	Lara	1
USA	MT	1
USA	NM	1
Venezuela	Nueva Esparta	1
USA	OR	4
Canada	Québec	1
Brazil	RJ	3
Brazil	SP	6
Venezuela	Táchira	1
USA	WA	3
USA	WY	1

iii. Country, Region, City

```
SELECT Country, Region, City, COUNT(*) AS CountryRegionCityCount
FROM Customers GROUP BY Country, Region, City HAVING Region IS NOT NULL
```

The screenshot shows the SSMS interface with the following details:

- EXPLORER**: Shows a folder structure with '1 UNSAVED' file.
- SQLQuery_2.sql - database...(admin)**: The current query window.
- SQLQuery_1.sql - database...(d...)**: Another query window.
- Run**: Run button.
- Disconnect**: Disconnect button.
- Change Connection**: Change connection dropdown.
- NorthWind**: Current connection name.
- Results**: Tab selected.
- Messages**: Tab available.
- Customer Data**: Results grid showing the following data:

	Country	Region	City	CountryRegionCityCount
1	Brazil	RJ	Rio de Janeiro	3
2	Brazil	SP	Campinas	1
3	Brazil	SP	Resende	1
4	Brazil	SP	Sao Paulo	4
5	Canada	BC	Tsawassen	1
6	Canada	BC	Vancouver	1
7	Canada	Québec	Montréal	1
8	Ireland	Co. Cork	Cork	1
9	UK	Isle of Wight	Cowes	1
10	USA	AK	Anchorage	1
11	USA	CA	San Francisco	1
12	USA	ID	Boise	1
13	USA	MT	Butte	1
14	USA	NM	Albuquerque	1
15	USA	OR	Elgin	1
16	USA	OR	Eugene	1
17	USA	OR	Portland	2
18	USA	WA	Kirkland	1
19	USA	WA	Seattle	1
20	USA	WA	Walla Walla	1
21	USA	WY	Lander	1
22	Venezuela	DF	Caracas	1
23	Venezuela	Lara	Barquisimeto	1
24	Venezuela	Nueva Esparta	I. de Margarita	1
25	Venezuela	Táchira	San Cristóbal	1

Outline: Shows 0 nodes.

Bottom status bar: 0 rows, SQL, Choose SQL Language, 25 rows, 00:00:00, database-1.c0lnneqcsiq.us-east-2.rds.amazonaws.com.

9. The Company wants to increase its relationship with higher ranking officers of Customer. Get the Count of Contact's person by their title. (Table: Customers)

```
SELECT ContactTitle, COUNT(ContactTitle) AS ContactTitleCount
FROM Customers Group BY ContactTitle
```

The screenshot shows the Microsoft Data Studio interface. On the left is the Explorer sidebar with a '1 UNSAVED' folder containing 'SQLQuery_2.sql - database-1.c0Inneqcg5iq.us-east-2.rds.amazonaws.com (admin)' and 'SQLQuery_1.sql - database-1.c0Inneqcg5iq.us-east-2.rds.amazonaws.com (extract.me)'. The main area displays a query result table titled 'Results'.

Results

ContactTitle	ContactTitleCount
Accounting Manager	10
Assistant Sales Agent	2
Assistant Sales Representative	1
Marketing Assistant	6
Marketing Manager	12
Order Administrator	2
Owner	17
Owner/Marketing Assistant	1
Sales Agent	5
Sales Associate	7
Sales Manager	11
Sales Representative	17

Below the table, the query text is shown:

```
9. The Company wants to increase its relationship with higher ranking officers of Customer. Get the Count of Contact's person by their title. (Table: Customers)

SELECT ContactTitle, COUNT(ContactTitle) AS ContactTitleCount
FROM Customers Group BY ContactTitle
```

At the bottom, the status bar shows: 0 0 LF SQL Choose SQL Language 12 rows 00:00:00 database-1.c0Inneqcg5iq.us-east-2.rds.amazonaws.com

10. Get the list of Customers and Number of Orders placed by each customer till date. (Table: Orders)

```
SELECT Orders.CustomerID, Customers.ContactName, COUNT(*) AS  
OrdersMade FROM Orders INNER JOIN Customers ON Orders.CustomerID  
= Customers.CustomerID GROUP BY Orders.CustomerID,  
Customers.ContactName
```

The screenshot shows the Microsoft Data Studio interface. On the left is the Explorer pane with a folder tree and a message about no folder being open. In the center is the Results pane displaying a table of customer data. The table has columns: CustomerID, ContactName, and OrdersMade. The data consists of 28 rows of customer information. On the right is the Query Editor pane containing the SQL query for question 10.

