



## Northwind Project

- Data Analysis GHR\_DAT1\_G1e
- DEPI Graduation Project
- Under Supervision of Eng/Alaa Essam



## Team members



**Abdelrahman Hesham**



**Fathy Ashraf**



**Mahmoud Elsharkawy**



**Ibrahim Eldakroury**

# Agenda



Business  
Question

Get Data

Explore Data

Prepare Data

Analyze Data

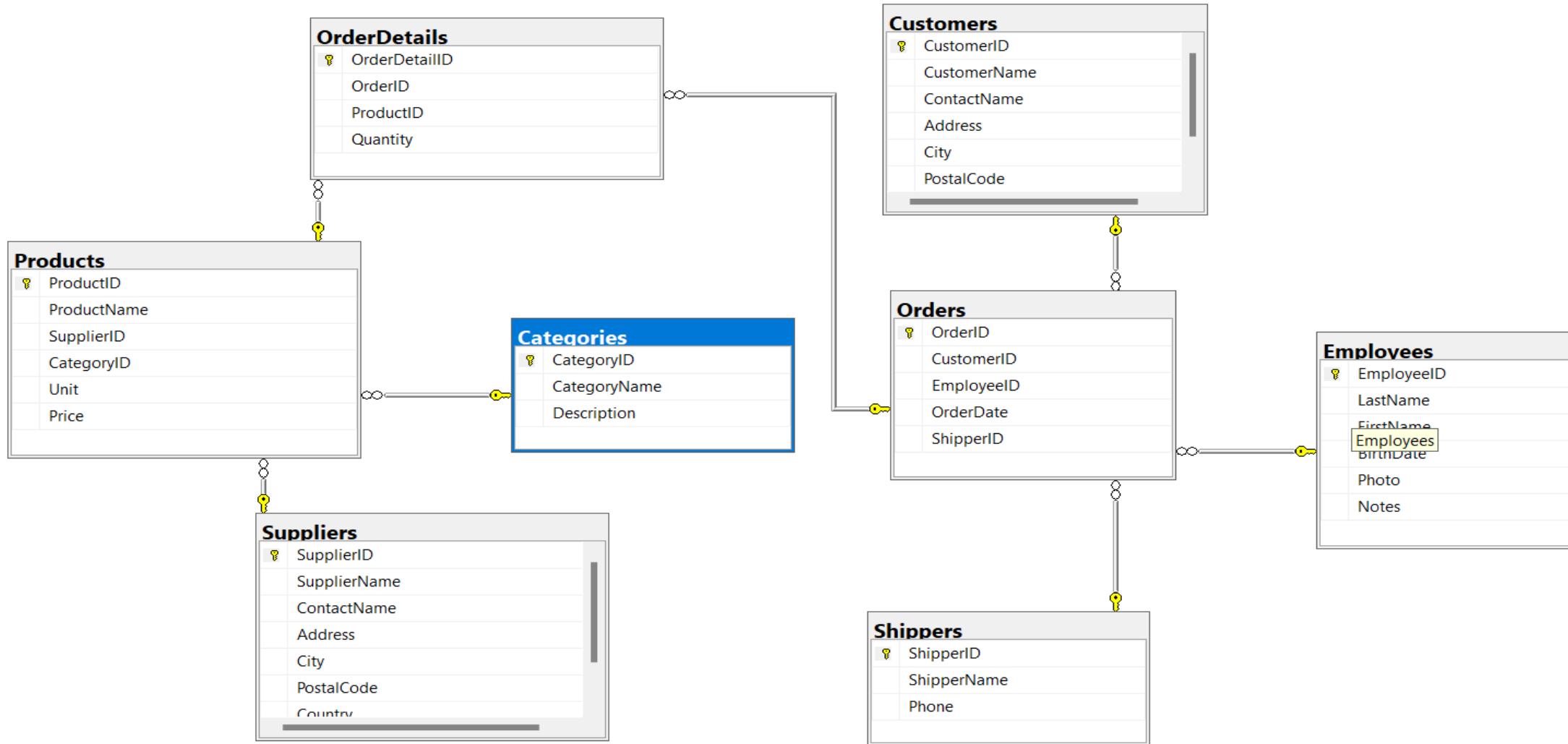
Present  
Findings



# Business Question

- Total Revenue : Show total revenue generated from all orders.(price \* quantity)
- Total Number of Orders : Display the total number of orders placed
- Number of Unique Customers : Count the number of unique customers who placed orders.
- Best-Selling 5 Products : Identify 5 Products with the highest sales volume.(price \* quantity)
- Total Revenue by Category : Identify The Total Revenue For Each Category
- Customer Distribution by Country : Show how customers are distributed across different countries
- Total sales(Price\*Quantity) by Country : Show The Sales Across Different Countries
- Total Suppliers : Show the total number of suppliers providing products
- Top 3 Employees by Sales : Show The Best 3 Employees In Sales
- How many customers have placed more than one order : Show the Loyal Customers
- Total Quantity Sold by Category : Show The Total Amount Sold For Each Category
- How many orders were shipped by each shipper? : Show The Count Of Orders For Each Shipper

# Data Modeling



# Agenda



Business  
Question

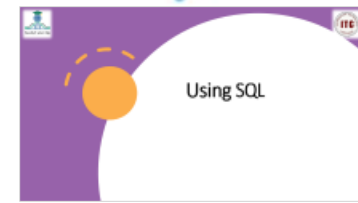
Get Data

Explore Data

Prepare Data

Analyze Data

Present  
Findings





# Using SQL



# Explore & Clean Data

use Northwind

```
select * from Categories
select * from Customers
select * from Employees
select * from Products
select * from OrderDetails
select * from Orders
select * from Shippers
select * from Suppliers
```

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and...
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	5021	Mexico
3	Antonio Moreno Taquería	Antonio Mor...	Mataderos 2312	México D.F.	5023	Mexico
4	Arround the Horn	Thomas Ma...	120 Mansour St.	London	W8A 1DP	UK

EmployeeID	LastName	FirstName	BirthDate	Photo	Notes
1	Diavolo	Nancy	1968-12-08 00:00:00.000	EmpID1.pic	Education includes a BA in psychology from Color...
2	Fuller	Andrew	1952-02-19 00:00:00.000	EmpID2.pic	Andrew received his BTS commercial and a Ph.D. ...
3	Levering	Janet	1963-08-30 00:00:00.000	EmpID3.pic	Janet has a BS degree in chemistry from Boston C...
4	Peacock	Margaret	1958-09-19 00:00:00.000	EmpID4.pic	Margaret holds a BA in English literature from Con...
5	Buchanan	Steven	1955-03-04 00:00:00.000	EmpID5.pic	Steven Buchanan graduated from St. Andrews Uni...

