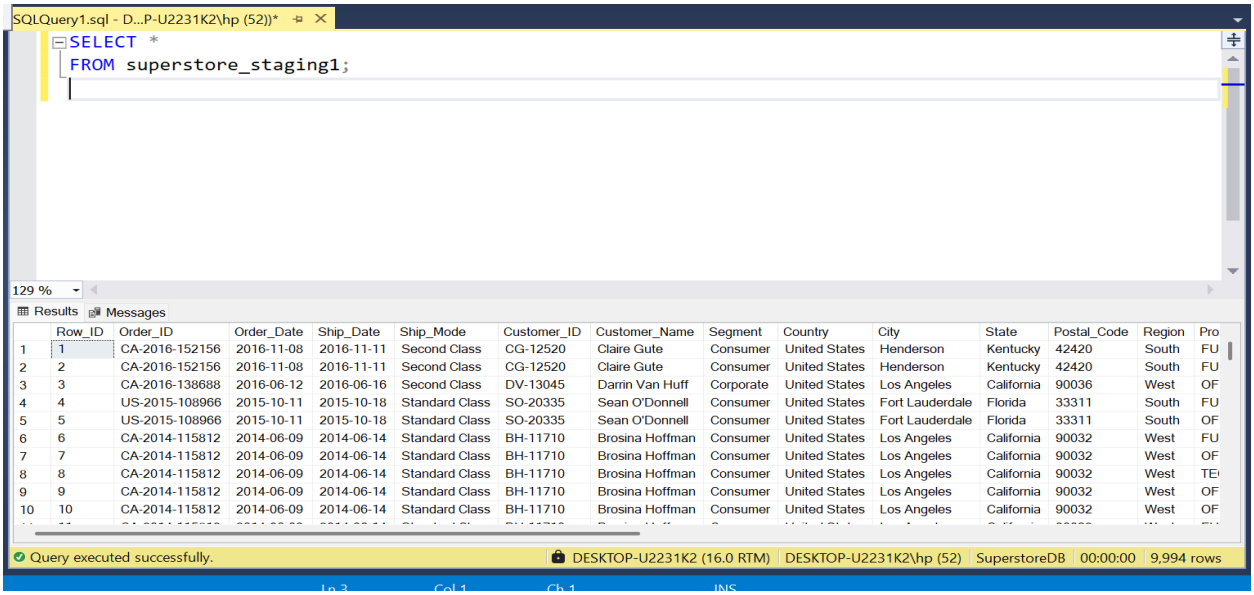
2025

TASK1: QUERYING DATA WITH SQL

SUBMITTED BY:

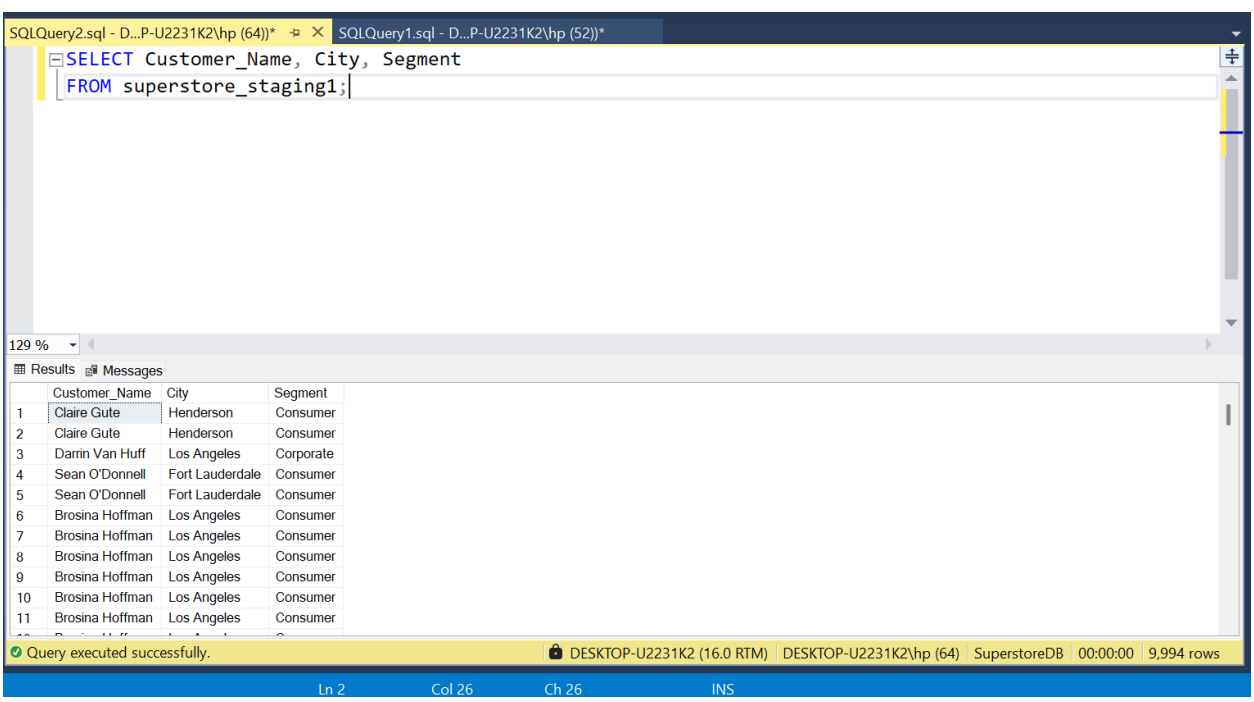
AYESHA SALEH

Query 1: I retrieved all the columns from the table using `SELECT`. This helped me see the complete dataset and understand all available fields.