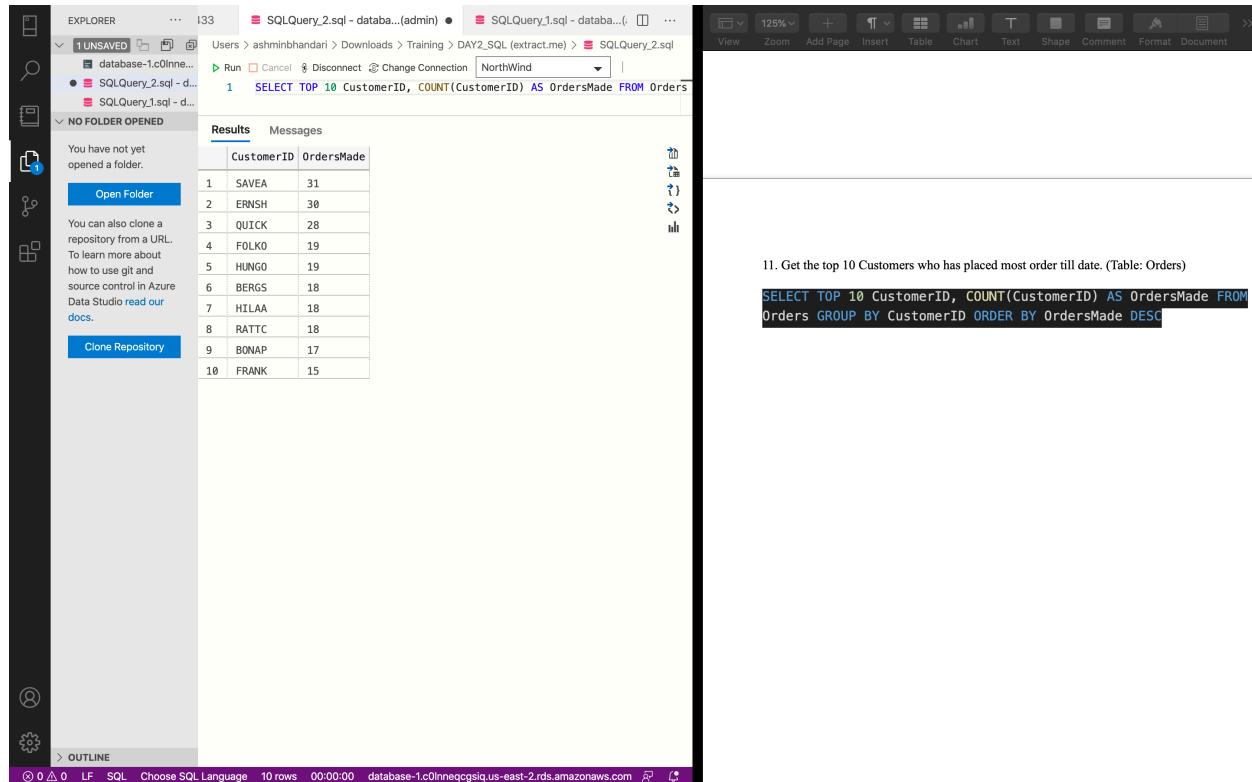
	CustomerID	ContactName	OrdersMade
1	ALFKI	Maria Anders	6
2	ANATR	Ana Trujillo	4
3	ANTON	Antonio Moreno	7
4	AROUT	Thomas Hardy	13
5	BERGS	Christina Berglund	18
6	BLAUS	Hanna Moos	7
7	BLONP	Frédérique Citeaux	11
8	BOLID	Martin Sommer	3
9	BONAP	Laurence Lebihan	17
10	BOTTM	Elizabeth Lincoln	14
11	BSBEV	Victoria Ashworth	10
12	CACTU	Patricia Simpson	6
13	CENTC	Francisco Chang	1
14	CHOPS	Yang Wang	8
15	COMMI	Pedro Afonso	5
16	CONSH	Elizabeth Brown	3
17	DRACD	Sven Ottlieb	6
18	DUMON	Janine Labrune	4
19	EASTC	Ann Devon	8
20	ERNSH	Roland Mendel	30
21	FAMIL	Aria Cruz	7
22	FOLIG	Martine Rancé	5
23	FOLKO	Maria Larsson	19
24	FRANK	Peter Franken	15
25	FRANR	Carine Schmitt	3
26	FRANS	Paolo Accorti	6
27	FURIB	Lino Rodriguez	8
28	GALED	Eduardo Saavedra	5

10. Get the list of Customers and Number of Orders placed by each customer till date. (Table: Orders)

```
SELECT Orders.CustomerID, Customers.ContactName, COUNT(*) AS  
OrdersMade FROM Orders INNER JOIN Customers ON Orders.CustomerID  
= Customers.CustomerID GROUP BY Orders.CustomerID,  
Customers.ContactName
```

11. Get the top 10 Customers who has placed most order till date. (Table: Orders)

```
SELECT TOP 10 CustomerID, COUNT(CustomerID) AS OrdersMade FROM Orders GROUP BY CustomerID ORDER BY OrdersMade DESC
```



The screenshot shows the SSMS interface with the following details:

- EXPLORER** pane on the left.
- RESULTS** pane on the right displaying the query results.
- Messages** pane at the bottom right.
- Toolbars** at the top and bottom.
- Toolbar Buttons** include Run, Cancel, Disconnect, Change Connection, View, Zoom, Add Page, Insert, Table, Chart, Text, Shape, Comment, Format, Document, etc.
- Query Editor** at the top with the query: `SELECT TOP 10 CustomerID, COUNT(CustomerID) AS OrdersMade FROM Orders GROUP BY CustomerID ORDER BY OrdersMade DESC`.
- Results Grid** showing the following data:

	CustomerID	OrdersMade
1	SAVEA	31
2	ERNSH	30
3	QUICK	28
4	FOLKO	19
5	HUNGO	19
6	BERGS	18
7	HILAA	18
8	RATTC	18
9	BONAP	17
10	FRANK	15

Message pane content:

11. Get the top 10 Customers who has placed most order till date. (Table: Orders)

```
SELECT TOP 10 CustomerID, COUNT(CustomerID) AS OrdersMade FROM Orders GROUP BY CustomerID ORDER BY OrdersMade DESC
```

12. Get Customers list who has placed 5 or more Orders. (Table:Orders)

```
SELECT CustomerID, COUNT(CustomerID) AS OrdersMade FROM Orders
GROUP BY CustomerID HAVING COUNT(CustomerID) >= 5
```

The screenshot shows the SSMS interface with two panes. The left pane is the Explorer pane, which is currently empty. The right pane is the Results pane, which displays the output of the query. The Results pane has tabs for 'Results' and 'Messages'. The 'Results' tab is selected and shows a table with two columns: 'CustomerID' and 'OrdersMade'. The data is as follows:

CustomerID	OrdersMade
ALFKI	6
ANTON	7
AROUT	13
BERGS	18
BLAUS	7
BLOWNP	11
BONAP	17
BOTTM	14
BSBEV	10
CACTU	6
CHOPS	8
COMMIX	5
DRACD	6
EASTC	8
ERNSH	30
FAMILIA	7
FOLIG	5
FOLKO	19
FRANK	15
FRANS	6
FURIB	8
GALED	5
GODOS	10
GOURL	9
GREAL	11
HANAR	14
HILAA	18
HUNGC	5

Below the table, there is a message: "12. Get Customers list who has placed 5 or more Orders. (Table:Orders)".

```
SELECT CustomerID, COUNT(CustomerID) AS OrdersMade FROM Orders
GROUP BY CustomerID HAVING COUNT(CustomerID) >= 5
```