ProductID	ProductName	SupplierID	Unit	Price
1	Chais	1	10 boxes x 20 bags	18
2	Chang	1	24 - 12 oz bottles	19
3	Aniseed Syr...	1	12 - 550 ml bottles	10
4	Chai Aniseed...	2	48 - 6 oz cans	22

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40
6	10250	41	10
7	10250	51	35

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04 00:00:00.000	3
10249	81	6	1996-07-05 00:00:00.000	1
10250	34	4	1996-07-08 00:00:00.000	2
10251	84	3	1996-07-08 00:00:00.000	1
10252	76	4	1996-07-09 00:00:00.000	2

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shippi...	(503) 555-9931

SupplierID	SupplierName	ContactName	Address	City	PostalCode	Country	Phone
1	Exotic Liquid	Charlotte Cooper	49 Gilbert St.	London	EC1 4SD	UK	(171) 555-2222
2	New Orleans Cajun Delights	Shelley Burke	P.O. Box 78934	New Orleans	70117	USA	(100) 555-4822
3	Grandma Kelly's Homeste...	Regina Murphy	707 Oxford Rd.	Ann Arbor	48104	USA	(313) 555-5735

Query executed successfully.

----- checking for Duplicates values Rows ----

```
select count (*) from Categories
select distinct count (*) from Categories
```

```
select count(*) from Customers
select distinct count(*) from Customers
```

```
select count (*) from Employees
select distinct count (*) from Employees
```

```
select count (*) from OrderDetails
select distinct count (*) from OrderDetails
```

```
select count(*) from Orders
select distinct count(*) from Orders
```

```
select count (*) from Products
select distinct count (*) from Products
```

```
select count (*) from shippers
select distinct count (*) from shippers
```

```
select count (*) from Suppliers
select distinct count (*) from Suppliers
```

----- checking for missing values ----

```
SELECT *
FROM Customers
where CustomerID is null or CustomerName is null
or ContactName is null
or Address is null or
city is null or
PostalCode is null
or Country is null ;
```

```
select * from OrderDetails
where OrderDetailID is null
or OrderID is null
or ProductID is null
or Quantity is null ;
```

```
select * from Orders
where OrderID is null
or CustomerID is null
or EmployeeID is null
or OrderDate is null
or ShipperID is null
```

```
select * from Products
where ProductID is null or
ProductID is null or
SupplierID is null or
CategoryID is null or
Unit is null or
Price is null
```



# Analyze Data

```

/* 1. Total Revenue
Insight: Show total revenue generated from all orders.(price * quantity)*/
select sum(price * Quantity) as Total_Revenue from Products p
join OrderDetails od on p.ProductID = od.ProductID

/*2. Total Number of Orders
Insight: Display the total number of orders placed*/
SELECT COUNT(*) AS Total_Orders
FROM Orders;

/*3. Number of Unique Customers
Insight: Count the number of unique customers who placed orders.*/
SELECT COUNT( distinct CustomerID) AS Unique_Customers
FROM Orders;

/*4. Best-Selling 5 Products
Insight: Identify 5 Products with the highest sales volume.(price * quantity)*/
select Top 5 ProductName ,sum(price * Quantity) as Total_Revenue from Products p
join OrderDetails od on p.ProductID = od.ProductID
group by ProductName order by sum(Price* Quantity) desc

/*5. Total Revenue by Category
Insight: Identify The Total Revenue For Each Category*/
select CategoryName ,sum(price * Quantity) as Total_Revenue
from Products p
join OrderDetails od on p.ProductID = od.ProductID
join Categories c on c.CategoryID = p.CategoryID
group by CategoryName
order by Total_Revenue

```

Results		Messages	
Total_Revenue			
1	387066		
Total_Orders			
1	196		
Unique_Customers			
1	74		
ProductName		Total_Revenue	
1	Côte de Blaye	63096	
2	Thüringer Rostbratwurst	20832	
3	Raclette Courdavault	19030	
4	Tarte au sucre	15925	
5	Camembert Pierrot	14620	
CategoryName		Total_Revenue	
1	Grains/Cereals	22370	
2	Produce	23439	
3	Seafood	29639	
4	Condiments	35024	
5	Meat/Poultry	51636	
6	Confections	54729	
7	Dairy Products	70530	
8	Beverages	99699	

Query executed successfully.

# Analyze Data

/\*6. Customer Distribution by Country

Insight: Show how customers are distributed across different countries\*/

```
-SELECT Country, COUNT(*) AS Customer_Count
FROM Customers
GROUP BY Country
order by Customer_Count desc;
```

/\*7. Total sales(Price\*Quantity) by Country

Insight: Show The Sales Across Different Countries\*/

```
-select Country ,sum(price * Quantity) as Total_Revenue from Products p
join OrderDetails od on p.ProductID = od.ProductID
join Orders o on o.OrderID = od.OrderID
join Customers c on c.CustomerID = o.CustomerID
group by Country
order by Total_Revenue desc
```

Results		Messages
	Country	Customer_Count
1	USA	13
2	France	11
3	Germany	11
4	Brazil	9
5	UK	7
6	Spain	5
7	Mexico	5
8	Venezu...	4
9	Italy	3
10	Canada	3
11	Argentina	3
12	Austria	2
13	Belgium	2
14	Denmark	2
15	Finland	2
16	Portugal	2
17	Sweden	2
18	Switzerl...	2

	Country	Total_Revenue
1	USA	69722
2	Austria	51687
3	Germ...	47316
4	Brazil	40272
5	Canada	31402
6	France	29559
7	Denm...	17954
8	UK	16764
9	Ireland	15405
10	Venez...	13559
11	Swed...	9736
12	Switz...	8109
13	Belgi...	8075
14	Finland	6478
15	Mexico	5882
16	Italy	4339
17	Switzerl...	4316

Query executed successfully.

