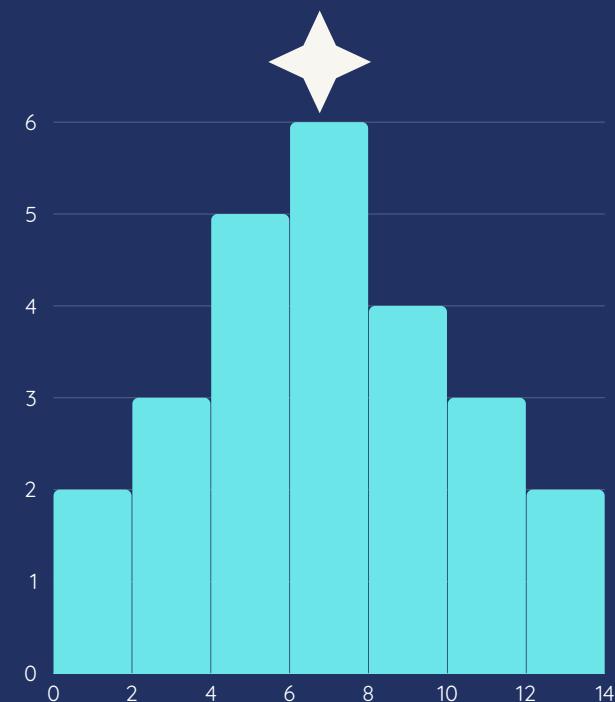


# AdventureWorks

## Sales Analysis



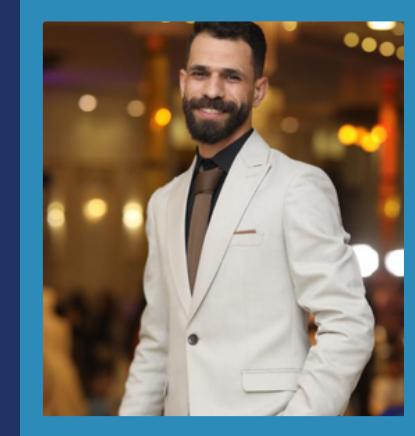
# Team Members



**Shorouq taha**  
data analyst



**Esraa ayman**  
data analyst



**Ahmed Gamal**  
data analyst





# Agenda

- 1 . Introduction
- 2 . Tools
- 3 . Dataset Overview
- 4 . Analysis & Insights
- 5 . Challenges & Solutions
- 6 . Visualization
- 7 . Conclusion
- 8 . References

## 1 . Introduction

- The sales analysis project aims to understand the sales performance of Adventure Works by analyzing its sales data.
- Emphasis is placed on aspects such as sales volume, store performance, and customer behavior, which helps identify factors affecting revenue and provides recommendations for performance improvement.
- Through analysis, it is possible to identify market trends, understand customer preferences, and identify areas or products that need improvement. This helps the company increase profits, reduce costs and develop more effective strategies.