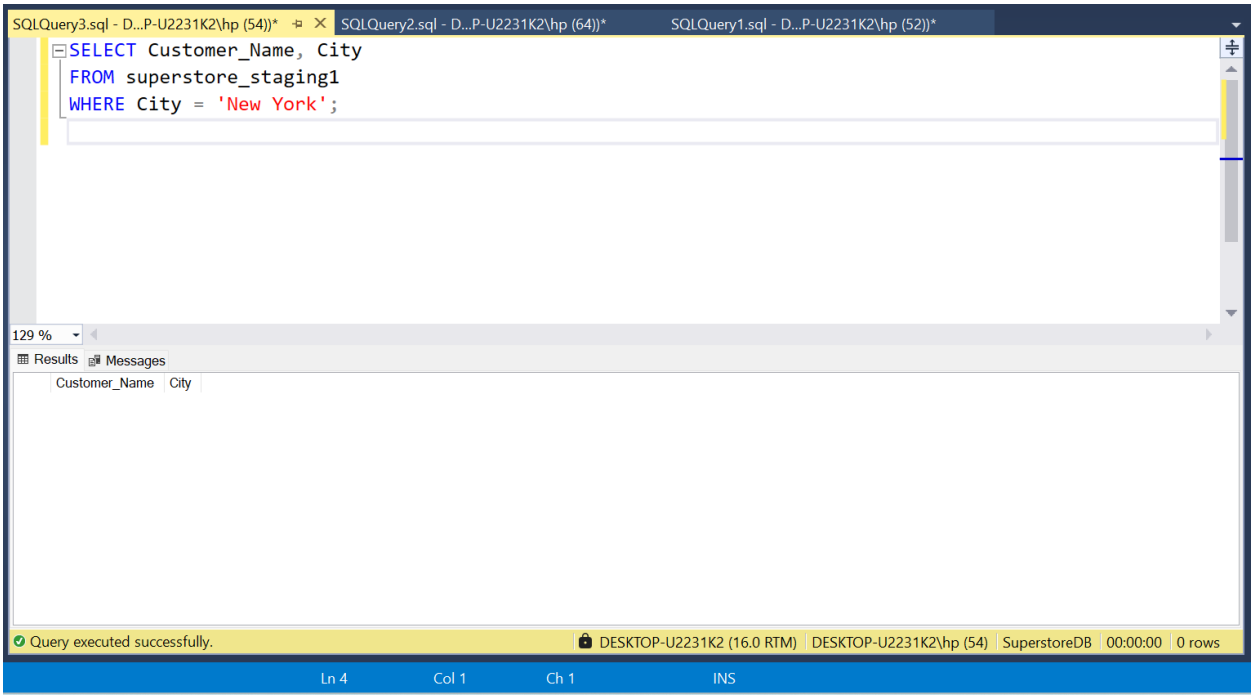
	Row_ID	Order_ID	Order_Date	Ship_Date	Ship_Mode	Customer_ID	Customer_Name	Segment	Country	City	State	Postal_Code	Region	Pro
1	1	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FU
2	2	CA-2016-152156	2016-11-08	2016-11-11	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	FU
3	3	CA-2016-138688	2016-06-12	2016-06-16	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036	West	OF
4	4	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	FU
5	5	US-2015-108966	2015-10-11	2015-10-18	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311	South	OF
6	6	CA-2014-115812	2014-06-09	2014-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	FU
7	7	CA-2014-115812	2014-06-09	2014-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	OF
8	8	CA-2014-115812	2014-06-09	2014-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	TE
9	9	CA-2014-115812	2014-06-09	2014-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	OF
10	10	CA-2014-115812	2014-06-09	2014-06-14	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032	West	OF

Query 2: I selected only specific columns such as `Customer Name`, `City`, and `Segment`. This made the output more focused and easier to analyze.

