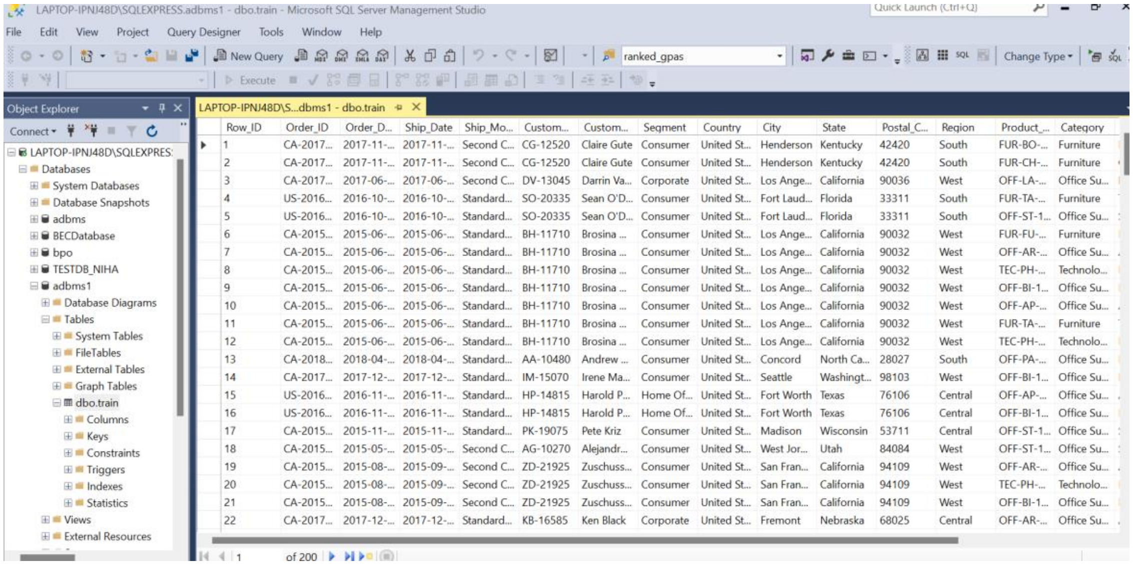
ASSIGNMENT– 3

Using Microsoft Sql Server, Power BI: Here, I have connected Sql Server to PowerBI to visualize the data

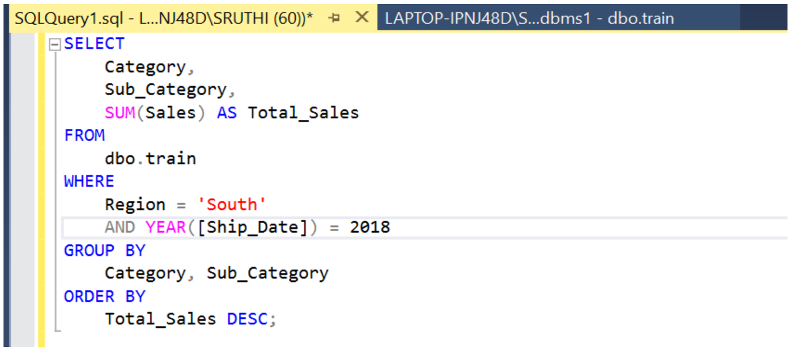


Row_ID	Order_ID	Order_D...	Ship_Date	Ship_Mo...	Custom...	Custom...	Segment	Country	City	State	Postal_C...	Region	Product_...	Category
1	CA-2017...	2017-11-...	2017-11-...	Second C...	CG-12520	Claire Gute	Consumer	United St...	Henderson	Kentucky	42420	South	FUR-BO-...	Furniture
2	CA-2017...	2017-11-...	2017-11-...	Second C...	CG-12520	Claire Gute	Consumer	United St...	Henderson	Kentucky	42420	South	FUR-CH-...	Furniture
3	CA-2017...	2017-06-...	2017-06-...	Second C...	DV-13045	Darrin Va...	Corporate	United St...	Los Ange...	California	90036	West	OFF-LA-...	Office Su...
4	US-2016...	2016-10-...	2016-10-...	Standard...	SO-20335	Sean O'D...	Consumer	United St...	Fort Laud...	Florida	33311	South	FUR-TA-...	Furniture
5	US-2016...	2016-10-...	2016-10-...	Standard...	SO-20335	Sean O'D...	Consumer	United St...	Fort Laud...	Florida	33311	South	OFF-ST-1...	Office Su...
6	CA-2015...	2015-06-...	2015-06-...	Standard...	BH-11710	Brosina ...	Consumer	United St...	Los Ange...	California	90032	West	FUR-FU-...	Furniture
7	CA-2015...	2015-06-...	2015-06-...	Standard...	BH-11710	Brosina ...	Consumer	United St...	Los Ange...	California	90032	West	OFF-AR-...	Office Su...
8	CA-2015...	2015-06-...	2015-06-...	Standard...	BH-11710	Brosina ...	Consumer	United St...	Los Ange...	California	90032	West	TEC-PH-...	Technolo...
9	CA-2015...	2015-06-...	2015-06-...	Standard...	BH-11710	Brosina ...	Consumer	United St...	Los Ange...	California	90032	West	OFF-BI-1...	Office Su...
10	CA-2015...	2015-06-...	2015-06-...	Standard...	BH-11710	Brosina ...	Consumer	United St...	Los Ange...	California	90032	West	OFF-AP-...	Office Su...
11	CA-2015...	2015-06-...	2015-06-...	Standard...	BH-11710	Brosina ...	Consumer	United St...	Los Ange...	California	90032	West	FUR-TA-...	Furniture
12	CA-2015...	2015-06-...	2015-06-...	Standard...	BH-11710	Brosina ...	Consumer	United St...	Los Ange...	California	90032	West	TEC-PH-...	Technolo...
13	CA-2018...	2018-04-...	2018-04-...	Standard...	AA-10480	Andrew ...	Consumer	United St...	Concord	North Ca...	28027	South	OFF-PA-...	Office Su...
14	CA-2017...	2017-12-...	2017-12-...	Standard...	IM-15070	Irene Ma...	Consumer	United St...	Seattle	Washingt...	98103	West	OFF-BI-1...	Office Su...
15	US-2016...	2016-11-...	2016-11-...	Standard...	HP-14815	Harold P...	Home Of...	United St...	Fort Worth	Texas	76106	Central	OFF-AP-...	Office Su...
16	US-2016...	2016-11-...	2016-11-...	Standard...	HP-14815	Harold P...	Home Of...	United St...	Fort Worth	Texas	76106	Central	OFF-BI-1...	Office Su...
17	CA-2015...	2015-11-...	2015-11-...	Standard...	PK-19075	Pete Kriz	Consumer	United St...	Madison	Wisconsin	53711	Central	OFF-ST-1...	Office Su...
18	CA-2015...	2015-05-...	2015-05-...	Second C...	AG-10270	Alejandr...	Consumer	United St...	West Jor...	Utah	84084	West	OFF-ST-1...	Office Su...
19	CA-2015...	2015-08-...	2015-09-...	Second C...	ZD-21925	Zuschuss...	Consumer	United St...	San Fran...	California	94109	West	OFF-AR-...	Office Su...
20	CA-2015...	2015-08-...	2015-09-...	Second C...	ZD-21925	Zuschuss...	Consumer	United St...	San Fran...	California	94109	West	TEC-PH-...	Technolo...
21	CA-2015...	2015-08-...	2015-09-...	Second C...	ZD-21925	Zuschuss...	Consumer	United St...	San Fran...	California	94109	West	OFF-BI-1...	Office Su...
22	CA-2017...	2017-12-...	2017-12-...	Standard...	KB-16585	Ken Black	Corporate	United St...	Fremont	Nebraska	68025	Central	OFF-AR-...	Office Su...