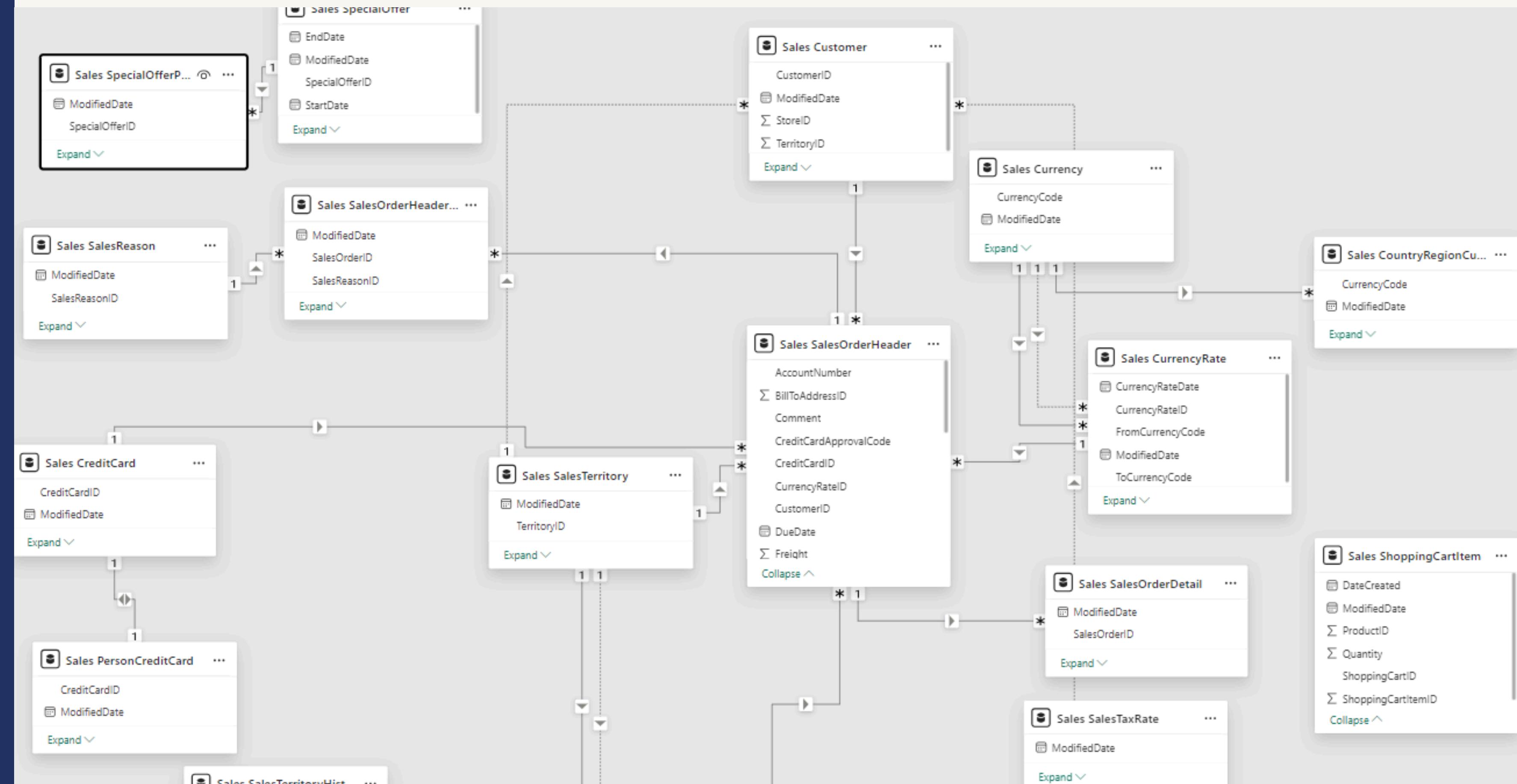
## 2. Tools



### 3 . Dataset Overview

- The Adventure Works Data Collection is an educational database that simulates the sales data of a global sports company.
- It contains a wide range of tables that include product data, sales, customers, employees, and stores.

# • Sales Data Model



## Main Tables

Table	Description
SalesOrderHeader	It contains basic information for each sales order such as order date, order status, and total amount paid.
SalesOrderDetail	Contains details of each item within the order, such as product number, quantity, and price. This table allows you to analyze best-selling products.
Store	It includes information about the stores through which sales are carried out. It helps analyze sales performance by store and identify the stores that generate the most revenue.
SalesPerson	Combine the data of the employees responsible for sales, allowing the analysis of the performance of the sales team and the identification of the most qualified employees.
Customers	Contain customer data such as name, address and city. Helps understand customer distribution and purchasing behaviors

# Data Exploration

```
[ ] #print sheet names
print(xls.sheet_names)

→ ['CountryRegionCurrency', 'CreditCard', 'Currency', 'CurrencyRate', 'Customer', 'PersonCreditCard']

[ ] # read data of sheet
df = pd.read_excel(xls, sheet_name='PersonCreditCard')
print(df.info())

→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 19118 entries, 0 to 19117
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   BusinessEntityID 19118 non-null    int64  
 1   CreditCardID     19118 non-null    int64  
 2   ModifiedDate      19118 non-null    datetime64[ns]

[ ] # read data of sheet
df = pd.read_excel(xls, sheet_name='Sales person')
df.head()

→   BusinessEntityID TerritoryID SalesQuota Bonus CommissionPct SalesYTD SalesLastYear          rowguid ModifiedDate
 0             274        NaN       NaN     0       0.000  5.596976e+05  0.000000e+00 48754992-9EE0-4C0E-8C94-9451604E3E02  2010-12-28
 1             275         2.0  300000.0  4100      0.012  3.763178e+06  1.750406e+06 1E0A7274-3064-4F58-88EE-4C6586C87169  2011-05-24
 2             276         4.0  250000.0  2000      0.015  4.251369e+06  1.439156e+06 4DD9EEE4-8E81-4F8C-AF97-683394C1F7C0  2011-05-24
 3             277         3.0  250000.0  2500      0.015  3.189418e+06  1.997186e+06 39012928-BFEC-4242-874D-423162C3F567  2011-05-24
 4             278         6.0  250000.0   500      0.010  1.453719e+06  1.620277e+06 7A0AE1AB-B283-40F9-91D1-167ABF06D720  2011-05-24

df.isnull().sum()

→
BusinessEntityID 0
TerritoryID 3
SalesQuota 3
Bonus 0
CommissionPct 0
SalesYTD 0
SalesLastYear 0
rowguid 0
ModifiedDate 0
```

