4/26/2025

Prestige Cars Sales Analysis Project

Group (1)

1- Hossameldin Hassan Abdelhady

2- Germine Yousry Abdelwahab

3- Ayatollah Ahmed Mahmoud

4- Salma Kamel Abdelaziz

Technical Instructor : Eng. Ahmed Alaa

Group : Alx2_Dat1_G1

Google Data Analyst Specialist

Table of Contents

1. Project Planning & Management

1.1 Project Overview

The Prestige Car Sales Project is designed to analyze historical and current sales data to uncover trends, support data-driven decision-making, and maximize revenue potential. This project focuses on transforming raw sales data into meaningful insights through systematic data processing, storage, and visualization.

Components of The Project Include:

- **Data Cleaning , Processing and Analysis**
  Utilizing Python libraries such as Pandas and NumPy to clean, transform, and prepare the sales dataset for analysis.
- **Interactive Data Visualization**
  Developing insightful and dynamic dashboards in Power BI to visualize key performance indicators (KPIs), sales trends, and regional performance for strategic business review.
- **SQL Database Integration**
  Migrating structured data from Excel files into a relational SQL database, enabling scalable storage, efficient querying, and improved data integrity.

1.2 Objectives

The project aims to apply both descriptive and diagnostic analytics to deliver valuable insights from sales data. These insights will inform business strategies, improve forecasting accuracy, and enhance overall operational efficiency.

1.2.1.Specific objectives include:

- **Analyze Sales Trends Over Time**
  Understand overall sales patterns and profit variations through time-series analysis.
- **Identify High-Performing Products and Customer Segments**
  Diagnose key revenue drivers by analyzing product performance and customer demographics.
- **Improve Demand Forecasting and Inventory Management**
  Use historical data to enhance cost and revenue control and meet market demand more effectively.
- **Develop Interactive Dashboards**
  Enable real-time exploration of sales data using dynamic, user-friendly visualizations in Power BI.

1.2.2.Key Analytical Questions for Prestige Cars Sales Analysis

Descriptive Questions (What happened?)

- **What are the total sales ,cost and profit over time?**
- **Which car's categories generate the highest sales and profit margins?**
- **Which car models generate the highest sales?**
- **Which car's make country generate the highest sales and profit margins?**
- **Which car colors are generating the most sales ?**
- **Who are our top customers or customer segments?**
- **Which regions or locations have the highest sales and profit margins?**

Diagnostic Questions (Why did it happen?)

- **Why did profit decrease in 2017 ?**

By answering these questions, the project will support decision-makers in making informed, strategic choices based on trends and data forecasts.

1.3 Scope

1.3.1 Data Coverage

- **Time Period:** Sales data from 2015 to 2018 (4 years analysis).
- **Geographical Coverage:** Sales across EMEA (Europe, Middle East, Africa), North America, and others.
- **Product Range:** Luxury car brands (Ferrari, Porsche, Aston Martin, Bentley, Jaguar, etc.).

1.3.2 Key Metrics Analyzed

Category	Key Metrics
Sales Performance	Total sales revenue, units sold, best/worst-selling models
Profitability	Gross profit, profit margin, discount impact, cost breakdown (repairs, transport, parts)
Customer Insights	Top customers, repeat buyers, regional sales distribution
Inventory & Pricing	Stock turnover, pricing trends, color preferences, model-wise demand
Time-Based Trends	Yearly sales patterns, seasonality effects

1.3.3 Deliverables

- **Data Cleaning & Processing Report (Python)**

  Handling missing data, outliers, and inconsistencies.

  Feature engineering (e.g., profit margin calculation, time-based aggregations).

- **Exploratory Data Analysis (EDA)**

  Statistical summaries, correlation analysis, trend identification.

- **Interactive Power BI Dashboard**

  Sales performance dashboard.

  Profitability heatmaps.

  Customer segmentation visuals.

  Inventory optimization recommendations.

- **Business Recommendations**

  Pricing strategy adjustments.