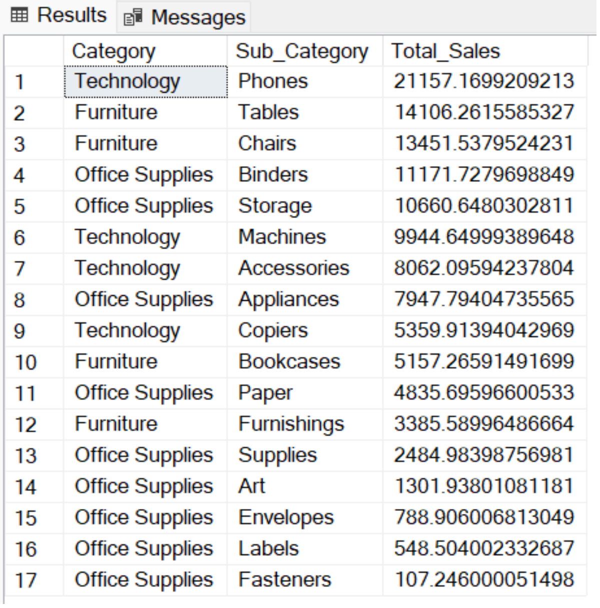
1) Analyze Product Sales

a) Perform basic drill-down operations to evaluate sales performance in a single region over the most recent year.

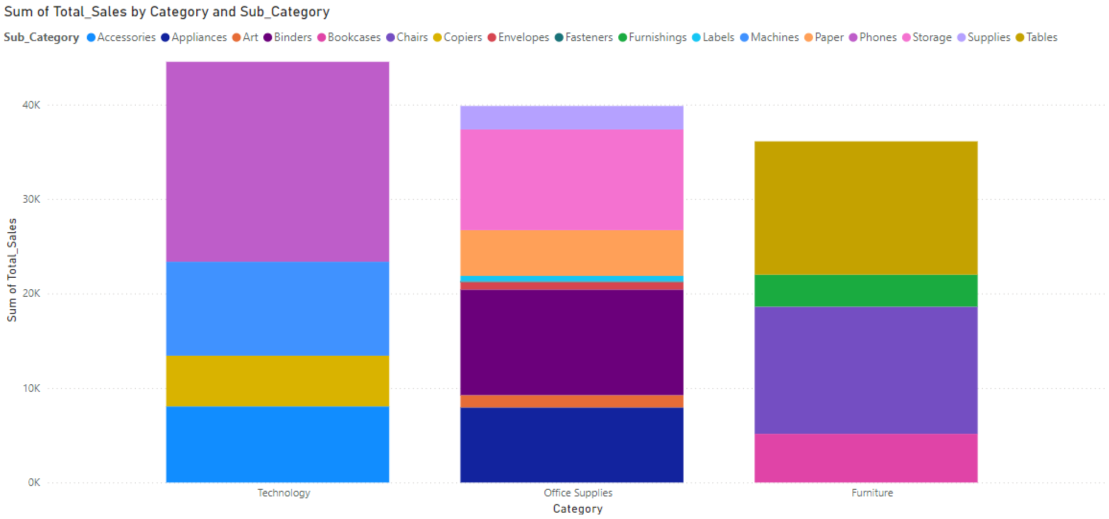
The below sql query analyzes sales data for the 'South' region in 2018. It selects the Category, Sub_Category, and the sum of Sales for each group. The WHERE clause filters the data to include only records from the 'South' region and the year 2018. The GROUP BY groups the data by Category and Sub_Category to calculate total sales for each combination. The results are ordered in descending order of Total_Sales to highlight top-performing categories. This query provides a detailed view of sales performance by category and subcategory. The plot gives the total sales for each category – for technology it is >40k, office Supplies = 40k, furniture >30k. The entire bar chart gives which sub categories under each category, the total sum of sales at each category, each category sale. Here, technology category has highest number of phones for sales and minimal is copiers, office supplies storage category highest number of sales and minimal is labels, for furniture tables are maximal and minimal is furnishings


SQLQuery1.sql - L...NJ48D\SRUTHI (60))*
LAPTOP-IPNJ48D\S...dbms1 - dbo.train

```sql
SELECT
    Category,
    Sub_Category,
    SUM(Sales) AS Total_Sales
FROM
    dbo.train
WHERE
    Region = 'South'
    AND YEAR([Ship_Date]) = 2018
GROUP BY
    Category, Sub_Category
ORDER BY
    Total_Sales DESC;
```

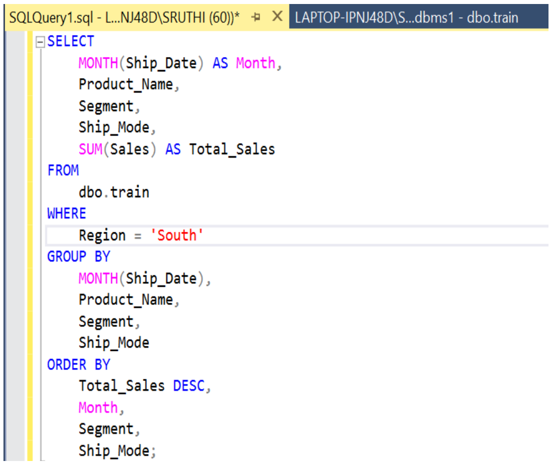


	Category	Sub_Category	Total_Sales
1	Technology	Phones	21157.1699209213
2	Furniture	Tables	14106.2615585327
3	Furniture	Chairs	13451.5379524231
4	Office Supplies	Binders	11171.7279698849
5	Office Supplies	Storage	10660.6480302811
6	Technology	Machines	9944.64999389648
7	Technology	Accessories	8062.09594237804
8	Office Supplies	Appliances	7947.79404735565
9	Technology	Copiers	5359.91394042969
10	Furniture	Bookcases	5157.26591491699
11	Office Supplies	Paper	4835.69596600533
12	Furniture	Furnishings	3385.58996486664
13	Office Supplies	Supplies	2484.98398756981
14	Office Supplies	Art	1301.93801081181
15	Office Supplies	Envelopes	788.906006813049
16	Office Supplies	Labels	548.504002332687
17	Office Supplies	Fasteners	107.246000051498

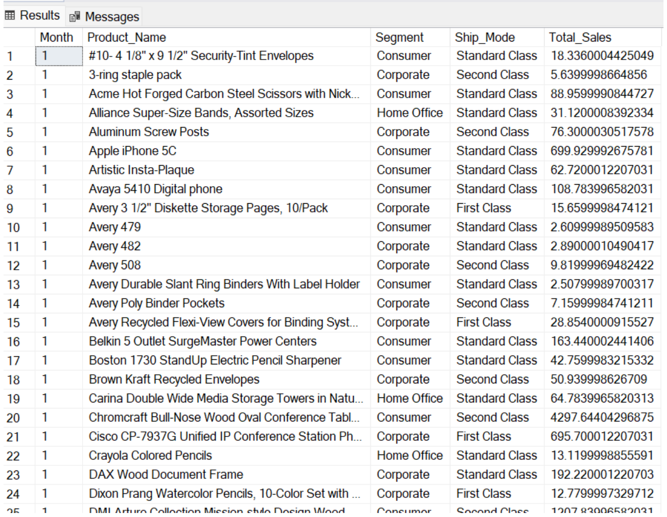


b) Identify top-selling products in this region.