- It is a fundamental stage in data analysis aimed at understanding the structure and nature of a data set before starting deep analysis or building models.
- During this phase, the data is carefully studied to get a comprehensive overview of its content, detect initial patterns, and identify missing or anomalous values.

```
[ ] # read data of sheet  
df = pd.read_excel(xls, sheet_name='Sales person')  
df.head()
```

→

	BusinessEntityID	TerritoryID	SalesQuota	Bonus	CommissionPct	SalesYTD	SalesLastYear	rowguid	ModifiedDate
0	274	NaN	NaN	0	0.000	5.596976e+05	0.000000e+00	48754992-9EE0-4C0E-8C94-9451604E3E02	2010-12-28
1	275	2.0	300000.0	4100	0.012	3.763178e+06	1.750406e+06	1E0A7274-3064-4F58-88EE-4C6586C87169	2011-05-24
2	276	4.0	250000.0	2000	0.015	4.251369e+06	1.439156e+06	4DD9EEE4-8E81-4F8C-AF97-683394C1F7C0	2011-05-24
3	277	3.0	250000.0	2500	0.015	3.189418e+06	1.997186e+06	39012928-BFEC-4242-874D-423162C3F567	2011-05-24
4	278	6.0	250000.0	500	0.010	1.453719e+06	1.620277e+06	7A0AE1AB-B283-40F9-91D1-167ABF06D720	2011-05-24

▶ df.isnull().sum()

→ 0

BusinessEntityID 0

TerritoryID 3

SalesQuota 3

Bonus 0

CommissionPct 0

SalesYTD 0

SalesLastYear 0

rowguid 0

ModifiedDate 0

## 4 . Analysis & Insights

- At this point, SQL was used to extract basic data from linked tables, such as sales data by stores and customers.
- Next, Python was used to perform advanced statistical analysis on the data.
- This helps detect important patterns and trends in store sales and identify factors affecting the financial performance of each store.



## Main KPI'S

Total sales per store were calculated using SQL, where sales were grouped at the store level or time period to analyze overall sales performance. This can be calculated using a query such as `SUM(SalesAmount)`.

Calculate average sales per customer, allowing you to identify purchasing behavior and identify high-value customers. This is done by dividing the client's total sales by the number of their orders.

**Most spenders:** The customers who spend the most were identified by grouping sales per customer and sorting them in descending order. This analysis helps target important customers by improving offerings and customer services.



- **DAX Formula**

- 1- Calculated column to collect gross profit

`1 #profit = Orders[OrderQty] * Orders[Unit Profit ]`

ProductID	OrderQty	SalesOrderDetailID	UnitPrice	UnitPriceDiscount	LineTotal	ProductCost	Unit Profit	#profit	Gocs
870	1	52242	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	52592	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	52694	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	52799	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	53799	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	54058	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663

- 2- calculated column to collect cost of goods sold

`1 Gocs = (Orders[ProductCost]) * Orders[OrderQty]`

ProductID	OrderQty	SalesOrderDetailID	UnitPrice	UnitPriceDiscount	LineTotal	ProductCost	Unit Profit	#profit	Gocs
870	1	52242	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	52592	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	52694	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	52799	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	53799	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	54058	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663
870	1	54950	4.99	0	4.99	1.8663	3.1237	3.1237	1.8663

### 3- Gross margin

```
1 GrossMargin =
2 VAR TotalRevenue = SUM(Orders[LineTotal]) -- Sum of revenue from SubTotal column
3 VAR TotalCost = SUM(Orders[Gocs])    -- Sum of cost from Total Cost column
4 RETURN
5 DIVIDE(
6   TotalRevenue - TotalCost,    -- Difference between revenue and cost
7   TotalRevenue,              -- Dividing by total revenue
8   0                         -- Return 0 if revenue is 0
9 )
```

- is an accounting term used to measure a company's profitability from its core operations.
- It is calculated by subtracting the cost of goods sold (COGS) from revenue (sales), and then dividing the result by revenue.
- This margin shows the percentage of profit a company makes after covering the direct costs of producing goods or providing services.

