



ADVENTURE WORKS 2019 ANALYSIS

Presentation by DEPI Students
OCT 2024

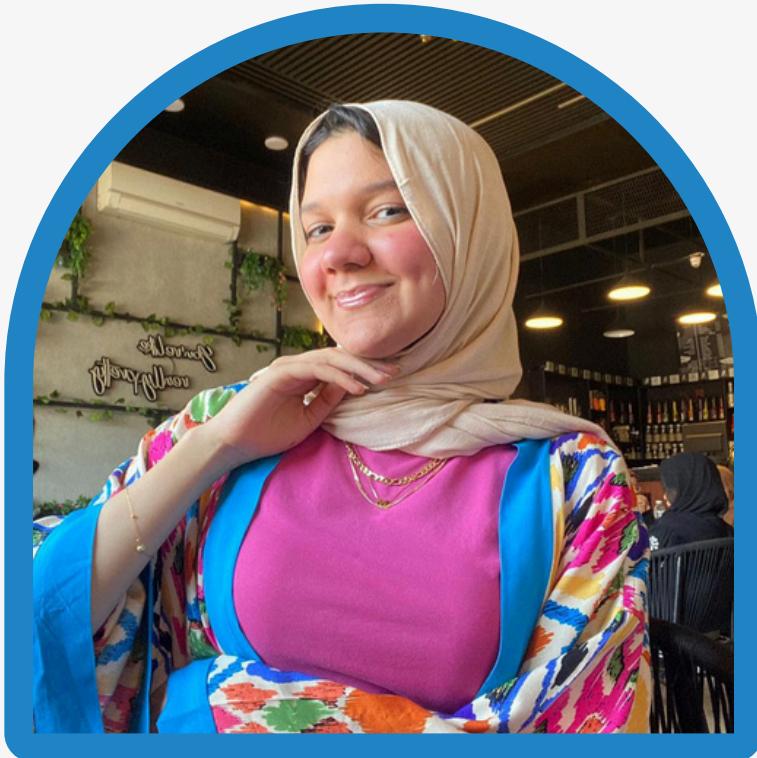
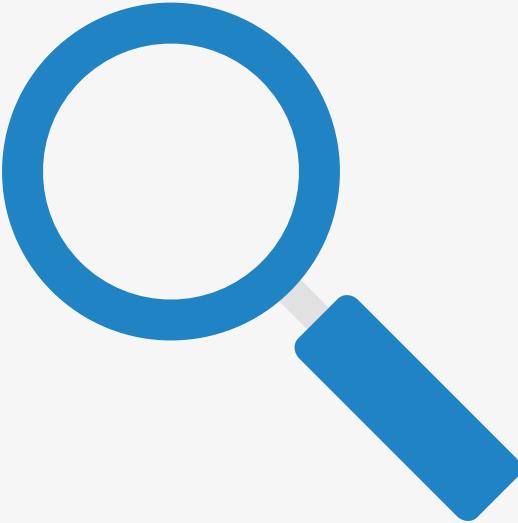
TABLE OF CONTENT

1. OUR TEAM
2. DATABASE DESCRIPTION
3. PROBLEM
4. OUR GOAL
5. DATA COLLECTION
6. DATA EXPLORATION
7. ASK QUESTIONS
8. DATA TRANSFORMATION
9. DATA VISUALIZATION
10. ENDING