# Analyze Data

```
/*8. Total Suppliers
Insight: Show the total number of suppliers providing products*/
```

```
SELECT COUNT(DISTINCT SupplierID) AS Total_Suppliers
FROM Products;
```

```
/*9. Top 5 Employees by Sales
Insight : Show The Best 3 Employees In Sales*/
```

```
SELECT Top 5
    FirstName + ' ' + LastName AS EmployeeName,
    SUM(Quantity * Price) AS TotalSales
FROM Employees e
JOIN Orders o ON e.EmployeeID = o.EmployeeID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY FirstName, LastName
ORDER BY TotalSales DESC
```

```
/* 10.How many customers have placed more than one order
Insight: Show the Loyal Customers*/
```

```
select customername ,count(OrderID) as number_of_orders
from Orders o join Customers c
on o.CustomerID =c.CustomerID
group by CustomerName
having COUNT(OrderID)>1
order by number_of_orders desc
```

Results

Messages

	Total_Suppliers
1	29

	EmployeeName	TotalSales
1	Margaret Peacock	105926
2	Nancy Davolio	57765
3	Janet Leverling	42823
4	Robert King	39843
5	Laura Callahan	39341

	customername	number_of_orders
1	Ernst Handel	10
2	QUICK-Stop	7
3	Rattlesnake C...	7
4	Wartian Herkku	7
5	Split Rail Bee...	6
6	Hungry Owl Al...	6
7	La maison d"...	5
8	LILA-Superm...	5
9	Mère Paillarde	5
10	Old World Del...	4
11	Que Delícia	4
12	Folk och få HB	4
13	Frankenversa...	4
14	Blondel père ...	4
15	Bottom-Dollar...	4
16	Die Wandern...	4
17	Save-a-lot Ma...	4
18	Tortuga Rest...	4
19	Seven Seas I...	3
20	Romero y to...	3
21	Bon app"	3
22	Berglunds sn...	3
23	Island Trading	3
24	Familia Arqui...	3
25	Piccolo und ...	3
26	Princesa Isab...	3

Query executed successfully.



# Analyze Data

```

/*11.Total Quantity Sold by Category
Insight: Show The Total Amount Sold For Each Category*/

select categoryname, sum(quantity) as total_quanitiy
from Products p join OrderDetails od on p.ProductID = od.ProductID
join Categories c on p.CategoryID = c.CategoryID
group by CategoryName
order by total_quanitiy

/*12. How many orders were shipped by each shipper?
Insight: Show The Count Of Orders For Each Shipper*/

select shippername , count(orderid) as total_orders
from Shippers s join Orders o on s.ShipperID= o.ShipperID
group by ShipperName
order by total_orders

```

Results Messages		
	CategoryName	total_quanitiy
1	Produce	715
2	Grains/Cereals	912
3	Meat/Poultry	1288
4	Condiments	1383
5	Seafood	1445
6	Confections	2110
7	Beverages	2289
8	Dairy Products	2601

	ShipperName	total_orders
1	Speedy Express	54
2	Federal Shipping	68
3	United Package	74

✓ Query executed successfully.

# Using Python

## First Way

- Using Sqllite3



## Second Way

- Working On Excel Sheet Using Python



# Analyze Data "First way"

```
[1]: import sqlite3

[2]: import pandas as pd

[3]: conn = sqlite3.connect('project.db')
    cursor = conn.cursor()

[4]: with open('project.sql', 'r', encoding='UTF-8') as sql_file:
    sql_script = sql_file.read()

[5]: sql_commands = sql_script.split(';')

[6]: for command in sql_commands:
    try:
        # ignore null space
        if command.strip():
            cursor.execute(command)
    except sqlite3.Error as e:
        print("error during process:", command)
        print("error:", e)

error during process:

----- checking for missing values -----
SELECT *
FROM Categories

SELECT *
FROM Customers
where CustomerID is null or CustomerName is null or ContactName is null or Address is null or city is null or PostalCode is null or Country is null
error: near "SELECT": syntax error
error during process:

SELECT *
FROM Employees

select * from OrderDetails
where OrderDetailID is null or OrderID is null or ProductID is null or Quantity is null

[7]: # save changes
    conn.commit()

# ensure that table insert
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
tables = cursor.fetchall()

print("tables done yeah !!:", tables)

tables done yeah !!: [('Categories'), ('Customers'), ('Employees'), ('Shippers'), ('Suppliers'), ('Products'), ('Orders'), ('OrderDetails')]
```

```
•[13]: # 1. Total Revenue
    cursor.execute('''
        select sum(price * Quantity) as Total_Revenue
        from Products p join OrderDetails od on p.ProductID = od.ProductID
    ''')
    total_revenue = cursor.fetchone()[0]

    # show me the results of it !
    print("Total Revenue: ", total_revenue)

Total Revenue: 386424.23

[22]: # 2. Total Number of Orders
    cursor.execute('''
        SELECT COUNT(*) AS TotalOrders
        FROM Orders;
    ''')
    total_orders = cursor.fetchone()[0]
    print("Total Number of Orders: ", total_orders)

Total Number of Orders: 196

[23]: # 3. Number of Unique Customers
    cursor.execute('''
        SELECT COUNT(DISTINCT CustomerID) AS UniqueCustomers
        FROM Orders;
    ''')
    unique_customers = cursor.fetchone()[0]
    print("Number of Unique Customers: ", unique_customers)

Number of Unique Customers: 74

[14]: # 4. Best-Selling 5 Products
    best_selling_products = pd.read_sql_query('''
        SELECT ProductName, SUM(Price * Quantity) AS Total_Revenue
        FROM Products p
        JOIN OrderDetails od ON p.ProductID = od.ProductID
        GROUP BY ProductName
        ORDER BY Total_Revenue DESC
        LIMIT 5;
    ''', conn)
    print("Best-Selling 5 Products:")
    print(best_selling_products)
```

```
Best-Selling 5 Products:
   ProductName  Total_Revenue
0   Côte de Blaye      62976.50
1 Thüringer Rostbratwurst    20796.72
2   Raclette Courdavault    19030.00
3   Tarte au sucre        16022.50
4   Camembert Pierrot     14620.00

•[15]: # 5. Total Revenue by Category
    revenue_by_category = pd.read_sql_query('''
        select CategoryName ,sum(price * Quantity) as Total_Revenue
        from Products p join OrderDetails od on p.ProductID = od.ProductID
        join Categories c on c.CategoryID = p.CategoryID group by CategoryName
        order by Total_Revenue
    ''', conn)
    print("Total Revenue by Category:")
    print(revenue_by_category)

Total Revenue by Category:
   CategoryName  Total_Revenue
0  Grains/Cereals      22327.75
1      Produce      23401.40
2      Seafood      29652.30
3  Condiments      35071.60
4  Meat/Poultry      51676.52
5  Confections      54909.16
6  Dairy Products      69921.00
7  Beverages      99464.50

[18]: # 6. Customer Distribution by Country
    customer_distribution = pd.read_sql_query('''
        SELECT Country, COUNT(*) AS Customer_Count
        FROM Customers
        GROUP BY Country
        order by Customer_Count ;
    ''', conn)
    print("Customer Distribution by Country:")
    print(customer_distribution)

Customer Distribution by Country:
   Country  Customer_Count
0   Ireland              1
1   Norway              1
2    Poland              1
3   Austria              2
4    Belgium              2
5   Denmark              2
```