# Sql Analysis

SalesPerson Performance during the current and last year

```

SELECT
    SP.BusinessEntityID,
    P.FirstName + ' ' + P.LastName AS SalesPersonName,
    SP.SalesYTD,
    SP.SalesLastYear,
    (SP.SalesYTD - SP.SalesLastYear) AS SalesGrowth
FROM
    Sales.SalesPerson SP
JOIN
    Person.Person P
ON
    SP.BusinessEntityID = P.BusinessEntityID
  
```

Results

	BusinessEntityID	SalesPersonName	SalesYTD	SalesLastYear	SalesGrowth
1	274	Stephen Jiang	559697.5639	0.00	559697.5639
2	275	Michael Blythe	3763178.1787	1750406.4785	2012771.7002
3	276	Linda Mitchell	4251368.5497	1439156.0291	2812212.5206
4	277	Jillian Carson	3189418.3662	1997186.2037	1192232.1625
5	278	Garrett Vargas	1453719.4653	1620276.8966	-166557.4313
6	279	Tevi Reiter	2315185.6111	1849640.9418	465544.6692
7	280	Pamela Anzman-Wolfe	1352577.1325	1927059.178	-574482.0455
8	281	Shu Ito	2458535.6169	2073505.9999	385029.617
9	282	José Saraiva	2604540.7172	2038234.6549	566306.0623
10	283	David Campbell	1573012.9383	1371635.3158	201377.6225
11	284	Tete Mensa-Annan	1576562.1966	0.00	1576562.1966
12	285	Syed Abbas	172524.4512	0.00	172524.4512
13	286	Lynn Tsolfias	1421810.9242	2278548.9776	-856738.0534
14	287	Amy Alberts	519905.932	0.00	519905.932
15	288	Rachel Valdez	1827066.7118	1307949.7917	519116.9201
16	289	Jae Pak	4116871.2277	1635823.3967	2481047.831
17	290	Banu Varkey Chudukatil	3121616.3202	2396539.7601	725076.5601

Top 10 product By total Revenue

```

SELECT TOP 10
    P.ProductID, -- معرف المنتج
    P.Name AS ProductName, -- اسم المنتج
    SUM(SOD.LineTotal) AS TotalRevenue, -- إجمالي الإيرادات لكل منتج
    SUM(SOD.OrderQty) AS TotalQuantity -- مجموع الكمية المباعة لكل منتج
FROM
    Sales.SalesOrderDetail SOD -- جدول تفاصيل الطلب
JOIN
    Production.Product P -- جدول المنتجات
ON
    SOD.ProductID = P.ProductID -- الربط عبر معرف المنتج
GROUP BY
    P.ProductID, P.Name -- تجميع النتائج حسب معرف المنتج واسم المنتج
ORDER BY
    TotalRevenue DESC;
  
```

Results

	ProductID	ProductName	TotalRevenue	TotalQuantity
1	782	Mountain-200 Black, 38	4400592.800400	2977
2	783	Mountain-200 Black, 42	4009494.761841	2664
3	779	Mountain-200 Silver, 38	3693678.025272	2394
4	780	Mountain-200 Silver, 42	3438478.860423	2234
5	781	Mountain-200 Silver, 46	3434256.941928	2216
6	784	Mountain-200 Black, 46	3309673.216908	2111
7	793	Road-250 Black, 44	2516857.314918	1642
8	794	Road-250 Black, 48	2347655.953454	1498
9	795	Road-250 Black, 52	2012447.775000	1245
10	753	Road-150 Red, 56	1847818.628000	664