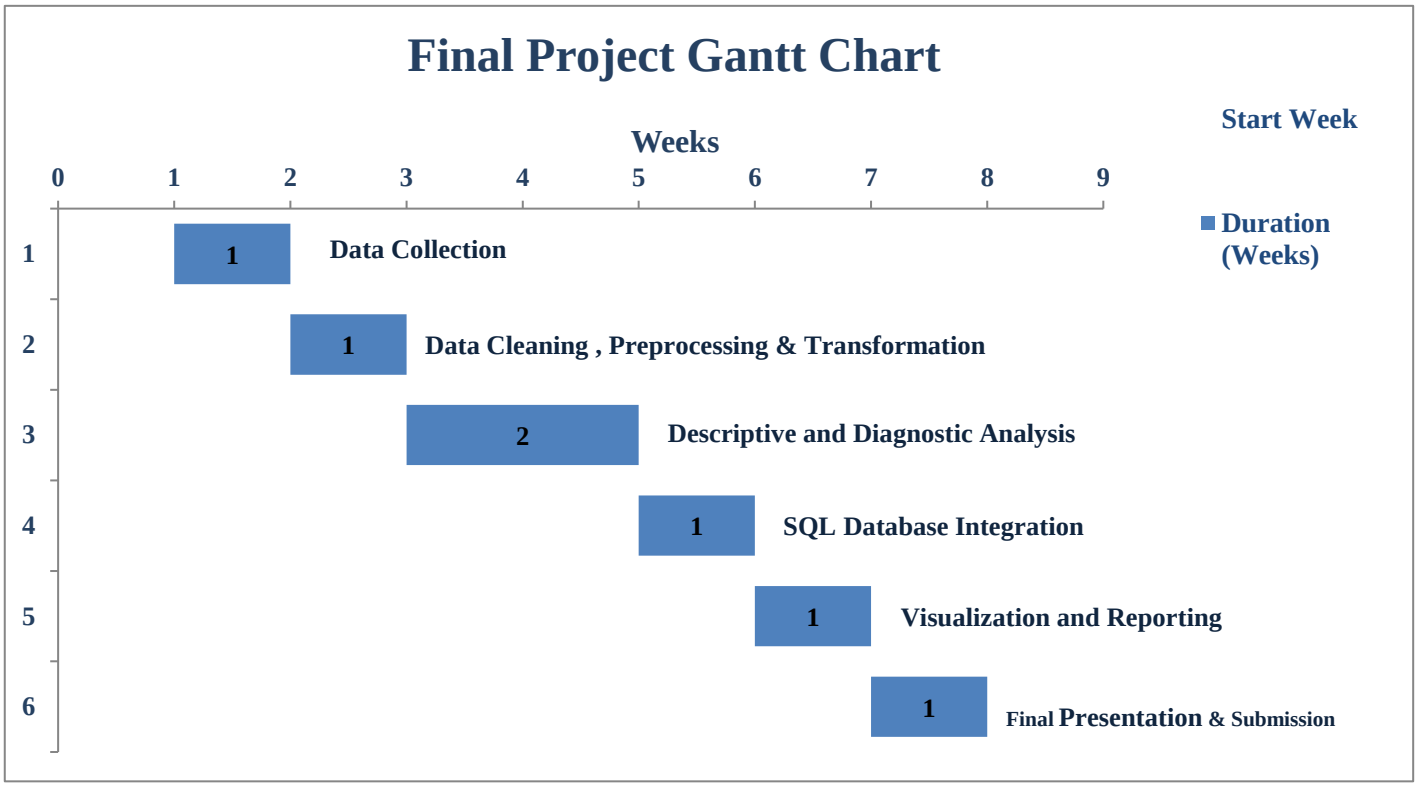
  Inventory management improvements.

  Regional sales expansion opportunities.

1.4 Project Timeline

To ensure the smooth execution of the project, we created a detailed timeline based on the phases of data analysis and presentation. The available duration is distributed as follows:

Task	Start Week	End Week	Duration (Weeks)
Data Collection	**1**	**2**	**1**
Data Cleaning , Preprocessing & Transformation	**2**	**3**	**1**
Descriptive and Diagnostic Analysis	**3**	**5**	**2**
SQL Database Integration	**5**	**6**	**1**
Visualization and Reporting	**6**	**7**	**1**
Final Presentation & Submission	**7**	**8**	**1**



Phase 1: Data Collection

Tools : Excel

Source:

Excel file from https://github.com / database for data analyst by Adam Aspin book

Data Coverage:

351 records of car sales with +30 columns including: (sales, inventory, customer, financial metrics). From (2015–2018).