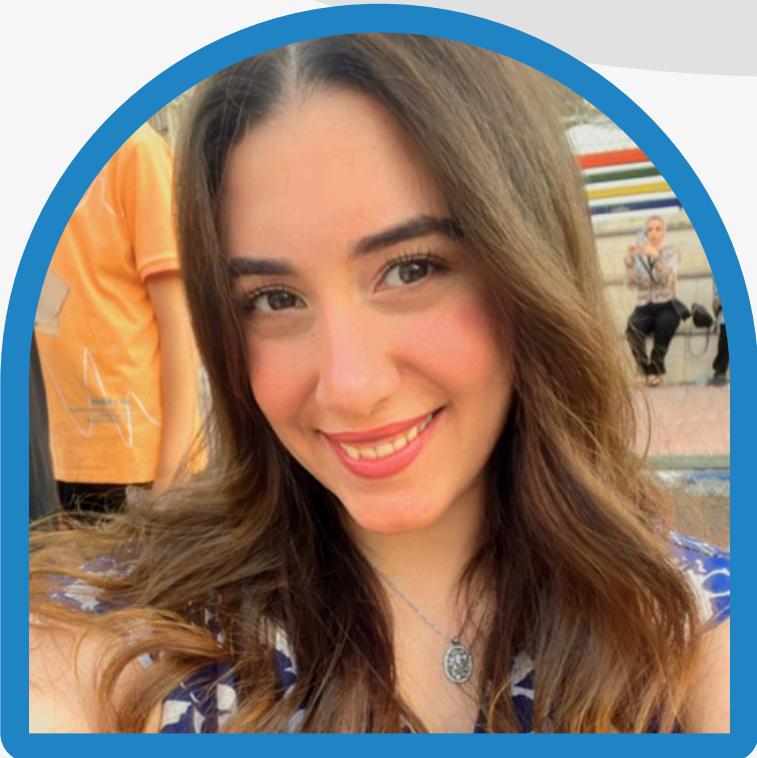




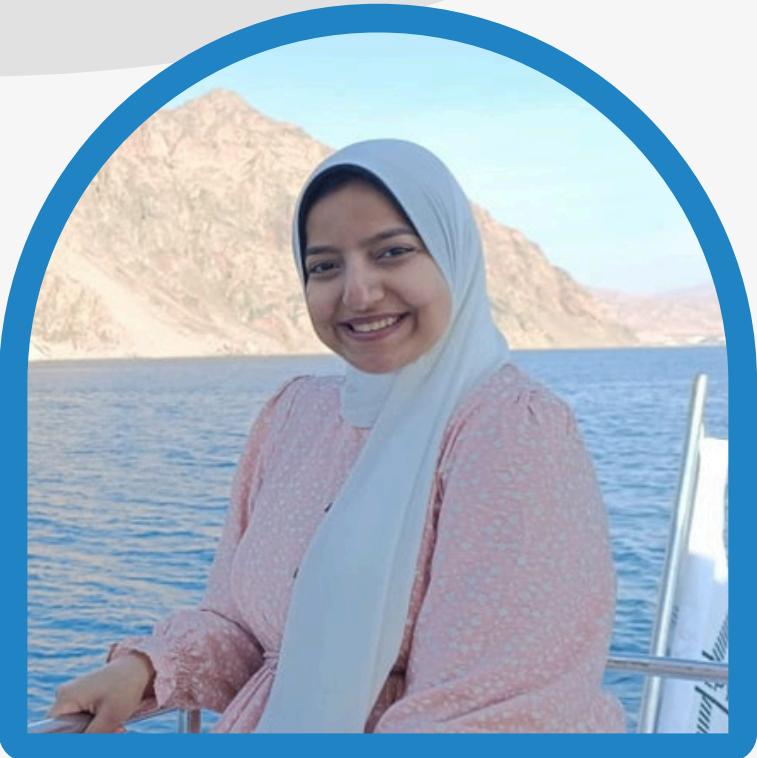
OUR TEAM



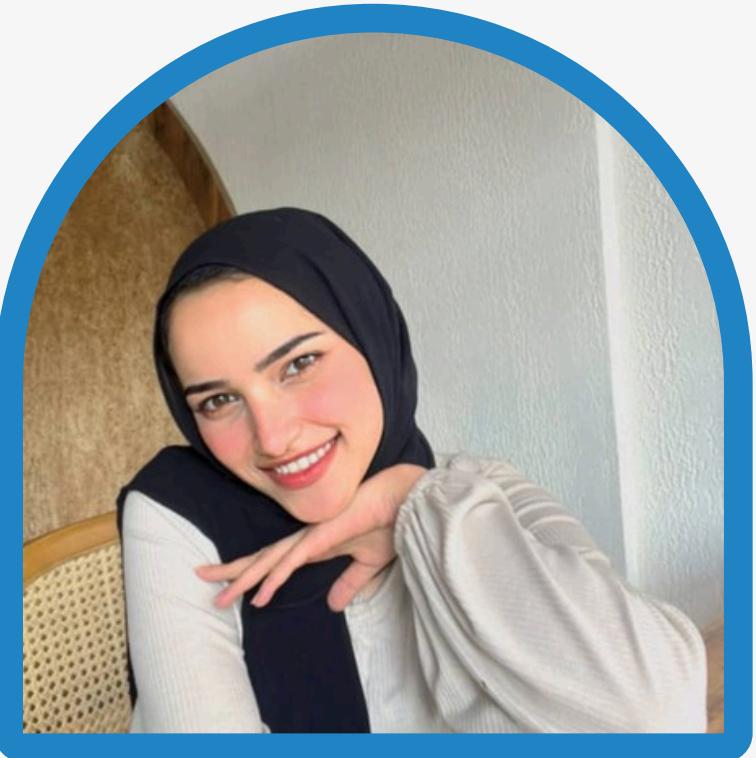
MANAR NAZEEH



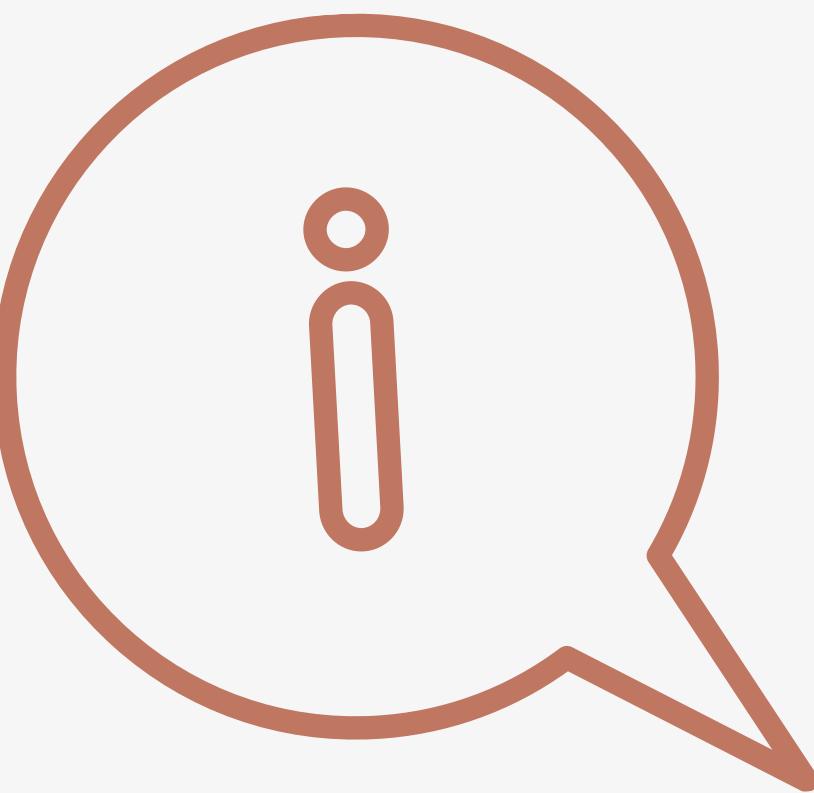
**NOURAN
WALEED**



TOQA MAHER



**NAGHAM
KHALED**



ABOUT DATABASE

Adventure Works 2019 is a sample database provided by Microsoft, primarily used for demonstrating SQL Server features and capabilities. It represents a fictional bicycle manufacturing company and includes a variety of data that showcases different aspects of a business.





COMMON TABLES AND COLUMNS

- **Products**

(ProductID,ProductName ,CategoryID,ListPrice,QuantityPerUnit)

- **SalesOrderHeaders**

(SalesOrderID,OrderDate,CustomerID,TotalDue)

- **Customers**

CustomerID,FirstName,LastName,EmailAddress)

- **Employee**

(EmployeeID,FirstName,LastName,JobTitle)

- **SalesTerritory:**

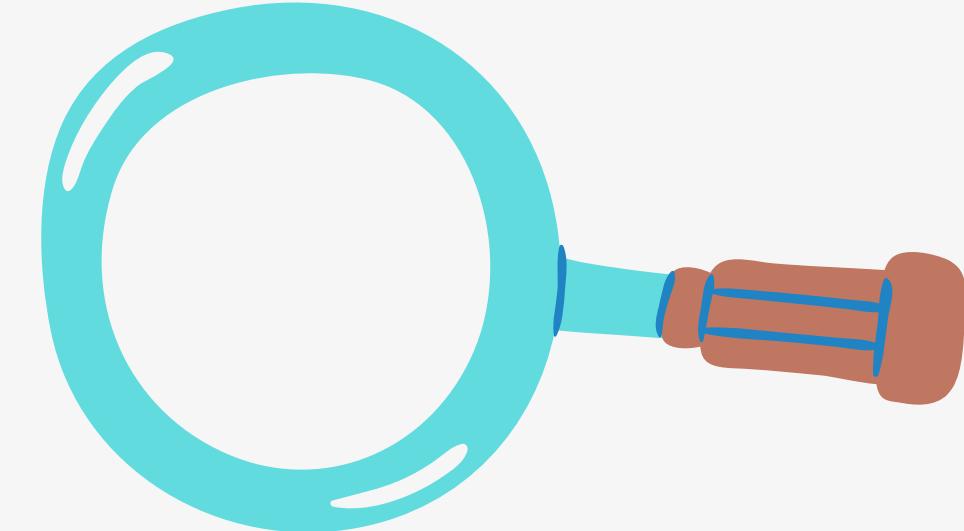
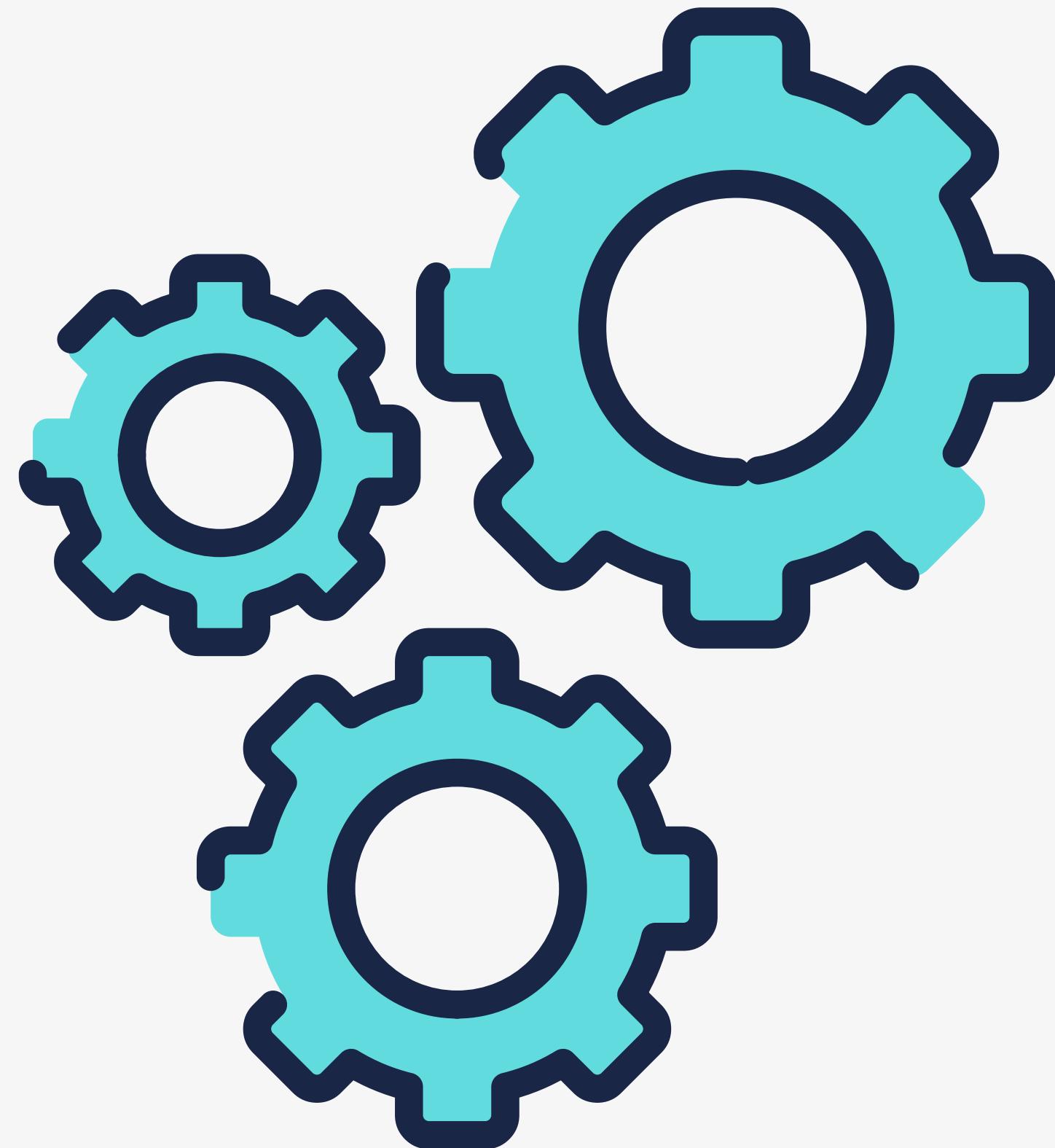
(TerritoryID,Name)



PROBLEM

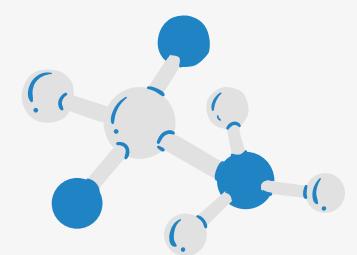
- The Adventure Works database shows a significant decline in sales across specific regions over the past few quarters.
- This trend raises concerns about market competitiveness and customer retention.



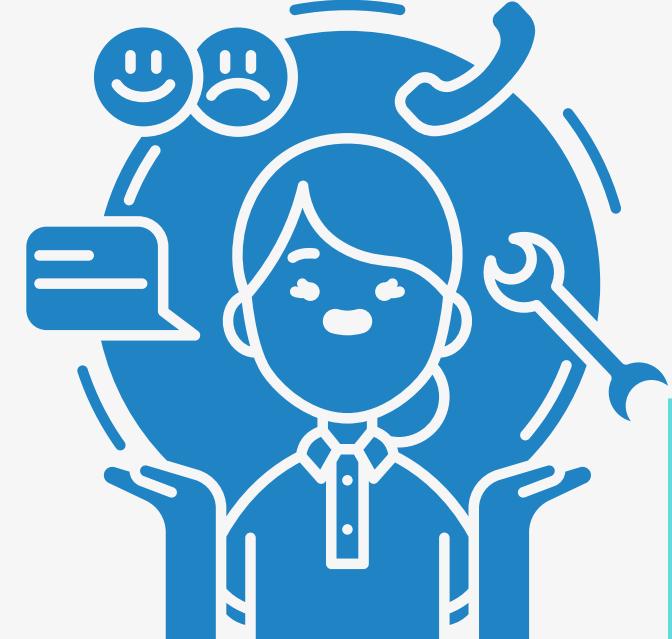


OUR GOAL

- To identify the root causes of the sales decline by analyzing sales data, customer demographics, product performance, and market trends.
- The goal is to provide actionable insights and recommendations to improve sales performance and enhance customer engagement.