SQLQuery1.sql - D...KLSD5EU\w 10 (72)\* SalesProject[1].sq...-KLSD5EU\w 10 (61)\*

```

----- total Revenue -----
select sum (LineTotal) AS Totalrevenue
from [Sales].[SalesOrderDetail]

----- Years and monthly orders Qty |-----
SELECT
    YEAR(SOH.OrderDate) AS OrderYear,
    MONTH(SOH.OrderDate) AS OrderMonth,
    SUM(SOD.OrderQty) AS TotalOrderQty -- مجموع الكمية لكل شهر --
FROM
    Sales.SalesOrderHeader SOH
JOIN
    Sales.SalesOrderDetail SOD ON SOH.SalesOrderID = SOD.SalesOrderID
GROUP BY
    YEAR(SOH.OrderDate), MONTH(SOH.OrderDate) -- تجميع النتائج حسب السنة والشهر --
ORDER BY
    OrderYear DESC,
    TotalOrderQty DESC;

```

99 %

Results Messages

	Totalrevenue		
1	109846381.399888		

	OrderYear	OrderMonth	TotalOrderQty
1	2014	3	22582
2	2014	5	15884
3	2014	1	11463
4	2014	4	5313
5	2014	2	4287
6	2014	6	2130
7	2013	7	18589
8	2013	6	16611

	OrderMonth	TotalOrderQty
1	5	34692
2	3	34016
3	6	30177
4	7	29940
5	10	26915
6	8	22027

Query executed successfully.

Number Of Products Sold in offers-----

```

SELECT
    SO.SpecialOfferID,
    SO.Description AS SpecialOfferDescription,
    COUNT(SOD.ProductID) AS NumberOfProductsSold
FROM
    Sales.SpecialOffer SO
JOIN
    Sales.SpecialOfferProduct SOP ON SO.SpecialOfferID = SOP.SpecialOfferID
JOIN
    Sales.SalesOrderDetail SOD ON SOP.ProductID = SOD.ProductID
WHERE
    SO.Description <> 'No Discount'
GROUP BY
    SO.SpecialOfferID,
    SO.Description
ORDER BY
    NumberOfProductsSold DESC;

```

99 %

Results Messages

	SpecialOfferID	SpecialOfferDescription	NumberOfProductsSold
1	2	Volume Discount 11 to 14	72473
2	3	Volume Discount 15 to 24	47999
3	4	Volume Discount 25 to 40	18424
4	8	Sport Helmet Discount-2002	9180
5	11	Sport Helmet Discount-2003	9180
6	10	Mountain Tire Sale	3419
7	16	Mountain-500 Silver Clearance Sale	3412
8	12	LL Road Frame Sale	2527
9	13	Touring-3000 Promotion	2443
10	7	Mountain-100 Clearance Sale	1846
11	15	Half-Price Pedal Sale	1517
12	14	Touring-1000 Promotion	1486
13	5	Volume Discount 41 to 60	1071
14	9	Road-650 Overstock	689

Query executed successfully.