Data Description:

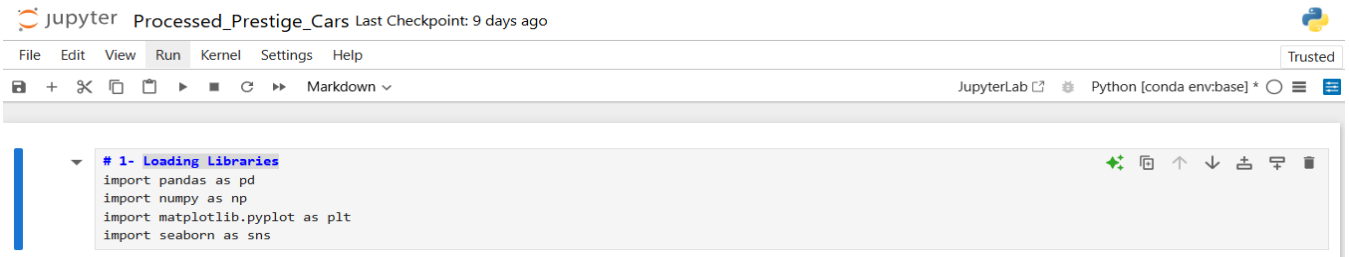
Key Fields:

- Sales details (ID, date, price, discounts)
- Vehicle information (make, model, color, costs)
- Customer data (name, location)
- Financial metrics (profit, profit percentage)
- Sales data

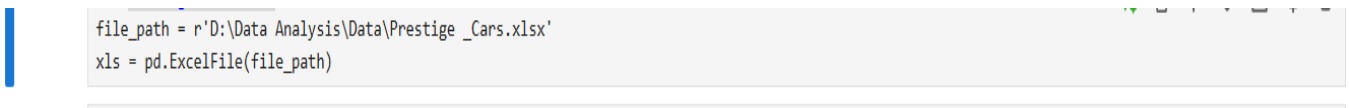
Phase 2: Exploratory Data Analysis, Data Cleaning , Data Processing & Transformation

Tools : Python

1- Loading Python Libraries That Support Data Cleaning, Transformation, Visualization, and Working With Database.



2- Loading Excel File and Connecting With Python



3- Exploratory Data Analysis to Explore, Visualize, and Better Understand Dataset Before Doing Deeper Analysis.

- ### Creating Data Frame

```
df = pd.read_excel(xls, sheet_name='Prestige_Cars')
```

- ### View first few rows

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

[35]:

	SalesDetailsID	SalesID	LineItemNumber	StockID	SalePrice	LineItemDiscount	SaleDate	Data Sales.ID	CustomerID	CustomerName	...	Stock.Cost	Stock.RepairsC
0	1	1	1	B1C3B95E-3005-4840-8CE3-A7BC5F9CFB3F	65000	2700.0	2015-01-02 08:00:00	1	1	Magic Motors	...	52000	2
1	2	2	1	A2C3B95E-3005-4840-8CE3-A7BC5F9CFB5F	220000	60000.0	2015-01-25 00:00:00	2	2	Snazzy Roadsters	...	176000	5
2	3	3	1	558620F5-B9E8-4FFF-8F73-A83FA9559C41	19500	NaN	2015-02-03 10:00:00	3	3	Birmingham Executive Prestige Vehicles	...	15600	
3	4	4	1	72443561-FAC4-4C25-B8FF-0C47361DDE2D	11500	NaN	2015-02-16 08:00:00	4	4	WunderKar	...	9200	
4	5	5	1	2189D556-D1C4-4BC1-B0C8-4053319E8E9D	19950	NaN	2015-01-02 10:33:00	5	5	Casseroles Chromes	...	15960	1

5 rows × 26 columns

- ### General information about the dataset

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 351 entries, 0 to 350
Data columns (total 26 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   SalesDetailsID         351 non-null    int64
 1   SalesID                351 non-null    int64
 2   LineItemNumber         351 non-null    int64
 3   StockID                351 non-null    object
 4   SalePrice              351 non-null    int64
 5   LineItemDiscount       119 non-null    float64
 6   SaleDate               351 non-null    datetime64[ns]
 7   Data Sales.ID          351 non-null    int64
 8   CustomerID             351 non-null    int64
 9   CustomerName           351 non-null    object
 10  Customer.Address1      351 non-null    object
 11  Customer.Town          351 non-null    object
 12  Customer.Country       351 non-null    object
 13  CountryName            351 non-null    object
 14  Country.SalesRegion    351 non-null    object
 15  ModelID                351 non-null    int64
 16  Stock.Cost             351 non-null    int64
 17  Stock.RepairsCost      351 non-null    int64
 18  Stock.PartsCost        351 non-null    int64
 19  Stock.TransportInCost  351 non-null    int64
 20  Stock.Color            351 non-null    object
 21  Stock.DateBought       351 non-null    datetime64[ns]
 22  MakeID                 348 non-null    float64
 23  ModelName              348 non-null    object
 24  MakeName               348 non-null    object
 25  MakeCountry            348 non-null    object
dtypes: datetime64[ns](2), float64(2), int64(11), object(11)
memory usage: 71.4+ KB
```

- ## Get descriptive statistics for numerical columns