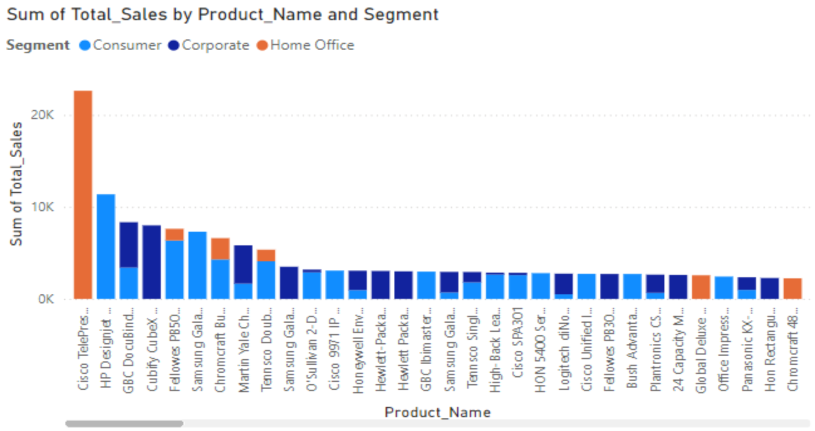
This query aggregates sales data from the dbo.train table for the 'South' region. It extracts the month from the Ship_Date using MONTH(), and groups the data by Segment and Ship_Mode. The SUM(Sales) function calculates the total sales for each combination of month, segment, and shipping mode, product_name. The results are ordered first by month, then by segment, ship mode, and finally by total sales in descending order. This query helps in analyzing sales performance across different months, customer segments, and shipping methods. The below graph clearly explains about “Cisco Telepresence system” product has highest number of sales – which is from segment home office. And the next is HP design – which is from Consumer segment. Some of the products comes from 2 or 3 segments.


SQLQuery1.sql - L...NJ48D\SRUTHI (60))* ⊐ X
LAPTOP-IPNJ48D\S...dbms1 - dbo.train

```
SELECT
    MONTH(Ship_Date) AS Month,
    Product_Name,
    Segment,
    Ship_Mode,
    SUM(Sales) AS Total_Sales
FROM
    dbo.train
WHERE
    Region = 'South'
GROUP BY
    MONTH(Ship_Date),
    Product_Name,
    Segment,
    Ship_Mode
ORDER BY
    Total_Sales DESC,
    Month,
    Segment,
    Ship_Mode;
```


Results
Messages

	Month	Product_Name	Segment	Ship_Mode	Total_Sales
1	1	#10- 4 1/8" x 9 1/2" Security-Tint Envelopes	Consumer	Standard Class	18.3360004425049
2	1	3-ring staple pack	Corporate	Second Class	5.6399998664856
3	1	Acme Hot Forged Carbon Steel Scissors with Nick...	Consumer	Standard Class	88.9599990844727
4	1	Alliance Super-Size Bands, Assorted Sizes	Home Office	Standard Class	31.1200008392334
5	1	Aluminum Screw Posts	Corporate	Second Class	76.3000030517578
6	1	Apple iPhone 5C	Consumer	Standard Class	699.929992675781
7	1	Artistic Insta-Plaque	Consumer	Standard Class	62.7200012207031
8	1	Avaya 5410 Digital phone	Consumer	Standard Class	108.783996582031
9	1	Avery 3 1/2" Diskette Storage Pages, 10/Pack	Corporate	First Class	15.6599998474121
10	1	Avery 479	Consumer	Standard Class	2.60999989509583
11	1	Avery 482	Corporate	Standard Class	2.89000010490417
12	1	Avery 508	Corporate	Second Class	9.81999969482422
13	1	Avery Durable Slant Ring Binders With Label Holder	Consumer	Standard Class	2.50799989700317
14	1	Avery Poly Binder Pockets	Corporate	Second Class	7.15999984741211
15	1	Avery Recycled Flexi-View Covers for Binding Syst...	Corporate	First Class	28.8540000915527
16	1	Belkin 5 Outlet SurgeMaster Power Centers	Consumer	Standard Class	163.440002441406
17	1	Boston 1730 StandUp Electric Pencil Sharpener	Consumer	Standard Class	42.7599983215332
18	1	Brown Kraft Recycled Envelopes	Corporate	Second Class	50.939998626709
19	1	Carina Double Wide Media Storage Towers in Natu...	Home Office	Standard Class	64.7839965820313
20	1	Chromcraft Bull-Nose Wood Oval Conference Tabl...	Consumer	Second Class	4297.64404296875
21	1	Cisco CP-7937G Unified IP Conference Station Ph...	Corporate	First Class	695.700012207031
22	1	Crayola Colored Pencils	Home Office	Standard Class	13.1199998855591
23	1	DAX Wood Document Frame	Corporate	Standard Class	192.220001220703
24	1	Dixon Prang Watercolor Pencils, 10-Color Set with ...	Corporate	First Class	12.7799997329712
25	1	DMI Arturo Collection Mission-style Design Wood...	Consumer	Second Class	1207.83996582031

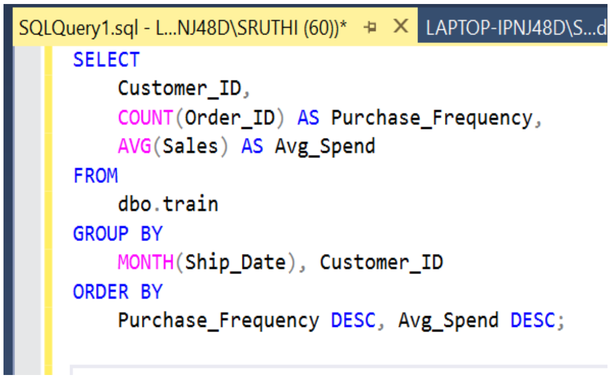


2) Customer Segmentation

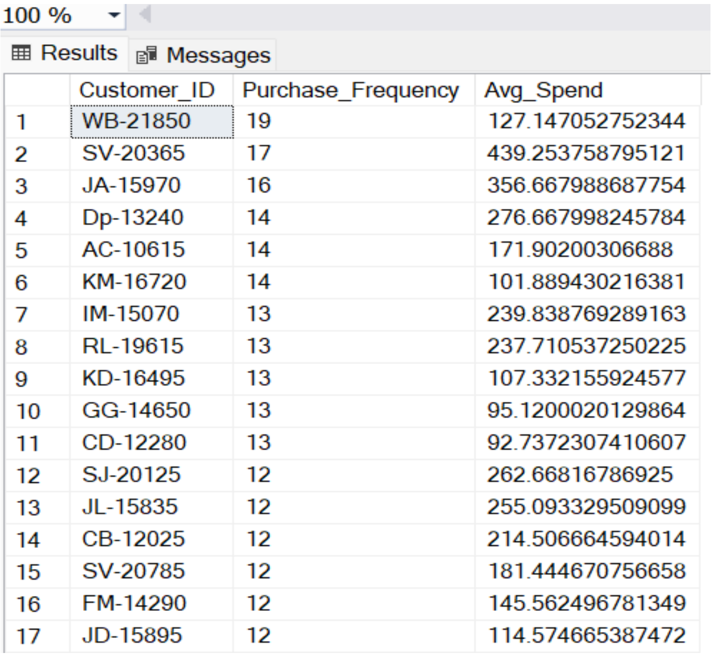
a) Use simple OLAP operations to segment customers based on a limited set of criteria (e.g., purchase frequency or average spend).

Drill-down Operation:

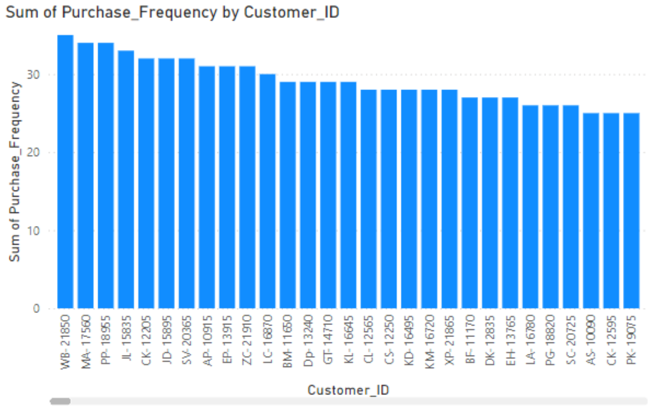
This query analyzes customer purchasing behavior by calculating two key metrics: purchase frequency and average spend. It first groups the data by Customer_ID and the month of the ship date. The COUNT(Order_ID) function counts the number of orders placed by each customer within a given month, which represents the purchase frequency. The AVG(Sales) function calculates the average amount spent by each customer, representing the average spend. The query then orders the results in descending order based on purchase frequency and average spend, helping to identify the most frequent and highest-spending customers.