## 5 . Challenges & Solutions

### 1. challenges

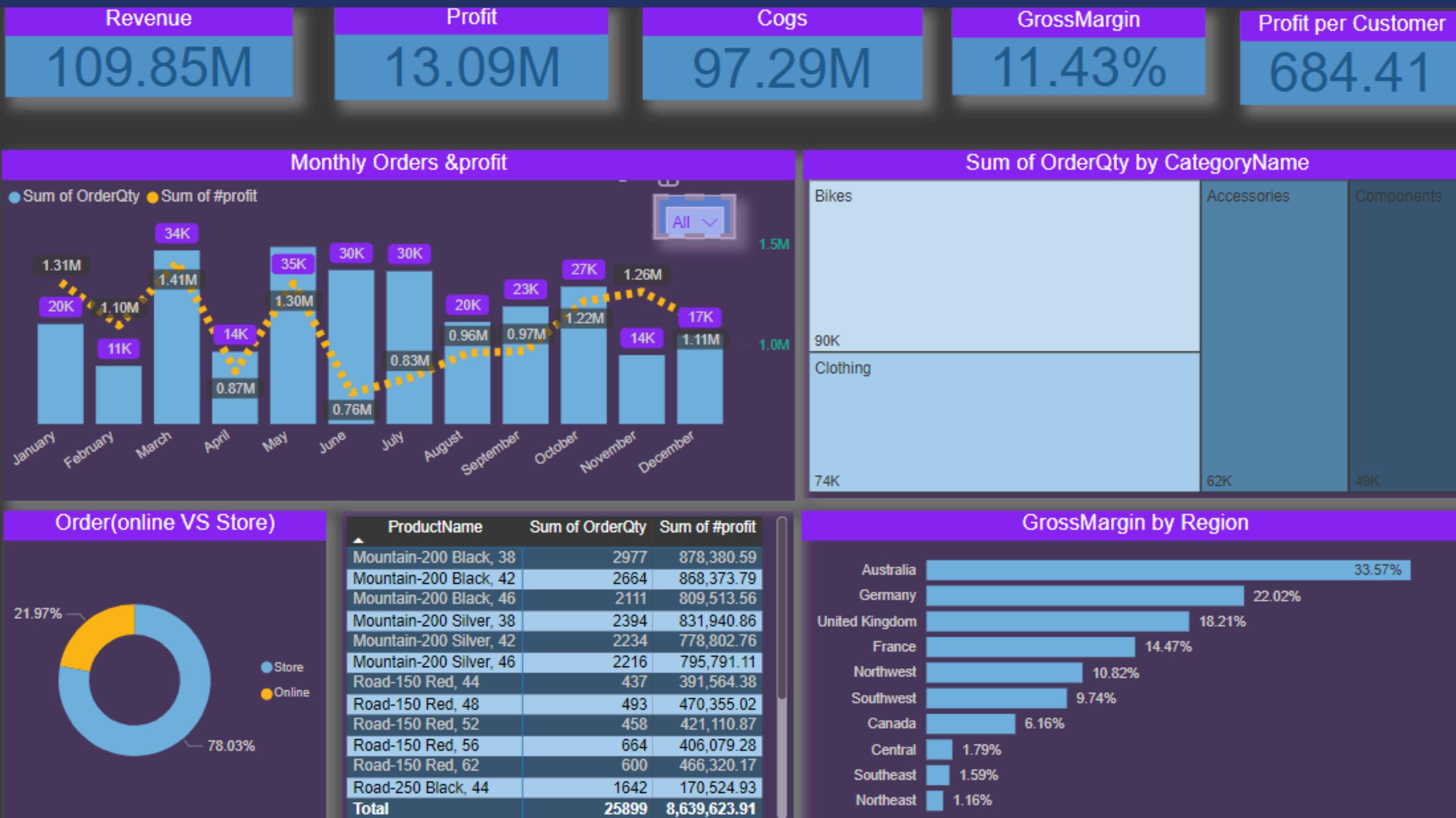
- The challenge was to understand how to properly construct relationships between tables to extract integrated and interconnected data between sales, customers, and stores.
- There has been difficulty in linking different tables such as Salisorderhader and Salisorditile with tables such as Customs, Storr and Salisperson, as these tables contain foreign keys that need to be thoroughly checked to ensure data integrity.
- had difficulties dealing with missing values (missing falls) and outliers in some columns. Missing values may lead to deviation in results if not handled properly.

## 2. Solutions

- used graphs to understand the database structure and relationships between tables, making it easier to link tables correctly and easily extract the required data. SQL was used to perform the necessary JOINs and extract integrated data.
- Libraries such as Pandas and Numpy have been used to clean up data and manipulate missing values through methods such as replacing missing values with an average or median.
- identified outliers using charts such as Box Plot and checked their impact on the analysis, and then decided to either delete or modify them.
- Power BI tools were used to clean up data (Data Transformation) to ensure its quality before being used in graphs.

# 6 . Visualization

This dashboard shows the basic performance indicators for sales operations and the performance of various products



## 7 . Recommendation

### 1-Exploiting seasonal fluctuations:

**Plan ahead:** The sales team must plan for high seasons by increasing inventory, implementing customized marketing campaigns, and training employees to handle the increase in demand.