# Analyze Data "First way"

```
[23]: # 7. Total sales (Price*Quantity) by Country
sales_by_country = pd.read_sql_query('''
SELECT c.Country, ROUND(SUM(p.Price * od.Quantity), 0) AS Total_Revenue
FROM Products p
JOIN OrderDetails od ON p.ProductID = od.ProductID
JOIN Orders o ON o.OrderID = od.OrderID
JOIN Customers c ON c.CustomerID = o.CustomerID
GROUP BY c.Country
ORDER BY c.Country ASC;

''', conn)
print("Total Sales by Country:")
print(sales_by_country)
```

	Country	Total_Revenue
0	Argentina	399.0
1	Austria	51672.0
2	Belgium	8051.0
3	Brazil	40215.0
4	Canada	31326.0
5	Denmark	17871.0
6	Finland	6439.0
7	France	29549.0
8	Germany	47242.0

```
[28]: # 8. Total Suppliers
cursor.execute('''
SELECT COUNT(DISTINCT SupplierID) AS Total_Suppliers
FROM Products;
''')
total_suppliers = cursor.fetchone()[0]
print("Total Suppliers: ", total_suppliers)
```

Total Suppliers: 29

```
[27]: # 9. Top 3 Employees by Sales
top_employees = pd.read_sql_query('''
SELECT (e.FirstName || ' ' || e.LastName) AS EmployeeName,
ROUND(SUM(od.Quantity * p.Price), 0) AS TotalSales
FROM Employees e
JOIN Orders o ON e.EmployeeID = o.EmployeeID
JOIN OrderDetails od ON o.OrderID = od.OrderID
JOIN Products p ON od.ProductID = p.ProductID
GROUP BY e.FirstName, e.LastName
ORDER BY TotalSales DESC
LIMIT 5;

''', conn)
print("Top 3 Employees by Sales:")
print(top_employees)
```

Top 3 Employees by Sales:

	EmployeeName	TotalSales
0	Margaret Peacock	105697.0
1	Nancy Davolio	57690.0
2	Janet Leverling	42838.0
3	Robert King	39772.0
4	Laura Callahan	39309.0

```
[34]: # 10. How many customers have placed more than one order
loyal_customers_query = '''
SELECT c.CustomerID, c.ContactName AS CustomerName, COUNT(o.OrderID) AS OrderCount
FROM Orders o
JOIN Customers c ON o.CustomerID = c.CustomerID
GROUP BY c.CustomerID
HAVING COUNT(o.OrderID) > 1
ORDER BY OrderCount DESC;

'''

# Execute the query
cursor.execute(loyal_customers_query)

# Fetch all results
loyal_customers_with_names = cursor.fetchall()

# Display the results
print("CustomerName", "OrderCount")
for row in loyal_customers_with_names:
    print(row[1], row[2]) # row[1] is CustomerName, row[2] is OrderCount
```

CustomerName	OrderCount
Roland Mendel	10
Pirkko Koskitalo	7
Paula Wilson	7
Horst Kloss	7
Art Braunschweiger	6
Patricia McKenna	6
Jean Fresnière	5
Carlos González	5
Annette Roulet	5
Rita Müller	4
Miguel Angel Paolino	4

```
[31]: # 11. Total Quantity Sold by Category
quantity_by_category = pd.read_sql_query('''
SELECT Categories.CategoryName, SUM(OrderDetails.Quantity) AS TotalQuantity
FROM OrderDetails
JOIN Products ON OrderDetails.ProductID = Products.ProductID
JOIN Categories ON Products.CategoryID = Categories.CategoryID
GROUP BY Categories.CategoryName;

''', conn)
print("Total Quantity Sold by Category:")
print(quantity_by_category)
```

Total Quantity Sold by Category:

	CategoryName	TotalQuantity
0	Beverages	2289
1	Condiments	1383
2	Confections	2110
3	Dairy Products	2601
4	Grains/Cereals	912
5	Meat/Poultry	1288
6	Produce	715
7	Seafood	1445

```
[35]: # 12. How many orders were shipped by each shipper
orders_by_shipper = pd.read_sql_query('''
SELECT Shippers.ShipperName, COUNT(*) AS OrderCount
FROM Orders
JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID
GROUP BY Shippers.ShipperName
ORDER BY OrderCount ASC;

''', conn)
print("Orders Shipped by Each Shipper (Sorted in Ascending Order):")
print(orders_by_shipper)
```

Orders Shipped by Each Shipper (Sorted in Ascending Order):

	ShipperName	OrderCount
0	Speedy Express	54
1	Federal Shipping	68
2	United Package	74