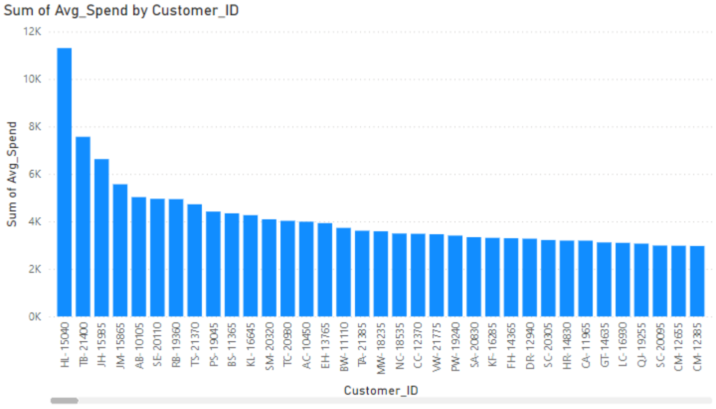


```sql
SELECT
    Customer_ID,
    COUNT(Order_ID) AS Purchase_Frequency,
    AVG(Sales) AS Avg_Spend
FROM
    dbo.train
GROUP BY
    MONTH(Ship_Date), Customer_ID
ORDER BY
    Purchase_Frequency DESC, Avg_Spend DESC;
```



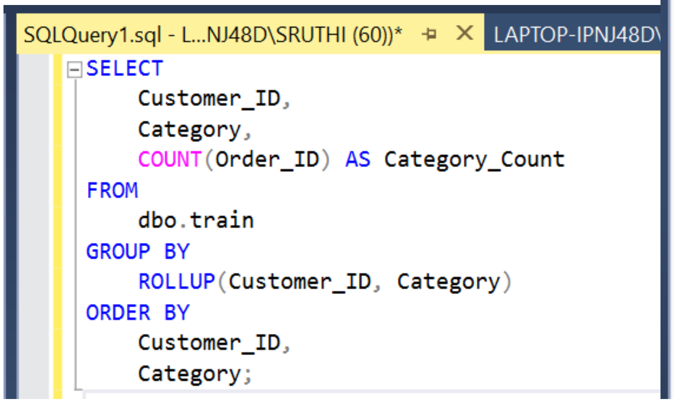
	Customer_ID	Purchase_Frequency	Avg_Spend
1	WB-21850	19	127.147052752344
2	SV-20365	17	439.253758795121
3	JA-15970	16	356.667988687754
4	Dp-13240	14	276.667998245784
5	AC-10615	14	171.90200306688
6	KM-16720	14	101.889430216381
7	IM-15070	13	239.838769289163
8	RL-19615	13	237.710537250225
9	KD-16495	13	107.332155924577
10	GG-14650	13	95.1200020129864
11	CD-12280	13	92.7372307410607
12	SJ-20125	12	262.66816786925
13	JL-15835	12	255.093329509099
14	CB-12025	12	214.50664594014
15	SV-20785	12	181.444670756658
16	FM-14290	12	145.562496781349
17	JD-15895	12	114.574665387472



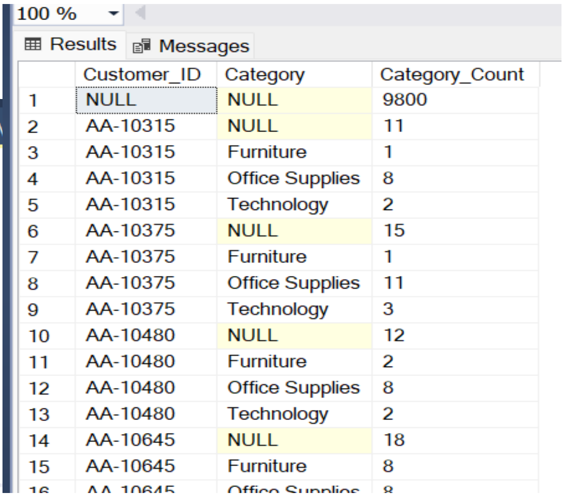


Roll-up Operation:

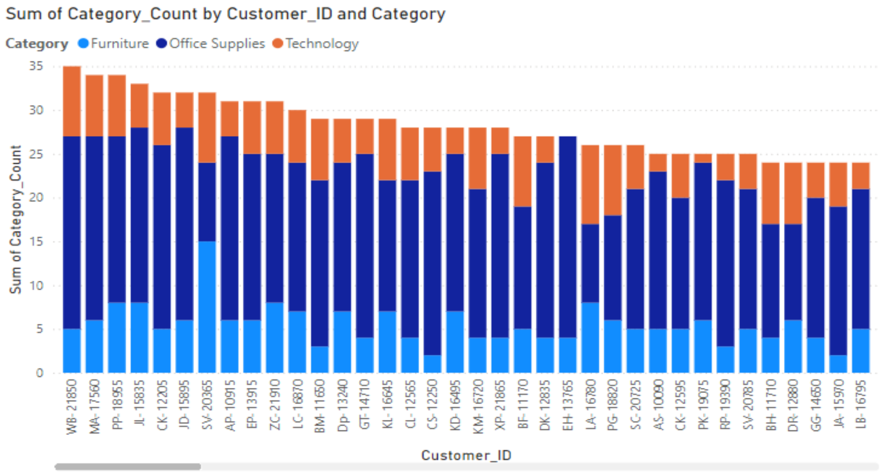
This SQL query performs a roll-up operation on the Customer_ID and Category columns from the dbo.train table. It counts the number of orders (Order_ID) for each combination of customer and category, and calculates the total count at both the customer and category levels. The HAVING clause filters out rows where either Customer_ID or Category is null, which can result from the roll-up operation. The results are then ordered by Customer_ID and Category. This allows for an organized view of the order counts by customer and category. As you can see highest number of category count in below chart, describing the count which customer bought from which categories.



```
SELECT
    Customer_ID,
    Category,
    COUNT(Order_ID) AS Category_Count
FROM
    dbo.train
GROUP BY
    ROLLUP(Customer_ID, Category)
ORDER BY
    Customer_ID,
    Category;
```

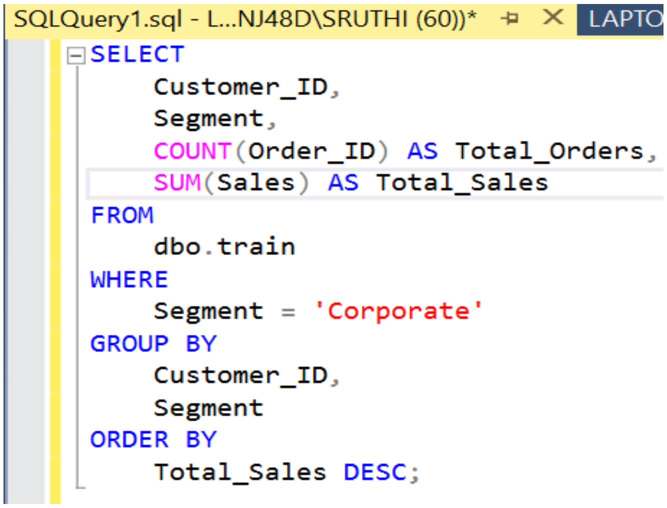


	Customer_ID	Category	Category_Count
1	NULL	NULL	9800
2	AA-10315	NULL	11
3	AA-10315	Furniture	1
4	AA-10315	Office Supplies	8
5	AA-10315	Technology	2
6	AA-10375	NULL	15
7	AA-10375	Furniture	1
8	AA-10375	Office Supplies	11
9	AA-10375	Technology	3
10	AA-10480	NULL	12
11	AA-10480	Furniture	2
12	AA-10480	Office Supplies	8
13	AA-10480	Technology	2
14	AA-10645	NULL	18
15	AA-10645	Furniture	8
16	AA-10645	Office Supplies	8