```
[33]: df.describe()
```

[33]:

	SalesDetailsID	SalesID	LineItemNumber	SalePrice	LineItemDiscount	SaleDate	Data Sales.ID	CustomerID	ModelID	Stock.Cost	Stock.Repa
count	351.000000	351.000000	351.000000	351.000000	119.000000	351	351.000000	351.000000	351.000000	351.000000	351
mean	176.000000	161.692308	1.096866	62813.247863	2266.596639	2017-06-01 04:49:09.401709568	161.692308	32.142450	49.603989	50227.806268	1568
min	1.000000	1.000000	1.000000	950.000000	25.000000	2015-01-02 08:00:00	1.000000	1.000000	1.000000	760.000000	140
25%	88.500000	81.500000	1.000000	12500.000000	750.000000	2016-09-04 00:00:00	81.500000	15.000000	21.000000	10000.000000	500
50%	176.000000	162.000000	1.000000	39500.000000	1250.000000	2017-06-01 12:21:00	162.000000	28.000000	32.000000	31600.000000	660
75%	263.500000	240.500000	1.000000	79500.000000	2450.000000	2018-04-13 00:00:00	240.500000	46.000000	72.500000	63600.000000	2000
max	351.000000	324.000000	4.000000	395000.000000	60000.000000	2018-12-31 00:00:00	324.000000	87.000000	985.000000	316000.000000	9250
std	101.469207	93.683064	0.365304	73834.447753	5874.488435	NaN	93.683064	21.276605	78.259458	58975.481066	1966

- ## Check for missing values

```
[98]: df.isnull().sum()
```

```
[98]: SalesDetailsID            0
      SalesID                   0
      LineItemNumber            0
      StockID                   0
      SalePrice                 0
      LineItemDiscount        232
      SaleDate                  0
      Data Sales.ID             0
      CustomerID                0
      CustomerName              0
      Customer.Address1         0
      Customer.Town             0
      Customer.Country          0
      CountryName               0
      Country.SalesRegion       0
      ModelID                   0
      Stock.Cost                0
      Stock.RepairsCost         0
      Stock.PartsCost           0
      Stock.TransportInCost     0
      Stock.Color               0
      Stock.DateBought          0
      MakeID                    3
      ModelName                 3
      MakeName                  3
      MakeCountry               3
      dtype: int64
```

- ## Check for duplicate records

```
: print(f"Duplicates: {df.duplicated().sum()}")
```

```
Duplicates: 0
```

4- Data Cleaning to Handle Missing Values, Remove Duplicates, Fix Formatting, and Correct Inconsistencies.

- **Handling Missing Values**

```
df['LineItemDiscount'].fillna(0, inplace=True)
df['MakeCountry'].fillna('Unkown',inplace = True )
df['MakeName'].fillna('Unkown',inplace = True )
df['MakeID'].fillna(0,inplace = True )
df['ModelName'].fillna('Unkown',inplace = True )
```

5- Data Transformation & Processing to Convert dates and Creating new columns (Total Sales , Total Cost , Profit & Profit Percentage)

```
## Convert to datetime
df['SaleDate'] = pd.to_datetime(df['SaleDate'])
df['Stock.DateBought'] = pd.to_datetime(df['Stock.DateBought'])
```

```
#  Calculate total cost
df['Total_Cost'] = df['Stock.Cost'] + df['Stock.RepairsCost'] + df['Stock.PartsCost'] + df['Stock.TransportInCost']
```

```
#  Calculate total sales
df['Total_Sales'] = df['LineItemNumber'] * df['SalePrice']
```

```
#  Calculate profit
df['Profit'] = df['Total_Sales'] - df['Total_Cost']-df['LineItemDiscount']
```