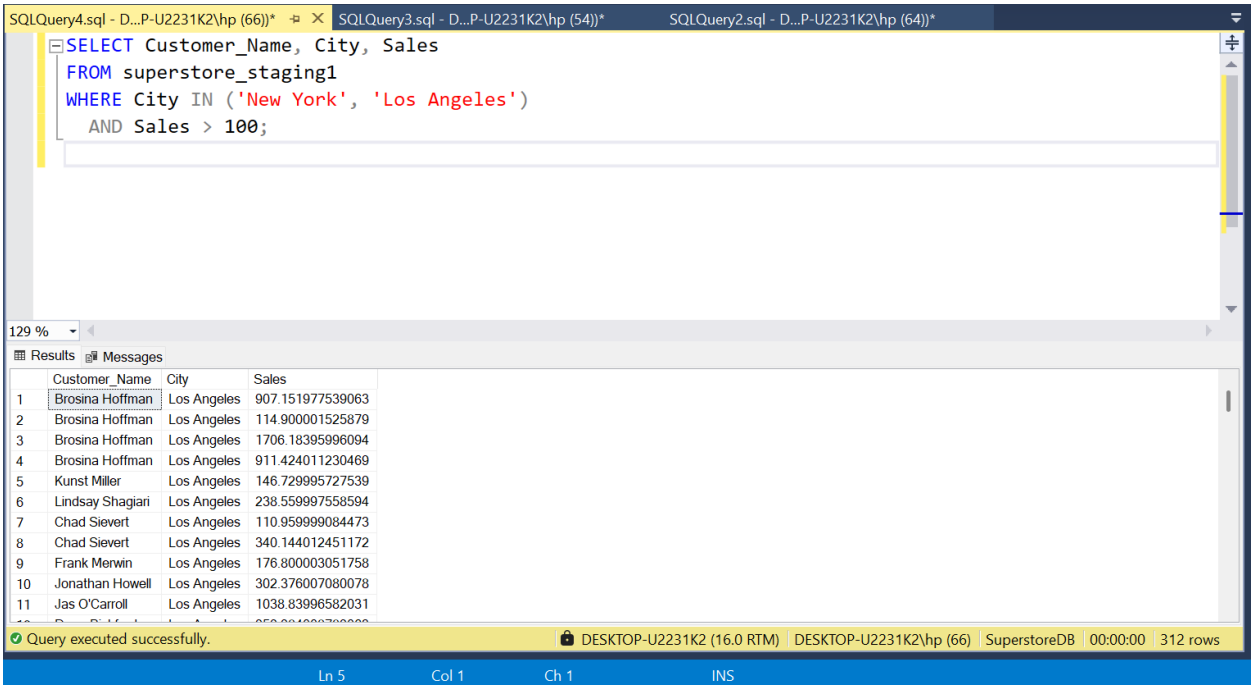


	Customer_Name	City	Segment
1	Claire Gute	Henderson	Consumer
2	Claire Gute	Henderson	Consumer
3	Darrin Van Huff	Los Angeles	Corporate
4	Sean O'Donnell	Fort Lauderdale	Consumer
5	Sean O'Donnell	Fort Lauderdale	Consumer
6	Brosina Hoffman	Los Angeles	Consumer
7	Brosina Hoffman	Los Angeles	Consumer
8	Brosina Hoffman	Los Angeles	Consumer
9	Brosina Hoffman	Los Angeles	Consumer
10	Brosina Hoffman	Los Angeles	Consumer
11	Brosina Hoffman	Los Angeles	Consumer

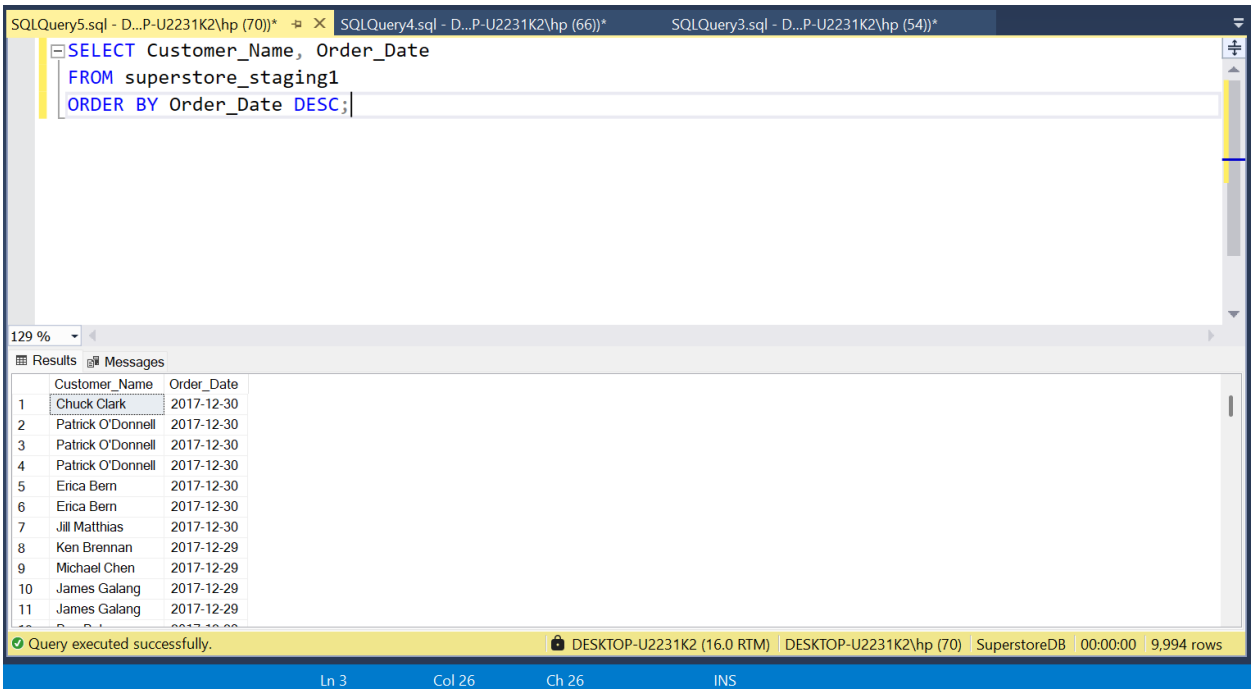
Query 3: I applied a `WHERE` condition to filter customers from a specific city. This allowed me to extract only the relevant records.



Query 4: I combined conditions with `AND`/`OR` to find customers from New York or Los Angeles with sales greater than 100. This showed how multiple filters can be used together.