ANALYSIS STEPS



1-Data Collection

★ IN THIS STAGE WE USED SQL TO RESTORE DATA

The screenshot shows two windows from Microsoft SQL Server Management Studio (Administrator) running on SQLVM1\SQL2019.

Object Explorer: Shows the connection to SQLVM1\SQL2019 (SQL Server 15.0.2000.5 - CONTOSO\myvmadmin). The **Databases** node is selected, and its context menu is open. The **Restore Database...** option is highlighted with a red rectangle.

Restore Database - AdventureWorks2019: This is the active window. It shows the configuration for restoring the **AdventureWorks2019** database. The **Source** section has **Device** selected, pointing to **C:\Program Files\Microsoft SQL Server\MSSQL15.SQL2019\MSSQL\Backup\AdventureWorks2019-Full Database Backup.bak**. The **Destination** section has **Database** set to **AdventureWorks2019**. The **Restore to:** field shows **The last backup taken (Monday, June 15, 2020 10:40:37 AM)**. The **Restore plan** table lists one backup set:

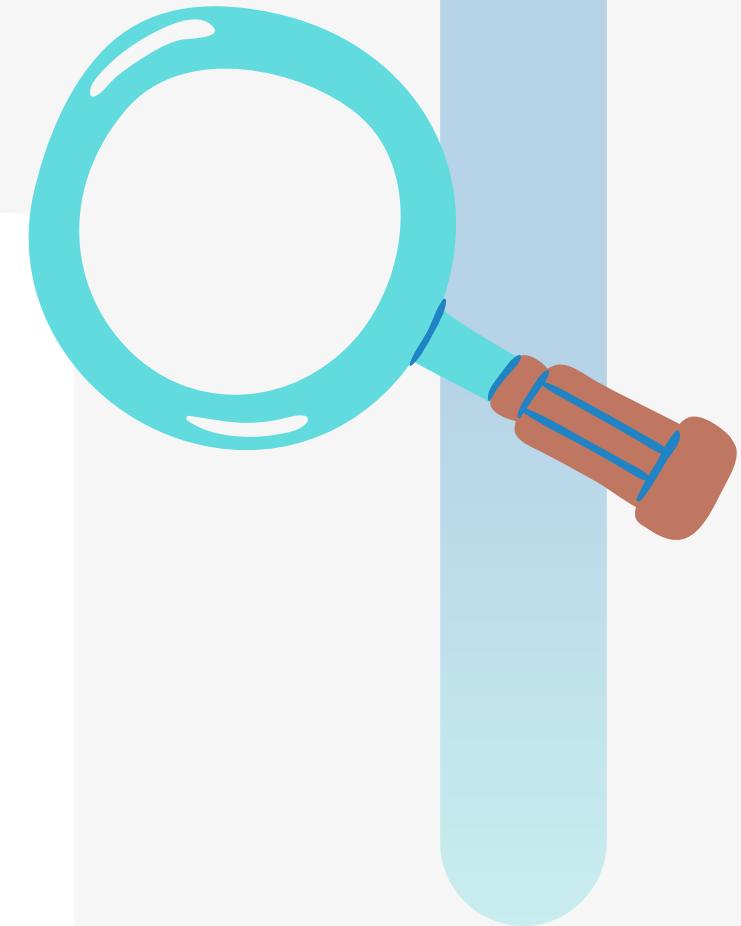
Restore	Name	Component	Type	S...	Database	Position	Firs
<input checked="" type="checkbox"/>	AdventureWorks2019-Full Database Backup	Database	Full	M...	AdventureWorks2019	1	39

The **Connection** pane shows the current connection is **SQLVM1\SQL2019 [CONTOSO\myvmadmin]**. The **Progress** pane shows the status is **Done**.

- ★ WE USED SQL ALSO TO WRITE THE SELECT QUERY WE WILL USE IN THE PROCESS OF ANALYZING DATA (EXCEL AND POWER PI)

```
SELECT
    OH.[SalesOrderID], [OrderDate], [DueDate], [ShipDate], [Status]
    , [OnlineOrderFlag], [CustomerID], [SalesPersonID],
    ter. [TerritoryID], meth.[ShipMethodID], [CreditCardID]
    , [SubTotal], [TaxAmt], [Freight], [TotalDue],
    [SalesOrderDetailID], [OrderQty], OD.[ProductID]
    , [UnitPrice], [LineTotal]
    , pp.[Name] AS ProductName , sub.[Name] AS SubCategoryName, cat.[Name] AS CategoryName
    , meth.[Name] AS ShipMethodName ,ter.[Name] AS TerritoryName, ter.[Group] AS TerritoryGroup

FROM
    [sales].[SalesOrderHeader] AS OH
JOIN
    [sales].[SalesOrderDetail] AS OD ON OH.[SalesOrderID]=OD.[SalesOrderID]
JOIN
    [Production].[Product] AS pp ON pp.[ProductID]=OD.[ProductID]
JOIN
    [Production].[ProductSubcategory] AS sub on sub.[ProductSubcategoryID]=pp.[ProductSubcategoryID]
JOIN
    [Production].[ProductCategory] AS cat ON cat.[ProductCategoryID]=sub.[ProductCategoryID]
JOIN
    [Purchasing].[ShipMethod] meth ON meth.[ShipMethodID]=OH.[ShipMethodID]
JOIN
    [Sales].[SalesTerritory] AS ter ON OH.[TerritoryID]=ter.[TerritoryID];
```



2-DATA EXPLORATION



- Exploring data involves a systematic approach to understanding a dataset's characteristics, structure, and underlying patterns.
- so we explored data using python.

3]Exploring Data

```
[111]: df.head()
```

	SalesOrderID	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	CustomerID	SalesPersonID	TerritoryID	ShipMethodID	...	UnitPrice	LineTotal	Si
0	43659	5/31/2011 0:00	6/12/2011 0:00	6/7/2011 0:00	5	False	29825	279.0	5	5	...	2024.994	2024.994	
1	43659	5/31/2011 0:00	6/12/2011 0:00	6/7/2011 0:00	5	False	29825	279.0	5	5	...	2024.994	6074.982	
2	43659	5/31/2011 0:00	6/12/2011 0:00	6/7/2011 0:00	5	False	29825	279.0	5	5	...	2024.994	2024.994	
3	43659	5/31/2011 0:00	6/12/2011 0:00	6/7/2011 0:00	5	False	29825	279.0	5	5	...	2039.994	2039.994	
4	43659	5/31/2011 0:00	6/12/2011 0:00	6/7/2011 0:00	5	False	29825	279.0	5	5	...	2039.994	2039.994	

5 rows × 28 columns

```
[113]: df.tail()
```

	SalesOrderID	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	CustomerID
121312	75122	6/30/2014 0:00	7/12/2014 0:00	7/7/2014 0:00	5	True	15868
121313	75122	6/30/2014 0:00	7/12/2014 0:00	7/7/2014 0:00	5	True	15868
121314	75123	6/30/2014 0:00	7/12/2014 0:00	7/7/2014 0:00	5	True	18759
121315	75123	6/30/2014 0:00	7/12/2014 0:00	7/7/2014 0:00	5	True	18759
121316	75123	6/30/2014 0:00	7/12/2014 0:00	7/7/2014 0:00	5	True	18759

5 rows × 28 columns

```
[7]: df.shape
```

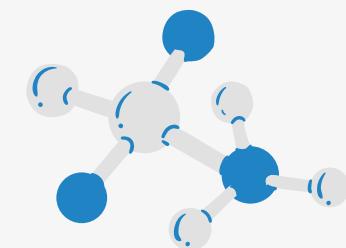
```
[7]: (121317, 28)
```

```
[9]: df.columns
```

```
[9]: Index(['SalesOrderID', 'OrderDate', 'DueDate', 'ShipDate', 'Status',
       'OnlineOrderFlag', 'CustomerID', 'SalesPersonID', 'TerritoryID',
       'ShipMethodID', 'CreditCardID', 'SubTotal', 'TaxAmt', 'Freight',
       'TotalDue', 'SalesOrderDetailID', 'OrderQty', 'ProductID', 'UnitP',
       'LineTotal', 'SubCategoryName', 'CategoryName', 'ShipMethodName',
       'TerritoryName', 'TerritoryGroup', 'ProductNameX', 'Unnamed: 26',
       'ProductName'],
      dtype='object')
```

```
[5]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 121317 entries, 0 to 121316
Data columns (total 28 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   SalesOrderID    121317 non-null   int64  
 1   OrderDate       121317 non-null   object 
 2   DueDate         121317 non-null   object 
 3   ShipDate        121317 non-null   object 
 4   Status          121317 non-null   int8   
 5   OnlineOrderFlag 121317 non-null   bool    
 6   CustomerID      121317 non-null   int64  
 7   SalesPersonID   121317 non-null   int64  
 8   TerritoryID     121317 non-null   int64  
 9   ShipMethodID    121317 non-null   int64  
 10  CreditCardID   121317 non-null   int64  
 11  SubTotal        121317 non-null   float64
 12  TaxAmt          121317 non-null   float64
 13  Freight         121317 non-null   float64
 14  TotalDue        121317 non-null   float64
 15  SalesOrderDetailID 121317 non-null   int64  
 16  OrderQty        121317 non-null   int64  
 17  ProductID       121317 non-null   int64  
 18  UnitP           121317 non-null   float64
 19  LineTotal        121317 non-null   float64
 20  SubCategoryName 121317 non-null   object 
 21  CategoryName    121317 non-null   object 
 22  ShipMethodName   121317 non-null   object 
 23  TerritoryName   121317 non-null   object 
 24  TerritoryGroup   121317 non-null   object 
 25  ProductNameX    121317 non-null   object 
 26  Unnamed: 26      121317 non-null   object 
 27  ProductName     121317 non-null   object 
```