# Using Python

## First Way

- Using Sqllite3



## Second Way

- Working On Excel Sheet Using Python



# Analyze Data "Second way"

```
[2]: import pandas as pd

# Load the Excel file to inspect the contents and see the available sheets
file_path = 'Data Pro.XLSX'
xls = pd.ExcelFile(file_path)

# Display sheet names to understand the structure of the file
xls.sheet_names
```

```
[2]: ['Suppliers',
      'Shippers',
      'Products',
      'Orders',
      'OrderDetails',
      'Employees',
      'Customers',
      'Categories']
```

```
[3]: # Load the necessary sheets for analysis

suppliers_df = pd.read_excel(xls, 'Suppliers')

shippers_df = pd.read_excel(xls, 'Shippers')

products_df = pd.read_excel(xls, 'Products')

orders_df = pd.read_excel(xls, 'Orders')

order_details_df = pd.read_excel(xls, 'OrderDetails')

employees_df = pd.read_excel(xls, 'Employees')

customers_df = pd.read_excel(xls, 'Customers')

categories_df = pd.read_excel(xls, 'Categories')
```

```
[4]: # Merging relevant data for analysis
# Joining OrderDetails with Products and Categories to get product details and category info
order_details_merged_df = pd.merge(order_details_df, products_df, on='ProductID')
order_details_merged_df = pd.merge(order_details_merged_df, categories_df, on='CategoryID')

# Joining with Orders to get customer and shipping details
order_details_merged_df = pd.merge(order_details_merged_df, orders_df, on='OrderID')

# Joining with Customers to get customer details
order_details_merged_df = pd.merge(order_details_merged_df, customers_df, on='CustomerID')

# Joining with Employees to get employee details
order_details_merged_df = pd.merge(order_details_merged_df, employees_df, on='EmployeeID')

# Calculating Total Revenue (Price * Quantity) for each order detail
order_details_merged_df['TotalRevenue'] = order_details_merged_df['Price'] * order_details_merged_df['Quantity']
```

```
[5]: # 1-Calculating the total revenue from all orders
total_revenue = order_details_merged_df['TotalRevenue'].sum()
total_revenue
```

```
[5]: 387066
```

```
[6]: #2- Total Number of Orders
total_orders = orders_df['OrderID'].nunique()
print("Total Number of Orders: ", total_orders)
```

```
Total Number of Orders: 196
```

```
[7]: # 3-Number of Unique Customers
# Number of Unique Customers who placed orders
unique_customers = order_details_merged_df['CustomerID'].nunique()
print("Number of Unique Customers: ", unique_customers)
```

```
Number of Unique Customers: 74
```

```
[8]: # 4-Best-Selling 5 Products
# Best-Selling 5 Products by Total Revenue (Price * Quantity)
best_selling_products = order_details_merged_df.groupby
('ProductName')['TotalRevenue'].sum().nlargest(5)
print("Best-Selling 5 Products by Revenue:\n", best_selling_products)
```

Best-Selling 5 Products by Revenue:

ProductName	TotalRevenue
C?te de Blaye	63096
Th?ringer Rostbratwurst	20832
Raclette Courdavault	19030
Tarte au sucre	15925
Camembert Pierrot	14620

Name: TotalRevenue, dtype: int64

# Analyze Data "Second way"

```
•[9]: #5- Total Revenue by Category
revenue_by_category = order_details_merged_df.groupby(
    'CategoryName')['TotalRevenue'].sum()
print("Total Revenue by Category:\n", revenue_by_category)
```

```
Total Revenue by Category:
CategoryName
Beverages      99699
Condiments     35024
Confections    54729
Dairy Products 70530
Grains/Cereals 22370
Meat/Poultry   51636
Produce        23439
Seafood        29639
Name: TotalRevenue, dtype: int64
```

```
[10]: # 6-Customer Distribution by Country
customer_distribution = customers_df['Country'].value_counts()
print("Customer Distribution by Country:\n", customer_distribution)
```

```
Customer Distribution by Country:
Country
USA      13
Germany  11
France   11
Brazil    9
UK        7
Spain     5
Mexico    5
Venezuela 4
Canada    3
Argentina 3
```

```
•[11]: # 7-Total Sales by Country (Revenue across different countries)
sales_by_country = order_details_merged_df.groupby(
    'Country')['TotalRevenue'].sum()
print("Total Sales by Country:\n", sales_by_country)
```

```
Total Sales by Country:
Country
Argentina      396
Austria       51687
Belgium        8075
Brazil       40272
Canada       31402
Denmark      17954
Finland       6478
France       29559
Germany      47316
Ireland      15405
Italy        4339
```

```
[12]: # 8-Total Number of Suppliers
total_suppliers = suppliers_df['SupplierID'].nunique()
print("Total Suppliers: ", total_suppliers)
```

Total Suppliers: 29

```
•[13]: # 9-Top 3 Employees by Sales
top_employees = order_details_merged_df.groupby(
    (['EmployeeID', 'LastName', 'FirstName']))['TotalRevenue'].sum().nlargest(3)
print("Top 3 Employees by Sales:\n", top_employees)
```

```
Top 3 Employees by Sales:
EmployeeID LastName  FirstName
4          Peacock  Margaret   105926
1           Davolio  Nancy     57765
3          Leverling Janet     42823
Name: TotalRevenue, dtype: int64
```

```
•[12]: # 10- Customers who placed more than one order (Loyal Customers)
# Customers who placed more than one order (Loyal Customers)
orders_per_customer = orders_df.groupby('CustomerID')['OrderID'].nunique()
# Filter customers with more than one order
loyal_customers = orders_per_customer[orders_per_customer > 1]
# Merge with customer names
loyal_customers_with_names = pd.merge(loyal_customers,
    customers_df[['CustomerID', 'ContactName']], on='CustomerID')
# Rename columns for clarity
loyal_customers_with_names.columns = ['CustomerID', 'OrderCount', 'CustomerName']
# Display the results
print(loyal_customers_with_names[['CustomerName', 'OrderCount']])
```

```
CustomerName OrderCount
0      Thomas Hardy      2
1    Christina Berglund    3
2    Fr?d?rique Citeaux    4
3    Laurence Lebihans     3
4    Elizabeth Lincoln     4
5         Yang Wang       2
6         Sven Ottlieb     2
7         Ann Devon       2
8      Roland Mendel      10
```

```
[10]: # 11-Total Quantity Sold by Category
quantity_by_category = order_details_merged_df.groupby('CategoryName')['Quantity'].sum()
print("Total Quantity Sold by Category:\n", quantity_by_category)
```

```
Total Quantity Sold by Category:
CategoryName
Beverages      2289
Condiments     1383
Confections    2110
Dairy Products 2601
Grains/Cereals  912
Meat/Poultry   1288
Produce        715
Seafood       1445
Name: Quantity, dtype: int64
```

```
•[14]: # 12-Orders Shipped by Each Shipper
# Count of orders shipped by each shipper (with Shipper Name)
orders_by_shipper = orders_df.groupby('ShipperID')['OrderID'].count()

# Merge with shipper names from the Shippers dataframe
orders_by_shipper_with_names = pd.merge(orders_by_shipper,
    shippers_df[['ShipperID', 'ShipperName']], on='ShipperID')

# Rename columns for clarity
orders_by_shipper_with_names.columns = ['ShipperID', 'OrderCount', 'ShipperName']

# Display the results
print(orders_by_shipper_with_names[['ShipperName', 'OrderCount']])
```