```
#  Calculate profit_percentage
df['Profit_Percentage'] = df.apply(lambda row: row['Profit'] / row['Total_Sales'] if row['Total_Sales'] != 0 else 0, axis=1)
```

Phase 3: Descriptive and Diagnostic Analysis (Python)

Tools : Python

Descriptive Questions (What happened?)

- **What are the total sales ,cost and profit over time ?**

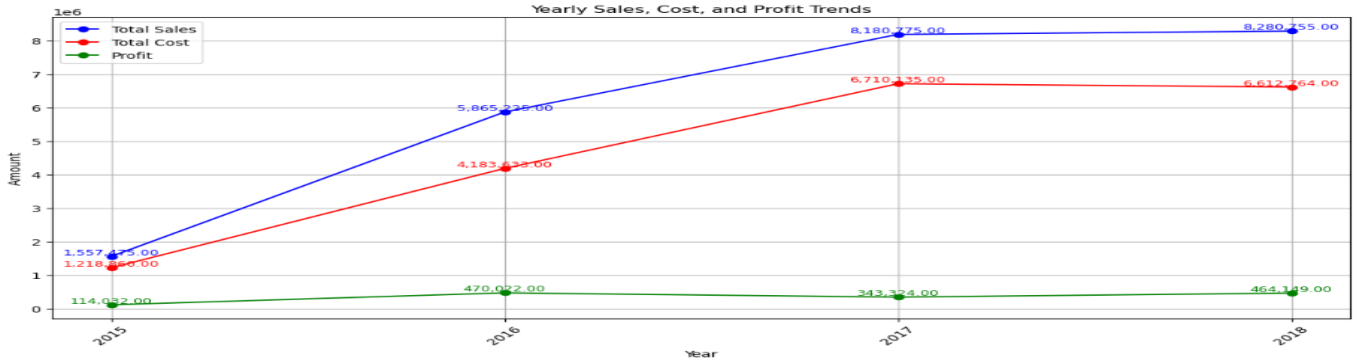
```
df['Month'] = df['SaleDate'].dt.to_period('M')
df['Quarter'] = df['SaleDate'].dt.to_period('Q')
df['Year'] = df['SaleDate'].dt.year
```

```
monthly_summary = df.groupby('Month')[['Total_Sales', 'Total_Cost', 'Profit']].sum()
quarterly_summary = df.groupby('Quarter')[['Total_Sales', 'Total_Cost', 'Profit']].sum()
yearly_summary = df.groupby('Year')[['Total_Sales', 'Total_Cost', 'Profit']].sum()
```

```
print(yearly_summary)
```

```
      Total_Sales  Total_Cost    Profit
Year
2015      1557475     1218860  114032.0
2016      5865225     4183633  470022.0
2017      8180775     6710135  343324.0
2018      8280755     6612764  464149.0
```

```
plt.figure(figsize=(12, 6))

plt.plot(yearly_summary['year'], yearly_summary['Total_Sales'], marker='o', label='Total Sales', color='blue')
plt.plot(yearly_summary['year'], yearly_summary['Total_Cost'], marker='o', label='Total Cost', color='red')
plt.plot(yearly_summary['year'], yearly_summary['Profit'], marker='o', label='Profit', color='green')
for i in range(len(yearly_summary)):
    plt.text(yearly_summary['year'][i], yearly_summary['Total_Sales'][i], f'{yearly_summary["Total_Sales"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='blue')
    plt.text(yearly_summary['year'][i], yearly_summary['Total_Cost'][i], f'{yearly_summary["Total_Cost"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='red')
    plt.text(yearly_summary['year'][i], yearly_summary['Profit'][i], f'{yearly_summary["Profit"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='green')