13. Get the Order Count by (Table:Orders)

I. Each Year (Hint: YEAR() function)

```
SELECT YEAR(OrderDate) AS Year, COUNT(CustomerID) AS OrdersCount  
FROM Orders GROUP BY YEAR(OrderDate)
```

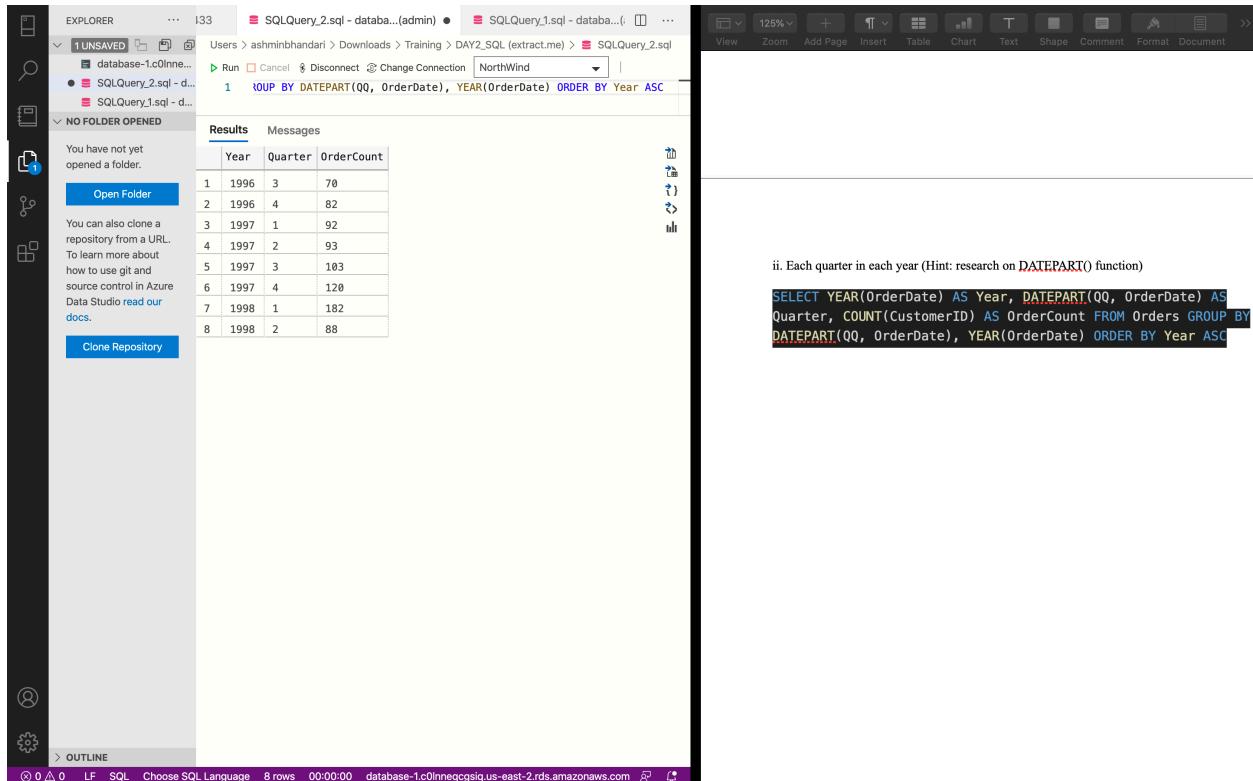
The screenshot shows the Microsoft Data Studio interface. On the left is the Explorer sidebar with a '1 UNSAVED' file named 'SQLQuery_2.sql'. The main area has two tabs: 'Results' and 'Messages'. The 'Results' tab displays a table with columns 'CustomerID' and 'OrdersMade'. The table contains 28 rows of data. The 'Messages' tab is empty. At the bottom, the status bar shows '0 rows' and 'Choose SQL Language'.

CustomerID	OrdersMade
ALFKI	6
ANTON	7
AROUT	13
BERGS	18
BLAUS	7
BLONP	11
BONAP	17
BOTTM	14
BSBEV	10
CACTU	6
CHOPS	8
COMMI	5
DRACD	6
EASTC	8
ERNSH	30
FAMIL	7
FOLIG	5
FOLKO	19
FRANK	15
FRANS	6
FURIB	8
GALED	5
GODOS	10
GOURL	9
GREAL	11
HANAR	14
HILAA	18
HUNGC	5

13. Get the Order Count by (Table:Orders)
I. Each Year (Hint: YEAR() function)
II.
SELECT YEAR(OrderDate) AS Year, COUNT(CustomerID) AS OrdersCount
FROM Orders GROUP BY YEAR(OrderDate)

ii. Each quarter in each year (Hint: research on DATEPART() function)

```
SELECT YEAR(OrderDate) AS Year, DATEPART(QQ, OrderDate) AS Quarter, COUNT(CustomerID) AS OrderCount FROM Orders GROUP BY DATEPART(QQ, OrderDate), YEAR(OrderDate) ORDER BY Year ASC
```



The screenshot shows the SSMS interface with the following details:

- EXPLORER** pane: Shows 1 UNSAVED file named "SQLQuery_2.sql".
- RESULTS** pane: Displays the query results in a table format.
- Messages** pane: Shows a message indicating no folder has been opened.
- Toolbars**: Standard SSMS toolbars for View, Zoom, Add Page, Insert, Table, Chart, Text, Shape, Comment, Format, Document.
- Toolbar Buttons**: Includes icons for Run, Cancel, Disconnect, Change Connection, NorthWind, and a dropdown menu.
- Bottom Status Bar**: Shows the connection status as "database-1.c0Inneqccgsi.us-east-2.rds.amazonaws.com", the number of rows (8), and the execution time (00:00:00).