3 - ASK QUESTIONS



★ WE USED SQL TO ASK QUESTIONS AND FIND ITS ANSWERS THAT HELP IN ANALYZING DATA

---1) Top 10 Sold Products:

```
SELECT TOP 10
    P.[Name] AS ProductName,
    SUM(OD.LineTotal) AS TotalRevenue
FROM
    [Sales].[SalesOrderDetail] AS OD
JOIN
    [Production].[Product] AS P ON OD.[ProductID] = P.[ProductID]
GROUP BY
    P.[Name]
ORDER BY
    TotalRevenue DESC;
```

---2) Top 10 Customers making orders?

```
SELECT TOP 10
    C.[FirstName] + ' ' + C.[LastName] AS CustomerName,
    COUNT(OH.[SalesOrderID]) AS OrderCount
FROM
    [Sales].[SalesOrderHeader] AS OH
JOIN
    [Person].[Person] AS C ON OH.[CustomerID] = C.[BusinessEntityID]
GROUP BY
    C.[FirstName], C.[LastName]
ORDER BY
    OrderCount DESC;
```

---16) Total orders by each customer:

```
SELECT
    OH.[CustomerID],
    COUNT(OH.SalesOrderID) AS NumberOfOrders
FROM
    [Sales].[SalesOrderHeader] AS OH
GROUP BY
    OH.[CustomerID]
ORDER BY
    NumberOfOrders DESC;
```

---17) What is the total number of customers in each territory?

```
SELECT
    Ter.[Name] AS TerritoryName,
    COUNT(C.CustomerID) AS NumberOfCustomers
FROM
    [Sales].[Customer] C
JOIN
    [Sales].[SalesTerritory] AS Ter ON C.[TerritoryID] = Ter.[TerritoryID]
GROUP BY
    Ter.[Name]
ORDER BY
    NumberOfCustomers DESC;
```

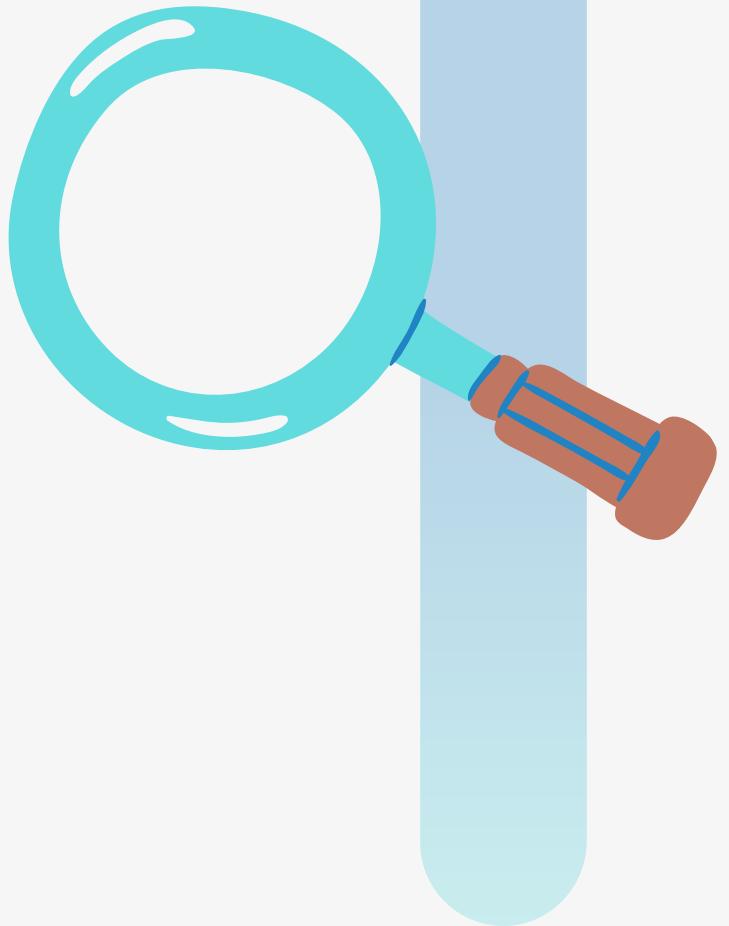
★ WE USED SQL TO ASK QUESTIONS AND FIND ITS ANSWERS THAT HELP IN ANALYZING DATA

---7) What is the total number of orders sold by each year?

```
SELECT  
    YEAR(OH.OrderDate) AS OrderYear,  
    COUNT(OH.SalesOrderID) AS TotalOrders  
FROM  
    [Sales].[SalesOrderHeader] AS OH  
GROUP BY  
    YEAR(OH.OrderDate)  
ORDER BY  
    OrderYear;
```

---8) What is the total quantity sold from each subcategory?

```
SELECT  
    SC.[Name] AS SubcategoryName,  
    SUM(OD.OrderQty) AS TotalQuantitySold  
FROM  
    [Sales].[SalesOrderDetail] AS OD  
JOIN  
    [Production].[Product] AS P ON OD.[ProductID] = P.[ProductID]  
JOIN  
    [Production].[ProductSubcategory] AS SC ON P.[ProductSubcategoryID] = SC.[ProductSubcategoryID]  
GROUP BY  
    SC.[Name]  
ORDER BY  
    SC.[Name]
```



★ QUESTIONS BY PYTHON

```
[153]: # What is the total sales for each product?  
df.groupby('ProductName').sum()['TotalDue']
```

```
[153]: ProductName  
AWC Logo Cap           4.978507e+07  
All-Purpose Bike Stand 1.658042e+05  
Bike Wash - Dissolver  1.841311e+07  
Cable Lock              1.503561e+07  
Chain                   1.024501e+07  
...  
Women's Mountain Shorts M 1.098193e+07  
Women's Mountain Shorts S 1.154731e+07  
Women's Tights L          1.918154e+07  
Women's Tights M          7.281956e+06  
Women's Tights S          1.915492e+07  
Name: TotalDue, Length: 266, dtype: float64
```

```
[155]: # What is the average cost by shipping?  
df.groupby('ShipMethodName')['Freight'].mean()
```

```
[155]: ShipMethodName  
CARGO TRANSPORT 5      1267.416031  
XRQ - TRUCK GROUND    24.515398  
Name: Freight, dtype: float64
```

```
[161]: #What is the total sales for each territory?  
df.groupby('TerritoryName').sum()['TotalDue']
```

```
[161]: TerritoryName  
Australia             7.057392e+07  
Canada                5.269695e+08  
Central               2.630991e+08  
France                1.981583e+08
```

```
[171]: #what is the total number of products in each category?  
df.groupby('CategoryName').count()['ProductName']
```

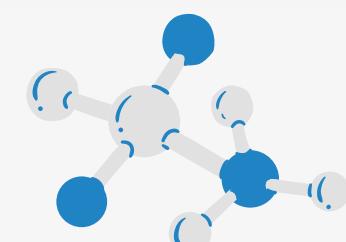
```
[171]: CategoryName  
Accessories            41194  
Bikes                  40031  
Clothing                21394  
Components              18698  
Name: ProductName, dtype: int64
```

```
[175]: #what is the average purchase for each category?  
df.groupby('CategoryName')['TotalDue'].mean()
```

```
[175]: CategoryName  
Accessories            6410.801154  
Bikes                  29723.099807  
Clothing                25356.121456  
Components              49768.387556  
Name: TotalDue, dtype: float64
```

```
[178]: #what is the total number of products shipped by each method?  
df.groupby('ShipMethodName').count()['ProductID']
```

```
[178]: ShipMethodName  
CARGO TRANSPORT 5      60919  
XRQ - TRUCK GROUND    60398  
Name: ProductID, dtype: int64
```