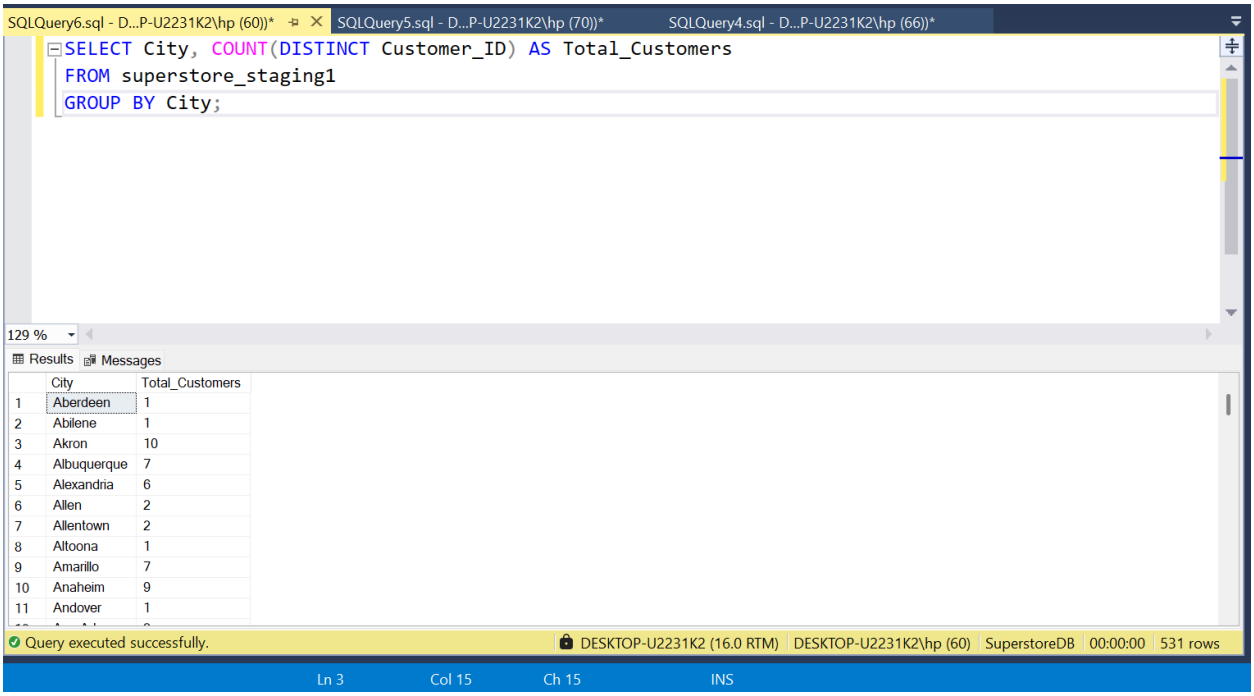


Query 5: I used `ORDER BY` on the `Order Date` column to sort results in descending order. This helped me organize the data by showing the most recent orders first.



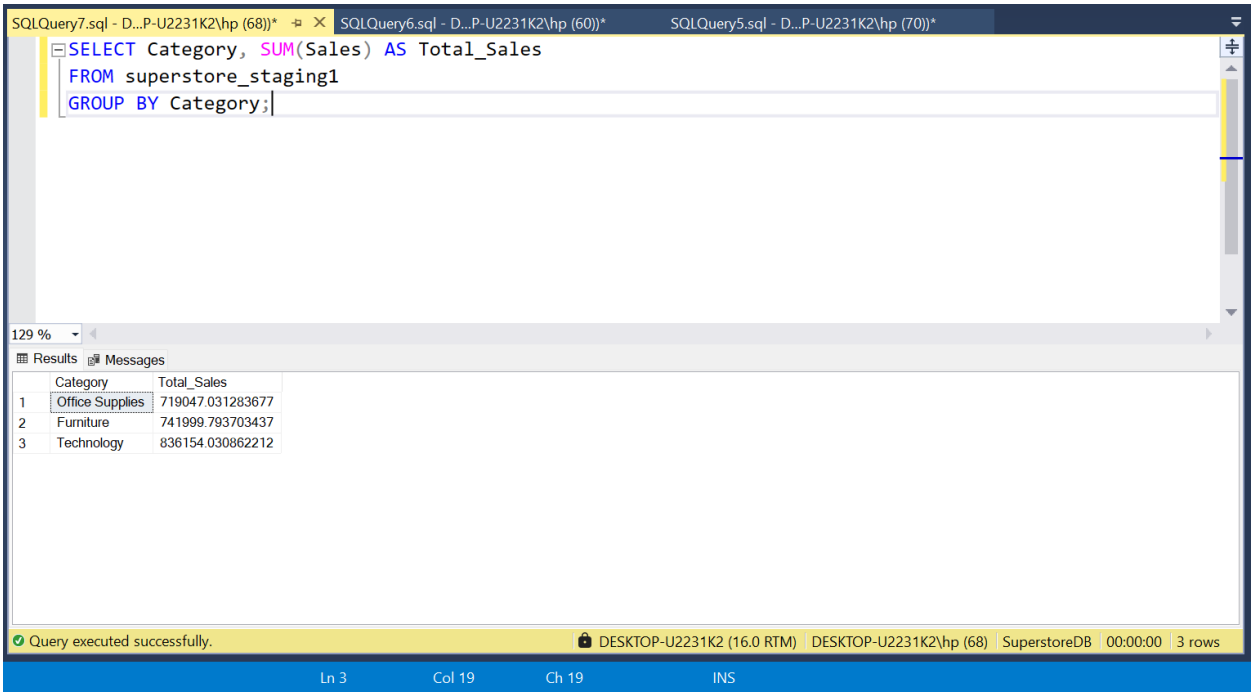
	Customer_Name	Order_Date
1	Chuck Clark	2017-12-30
2	Patrick O'Donnell	2017-12-30
3	Patrick O'Donnell	2017-12-30
4	Patrick O'Donnell	2017-12-30
5	Erica Bern	2017-12-30
6	Erica Bern	2017-12-30
7	Jill Matthias	2017-12-30
8	Ken Brennan	2017-12-29
9	Michael Chen	2017-12-29
10	James Galang	2017-12-29
11	James Galang	2017-12-29

Query 6: I counted the number of customers grouped by city using `COUNT` with `GROUP BY`. This helped me understand customer distribution across different cities.