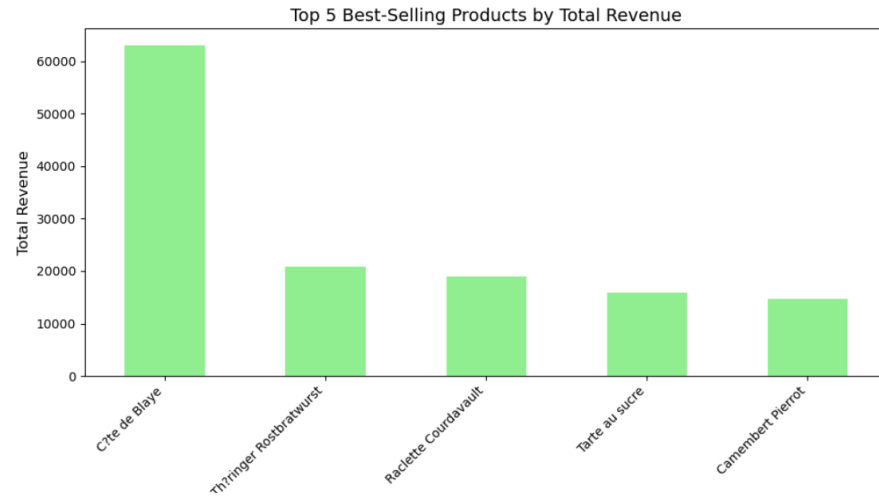
```
ShipperName OrderCount
0    Speedy Express      54
1    United Package      74
2    Federal Shipping     68
```

```
[29]: # Plotting the Best-Selling 5 Products by Total Revenue
plt.figure(figsize=(10, 6))
best_selling_products.plot(kind='bar', color='lightgreen')
```

```
# Adding title and axis Labels
plt.title('Top 5 Best-Selling Products by Total Revenue', fontsize=14)
plt.xlabel('Product Name', fontsize=12)
plt.ylabel('Total Revenue', fontsize=12)
```

```
# Rotating product names for better readability
plt.xticks(rotation=45, ha='right')
```

```
# Display the plot
plt.tight_layout()
plt.show()
```

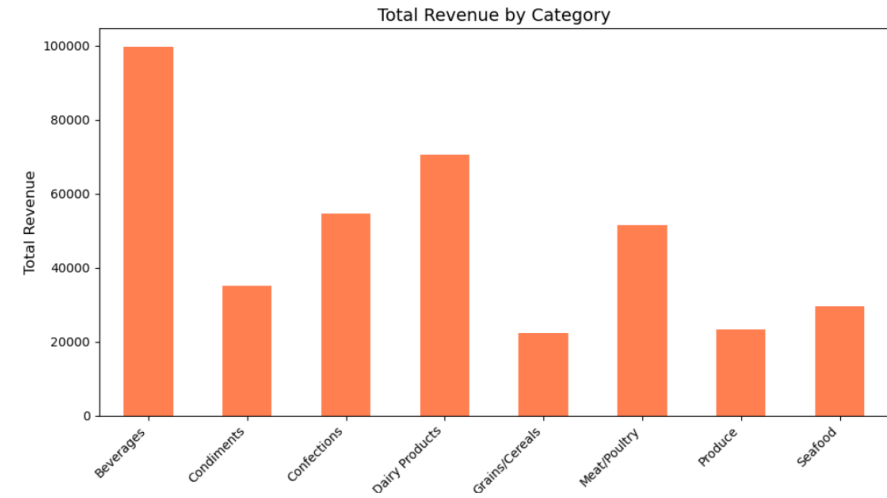


```
[30]: # Plotting the Total Revenue by Category
plt.figure(figsize=(10, 6))
revenue_by_category.plot(kind='bar', color='coral')
```

```
# Adding title and axis Labels
plt.title('Total Revenue by Category', fontsize=14)
plt.xlabel('Category Name', fontsize=12)
plt.ylabel('Total Revenue', fontsize=12)
```

```
# Rotating category names for better readability
plt.xticks(rotation=45, ha='right')
```

```
# Display the plot
plt.tight_layout()
plt.show()
```



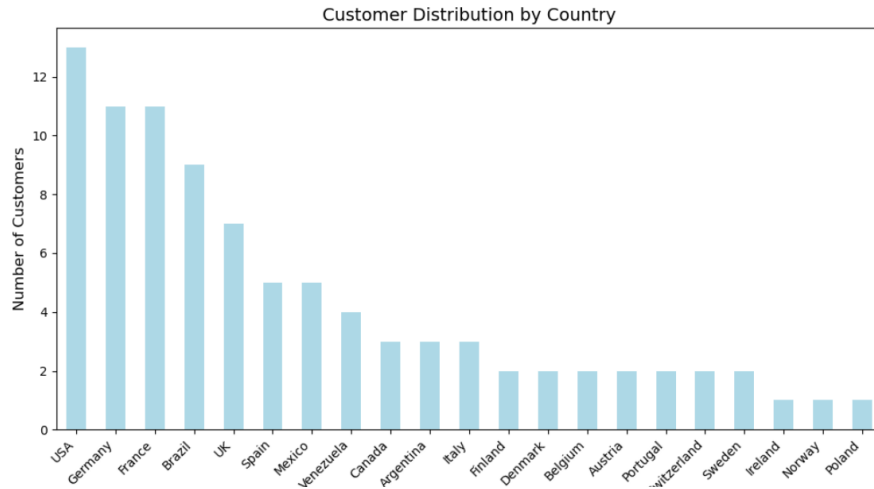
# Samples for python visuals

```
[31]: # Plotting Customer Distribution by Country
plt.figure(figsize=(10, 6))
customer_distribution.plot(kind='bar', color='lightblue')

# Adding title and axis labels
plt.title('Customer Distribution by Country', fontsize=14)
plt.xlabel('Country', fontsize=12)
plt.ylabel('Number of Customers', fontsize=12)

# Rotating country names for better readability
plt.xticks(rotation=45, ha='right')

# Display the plot
plt.tight_layout()
plt.show()
```