4 - DATA TRANSFORMATION & CLEANING



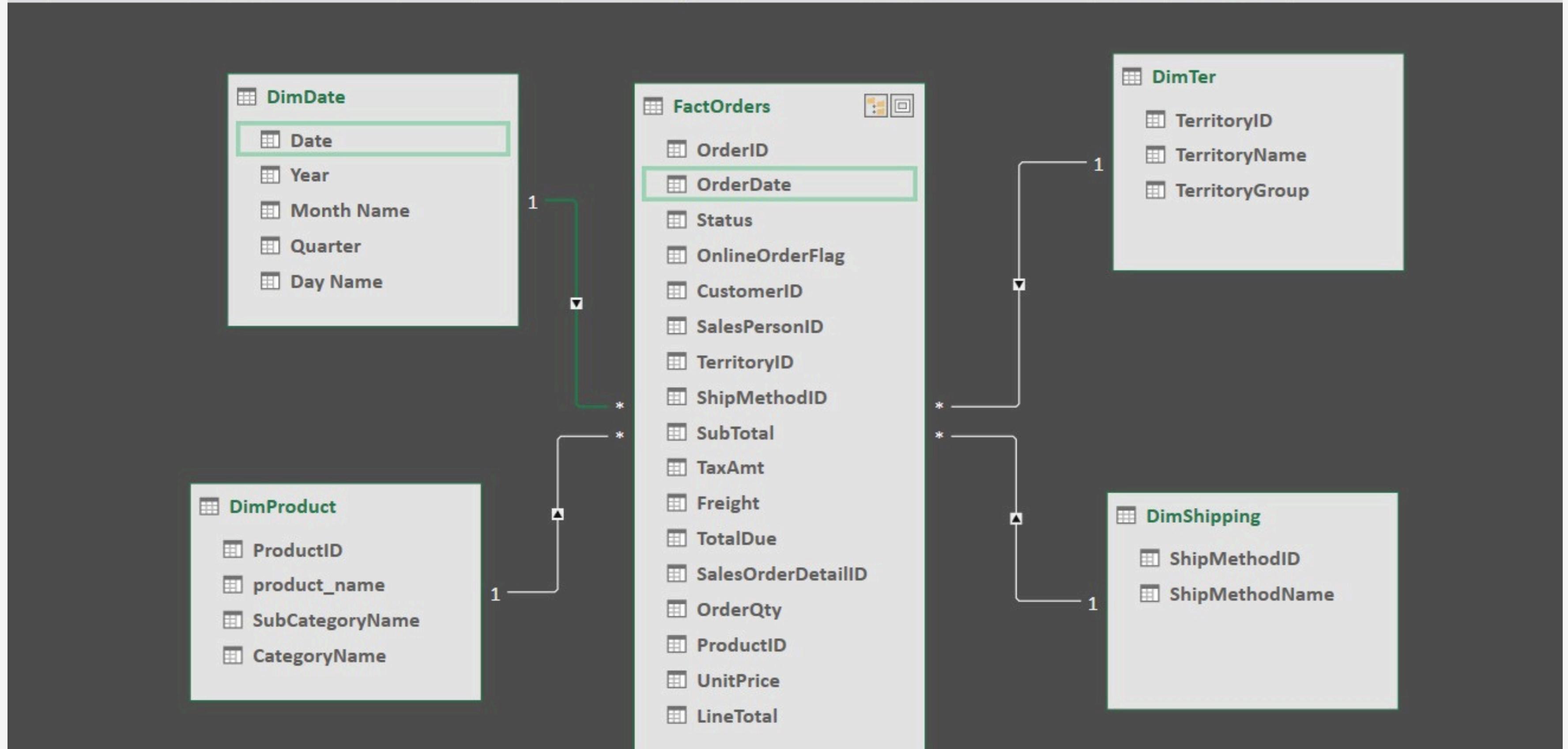
FOR DATA CLEANING WE USED EXCEL POWER QUERY BY ENTERING SELECT QUERY

CLEANING STEPS DONE

1. SELECT THE REQUIRED COLUMNS
2. REMOVED OTHER COLUMNS THAT ISN'T USEFUL
3. REMOVED DUPLICATES
4. CHANGED DATA TYPE
5. CREATED DIMS AND FACTS
6. STAR SCHEMA



STAR SCHEMA





CLEANING BY PYTHON

4] Cleaning Data

```
# check if there are missing values  
df.isna().sum()
```

```
SalesOrderID          0  
OrderDate            0  
DueDate              0  
ShipDate              0  
Status                0  
OnlineOrderFlag      0  
CustomerID           0  
SalesPersonID        60398  
TerritoryID          0  
ShipMethodID         0  
CreditCardID         2573  
SubTotal              0  
TaxAmt                0  
Freight               0  
TotalDue              0  
SalesOrderDetailID   0  
OrderQty              0  
ProductID             0  
UnitPrice             0  
LineTotal              0  
SubCategoryName       0  
CategoryName          0  
ShipMethodName        0  
TerritoryName         0  
TerritoryGroup        0  
ProductNameX          0  
Unnamed: 26            43476  
ProductName           0  
dtype: int64
```

```
[9]: df.drop('Unnamed: 26', axis = 1, inplace = True)  
  
[11]: df.drop('CreditCardID', axis = 1, inplace = True)  
  
[13]: df.columns  
  
[13]: Index(['SalesOrderID', 'OrderDate', 'DueDate', 'ShipDate', 'Status',  
           'OnlineOrderFlag', 'CustomerID', 'SalesPersonID', 'TerritoryID',  
           'ShipMethodID', 'SubTotal', 'TaxAmt', 'Freight', 'TotalDue',  
           'SalesOrderDetailID', 'OrderQty', 'ProductID', 'UnitPrice', 'LineTotal',  
           'SubCategoryName', 'CategoryName', 'ShipMethodName', 'TerritoryName',  
           'TerritoryGroup', 'ProductName'],  
           dtype='object')  
  
[15]: df.isna().sum()  
  
[15]: SalesOrderID          0  
      OrderDate            0  
      DueDate              0  
      ShipDate             0  
      Status                0  
      OnlineOrderFlag      0  
      CustomerID           0  
      SalesPersonID        60398  
      TerritoryID          0  
      ShipMethodID         0  
      SubTotal              0  
      TaxAmt                0  
      Freight               0  
      TotalDue              0  
      SalesOrderDetailID   0  
      OrderQty              0  
      ProductID             0  
      UnitPrice             0  
      LineTotal              0  
      SubCategoryName       0
```

★ CLEANING BY PYTHON

```
[15]: df.isna().sum()
```

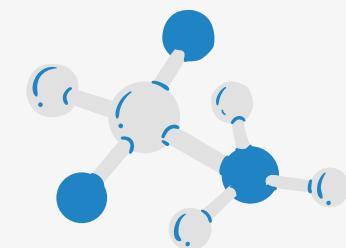
```
[15]: SalesOrderID          0  
OrderDate            0  
DueDate              0  
ShipDate              0  
Status                0  
OnlineOrderFlag       0  
CustomerID           0  
SalesPersonID         60398  
TerritoryID           0  
ShipMethodID          0  
SubTotal              0  
TaxAmt                0  
Freight               0  
TotalDue              0  
SalesOrderDetailID    0  
OrderQty              0  
ProductID             0  
UnitPrice              0  
LineTotal              0  
SubCategoryName        0  
CategoryName           0  
ShipMethodName         0  
TerritoryName          0  
TerritoryGroup         0  
ProductName            0  
dtype: int64
```

```
[151]: # Check if there are Duplicates  
df.duplicated().sum()
```

```
[151]: 0
```

NO NULLS OR DUPLICATES DATA IS CLEAN!

(WE DIDN'T REMOVE OR REPLACE THE NULLS IN SALESPERSONID COLUMN AS 'NULL'
HERE MEANS THAT THERE WEREN'T A SELLING TRANSACTION)