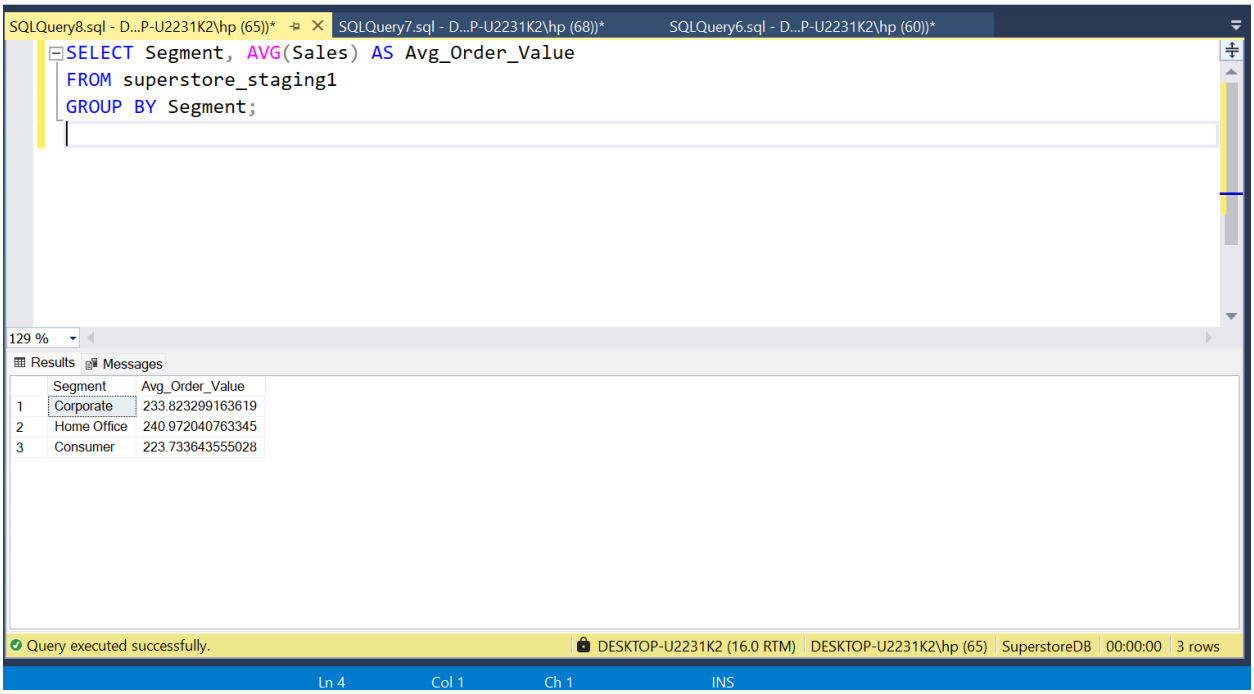


	City	Total_Customers
1	Aberdeen	1
2	Abilene	1
3	Akron	10
4	Albuquerque	7
5	Alexandria	6
6	Allen	2
7	Allentown	2
8	Altoona	1
9	Amarillo	7
10	Anaheim	9
11	Andover	1

Query 7: I calculated the total sales for each product category using `SUM` with `GROUP BY`. This allowed me to compare how different categories contribute to sales.



Query 8: I used `AVG` with `GROUP BY` to find the average order value for each segment. This showed me how customer segments differ in their purchasing behavior.



Query 9: I applied a `HAVING` clause to filter cities with more than 10 customers. This helped me focus on cities with a larger customer base.

SQLQuery9.sql - D...P-U2231K2\hp (69))* | SQLQuery8.sql - D...P-U2231K2\hp (65))* | SQLQuery7.sql - D...P-U2231K2\hp (68))*

```sql
SELECT City, COUNT(DISTINCT Customer_ID) AS Customer_Count
FROM superstore_staging1
GROUP BY City
HAVING COUNT(DISTINCT Customer_ID) > 10;
```

129 %

Results | Messages

	City	Customer_Count
1	Arlington	31
2	Atlanta	17
3	Aurora	34
4	Austin	18
5	Baltimore	21
6	Burlington	14
7	Charlotte	24
8	Chicago	155
9	Cincinnati	12
10	Cleveland	21
11	Columbia	33

Query executed successfully. | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (69) | SuperstoreDB | 00:00:00 | 76 rows

Ln 4 | Col 41 | Ch 41 | INS

Query 10: I calculated the highest and lowest order values for each month using MAX and MIN. This gave me insights into sales variation across time.

SQLQuery10.sql -...P-U2231K2\hp (71))* | SQLQuery9.sql - D...P-U2231K2\hp (69))* | SQLQuery8.sql - D...P-U2231K2\hp (65))*

```sql
SELECT YEAR(Order_Date) AS Order_Year, MONTH(Order_Date) AS Order_Month,
       MAX(Sales) AS Highest_Order, MIN(Sales) AS Lowest_Order
FROM superstore_staging1
GROUP BY YEAR(Order_Date), MONTH(Order_Date);
```

129 %

Results | Messages

	Order_Year	Order_Month	Highest_Order	Lowest_Order
1	2017	5	3359.95190429688	1.18799996376038
2	2014	1	2573.82006835938	2.89000010490417
3	2016	12	9892.740234375	1.72800004482269
4	2014	7	8187.64990234375	1.0440000295639
5	2015	11	4643.7998046875	1.69599997997284
6	2015	5	2973.32006835938	2.69600009918213
7	2014	4	4164.0498046875	2.36800003051758
8	2016	9	4355.16796875	1.80999994277954
9	2017	2	1640.69995117188	1.78799998760223
10	2014	10	2735.95190429688	0.898000001907349
11	2016	1	1592.84997558594	1.72800004482269

Query executed successfully. | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (71) | SuperstoreDB | 00:00:00 | 48 rows

Ln 4 | Col 46 | Ch 46 | INS

Query 11: I matched customers to their specific orders by linking on their unique ID and name. This shows only customers who have actually placed orders and all the details for those transactions.