Results Table Data:

	Year	Quarter	OrderCount
1	1996	3	70
2	1996	4	82
3	1997	1	92
4	1997	2	93
5	1997	3	103
6	1997	4	120
7	1998	1	182
8	1998	2	88

Query Results:

```
SELECT YEAR(OrderDate) AS Year, DATEPART(QQ, OrderDate) AS Quarter, COUNT(CustomerID) AS OrderCount FROM Orders GROUP BY DATEPART(QQ, OrderDate), YEAR(OrderDate) ORDER BY Year ASC
```

iii. Each Month in each year (Hint: research on DATEPART() function)

```
SELECT YEAR(OrderDate) AS Year, DATEPART(MM, OrderDate) AS Month, COUNT(CustomerID) AS OrderCount FROM Orders GROUP BY DATEPART(MM, OrderDate), YEAR(OrderDate) ORDER BY Year ASC
```

The screenshot shows the SSMS interface with the following details:

- EXPLORER** pane: Shows 1 UNSAVED file named "SQLQuery_2.sql".
- RESULTS** pane: Displays the query results in a table format.
- Messages** pane: Shows a message indicating no folder has been opened.
- Toolbars**: Standard SSMS toolbars for View, Zoom, Add Page, Insert, Table, Chart, Text, Shape, Comment, Format, and Document.
- Toolbar Buttons**: Includes Run, Cancel, Disconnect, Change Connection, and a dropdown for NorthWind.
- Table Headers**: The results table has columns: Year, Month, and OrderCount.
- Data Rows**: The table contains 23 rows of data, starting from Year 1996, Month 7, OrderCount 22, up to Year 1998, Month 5, OrderCount 14.
- Bottom Status Bar**: Shows 0 rows affected, SQL chosen as language, 23 rows total, 00:00:00 duration, and the connection string database-1.c0lnneqcgsql.us-east-2.rds.amazonaws.com.

iii. Each Month in each year (Hint: research on DATEPART() function)

```
SELECT YEAR(OrderDate) AS Year, DATEPART(MM, OrderDate) AS Month, COUNT(CustomerID) AS OrderCount FROM Orders GROUP BY DATEPART(MM, OrderDate), YEAR(OrderDate) ORDER BY Year ASC
```

14. Calculate Average, Total, Minimum, and Maximum Freight paid (Table:Orders)

i. For each Order

```
SELECT AVG(Freight) AS Average, SUM(Freight) AS Total,  
MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders  
GROUP BY OrderID
```

AVG, SUM, MAX is not necessary here

The screenshot shows the SSMS interface with the following details:

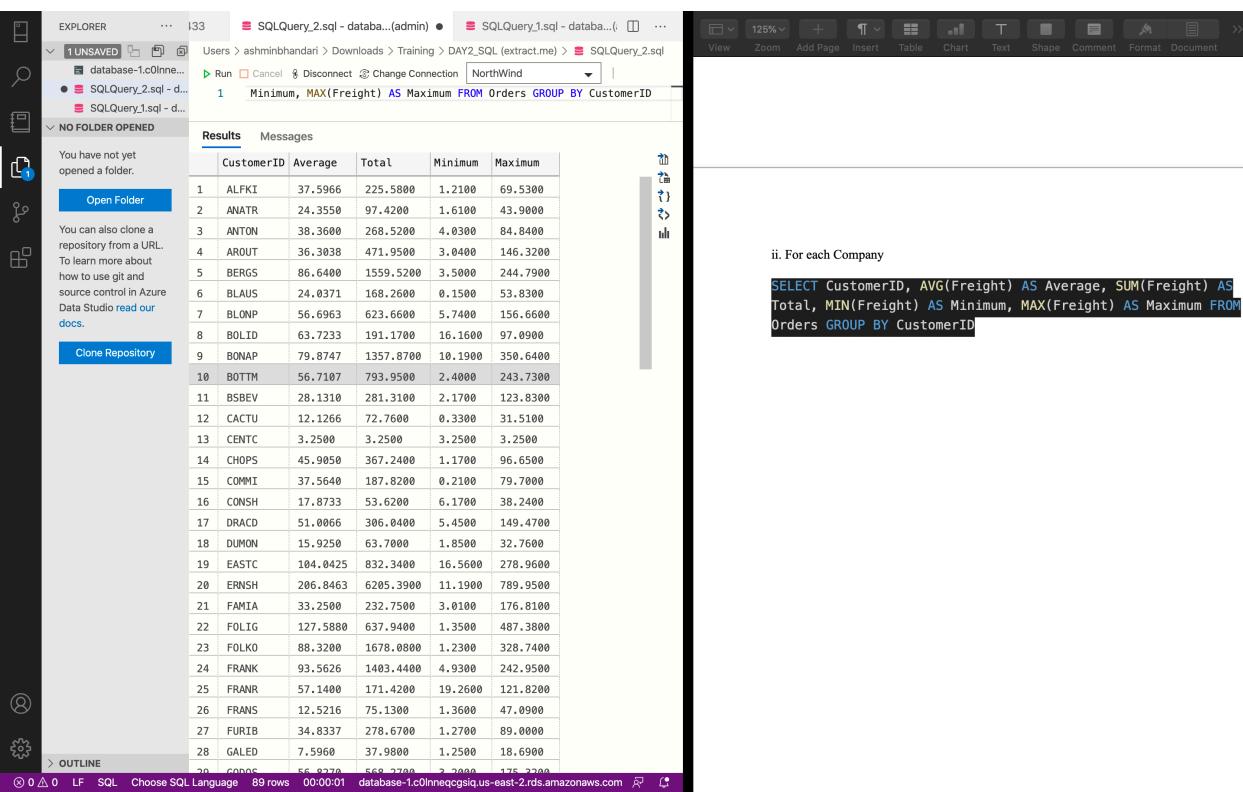
- EXPLORER** pane: Shows a folder structure with 'UNSAVED' and 'SQLQuery_2.sql'.
- RESULTS** pane: Displays the query results as a table:

	Year	Month	OrderCount
1	1996	7	22
2	1996	8	25
3	1996	9	23
4	1996	10	26
5	1996	11	25
6	1996	12	31
7	1997	1	33
8	1997	2	29
9	1997	3	30
10	1997	4	31
11	1997	5	32
12	1997	6	30
13	1997	7	33
14	1997	8	33
15	1997	9	37
16	1997	10	38
17	1997	11	34
18	1997	12	48
19	1998	1	55
20	1998	2	54
21	1998	3	73
22	1998	4	74
23	1998	5	14

- MESSAGES** pane: Shows the query text: 'SELECT AVG(Freight) AS Average, SUM(Freight) AS Total, MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders GROUP BY OrderID'.
- NOTIFICATIONS** pane: Shows the message: '14. Calculate Average, Total, Minimum, and Maximum Freight paid (Table:Orders)'.