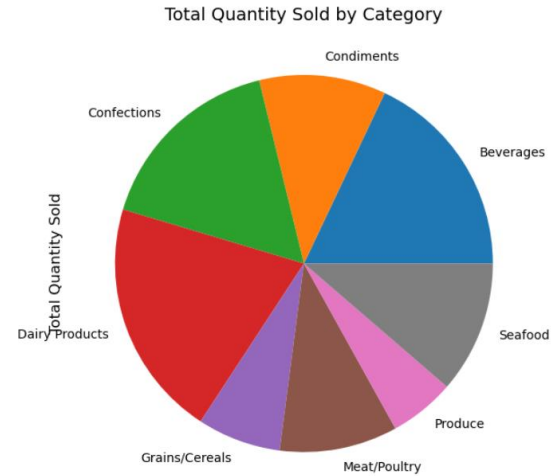


```
[38]: # Plotting Total Quantity Sold by Category
plt.figure(figsize=(10, 6))
quantity_by_category.plot(kind='pie', color='dodgerblue')

# Adding title and axis labels
plt.title('Total Quantity Sold by Category', fontsize=14)
plt.xlabel('Category Name', fontsize=12)
plt.ylabel('Total Quantity Sold', fontsize=12)

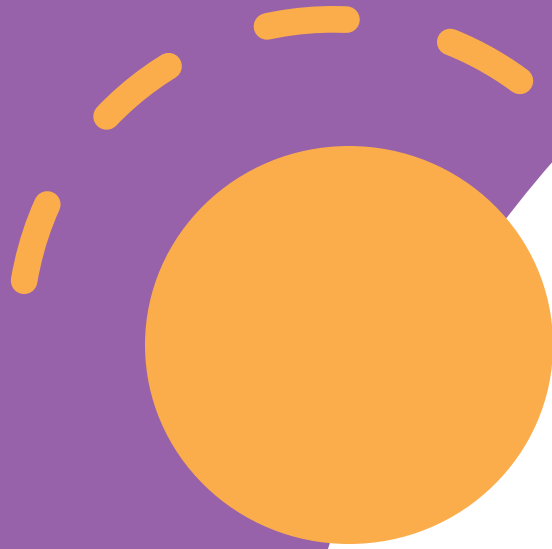
# Rotating category names for better readability
plt.xticks(rotation=45, ha='right')

# Display the plot
plt.tight_layout()
plt.show()
```



# Samples for python visuals





# Using Power BI



## SQL Server database

Server

.

Database (optional)

Northwind

Advanced options

OK

Cancel

Navigator

Select multiple items

Display Options

Local SQL database: Northwind [10]