SQLQuery1.sql - D...P-U2231K2\hp (75))*

```sql
SELECT
    c.Customer_ID, c.Customer_Name,o.Order_ID,o.Order_Date,o.Ship_Date,o.Product_Name,o.Sales,o.Quantity
FROM dbo.superstore_staging1 c
INNER JOIN dbo.superstore_staging1 o
    ON c.Customer_ID = o.Customer_ID
    AND c.Customer_Name = o.Customer_Name
WHERE c.Row_ID < o.Row_ID
ORDER BY c.Customer_Name, o.Order_Date;
```

116 %

Results | Messages

	Customer_ID	Customer_Name	Order_ID	Order_Date	Ship_Date	Product_Name	Sales	Quantity
1	AB-10015	Aaron Bergman	CA-2014-152905	2014-02-18	2014-02-24	Akro Stacking Bins	12.6239995956421	2
2	AB-10015	Aaron Bergman	CA-2014-152905	2014-02-18	2014-02-24	Akro Stacking Bins	12.6239995956421	2
3	AB-10015	Aaron Bergman	CA-2014-152905	2014-02-18	2014-02-24	Akro Stacking Bins	12.6239995956421	2
4	AB-10015	Aaron Bergman	CA-2014-156587	2014-03-07	2014-03-08	Newell 330	17.9400005340576	3
5	AB-10015	Aaron Bergman	CA-2014-156587	2014-03-07	2014-03-08	Carina 42"Hx23 3/4"W Media Storage Unit	242.940002441406	3
6	AB-10015	Aaron Bergman	CA-2014-156587	2014-03-07	2014-03-08	Carina 42"Hx23 3/4"W Media Storage Unit	242.940002441406	3
7	AB-10015	Aaron Bergman	CA-2016-140935	2016-11-10	2016-11-12	Samsung Convoy 3	221.979995727539	2
8	AB-10015	Aaron Bergman	CA-2016-140935	2016-11-10	2016-11-12	Samsung Convoy 3	221.979995727539	2
9	AB-10015	Aaron Bergman	CA-2016-140935	2016-11-10	2016-11-12	Samsung Convoy 3	221.979995727539	2
10	AB-10015	Aaron Bergman	CA-2016-140935	2016-11-10	2016-11-12	Samsung Convoy 3	221.979995727539	2
11	AB-10015	Aaron Bergman	CA-2016-140935	2016-11-10	2016-11-12	Sauder Facets Collection Library, Sky Alder Finish	341.959991455078	2

Query executed successfully. | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (75) | SuperstoreDB | 00:00:01 | 73,411 rows

Ln 2 | Col 95 | Ch 95 | INS

Query 12: I kept every single customer from our master list on the left, then attached any order data they might have. This includes customers who've never ordered, showing them with blank order fields.

SQLQuery2.sql - D...P-U2231K2\hp (59))* | SQLQuery1.sql - D...P-U2231K2\hp (75))*

```sql
SELECT
    c.Customer_ID,c.Customer_Name,o.Order_ID,o.Order_Date,o.Product_Name,o.Sales,
    COALESCE(o.Quantity, 0) as Quantity
FROM (
    SELECT DISTINCT Customer_ID, Customer_Name
    FROM dbo.superstore_staging1
) c
LEFT JOIN dbo.superstore_staging1 o
    ON c.Customer_ID = o.Customer_ID
ORDER BY c.Customer_Name, o.Order_Date;
```

116 %

Results | Messages

	Customer_ID	Customer_Name	Order_ID	Order_Date	Product_Name	Sales	Quantity
1	AB-10015	Aaron Bergman	CA-2014-152905	2014-02-18	Akro Stacking Bins	12.6239995956421	2
2	AB-10015	Aaron Bergman	CA-2014-156587	2014-03-07	Global Push Button Manager's Chair, Indigo	48.7120018005371	1
3	AB-10015	Aaron Bergman	CA-2014-156587	2014-03-07	Newell 330	17.9400005340576	3
4	AB-10015	Aaron Bergman	CA-2014-156587	2014-03-07	Carina 42"Hx23 3/4"W Media Storage Unit	242.940002441406	3
5	AB-10015	Aaron Bergman	CA-2016-140935	2016-11-10	Samsung Convoy 3	221.979995727539	2
6	AB-10015	Aaron Bergman	CA-2016-140935	2016-11-10	Sauder Facets Collection Library, Sky Alder Finish	341.959991455078	2
7	AH-10030	Aaron Hawkins	CA-2014-122070	2014-04-22	Staple envelope	247.839996337891	8
8	AH-10030	Aaron Hawkins	CA-2014-122070	2014-04-22	ACCOHIDE 3-Ring Binder, Blue, 1"	9.91199970245361	3
9	AH-10030	Aaron Hawkins	CA-2014-113768	2014-05-13	Iceberg Nesting Folding Chair, 19w x 6d x 43h	279.455993652344	6
10	AH-10030	Aaron Hawkins	CA-2014-113768	2014-05-13	EcoTones Memo Sheets	8	2
11	AH-10030	Aaron Hawkins	US-2014-158400	2014-10-25	GBC VeloBind Cover Sets	49.4080009460449	4

Query executed successfully. | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (59) | SuperstoreDB | 00:00:00 | 9,994 rows

Ln 10 | Col 34 | Ch 34 | INS

Query 13: I used our complete product catalog on the right and joined it with sales records. This ensures every product is shown, even those that have never been sold, with their sales data left blank.