5 - DATA VISUALIZATION





EXCEL PIVOT TABLES



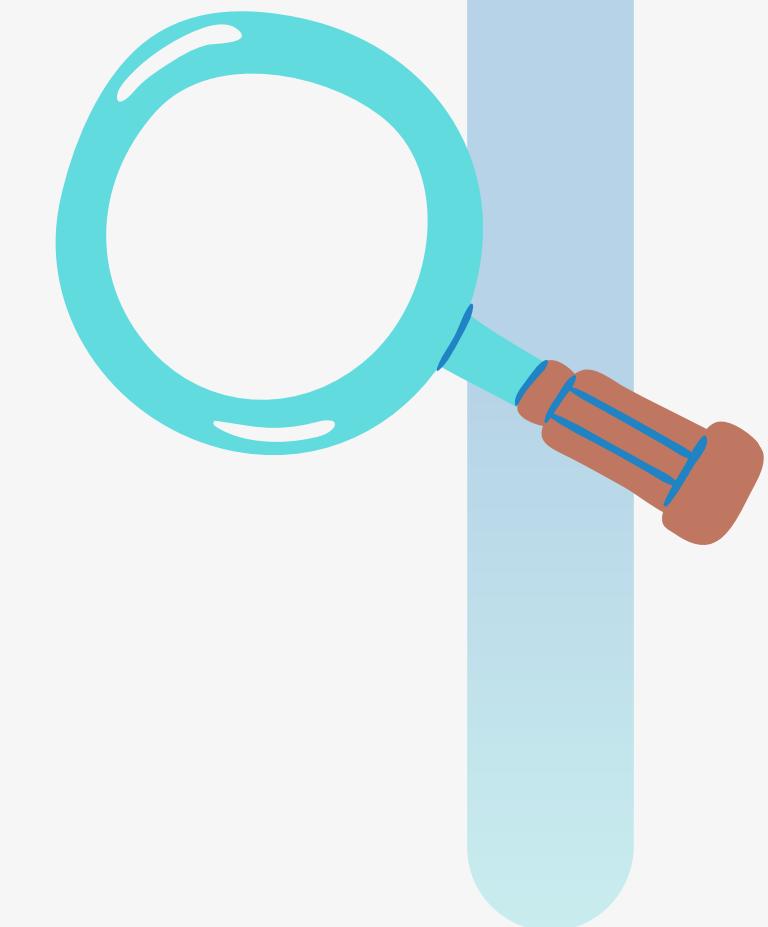
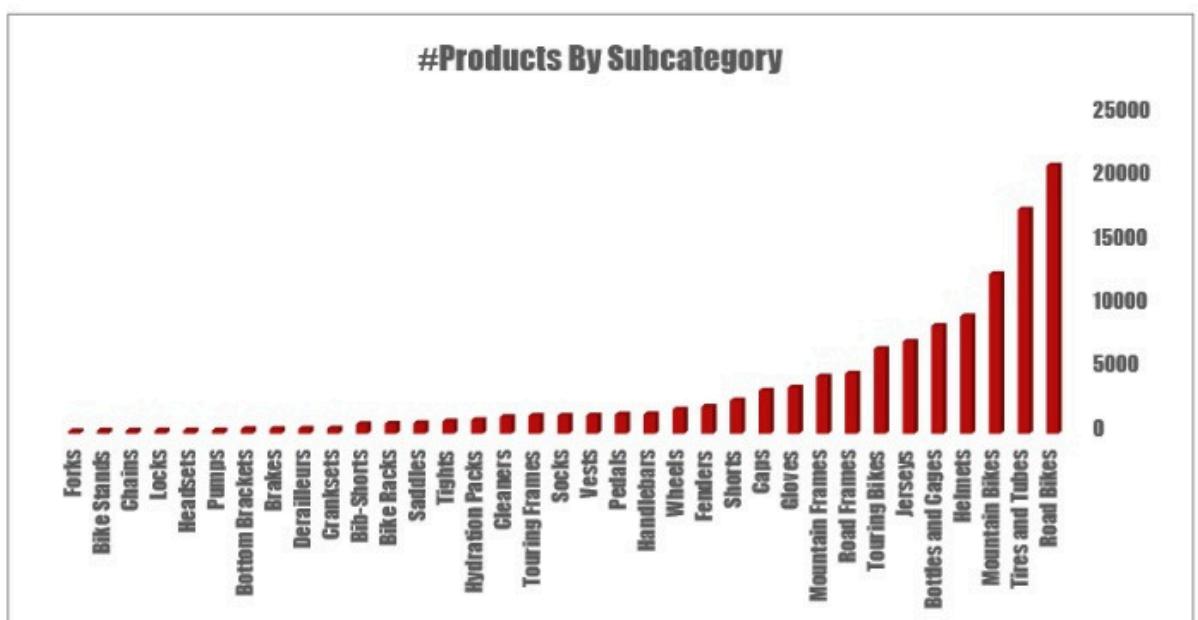
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
SubCategories	Count of ProductID	Products	Sum of OrderQty	Sum of UnitPrice	Territory Name	Sum of TotalDue	ShipMethod	Count of SalesOrderID		OrderDate (Year)						
Road Bikes	20944	AWC Logo Cap	8311	\$25,933.96	Southwest	\$696,896,625.81	CARGO TRANSPOR	60919		2011						
Tires and Tubes	17495	Water Bottle - 30 oz.	6815	\$22,498.01	Canada	\$526,969,463.24	XRQ - TRUCK GROL	60398		2012						
Mountain Bikes	12457	Sport-100 Helmet, Blue	6743	\$93,827.86	Northwest	\$411,207,275.13	Grand Total	121317	Total order by ship method	2013						
Helmets	9180	Long-Sleeve Logo Jersey, L	6592	\$57,063.17	Central	\$263,099,145.85				2014						
Bottles and Cages	8425	Sport-100 Helmet, Black	6532	\$91,561.73	Northeast	\$253,771,294.44	ShipMethod	Sum of Freight								
Jerseys	7191	Sport-100 Helmet, Red	6266	\$95,223.18	Southeast	\$226,427,460.25	CARGO TRANSPOR	\$77,209,717.21								
Touring Bikes	6630	Classic Vest, S	4247	\$29,917.39	France	\$198,158,287.87	XRQ - TRUCK GROL	\$1,480,680.99								
Road Frames	4713	Patch Kit/8 Patches	3865	\$7,531.31	United Kingdom	\$187,208,421.64	Grand Total	\$78,690,398.20	total freight for each ship method							
Mountain Frames	4477	Short-Sleeve Classic Jersey, XL	3864	\$37,925.82	Germany	\$92,658,225.65										
Gloves	3605	Long-Sleeve Logo Jersey, M	3636	\$44,799.43	Australia	\$70,573,924.16	Years	Count of SalesOrderID								
Caps	3382	Grand Total	56871	\$506,281.85	Grand Total	\$2,926,970,124.04	2013	56573								
Shorts	2610	Top 10 Products by order quantity & unit price					2014	37339								
Fenders	2121	total sales by territory					2012	21689								
Wheels	1904	total sales by each category					2011	5716								
Handlebars	1531	Categories	Sum of TotalDue		Territory Name	Count of SalesOrderID	Grand Total	121317	total orders by each year							
Pedals	1517	Bikes	\$1,189,845,408.36		Southwest	25644										
Vests	1438	Components	\$930,569,310.51		Canada	19064	Years	Sum of TotalDue								
Socks	1429	Clothing	\$542,468,862.44		Northwest	16865	2013	\$1,342,968,284.46								
Touring Frames	1428	Accessories	\$264,086,542.73		Australia	15058	2012	\$972,777,341.23								
Cleaners	1327	Grand Total	\$2,926,970,124.04		United Kingdom	10426	2014	\$440,440,763.40								
Hydration Packs	1074	total sales by each category					2011	\$170,783,734.96								
Tights	977	total orders by territory					Grand Total	\$2,926,970,124.04	total sales by each year							
Saddles	840	total number of product by each category														
Bike Racks	796	Categories	Count of ProductID		Territory Name	Sum of TotalDue										
Bib-Shorts	762	Accessories	41194		Southwest	\$696,896,625.81										
Cranksets	423	Bikes	40031		Canada	\$526,969,463.24										
Derailleurs	402	Clothing	21394		Northwest	\$411,207,275.13										
Brakes	379	Components	18698		Central	\$263,099,145.85										
Bottom Brackets	368	Grand Total	121317		Northeast	\$253,771,294.44										
Pumps	267	number of products by each sub category														
Headsets	266	Total orders by territory														
Locks	260	Territory Name														
Chains	250		Sum of TotalDue		Southwest	\$696,896,625.81										
Bike Stands	249				Canada	\$526,969,463.24										
Forks	200				Northwest	\$411,207,275.13										
Grand Total	121317				Central	\$263,099,145.85										
					Northeast	\$253,771,294.44										
					Southeast	\$226,427,460.25										
					France	\$198,158,287.87										
					United Kingdom	\$187,208,421.64										
					Germany	\$92,658,225.65										



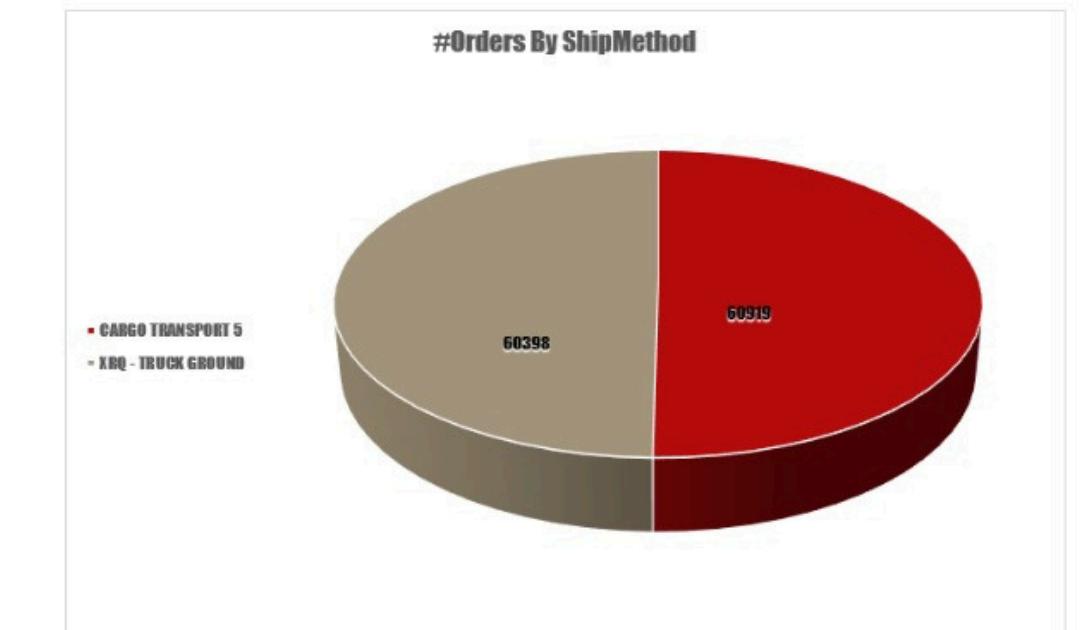
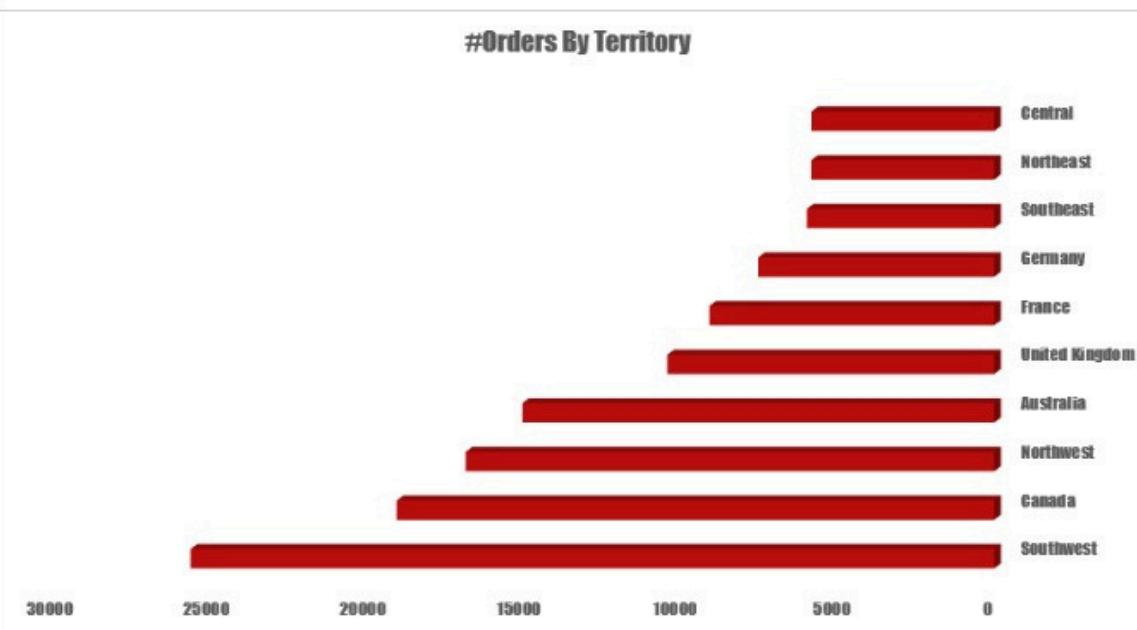
EXCEL PIVOT CHARTS