Categories

Customers

Employees

OrderDetails

Orders

Products

Shippers

Suppliers

Tables

Relationships

Views

Stored Procedures

Functions

Aggregate Functions

Scalar Functions

System Functions

System Stored Procedures

System Views

System Tables

System Columns

System Indexes

System Constraints

System Triggers

System Rules

System Defaults

System Schemas

System Databases

System Servers

System Groups

System Users

System Roles

System Permissions

System Security

System Audit

System Backup

System Restore

System Maintenance

System Configuration

System Tools

System Utilities

System Services

System Agents

System Alerts

System Events

System Jobs

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

System Operators

# Analyze Data Using Power BI

DEPI GR Product • Last saved: Today at 12:06 AM

Search

Ibrahim Eidakrorey

Share

File Home Help

Name Date Map

Table tools

Mark as date table

Manage relationships

New measure

Quick measure

New column

New table

Structure

1 Date Map - CALENDAR(AUTO())

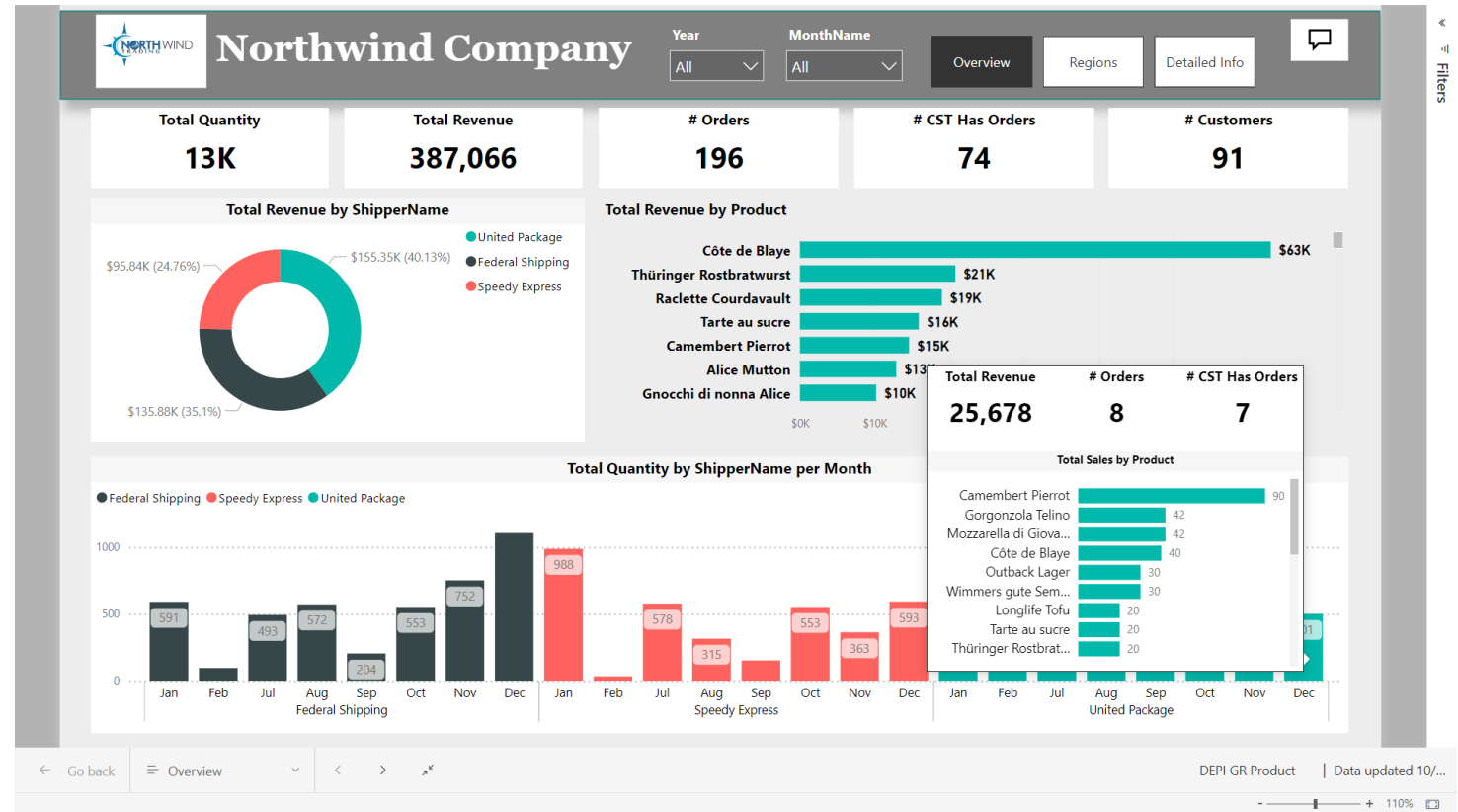
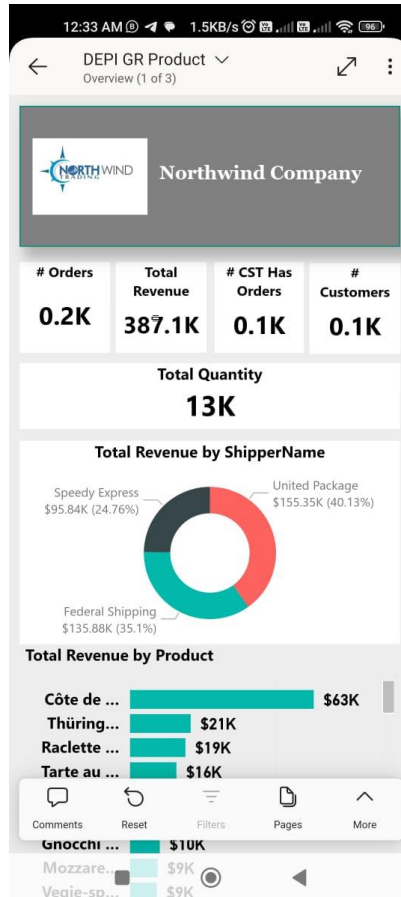
Date	Year	Week	Month	Day	MonthName	Week-Year
1/1/1928	1928	1	1	1	Jan	W1-1928
1/2/1928	1928	1	1	2	Jan	W1-1928
1/3/1928	1928	1	1	3	Jan	W1-1928
1/4/1928	1928	1	1	4	Jan	W1-1928
1/5/1928	1928	1	1	5	Jan	W1-1928
1/6/1928	1928	1	1	6	Jan	W1-1928
1/7/1928	1928	1	1	7	Jan	W1-1928
1/8/1928	1928	2	1	8	Jan	W2-1928
1/9/1928	1928	2	1	9	Jan	W2-1928
1/10/1928	1928	2	1	10	Jan	W2-1928
1/11/1928	1928	2	1	11	Jan	W2-1928
1/12/1928	1928	2	1	12	Jan	W2-1928
1/13/1928	1928	2	1	13	Jan	W2-1928
1/14/1928	1928	2	1	14	Jan	W2-1928
1/15/1928	1928	3	1	15	Jan	W3-1928
1/16/1928	1928	3	1	16	Jan	W3-1928
1/17/1928	1928	3	1	17	Jan	W3-1928
1/18/1928	1928	3	1	18	Jan	W3-1928
1/19/1928	1928	3	1	19	Jan	W3-1928
1/20/1928	1928	3	1	20	Jan	W3-1928
1/21/1928	1928	3	1	21	Jan	W3-1928
1/22/1928	1928	4	1	22	Jan	W4-1928
1/23/1928	1928	4	1	23	Jan	W4-1928
1/24/1928	1928	4	1	24	Jan	W4-1928
1/25/1928	1928	4	1	25	Jan	W4-1928
1/26/1928	1928	4	1	26	Jan	W4-1928
1/27/1928	1928	4	1	27	Jan	W4-1928
1/28/1928	1928	4	1	28	Jan	W4-1928

Table: Date Map (25,568 rows)

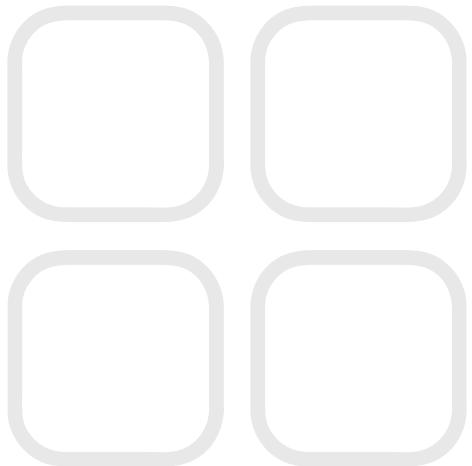
Type here to search

23°C عاتمة 10/11/2024

# Analyze Data Using Power BI



# Question Time





**Thank You**