SQLQuery3.sql - D...P-U2231K2\hp (64))* | SQLQuery2.sql - D...P-U2231K2\hp (59))* | SQLQuery1.sql - D...P-U2231K2\hp (75))*

```sql
SELECT
    p.Product_ID,p.Product_Name,p.Category,p.Sub_Category,s.Order_ID,s.Order_Date,s.Sales,
    COALESCE(s.Quantity, 0) as Quantity,
    COALESCE(s.Profit, 0) as Profit
FROM (
    SELECT DISTINCT Product_ID, Product_Name, Category, Sub_Category
    FROM dbo.superstore_staging1
) p
RIGHT JOIN dbo.superstore_staging1 s
    ON p.Product_ID = s.Product_ID
ORDER BY p.Product_Name, s.Order_Date;
```

116 %

Results | Messages

	Product_ID	Product_Name	Category	Sub_Category	Order_ID	Order_Date	Sales	Quantity	Profit
1	OFF-PA-10003424	"While you Were Out" Message Book, One Form per ...	Office Supplies	Paper	CA-2017-167626	2017-09-03	8.9040002822876	3	3.3390000000
2	OFF-PA-10003424	"While you Were Out" Message Book, One Form per ...	Office Supplies	Paper	CA-2017-123491	2017-10-30	7.42000007629395	2	3.7100000000
3	OFF-PA-10003424	"While you Were Out" Message Book, One Form per ...	Office Supplies	Paper	CA-2017-165204	2017-11-13	8.9040002822876	3	3.3390000000
4	OFF-EN-10000461	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	US-2014-161305	2014-06-06	13.9840002059937	2	4.7196000000
5	OFF-EN-10000781	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	CA-2014-131541	2014-07-28	48.9440002441406	7	16.5186000000
6	OFF-EN-10000461	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	CA-2015-108588	2015-01-05	17.4799995422363	2	8.2156000000
7	OFF-EN-10000461	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	CA-2015-100573	2015-09-25	17.4799995422363	2	8.2156000000
8	OFF-EN-10000781	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	CA-2015-127509	2015-11-09	26.2199993133545	3	12.3234000000
9	OFF-EN-10000461	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	CA-2015-140830	2015-11-30	17.4799995422363	2	8.2156000000
10	OFF-EN-10000461	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	CA-2015-153038	2015-12-18	55.9360008239746	8	18.8784000000
11	OFF-EN-10000461	#10- 4 1/8" x 9 1/2" Recycled Envelopes	Office Supplies	Envelopes	US-2017-159562	2017-09-09	17.4799995422363	2	8.2156000000

Query executed successfully. | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (64) | SuperstoreDB | 00:00:00 | 10,331 rows

Media Player Classic

Ln 2 | Col 83 | INS

Query 14: I connected three key data points: customer profiles, their order headers, and the product details for each item purchased. This creates a complete view of who bought what, when, and for how much.

SQLQuery5.sql - D...P-U2231K2\hp (55))* | SQLQuery3.sql - D...P-U2231K2\hp (64))* | SQLQuery2.sql - D...P-U2231K2\hp (59))*

```sql
SELECT
c.Customer_ID,c.Customer_Name,c.Segment,c.Region,o.Order_ID,o.Order_Date,o.Ship_Date,p.Product_ID,
p.Product_Name,p.Category,p.Sub_Category,o.Sales,o.Quantity,o.Discount,o.Profit
FROM (
    SELECT DISTINCT Customer_ID, Customer_Name, Segment, Region
    FROM dbo.superstore_staging1
) c
INNER JOIN dbo.superstore_staging1 o
    ON c.Customer_ID = o.Customer_ID
INNER JOIN (
    SELECT DISTINCT Product_ID, Product_Name, Category, Sub_Category
    FROM dbo.superstore_staging1
) p
    ON o.Product_ID = p.Product_ID
ORDER BY c.Customer_Name, o.Order_Date, p.Product_Name;
```

87 %

Results | Messages

	tomer_ID	Customer_Name	Segment	Region	Order_ID	Order_Date	Ship_Date	Product_ID	Product_Name	Category	Sub_Ca
1	10015	Aaron Bergman	Consumer	West	CA-2014-152905	2014-02-18	2014-02-24	OFF-ST-10000321	Akro Stacking Bins	Office Supplies	Storag
2	10015	Aaron Bergman	Consumer	Central	CA-2014-152905	2014-02-18	2014-02-24	OFF-ST-10000321	Akro Stacking Bins	Office Supplies	Storag
3	10015	Aaron Bergman	Consumer	West	CA-2014-156587	2014-03-07	2014-03-08	OFF-ST-10002344	Carina 42"Hx23 3/4"W Media Storage Unit	Office Supplies	Storag
4	10015	Aaron Bergman	Consumer	Central	CA-2014-156587	2014-03-07	2014-03-08	OFF-ST-10002344	Carina 42"Hx23 3/4"W Media Storage Unit	Office Supplies	Storag
5	10015	Aaron Bergman	Consumer	West	CA-2014-156587	2014-03-07	2014-03-08	FUR-CH-10004477	Global Push Button Manager's Chair, Indigo	Furniture	Chairs
6	10015	Aaron Bergman	Consumer	Central	CA-2014-156587	2014-03-07	2014-03-08	FUR-CH-10004477	Global Push Button Manager's Chair, Indigo	Furniture	Chairs
7	10015	Aaron Bergman	Consumer	West	CA-2014-156587	2014-03-07	2014-03-08	OFF-AR-10001427	Newell 330	Office Supplies	Art
8	10015	Aaron Bergman	Consumer	Central	CA-2014-156587	2014-03-07	2014-03-08	OFF-AR-10001427	Newell 330	Office Supplies	Art
9	10015	Aaron Bergman	Consumer	West	CA-2016-140935	2016-11-10	2016-11-12	TEC-PH-10000562	Samsung Convoy 3	Technology	Phones
10	10015	Aaron Bergman	Consumer	Central	CA-2016-140935	2016-11-10	2016-11-12	TEC-PH-10000562	Samsung Convoy 3	Technology	Phones

Query executed successfully. | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (55) | SuperstoreDB | 00:00:00 | 34,640 rows

Ln 3 | Col 27 | Ch 27 | INS

Query 15: I analyzed sales by month and customer segment, then ranked products within each group to identify top performers. The final report shows the percentage each top product contributed to its segment's total monthly sales.