Product Analysis

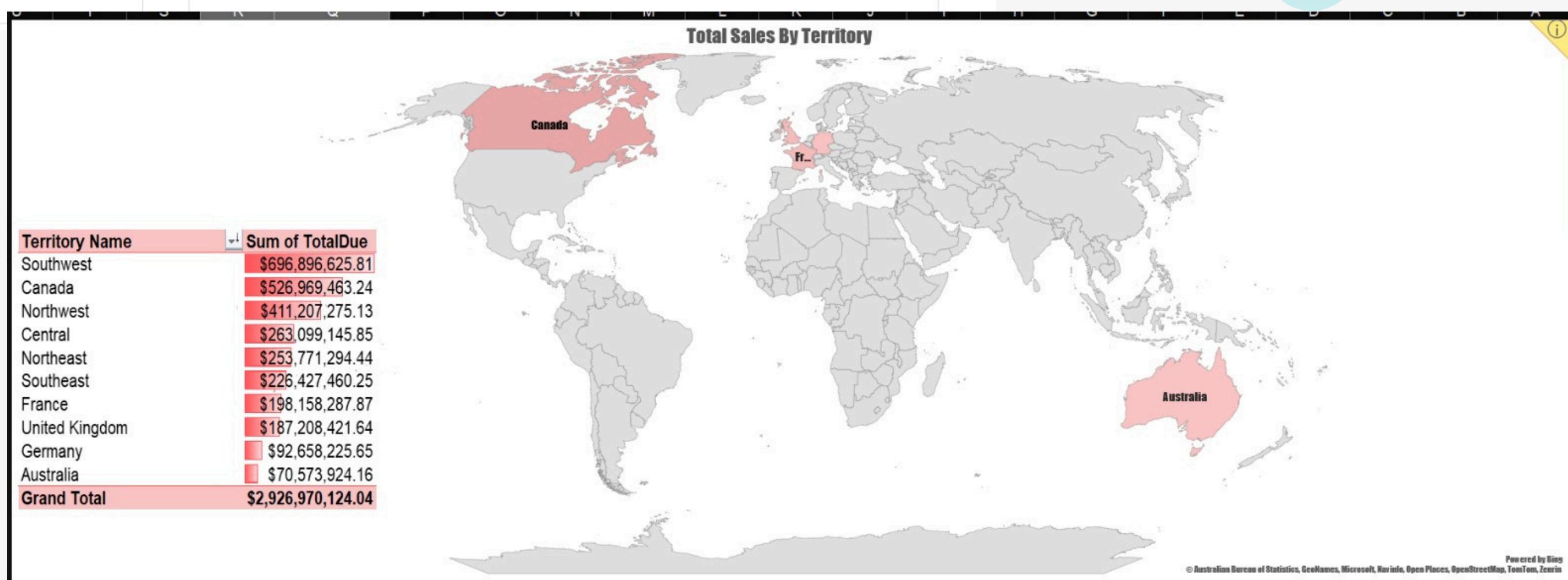
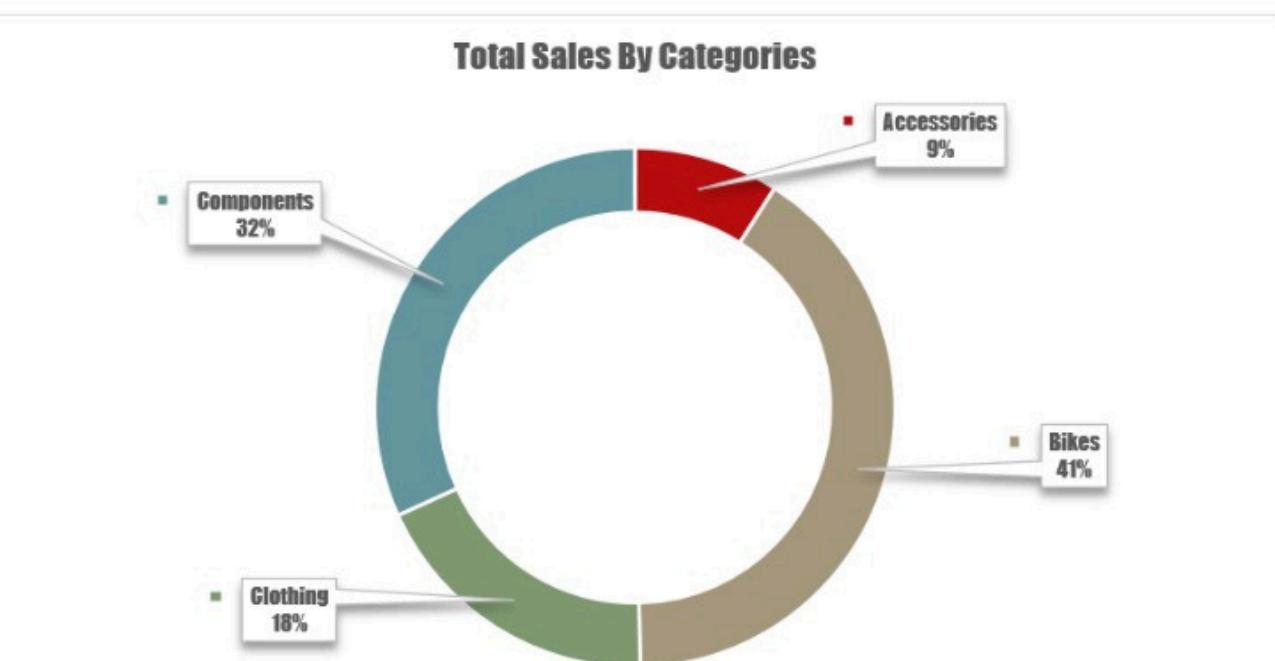


Orders Analysis

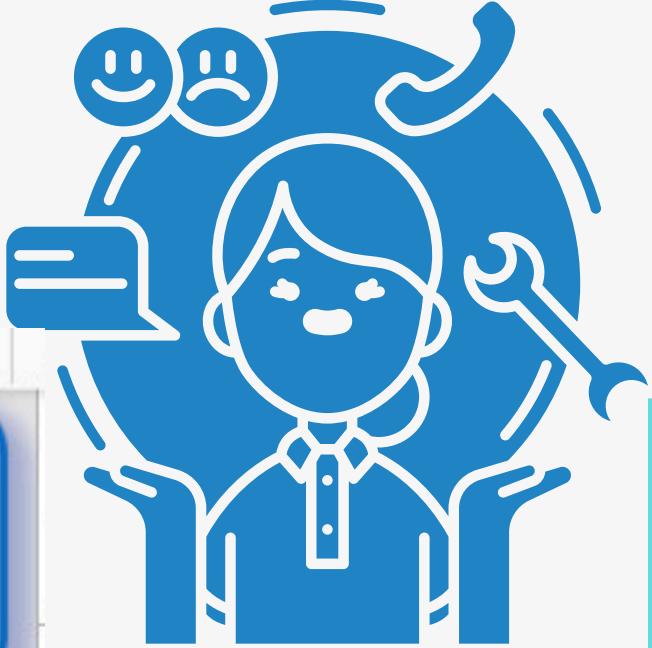


EXCEL PIVOT CHARTS

Sales Analysis



POWER BI VISUALS



Clipboard Data Queries Insert Calculations

ADVENTURE WORKS

Home
Categories
Time
Territory

2.93bn
Total Sales

251.81M
Total taxes

78.69M
Total freight

266
#Products

4
#Categories

35
#SubCategories

#Orders by ShipMethodName

ShipMethodName
● XRQ - TRUCK ...
● CARGO TRAN...