ii. For each Company

```
SELECT CustomerID, AVG(Freight) AS Average, SUM(Freight) AS Total, MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders GROUP BY CustomerID
```



The screenshot shows the SSMS interface with the following details:

- EXPLORER**: Shows a folder structure with '1 UNSAVED' file.
- SQLQuery_2.sql - database... (admin)**: The current query window.
- SQLQuery_1.sql - database... (d...)**: Another query window.
- Results**: The results pane displays the query results as a table.
- Messages**: The messages pane is empty.
- Toolbar**: Standard SSMS toolbar with icons for Run, Cancel, Connection, View, Zoom, Add Page, Insert, Table, Chart, Text, Shape, Comment, Format, and Document.
- Table Headers**: CustomerID, Average, Total, Minimum, Maximum.
- Table Data** (28 rows):

CustomerID	Average	Total	Minimum	Maximum
ALFKI	37.5966	225.5800	1.2100	69.5300
ANATR	24.3550	97.4200	1.6100	43.9000
ANTON	38.3600	268.5200	4.0300	84.8400
AROUT	36.3038	471.9500	3.0400	146.3200
BERGS	86.6400	1559.5200	3.5000	244.7900
BLAUS	24.0371	168.2600	0.1500	53.8300
BLONI	56.6963	623.6600	5.7400	156.6600
BOLID	63.7233	191.1700	16.1600	97.0900
BONAP	79.8747	1357.8700	18.1900	350.6400
BOTTM	56.7107	793.9500	2.4000	243.7300
BSBEV	28.1310	281.3100	2.1700	123.8300
CACTU	12.1266	72.7600	0.3300	31.5100
CENTN	3.2500	3.2500	3.2500	3.2500
CHOPS	45.9050	367.2400	1.1700	96.6500
COMMI	37.5640	187.8200	0.2100	79.7000
CONSH	17.8733	53.6200	6.1700	38.2400
DRACD	51.0065	306.8400	5.4500	149.4700
DUMON	15.9250	63.7000	1.8500	32.7600
EASTC	104.0425	832.3400	16.5600	278.9600
ERNSH	206.8463	6205.3900	11.1900	789.9500
FAMIL	33.2500	232.7500	3.0100	176.8100
FOLIG	127.5888	637.9400	1.3500	487.3800
FOLKO	88.3200	1678.0800	1.2300	328.7400
FRANL	93.5626	1483.4400	4.9300	242.9500
FRANR	57.1400	171.4200	19.2600	121.8200
FRANS	12.5216	75.1300	1.3600	47.0900
FURIB	34.8337	278.6700	1.2700	89.0000
GALED	7.5960	37.9800	1.2500	18.6900
GODOC	66.8270	560.7700	2.2000	175.2200
- Outline**: Shows the outline of the current query.
- Status Bar**: Shows the number of rows (89), execution time (00:00:01), and connection information (database-1.c0lnneqcgsql.us-east-2.rds.amazonaws.com).

iii. For each Country on all orders

```
SELECT ShipCountry, AVG(Freight) AS Average, SUM(Freight) AS Total, MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders GROUP BY ShipCountry
```

The screenshot shows the SSMS interface with two panes. The left pane is the Explorer pane, which is currently empty under 'NO FOLDER OPENED'. The right pane is the Results pane, which displays the output of the SQL query. The Results pane has tabs for 'Results' and 'Messages', with 'Results' selected. The output is a table with the following columns: ShipCountry, Average, Total, Minimum, and Maximum. The data consists of 21 rows, each representing a country and its shipping statistics.

	ShipCountry	Average	Total	Minimum	Maximum
1	Finland	41.4040	910.8900	0.5900	180.4500
2	USA	112.8794	13771.2900	0.2000	830.7500
3	Italy	30.8728	864.4400	0.4800	155.9700
4	Brazil	58.7974	4880.1900	0.1400	890.7800
5	Germany	92.4859	11283.2800	0.1500	1007.6400
6	Switzerland	76.0294	1368.5300	1.1700	232.4200
7	Mexico	40.0992	1122.7800	0.4000	218.1500
8	Sweden	87.5027	3237.6000	1.2300	328.7400
9	Argentina	37.4112	598.5800	0.3300	217.8600
10	Austria	184.7875	7391.5000	5.2900	789.9500
11	UK	52.7548	2954.2700	0.9000	288.4300
12	Poland	25.1057	175.7400	3.9400	80.6500
13	Canada	73.2696	2198.0900	0.9400	379.1300
14	Ireland	145.0126	2755.2400	16.7400	603.5400
15	Norway	45.9166	275.5000	4.6200	93.6300
16	France	55.0368	4237.8400	0.0200	487.3800
17	Belgium	67.3757	1280.1400	0.1700	424.3000
18	Spain	37.4734	861.8900	1.2500	175.3200
19	Venezuela	59.4604	2735.1800	0.1200	210.1900
20	Denmark	77.5661	1396.1900	2.9200	299.0900
21	Portugal	49.5023	643.5300	1.2700	202.2400

iii. For each Country on all orders

```
SELECT ShipCountry, AVG(Freight) AS Average, SUM(Freight) AS Total, MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders GROUP BY ShipCountry
```

iv. for Each Carrier (ShipVia)

```
SELECT ShipVia, AVG(Freight) AS Average, SUM(Freight) AS Total,  
MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders  
GROUP BY ShipVia
```

The screenshot shows the SSMS interface with the following details:

- EXPLORER**: Shows 1 UNSAVED file named SQLQuery_2.sql.
- SQL Query Editor**: Contains the SQL query provided above.
- Results**: Displays the query results in a table:

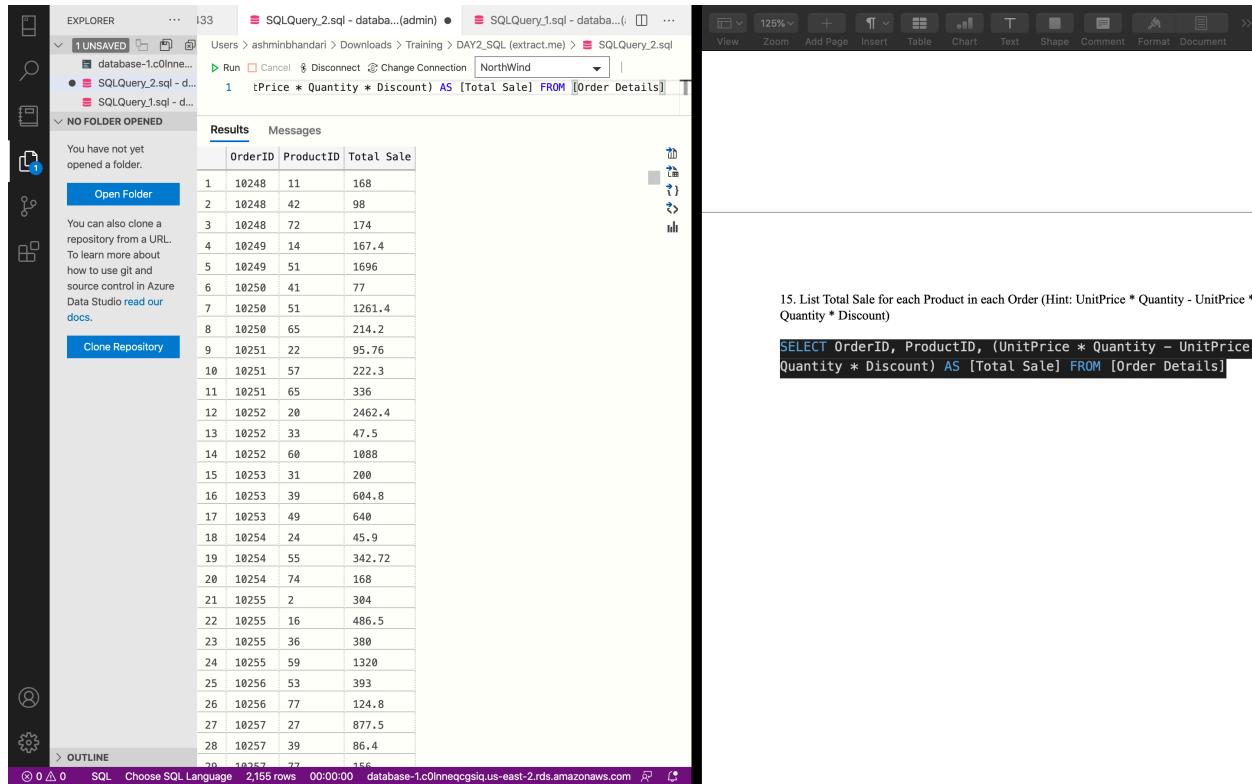
	ShipVia	Average	Total	Minimum	Maximum
1	3	80.4412	20512.5100	0.4000	1007.6400
2	1	65.0013	16185.3300	0.1200	458.7800
3	2	86.6406	28244.8500	0.0200	890.7800

iv. for Each Carrier (ShipVia)

```
SELECT ShipVia, AVG(Freight) AS Average, SUM(Freight) AS Total,  
MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders  
GROUP BY ShipVia
```

15. List Total Sale for each Product in each Order (Hint: UnitPrice * Quantity - UnitPrice * Quantity * Discount)

```
SELECT OrderID, ProductID, (UnitPrice * Quantity - UnitPrice * Quantity * Discount) AS [Total Sale] FROM [Order Details]
```



The screenshot shows the Microsoft Data Studio interface. On the left is the Explorer pane with a single unsaved file named 'SQLQuery_2.sql'. The main area contains a query editor with the following SQL code:

```
SELECT OrderID, ProductID, (UnitPrice * Quantity - UnitPrice * Quantity * Discount) AS [Total Sale] FROM [Order Details]
```

Below the query is a results grid titled 'Results' with three columns: OrderID, ProductID, and Total Sale. The data consists of 21 rows of sales information. At the bottom of the interface, there is a status bar showing '0 0 0 SQL Choose SQL Language 2,155 rows 00:00:00 database-1.c0lnneqqgsiq.us-east-2.rds.amazonaws.com'.

OrderID	ProductID	Total Sale
1	10248	11
2	10248	42
3	10248	72
4	10249	14
5	10249	51
6	10250	41
7	10250	51
8	10250	65
9	10251	22
10	10251	57
11	10251	65
12	10252	20
13	10252	33
14	10252	60
15	10253	31
16	10253	39
17	10253	49
18	10254	24
19	10254	55
20	10254	74
21	10255	2
22	10255	16
23	10255	36
24	10255	59
25	10256	53
26	10256	77
27	10257	27
28	10257	39
29	10257	77
30	10257	156

16. For each Order Calculate (Table: [Orders Details])

i. Types of Products Ordered (Hint: Count on Product)

DISTINCT is not really necessary here

```
SELECT OrderID, COUNT(DISTINCT(ProductID)) AS "Types of products ordered" FROM [Order Details] GROUP BY OrderID
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left is the Explorer pane, which lists several database files and connections. The main area has a toolbar at the top with various icons like Run, Cancel, and Disconnect. Below the toolbar is a code editor window containing the SQL query:

```
1 "Types of products ordered" FROM [Order Details] GROUP BY OrderID
```

Below the code editor is a results grid titled "Results". The grid has two columns: "OrderID" and "Types of products ordered". The data consists of 29 rows, each showing an OrderID and its corresponding count of distinct product types. The counts range from 1 to 4. The bottom of the results grid shows the following footer text:

16. For each Order Calculate (Table: [Orders Details])
i. Types of Products Ordered (Hint: Count on Product)
DISTINCT is not really necessary here

On the right side of the interface, there is another code editor window with the same query and some additional text at the bottom.

ii. Total Sale for each Order

```
SELECT OrderID, SUM((UnitPrice * Quantity - UnitPrice * Quantity * Discount)) AS TotalSale FROM [Order Details] GROUP BY OrderID
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left is the Explorer pane, which is currently empty. The center-left area shows the Query Editor with the following code:

```
SELECT OrderID, SUM((UnitPrice * Quantity - UnitPrice * Quantity * Discount)) AS TotalSale FROM [Order Details] GROUP BY OrderID
```