plt.title('Yearly Sales, Cost, and Profit Trends')
plt.xlabel('Year')
plt.ylabel('Amount')
plt.legend()
plt.grid(True)
plt.xticks(yearly_summary['year'], rotation=45)
plt.tight_layout()
plt.show()
```



Answer :

- **Steady Growth in Total Sales**

Sales increased consistently from 1.57 million in 2015 to a peak of 8.28 million in 2018. (This reflects a strong upward sales trend, with more than a 5x increase over four years.)

- **Cost of Goods Sold (COGS)**

Total costs increased from 1.22 million in 2015 to 4.2 million in 2016 then increased to 6.7 million in 2017 then decreased to 6.61 million in 2018. (This indicates consistent operational scaling to meet rising demand.)

- **Profit Trends and Fluctuations**

Profits grew significantly from 0.3 million in 2015 to 1.6 million in 2016, marking the highest year-on-year growth.

However, in 2017, despite higher sales, profit declined to 1.4 million due to increased costs in 2017.

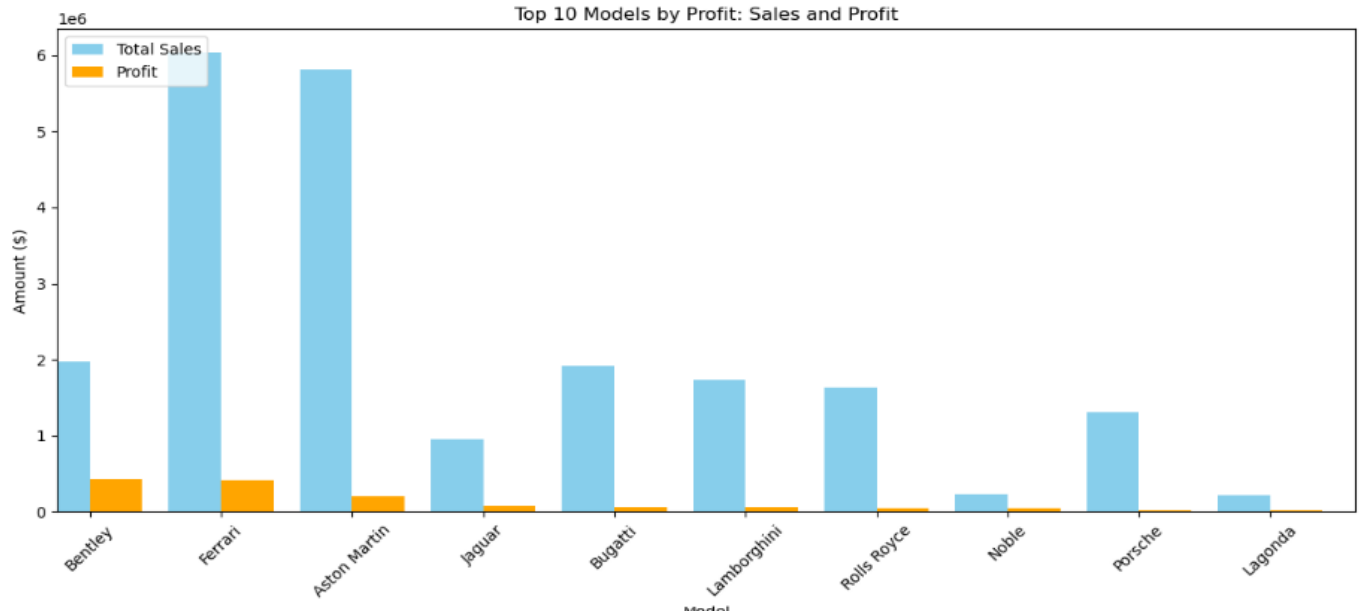
In 2018, profit recovered to 1.6 million, showing signs of operational efficiency and margin improvement.

- **Which car's categories generate the highest sales and profit margins?**

[156]:
```
model_summary = df.groupby('MakeName').agg({
    'Total_Sales': 'sum',
    'Profit': 'sum',
    'Profit_Percentage': 'mean'
}).sort_values(by='Profit', ascending=False)

print(model_summary.head(10))  # Top 10 models
```

```
             Total_Sales    Profit  Profit_Percentage
MakeName
Bentley          1968740  431378.0           0.239685
Ferrari          6043850  417595.0           0.141940
Aston Martin     5808405  204671.0           0.116980
Jaguar            950345   75145.0           0.126876
Bugatti          1915500   59600.0           0.177910
Lamborghini      1727150   58500.0           0.155587
Rolls Royce      1637000   54410.0           0.119815
Noble             236400   46770.0           0.216916
Porsche          1303540   20243.0           0.050096
Lagonda           218000   15850.0           0.101278
```

```
top_10 = model_summary.head(10)
fig, ax1 = plt.subplots(figsize=(12, 6))

# Bar plots for Total_Sales and Profit
top_10['Total_Sales'].plot(kind='bar', color='skyblue', ax=ax1, position=1, width=0.4, label='Total Sales')
top_10['Profit'].plot(kind='bar', color='orange', ax=ax1, position=0, width=0.4, label='Profit')

ax1.set_title('Top 10 Models by Profit: Sales and Profit')
ax1.set_xlabel('Model')
ax1.set_ylabel('Amount ($)')
ax1.legend(loc='upper left')  # Keep this only
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Answer :

Top Performing Car Category by Total Sales:

Ferrari leads with $6.04 million in total sales, followed by Aston Martin at $5.81 million. These two brands significantly outpace others, showing a strong presence in the market.

Mercedes and Noble represent the lowest sales at $484K and $236K, respectively, indicating relatively weaker market performance.

Profit Contribution by Car Category:

Aston Martin stands out with the highest profit of $1.5 million despite having slightly lower sales than Ferrari. This indicates that Bentley is not only selling high-priced vehicles but also maintaining healthy margins.