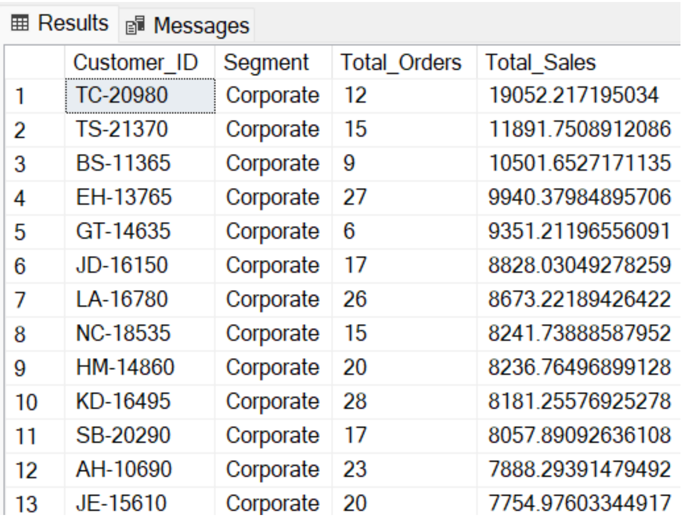


Slice Operation:

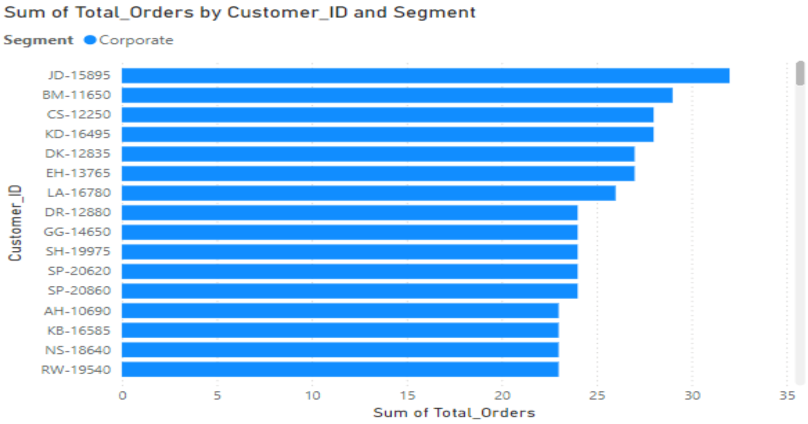
This query performs customer segmentation by focusing on the Corporate segment. It calculates the total number of orders and total sales for each customer within this segment. Using the WHERE clause, it filters records to include only customers from the Corporate segment. The GROUP BY clause groups data by Customer_ID and Segment, allowing aggregation of sales and orders. Finally, the results are sorted in descending order of Total_Sales, providing a ranking of Corporate customers by their sales contribution.



```
SELECT
    Customer_ID,
    Segment,
    COUNT(Order_ID) AS Total_Orders,
    SUM(Sales) AS Total_Sales
FROM
    dbo.train
WHERE
    Segment = 'Corporate'
GROUP BY
    Customer_ID,
    Segment
ORDER BY
    Total_Sales DESC;
```

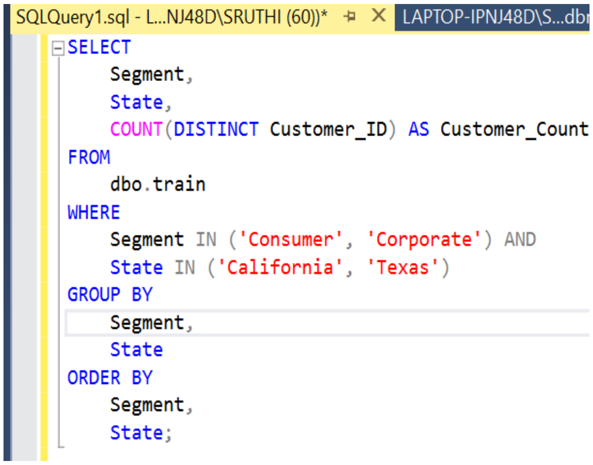


	Customer_ID	Segment	Total_Orders	Total_Sales
1	TC-20980	Corporate	12	19052.217195034
2	TS-21370	Corporate	15	11891.7508912086
3	BS-11365	Corporate	9	10501.6527171135
4	EH-13765	Corporate	27	9940.37984895706
5	GT-14635	Corporate	6	9351.21196556091
6	JD-16150	Corporate	17	8828.03049278259
7	LA-16780	Corporate	26	8673.22189426422
8	NC-18535	Corporate	15	8241.73888587952
9	HM-14860	Corporate	20	8236.76496899128
10	KD-16495	Corporate	28	8181.25576925278
11	SB-20290	Corporate	17	8057.89092636108
12	AH-10690	Corporate	23	7888.29391479492
13	JE-15610	Corporate	20	7754.97603344917

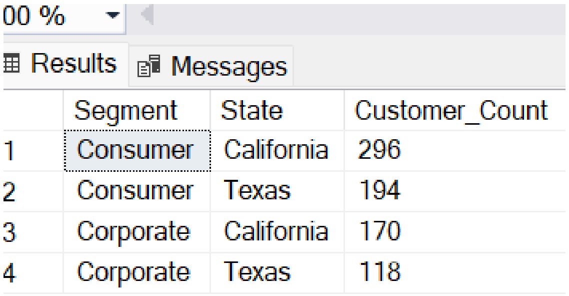


Dice Operation:

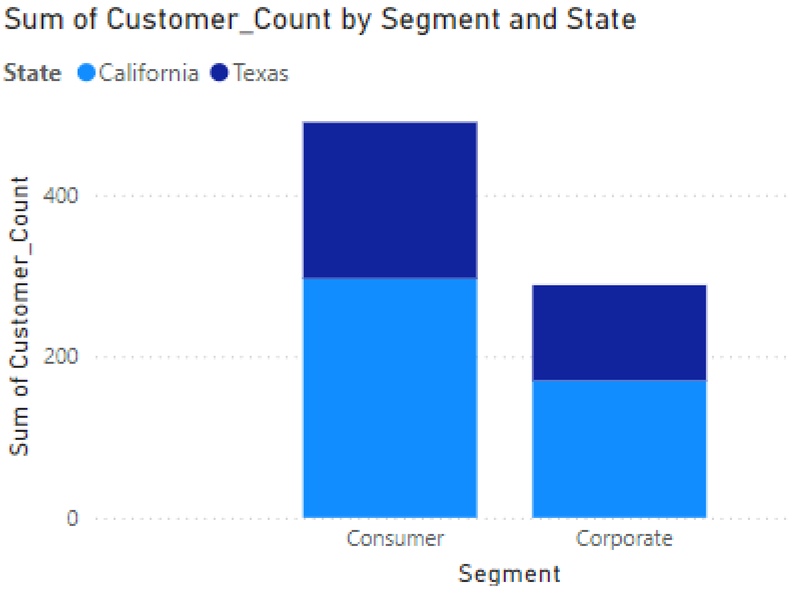
This SQL query retrieves the number of distinct customers and total sales for specific segments and states. It filters the data for customers in the 'Consumer' and 'Corporate' segments and located in 'California' and 'Texas'. The query counts distinct Customer_ID for each combination of segment and state. The results are grouped by Segment and State, allowing segmentation of customers based on these criteria. The bar chart clearly explains about states and type of segment and the customer count.



```sql
SELECT
    Segment,
    State,
    COUNT(DISTINCT Customer_ID) AS Customer_Count
FROM
    dbo.train
WHERE
    Segment IN ('Consumer', 'Corporate') AND
    State IN ('California', 'Texas')
GROUP BY
    Segment,
    State
ORDER BY
    Segment,
    State;
```