12.1%
87.9%

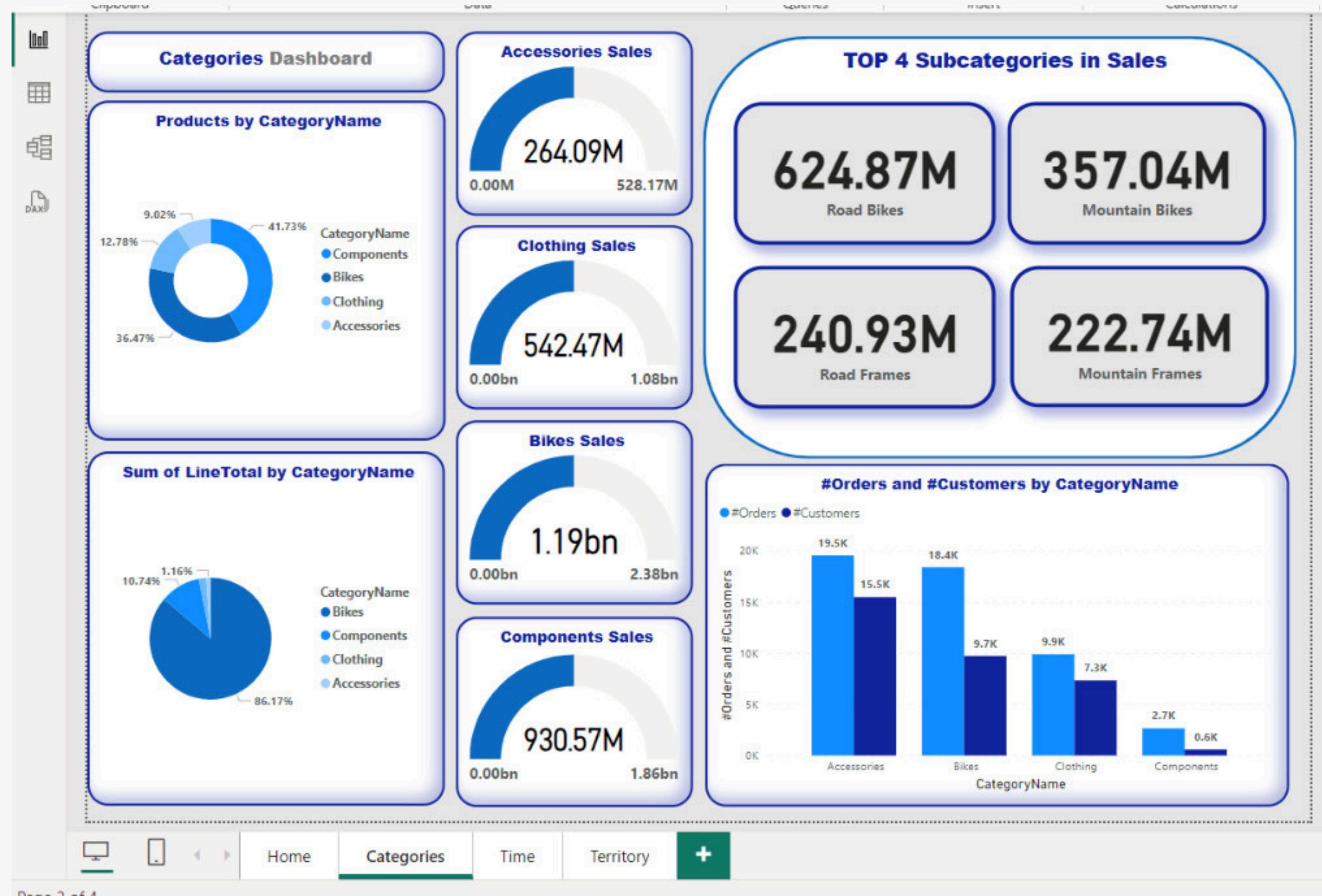
Total freight by ShipMethodName

ShipMethodNa...
● CARGO TRAN...
● XRQ - TRUCK ...

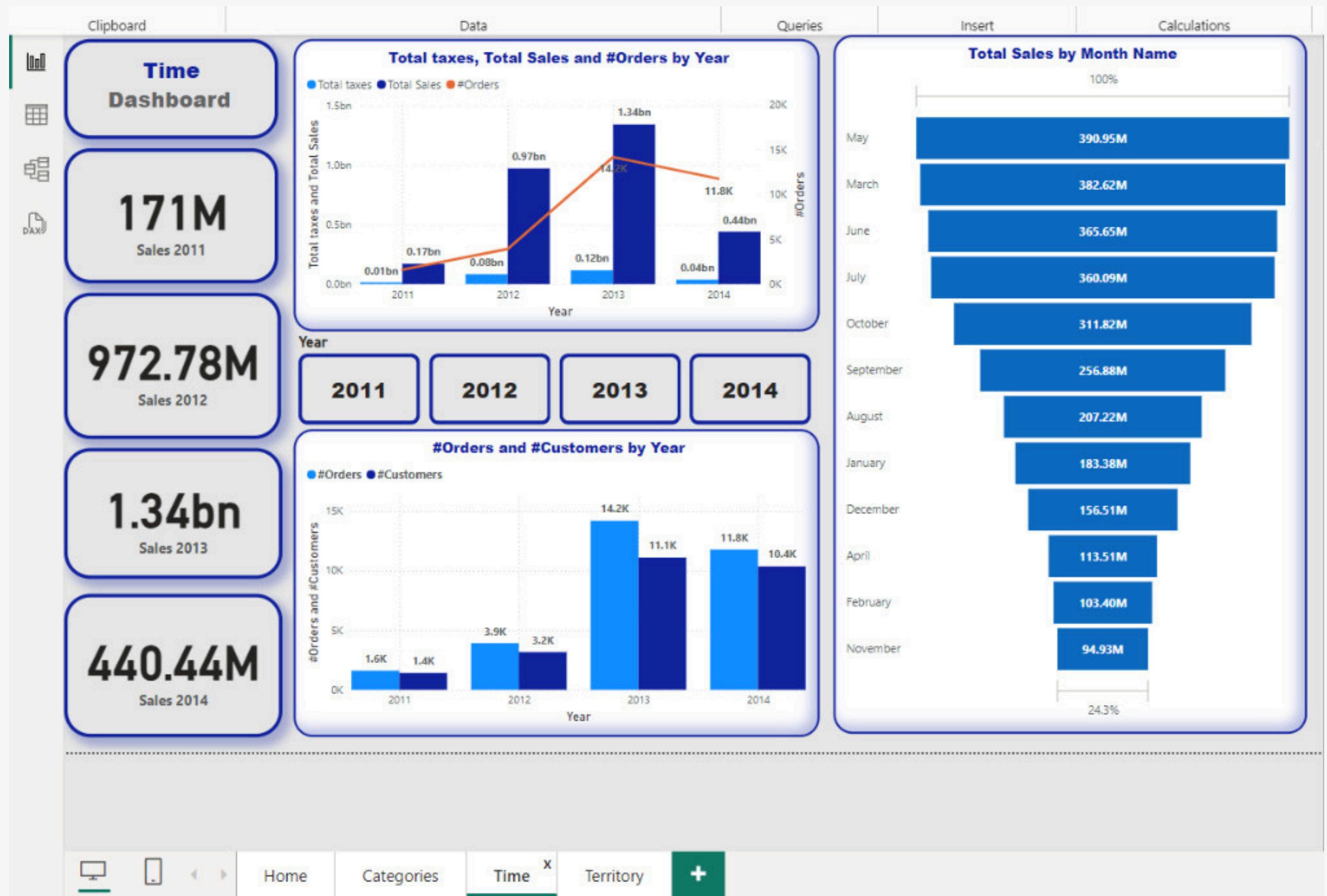
1.88%
98.12%

A screenshot of a Power BI dashboard. At the top, there's a navigation bar with tabs: Clipboard, Data, Queries, Insert, and Calculations. Below the navigation bar, there's a sidebar on the left with a logo for "ADVENTURE WORKS" featuring a mountain and sun, and four buttons: Home, Categories, Time, and Territory. The main area contains several data summary cards and visualizations. On the left, there are four cards with large numbers: "2.93bn" (Total Sales), "251.81M" (Total taxes), "78.69M" (Total freight), and "266" (#Products). Below these are two more cards: "4" (#Categories) and "35" (#SubCategories). Further down, there are two donut charts. The first chart, titled "#Orders by ShipMethodName", shows a distribution where 87.9% is represented by a large blue slice and 12.1% by a smaller dark blue slice. The second chart, titled "Total freight by ShipMethodName", shows a distribution where 98.12% is represented by a large dark blue slice and 1.88% by a very small light blue slice. Both charts have legends indicating categories like "XRQ - TRUCK ..." and "CARGO TRAN...".

POWER BI VISUALS



POWER BI VISUALS



POWER BI VISUALS

Clipboard Data Queries Insert Calculations Sensors

Territory Dashboard

Territory	Sales
Southeast	226.43M
France	198.16M
UK	187.21M
Germany	92.66M
Australia	70.57M
Northeast	253.77M
Central	263.10M
Northwest	411.21M
Canada	526.97M
Southwest	696.90M

Total freight by TerritoryName

This horizontal bar chart displays the total freight volume for various territories. The y-axis lists the territories, and the x-axis shows the freight amount in millions. The bars are blue, and the chart includes numerical labels for each bar.

Territory	Freight (M)
Southwest	18.76M
Canada	14.17M
Northwest	11.08M
Central	7.05M
Northeast	6.81M
Southeast	6.08M
France	5.34M
United Kingdom	5.02M
Germany	2.51M
Australia	1.85M

Total taxes by TerritoryName

This world map illustrates the total taxes paid by different territories. Major cities are marked with blue dots, and the map is labeled with regions like North America, Europe, Asia, Africa, South America, and Australia. The Microsoft logo is visible in the bottom left corner.

#Orders and #Customers by TerritoryName

This grouped bar chart compares the number of orders and customers across territories. The y-axis represents the count, ranging from 0K to 8K. Blue bars represent the number of orders, and dark blue bars represent the number of customers. The chart shows a general downward trend from Australia to Northeast.

Territory	#Orders	#Customers
Australia	~7.5K	~3.8K
Southwest	~6.8K	~4.5K
Northwest	~4.8K	~3.5K
Canada	~4.2K	~1.8K
United Kingdom	~3.2K	~2.0K
France	~2.8K	~1.8K
Germany	~2.5K	~1.8K
Southeast	~1.0K	~500
Central	~800	~500
Northeast	~800	~500

Home Categories Time Territory +

CONCLUSION

- OUR ANALYSIS OF THE ADVENTURE WORKS 2019 DATA HAS PROVIDED VALUABLE INSIGHTS INTO CUSTOMER BEHAVIOR, SALES PERFORMANCE, PRODUCTS AND OPERATIONAL EFFICIENCY. BY LEVERAGING KEY METRICS AND VISUALIZATIONS, WE IDENTIFIED TRENDS AND OPPORTUNITIES FOR GROWTH.
 - KEY TAKEAWAYS INCLUDE THE IMPORTANCE OF TARGETED MARKETING STRATEGIES TO ENHANCE CUSTOMER ENGAGEMENT, OPTIMIZING INVENTORY MANAGEMENT TO REDUCE COSTS, AND FOCUSING ON HIGH-PERFORMING PRODUCT LINES TO DRIVE REVENUE.
 - MOVING FORWARD, WE RECOMMEND IMPLEMENTING DATA-DRIVEN STRATEGIES TO IMPROVE DECISION-MAKING AND FOSTER A CULTURE OF CONTINUOUS IMPROVEMENT.
 - BY DOING SO, ADVENTURE WORKS CAN NOT ONLY ENHANCE ITS COMPETITIVE EDGE BUT ALSO ENSURE SUSTAINED GROWTH IN AN EVER-EVOLVING MARKET.
-

**THANK YOU FOR YOUR
ATTENTION, AND WE
LOOK FORWARD TO
ANY QUESTIONS OR
DISCUSSIONS!**