The center-right area shows the Results pane with the following output:

OrderID	TotalSale
1	10248
2	10249
3	10250
4	10251
5	10252
6	10253
7	10254
8	10255
9	10256
10	10257
11	10258
12	10259
13	10260
14	10261
15	10262
16	10263
17	10264
18	10265
19	10266
20	10267
21	10268
22	10269
23	10270
24	10271
25	10272
26	10273
27	10274
28	10275

17. List total Quantity Ordered for Each Product on all orders. (Table: [Orders Details])

```
SELECT ProductID, SUM(Quantity) AS "Total Quantity Ordered" FROM  
[Order Details] GROUP BY ProductID
```

Results

	ProductID	Total Quantity Ordered
1	23	580
2	46	548
3	69	714
4	29	746
5	75	1155
6	15	122
7	9	95
8	3	328
9	52	500
10	72	806
11	66	239
12	32	297
13	26	753
14	12	344
15	35	883
16	63	445
17	6	381
18	55	983
19	43	580
20	49	520
21	67	184
22	27	365
23	21	1016
24	58	534
25	64	740
26	38	623
27	7	763
28	44	601

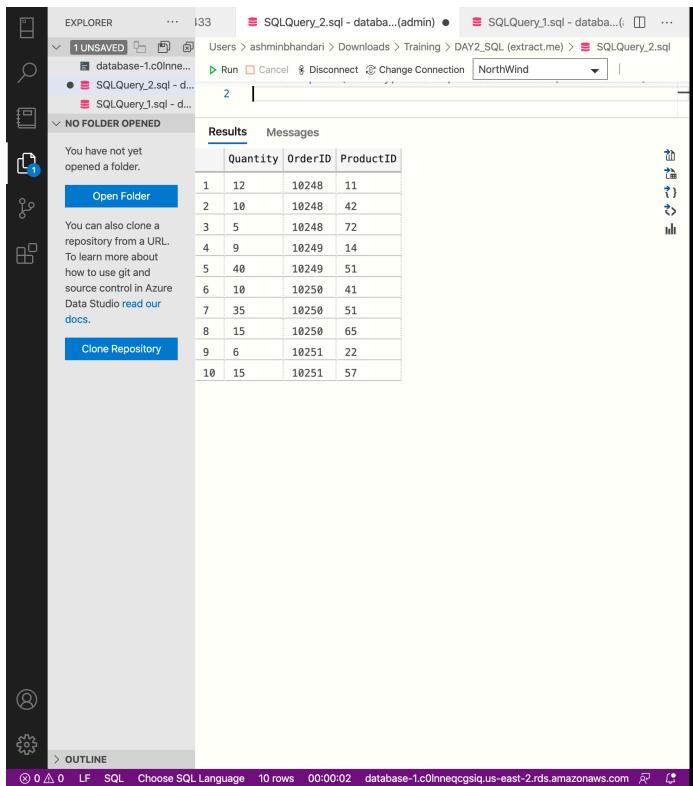
14. Calculate Average, Total, Minimum, and Maximum Freight paid (Table:Orders)
i. For each Order

```
SELECT AVG(Freight) AS Average, SUM(Freight) AS Total,  
MIN(Freight) AS Minimum, MAX(Freight) AS Maximum FROM Orders  
GROUP BY OrderID
```

Avg, Sum, Max is not necessary here

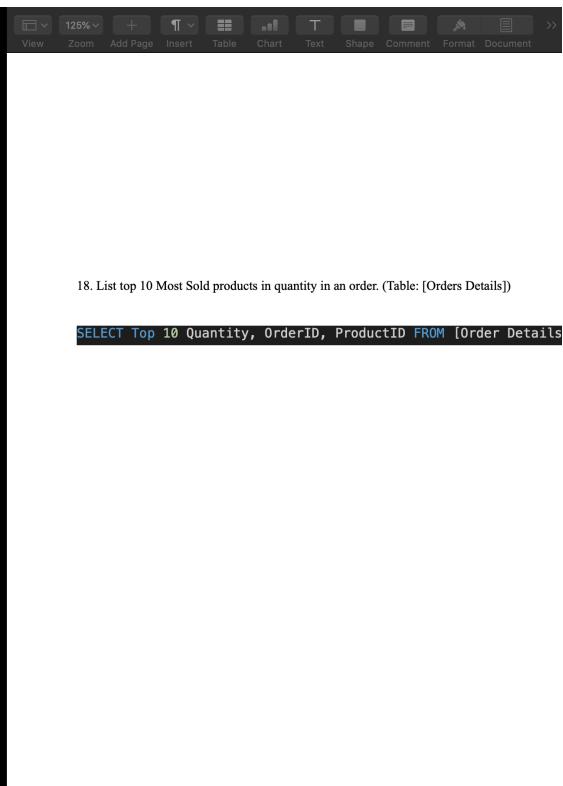
18. List top 10 Most Sold products in quantity in an order. (Table: [Orders Details])

```
SELECT Top 10 Quantity, OrderID, ProductID FROM [Order Details]
```



The screenshot shows the GitHub desktop application interface. On the left, the Explorer sidebar displays a repository named 'SQLQuery_2.sql - database-1cOnline...' with one file listed: 'SQLQuery_2.sql'. The main area shows a table titled 'Results' with the following data:

	Quantity	OrderID	ProductID
1	12	10248	11
2	10	10248	42
3	5	10248	72
4	9	10249	14
5	40	10249	51
6	10	10250	41
7	35	10250	51
8	15	10250	65
9	6	10251	22
10	15	10251	57