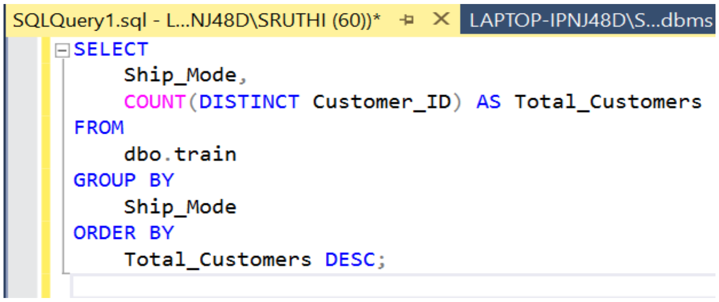


	Segment	State	Customer_Count
1	Consumer	California	296
2	Consumer	Texas	194
3	Corporate	California	170
4	Corporate	Texas	118

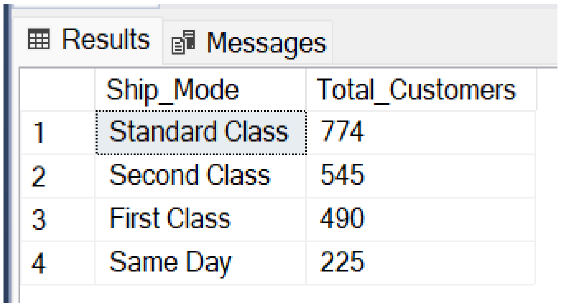


b) Identify general trends in customer preferences

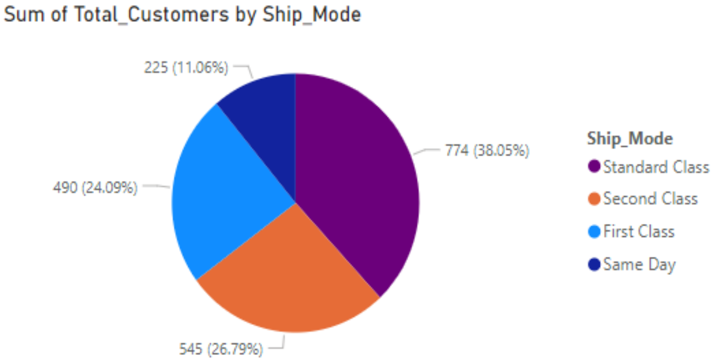
This query calculates the total number of unique customers using each shipping mode from the dbo.train table. The COUNT(DISTINCT Customer_ID) function ensures that duplicate customers are not counted multiple times. The results are grouped by Ship_Mode to show customer totals for each shipping method. The ORDER BY Total_Customers DESC clause sorts the output in descending order of customer count, highlighting the most popular shipping modes first. The below analysis provides insights into customer preferences based on shipping modes.



```
SELECT
    Ship_Mode,
    COUNT(DISTINCT Customer_ID) AS Total_Customers
FROM
    dbo.train
GROUP BY
    Ship_Mode
ORDER BY
    Total_Customers DESC;
```



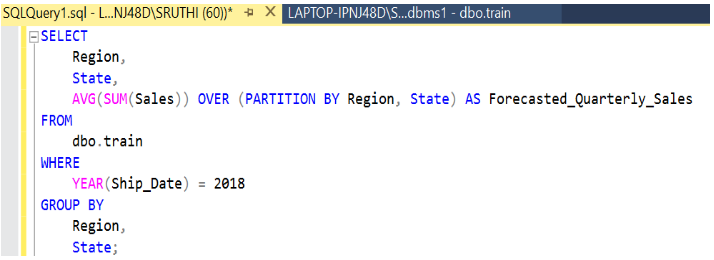
	Ship_Mode	Total_Customers
1	Standard Class	774
2	Second Class	545
3	First Class	490
4	Same Day	225



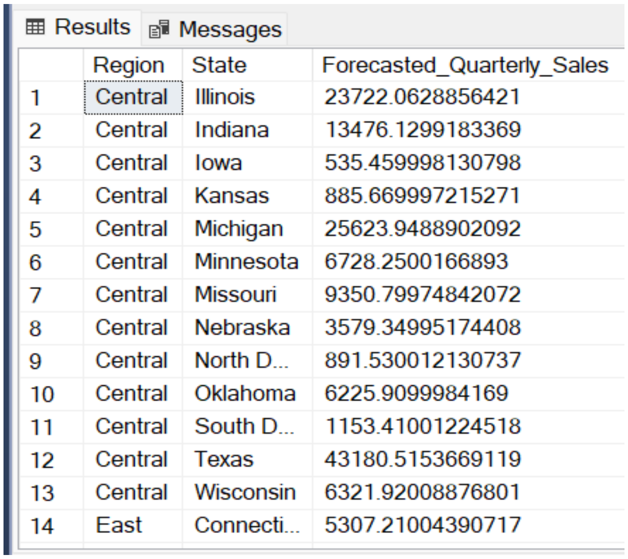
3) **Sales Forecasting**

a) Use basic aggregation techniques to forecast sales for the next quarter, considering simple factors like historical sales trends.

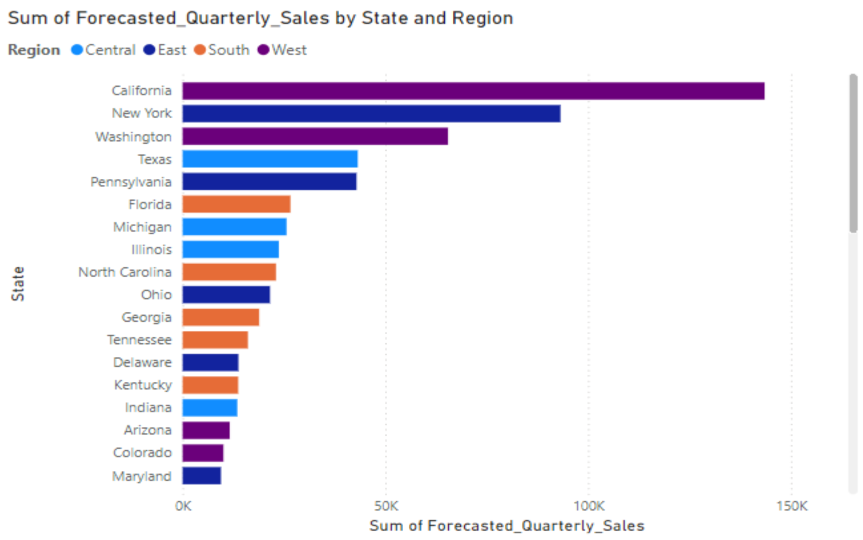
This query calculates the forecasted quarterly sales for each region and state. It aggregates the total sales per region and state in 2018. The SUM(Sales) calculates the total sales for each group, and AVG over partition by computes the average quarterly sales for each region and state. The WHERE clause filters the data for the year 2018. The GROUP BY clause ensures the sales are aggregated by region and state for meaningful analysis.


SQLQuery1.sql - L...NJ48D\SRUTHI (60))*
LAPTOP-IPNJ48D\S...dbms1 - dbo.train

```
SELECT
    Region,
    State,
    AVG(SUM(Sales)) OVER (PARTITION BY Region, State) AS Forecasted_Quarterly_Sales
FROM
    dbo.train
WHERE
    YEAR(Ship_Date) = 2018
GROUP BY
    Region,
    State;
```



	Region	State	Forecasted_Quarterly_Sales
1	Central	Illinois	23722.0628856421
2	Central	Indiana	13476.1299183369
3	Central	Iowa	535.459998130798
4	Central	Kansas	885.669997215271
5	Central	Michigan	25623.9488902092
6	Central	Minnesota	6728.2500166893
7	Central	Missouri	9350.79974842072
8	Central	Nebraska	3579.34995174408
9	Central	North D...	891.530012130737
10	Central	Oklahoma	6225.9099984169
11	Central	South D...	1153.41001224518
12	Central	Texas	43180.5153669119
13	Central	Wisconsin	6321.92008876801
14	East	Connecti...	5307.21004390717

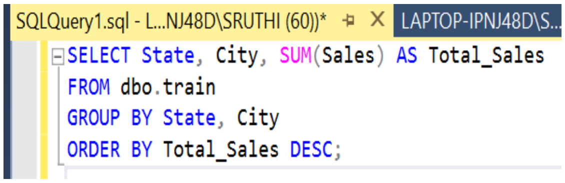


4) Basic OLAP Queries

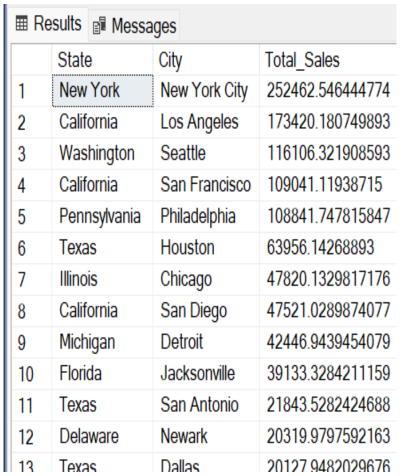
a) Construct basic OLAP queries involving a single dimension and measure.