Ferrari, while leading in total sales, generates $1.3 million in profit, maintaining solid profit despite slightly lower margins compared to Bentley.

Mercedes and Noble show significantly lower profits, with $70 K and $53K respectively, which could be attributed to either lower sales or higher production and operational costs.

- **Recommendations:**

  Aston Martin and Ferrari are the top contenders in both sales and profit, but Bentley's higher profit margin suggests better overall profitability. Strategies that can be considered for Ferrari would include improving operational efficiency or exploring pricing strategies to enhance margins.

- **Brands like Mercedes and Noble, which are underperforming in both profit and margin, may benefit from a detailed cost analysis, reviewing production costs and exploring ways to either increase sales or reduce operational expenses.**

- **Which car models generate the highest sales?**

```
model_summary = df.groupby('ModelName').agg({
    'Total_Sales': 'sum',

}).sort_values(by='Total_Sales', ascending=False)

print(model_summary.head(20))
```

```
             Total_Sales
ModelName
57C              1695000
Virage           1558790
355              1347950
DB6              1120595
DB9              1083500
Enzo             1015000
Flying Spur       958440
Diabolo           905000
Testarossa        870000
F50               760950
911               648050
Dino              565500
Jarama            550000
F40               519500
Phantom           481100
DB2               456440
XK120             374530
360               363000
Vanquish          361040
Silver Seraph     359450
```

Answer :

- **Top Sellers**

Bugatti 57C leads all models with $1.69 million in total sales — a clear favorite among customers, likely due to its luxury appeal and strong market positioning.

Aston Martin Virage follows closely with $1.56M, reinforcing the brand's prestige and customer loyalty in the high-end segment.

Ferrari 355 ($1.35M), Aston Martin DB6 ($1.12M), and DB9 ($1.08M) also rank high, confirming the dominance of British and Italian sports/luxury cars in sales performance.

Models like the Ferrari Enzo ($1.02M), Diabolo ($905K), and Testarossa ($870K) confirm the high demand for exotic sports cars.

Rolls Royce Phantom and Silver Seraph also make the list, highlighting a niche but powerful luxury car segment.

Although models like the Vanquish ($361K) and Silver Seraph ($359K) generate lower sales compared to top sellers, they still reflect meaningful demand in the ultra-luxury segment.

- **Recommendations:**

Prioritize stocking and marketing top performers like the 57C, Virage, and DB models generate the highest ROI.

Highlight the appeal of exotic and collector cars (e.g., Enzo, F50, F40) through targeted campaigns for enthusiasts.

Monitor mid-tier performers for pricing optimization or bundling strategies to boost their market traction.

- **Which car's make country generate the highest sales and profit margins?**

```
Make_summary = df.groupby('MakeCountry').agg({
    'Total_Sales': 'sum',
    'Profit': 'sum'
}).sort_values(by='Total_Sales', ascending=False)

print(Make_summary)
```

```
             Total_Sales    Profit
MakeCountry
GBR              11404110  806998.0
ITA               8274010  489037.0
FRA               2150980   62130.0
GER               1886130   31012.0
USA                 99500       0.0
```

Answer :

GBR dominates both in sales volume and profit generation, indicating high customer demand and strong profit margins for British-manufactured cars. This suggests brand preference and premium positioning.

Italian brands show high sales and healthy profit, making them a key driver of business performance. Likely tied to luxury or sports segments such as Ferrari and Lamborghini.

French brands contribute moderately to sales but show lower profitability. A detailed look into pricing, cost, or product mix is needed to improve margins.

Despite Germany's reputation in automotive excellence, the low profit suggests either high production costs or limited markup, warranting closer investigation.

U.S. brands contribute minimally to sales and offer no profit, possibly due to introductory models, high import costs, or limited market penetration.

- ## Recommendations:

Double down on GBR and Italian brands, which are your most profitable assets.

Review French and German models for pricing strategies or cost inefficiencies to improve margins.

Re-evaluate U.S. brand offerings or market strategy to assess viability or repositioning needs.

- **Which car colors are generating the most sales ?**

```
color_summary = df.groupby('Stock.Color').agg({
    'Total_Sales': 'sum',

}).sort_values(by='Total_Sales', ascending=False)

print(color_summary.head(10))  # Top 10 models
```

```
                      Total_Sales
Stock.Color
Black                     6969875
Silver                    2999135
Red                       2612845
Blue                      2608070
British Racing Green      2502975
Green                     1854930
Night Blue                1521820
Dark Purple               1437315
Canary Yellow              791365
Pink                       585900
```

Answer :

Black is the top-performing color, with $6.97 million in total sales, significantly outpacing other colors. This suggests that Black is a popular choice among customers, likely due to its timeless and versatile appeal.