SQLQuery7.sql - D...P-U2231K2\hp (77))* | SQLQuery5.sql - D...P-U2231K2\hp (55))* | SQLQuery3.sql - D...P-U2231K2\hp (64))*

```sql
WITH MonthlySales AS (
    SELECT
        YEAR(Order_Date) as SalesYear,
        MONTH(Order_Date) as SalesMonth,
        Segment,
        Product_ID,
        Product_Name,
        Category,
        SUM(Sales) as TotalSales,
        SUM(Quantity) as TotalQuantity,
        SUM(Profit) as TotalProfit,
        COUNT(Order_ID) as OrderCount,
        ROW_NUMBER() OVER (PARTITION BY YEAR(Order_Date), MONTH(Order_Date), Segment
                           ORDER BY SUM(Sales) DESC) as SalesRank
    FROM dbo.superstore_staging1
    WHERE Order_Date IS NOT NULL
    GROUP BY YEAR(Order_Date), MONTH(Order_Date), Segment, Product_ID, Product_Name, Category
),
TopProducts AS (
    SELECT
        SalesYear,
        SalesMonth,
        Segment,
        Product_ID,
        Product_Name,
        Category,
        TotalSales,
        TotalQuantity,
        TotalProfit,
        OrderCount
    FROM MonthlySales
    WHERE SalesRank <= 5 -- Top 5 products per segment per month
)
```

79 %

Connected. (1/1) | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (77) | SuperstoreDB | 00:00:00 | 0 rows

Ln 60 | Col 75 | Ch 75 | INS

SQLQuery7.sql - D...P-U2231K2\hp (77))* | SQLQuery5.sql - D...P-U2231K2\hp (55))* | SQLQuery3.sql - D...P-U2231K2\hp (64))*

```sql
SELECT
    tp.SalesYear,
    tp.SalesMonth,
    DATENAME(MONTH, DATEFROMPARTS(tp.SalesYear, tp.SalesMonth, 1)) as MonthName,
    tp.Segment,
    tp.Product_ID,
    tp.Product_Name,
    tp.Category,
    tp.TotalSales,
    tp.TotalQuantity,
    tp.TotalProfit,
    tp.OrderCount,
    ROUND((tp.TotalSales / ms.TotalSegmentSales) * 100, 2) as SegmentSalesPercentage
FROM TopProducts tp
INNER JOIN (
    SELECT
        SalesYear,
        SalesMonth,
        Segment,
        SUM(TotalSales) as TotalSegmentSales
    FROM MonthlySales
    GROUP BY SalesYear, SalesMonth, Segment
) ms ON tp.SalesYear = ms.SalesYear
    AND tp.SalesMonth = ms.SalesMonth
    AND tp.Segment = ms.Segment
ORDER BY tp.SalesYear DESC, tp.SalesMonth, tp.Segment, tp.TotalSales DESC;
```

49 %

Results | Messages

	SalesYear	SalesMonth	MonthName	Segment	Product_ID	Product_Name	Category	TotalSales	TotalQuantity	TotalProf
1	2017	1	January	Consumer	OFF-BI-10004995	GBC DocuBind P400 Electric Binding System	Office Supplies	5443.9599609375	4	2504.22
2	2017	1	January	Consumer	TEC-PH-10001459	Samsung Galaxy Mega 6.3	Technology	2939.92993164063	7	764.381
3	2017	1	January	Consumer	OFF-BI-10004584	GBC ProClick 150 Presentation Binding System	Office Supplies	2022.27197265625	8	682.516
4	2017	1	January	Consumer	TEC-PH-10002584	Samsung Galaxy S4	Technology	2003.16796875	4	250.396
5	2017	1	January	Consumer	OFF-AP-10002945	Honeywell Enviracaire Portable HEPA Air Cleaner for...	Office Supplies	1924.16003417969	8	312.676
6	2017	1	January	Corporate	OFF-SU-10002881	Martin Yale Chadless Opener Electric Letter Opener	Office Supplies	4164.0498046875	5	83.2810
7	2017	1	January	Corporate	TEC-MA-10000822	Lexmark MX611dhe Monochrome Laser Printer	Technology	3059.98193359375	2	679.996
8	2017	1	January	Corporate	FUR-TA-10001950	Balt Solid Wood Round Tables	Furniture	892.97998046875	2	80.3682
9	2017	1	January	Corporate	OFF-BI-10000545	GBC Ibimaster 500 Manual ProClick Binding System	Office Supplies	760.97998046875	5	-1141.47
10	2017	1	January	Corporate	TEC-MA-10002178	Cisco CP-7937G Unified IP Conference Station Phone	Technology	695.700012207031	2	-27.8280

Query executed successfully. | DESKTOP-U2231K2 (16.0 RTM) | DESKTOP-U2231K2\hp (77) | SuperstoreDB | 00:00:00 | 718 rows

Ln 60 | Col 75 | Ch 75 | INS