Drill down:

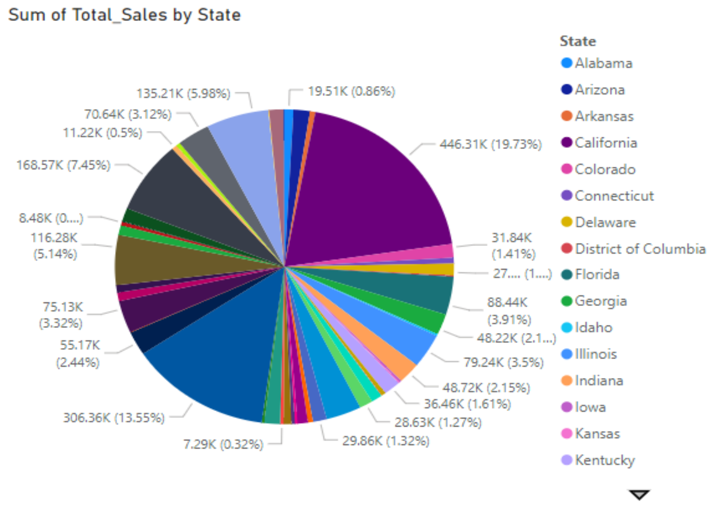
This query calculates the total sales for each combination of State and City from the dataset. It groups the data by State and City, summing up the sales for each group. The SUM(Sales) function aggregates sales figures for each State-City pair. The results are ordered in descending order based on the total sales, showing the highest sales first. This allows you to identify which states and cities contribute the most to total sales.



```
SELECT State, City, SUM(Sales) AS Total_Sales
FROM dbo.train
GROUP BY State, City
ORDER BY Total_Sales DESC;
```

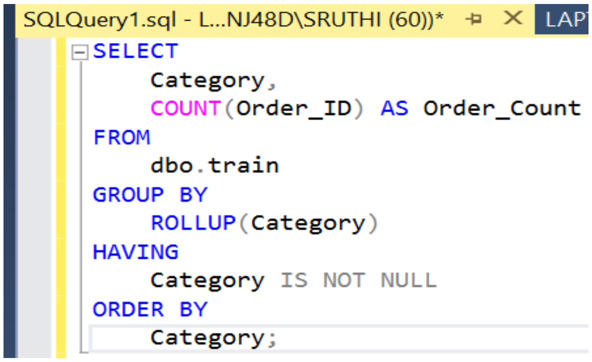


	State	City	Total_Sales
1	New York	New York City	252462.546444774
2	California	Los Angeles	173420.180749893
3	Washington	Seattle	116106.321908593
4	California	San Francisco	109041.11938715
5	Pennsylvania	Philadelphia	108841.747815847
6	Texas	Houston	63956.14268893
7	Illinois	Chicago	47820.1329817176
8	California	San Diego	47521.0289874077
9	Michigan	Detroit	42446.9439454079
10	Florida	Jacksonville	39133.3284211159
11	Texas	San Antonio	21843.5282424688
12	Delaware	Newark	20319.9797592163
13	Texas	Dallas	20127.9482029676

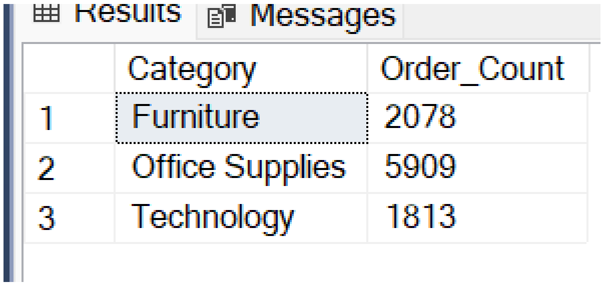


Roll up:

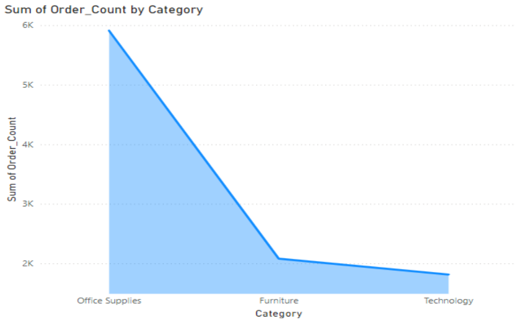
This query performs a ROLLUP operation on the Category column. It calculates the count of orders (Order_Count) for each category in the dbo.train table. The ROLLUP function aggregates the data hierarchically, providing both individual category counts and a grand total at the end. The HAVING clause filters out any null rows that appear as a result of the ROLLUP operation. Finally, the results are ordered by Category for a clearer presentation of the data.



```
SELECT
    Category,
    COUNT(Order_ID) AS Order_Count
FROM
    dbo.train
GROUP BY
    ROLLUP(Category)
HAVING
    Category IS NOT NULL
ORDER BY
    Category;
```

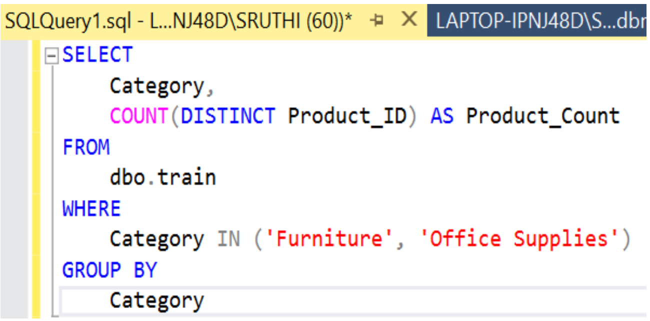


	Category	Order_Count
1	Furniture	2078
2	Office Supplies	5909
3	Technology	1813

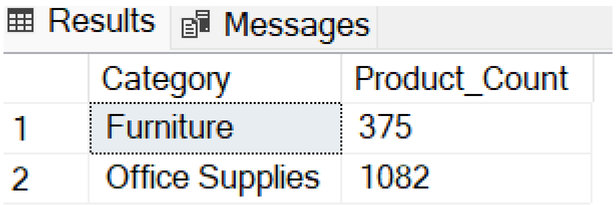


Dice:

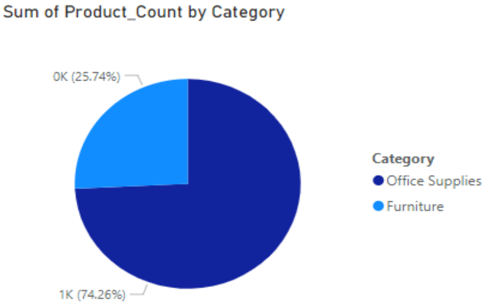
This query selects the Category and counts the distinct Product_IDs for each category from the dbo.train table. It filters the data to include only the categories 'Furniture' and 'Office Supplies' using the WHERE clause. The COUNT(DISTINCT Product_ID) function calculates the number of unique products within each category. The results are grouped by Category, meaning a separate count is returned for each specified category. The query provides an overview of how many unique products exist in each of the specified categories.



```
SELECT
    Category,
    COUNT(DISTINCT Product_ID) AS Product_Count
FROM
    dbo.train
WHERE
    Category IN ('Furniture', 'Office Supplies')
GROUP BY
    Category
```