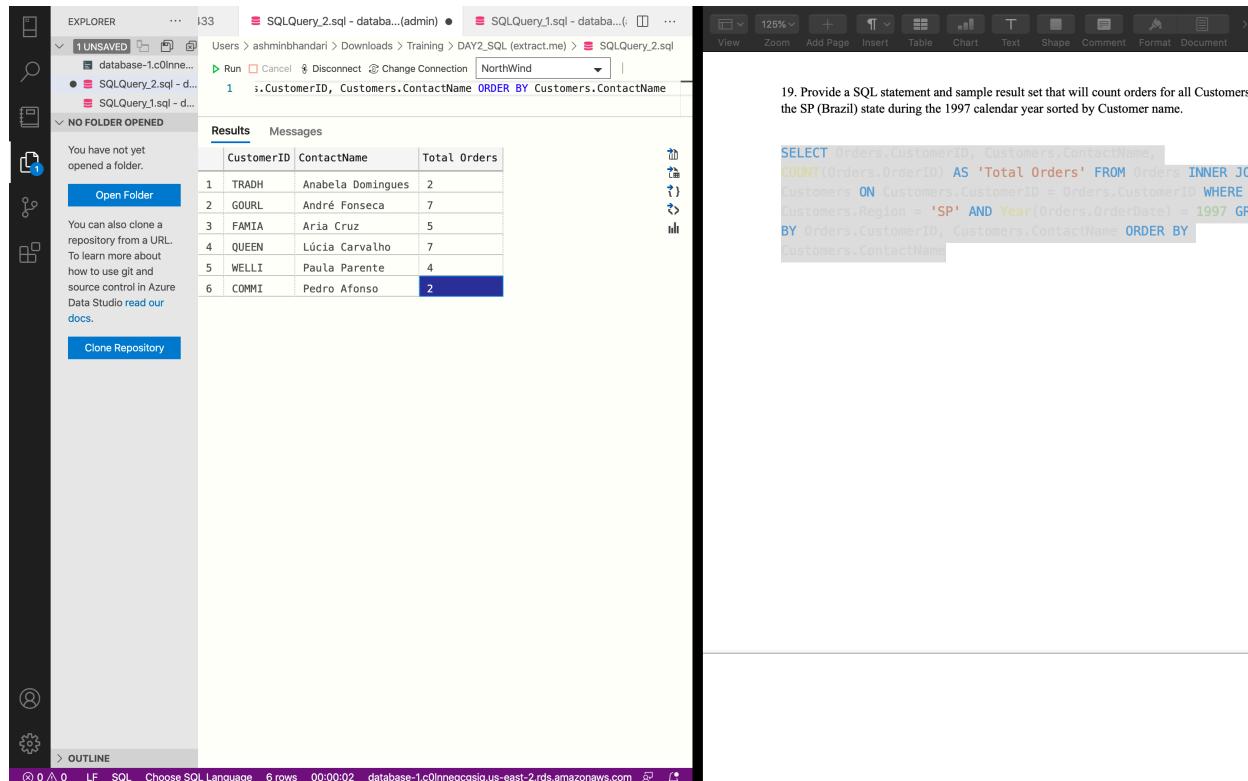
The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. At the top, there are tabs for 'View', 'Zoom', 'Add Page', 'Insert', 'Table', 'Chart', 'Text', 'Shape', 'Comment', 'Format', and 'Document'. Below the tabs, the status bar shows '125%'. The main pane displays the results of the query execution:

18. List top 10 Most Sold products in quantity in an order. (Table: [Orders Details])

```
SELECT Top 10 Quantity, OrderID, ProductID FROM [Order Details]
```

19. Provide a SQL statement and sample result set that will count orders for all Customers within the SP (Brazil) state during the 1997 calendar year sorted by Customer name.

```
SELECT Orders.CustomerID, Customers.ContactName,
COUNT(Orders.OrderID) AS 'Total Orders' FROM Orders INNER JOIN
Customers ON Customers.CustomerID = Orders.CustomerID WHERE
Customers.Region = 'SP' AND Year(Orders.OrderDate) = 1997 GROUP BY
Orders.CustomerID, Customers.ContactName ORDER BY
Customers.ContactName
```



The screenshot shows the Microsoft Azure Data Studio interface. On the left is the Explorer sidebar with a '1 UNSAVED' file named 'SQLQuery_2.sql'. The main area has two tabs: 'Results' and 'Messages'. The 'Results' tab displays a table with six rows, showing CustomerID, ContactName, and Total Orders. The 'Messages' tab contains the SQL query provided in the question.

	CustomerID	ContactName	Total Orders
1	TRADH	Anabela Domingues	2
2	GOURL	André Fonseca	7
3	FAMIL	Aria Cruz	5
4	QUEEN	Lúcia Carvalho	7
5	WELLI	Paula Parente	4
6	COMMI	Pedro Afonso	2

SQL Query in the Messages tab:

```
SELECT Orders.CustomerID, Customers.ContactName,
COUNT(Orders.OrderID) AS 'Total Orders' FROM Orders INNER JOIN
Customers ON Customers.CustomerID = Orders.CustomerID WHERE
Customers.Region = 'SP' AND Year(Orders.OrderDate) = 1997 GROUP BY
Orders.CustomerID, Customers.ContactName ORDER BY
Customers.ContactName
```

20. Provide a SQL statement and sample result set that will list all Customers within the SP (Brazil) state that have placed 7 or more orders during the 1998 calendar year.

```
SELECT Customers.ContactName, COUNT(Orders.CustomerID) AS  
'Orders Made' FROM Orders INNER JOIN Customers ON  
Customers.CustomerID = Orders.CustomerID WHERE Customers.Region  
= 'SP' AND Year(Orders.OrderDate) = 1998 GROUP BY  
Orders.CustomerID, Customers.ContactName HAVING  
COUNT(Orders.CustomerID) >= 7
```

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left is the Explorer pane, which lists several database files and a folder named 'NO FOLDER OPENED'. In the center is the Query Editor window. The title bar shows the connection details: 'ashminbandari' (admin) at 'NorthWind'. The query itself is displayed in the editor:

```
SELECT Customers.ContactName, COUNT(Orders.CustomerID) AS  
'Orders Made' FROM Orders INNER JOIN Customers ON  
Customers.CustomerID = Orders.CustomerID WHERE Customers.Region  
= 'SP' AND Year(Orders.OrderDate) = 1998 GROUP BY  
Orders.CustomerID, Customers.ContactName HAVING  
COUNT(Orders.CustomerID) >= 7
```

Below the query, the results pane shows the output:

ContactName	Orders Made
Antonio Moreno	7

At the bottom of the SSMS window, the status bar displays: '0 rows 00:00:02 database-1c0lnneqeqgsiq.us-east-2.rds.amazonaws.com'.