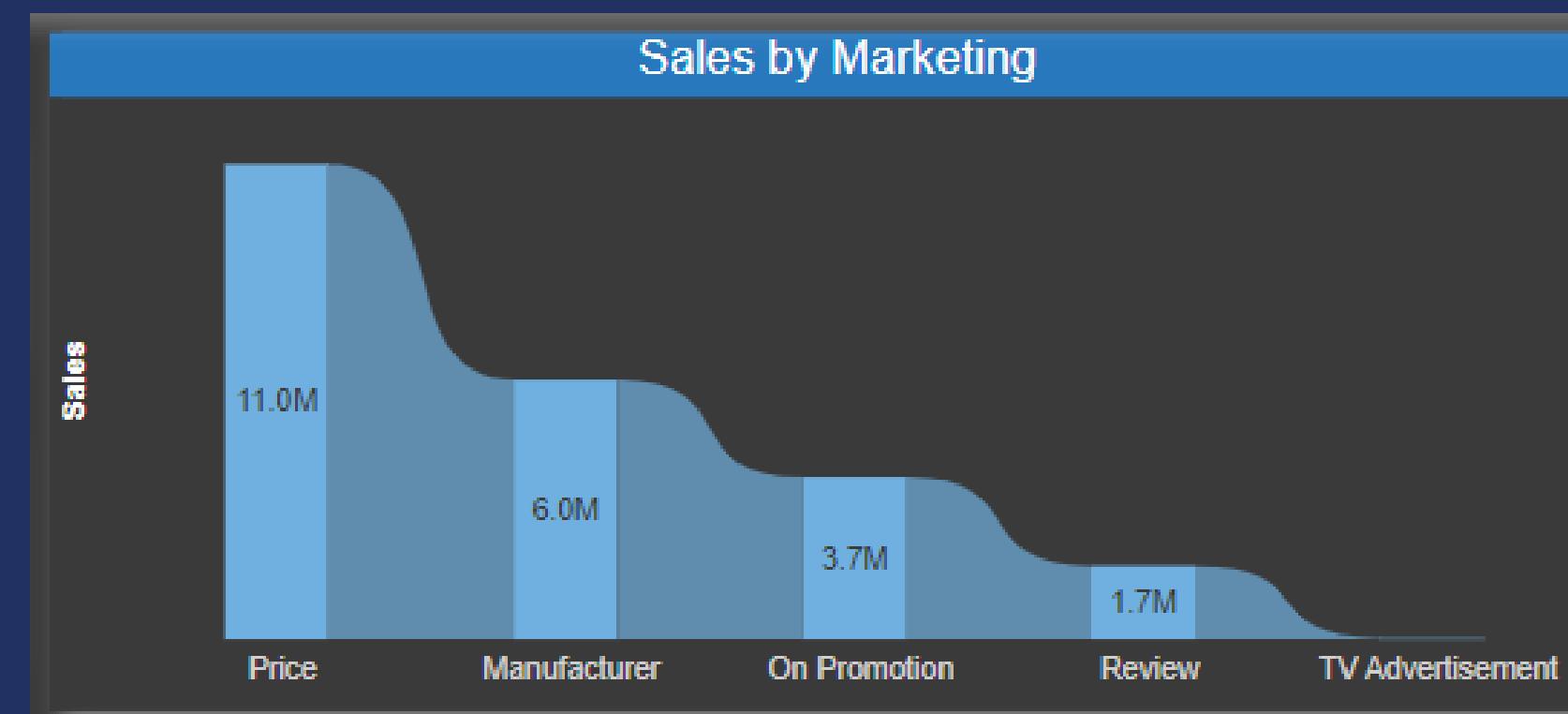
**Take advantage of quiet seasons:** Quiet seasons can be used to develop products, train employees, and perform equipment maintenance.



### 2-Analysis of the reasons for the decrease in sales during some holidays:

**Performance comparison:** Sales performance must be compared during different holidays to determine which holidays the company achieves good results and which holidays need improvement.

**Modifying strategies:** Marketing strategies used during the holidays that achieve less than expected results can be modified.

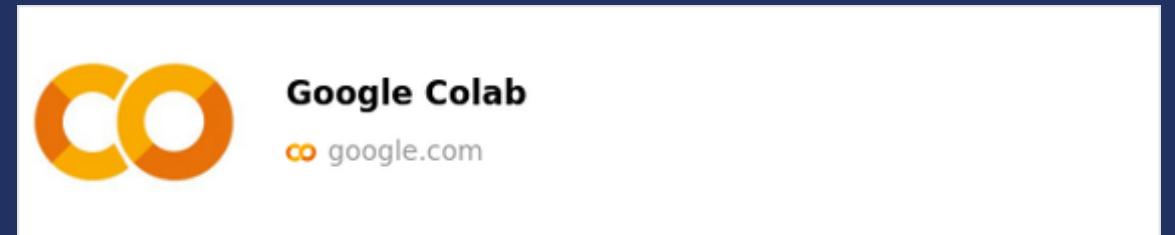


# 8 . Reference

- Data Set



- python analysis code





# Thanks