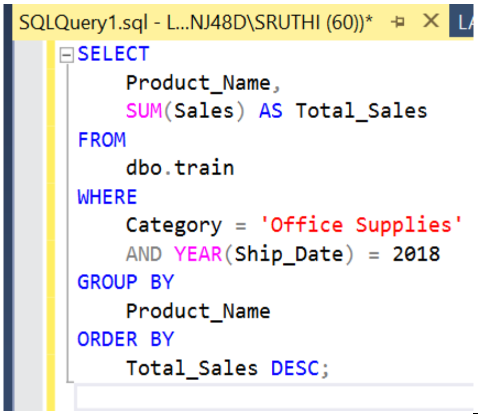


	Category	Product_Count
1	Furniture	375
2	Office Supplies	1082

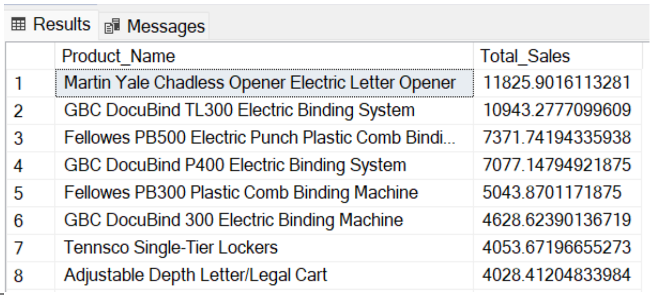


Slice:

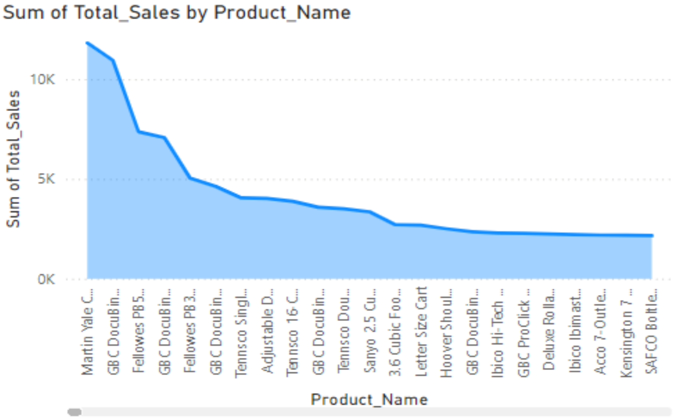
This query identifies top-selling products in the "Office Supplies" category for 2018. It filters data by category and year, groups sales by product name, and calculates total sales for each product. The results are sorted in descending order to prioritize high-performing products.



```sql
SELECT
    Product_Name,
    SUM(Sales) AS Total_Sales
FROM
    dbo.train
WHERE
    Category = 'Office Supplies'
    AND YEAR(Ship_Date) = 2018
GROUP BY
    Product_Name
ORDER BY
    Total_Sales DESC;
```

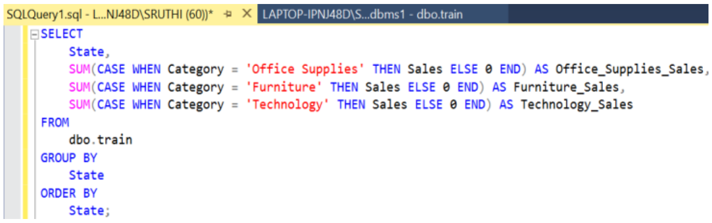


	Product_Name	Total_Sales
1	Martin Yale Chadless Opener Electric Letter Opener	11825.9016113281
2	GBC DocuBind TL300 Electric Binding System	10943.2777099609
3	Fellowes PB500 Electric Punch Plastic Comb Bindi...	7371.74194335938
4	GBC DocuBind P400 Electric Binding System	7077.14794921875
5	Fellowes PB300 Plastic Comb Binding Machine	5043.8701171875
6	GBC DocuBind 300 Electric Binding Machine	4628.62390136719
7	Tennsco Single-Tier Lockers	4053.67196655273
8	Adjustable Depth Letter/Legal Cart	4028.41204833984



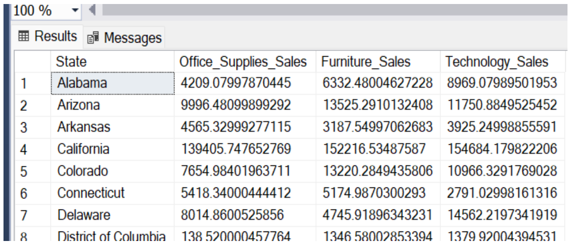
Pivot:

This query aggregates sales data by state and separates it by category. It uses a CASE statement to conditionally sum sales for each category (Office Supplies, Furniture, and Technology). The SUM function calculates the total sales for each category within each state. The GROUP BY clause groups the results by state, ensuring the sales data is segmented by state. This allows the analysis of sales performance across different categories for each state.

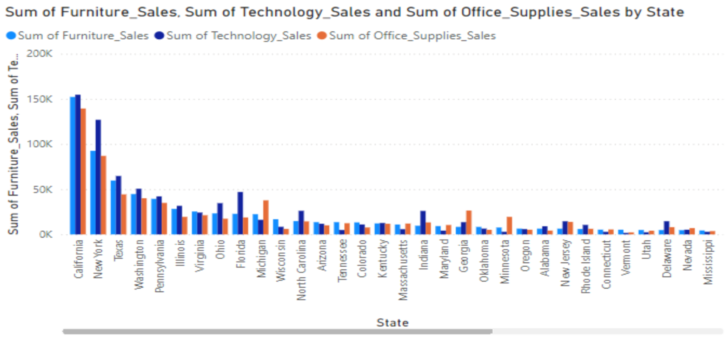


SQLQuery1.sql - L...NJ48D\SRUTHI (60))*
LAPTOP-IPNJ48D\S...dbms1 - dbo.train

```sql
SELECT
    State,
    SUM(CASE WHEN Category = 'Office Supplies' THEN Sales ELSE 0 END) AS Office_Supplies_Sales,
    SUM(CASE WHEN Category = 'Furniture' THEN Sales ELSE 0 END) AS Furniture_Sales,
    SUM(CASE WHEN Category = 'Technology' THEN Sales ELSE 0 END) AS Technology_Sales
FROM
    dbo.train
GROUP BY
    State
ORDER BY
    State;
```



	State	Office_Supplies_Sales	Furniture_Sales	Technology_Sales
1	Alabama	4209.07997870445	6332.48004627228	8969.07989501953
2	Arizona	9996.48099899292	13525.2910132408	11750.8849525452
3	Arkansas	4565.32999277115	3187.54997062683	3925.24998855591
4	California	139405.747652769	152216.53487587	154684.179822206
5	Colorado	7654.98401963711	13220.2849435806	10966.3291769028
6	Connecticut	5418.34000444412	5174.9870300293	2791.02998161316
7	Delaware	8014.8600525856	4745.91896343231	14562.2197341919
8	District of Columbia	138.520000457764	1346.58002853394	1379.92004394531



b) Focus on queries that are straightforward and demonstrate basic under-standing.

In OLAP (Online Analytical Processing), the basic operations focus on analyzing data across multiple dimensions, below are the olap operations explained in above examples, below is the functionality:
Roll-up: This operation aggregates data by climbing up a hierarchy, like summing sales data from months to quarters or years.
Drill-down: The opposite of roll-up, drill-down allows you to go into finer details by breaking down data into more granular levels, such as viewing sales by day instead of by month.
Slice: This operation selects a single level of data (one dimension) from a multidimensional dataset, like viewing sales for only a specific region or product category.
Dice: A more advanced operation that allows you to perform a slice along multiple dimensions, for example, examining sales data for a specific product category and region in a given time period.
Pivot: This operation reorganizes or rotates data to view it from different perspectives, such as swapping rows and columns in a report for better analysis.