Silver follows with $2.99 million in sales, indicating that it is also a preferred color choice, albeit at a lower level than Black.

Red and Blue generate substantial sales of $2.61 million and $2.61 million, respectively. These colors tend to appeal to a dynamic and bold customer base, commonly associated with high-performance or luxury models.

British Racing Green also shows notable performance with $2.50 million, which may reflect its appeal in premium or sports car segments.

Canary Yellow and Pink are the lowest-performing colors, with $791K and $586K in sales, respectively. These colors tend to be more niche, appealing to specific customer segments, which may limit their marketability and overall sales.

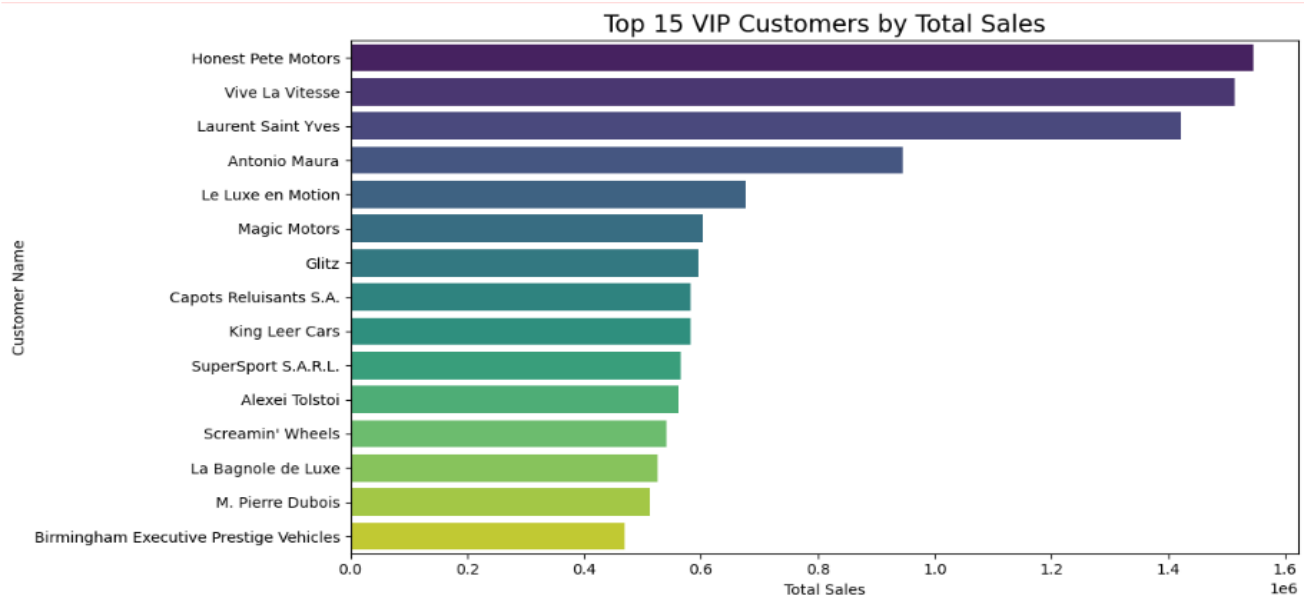
Opportunities for Improvement:

- **While Black remains the dominant color in terms of total sales, bold colors like Red and Blue also contribute significantly to revenue. Car manufacturers might consider expanding their offerings in these popular colors while also exploring strategies to boost sales for niche colors like Canary Yellow and Pink, either through special editions or targeted marketing campaigns.**

- **Who are our top customers ?**

```
customer_summary = df.groupby('CustomerName').agg({
    'Total_Sales': 'sum',
    'Total_Cost': 'sum',
    'Profit': 'sum',
    'Profit_Percentage': 'mean',
    'LineItemNumber': 'count'  # number of purchases
}).rename(columns={'LineItemNumber': 'number_of_purchases'}).sort_values(by='Total_Sales', ascending=False)

print(customer_summary.head(15))  # VIP customers
```

```
                                  Total_Sales  Total_Cost    Profit  \
CustomerName
Honest Pete Motors                    1545640      803442   24660.0
Vive La Vitesse                       1514100     1079875  240075.0
Laurent Saint Yves                    1421450     1129810   43490.0
Antonio Maura                          946330      471429  266203.0
Le Luxe en Motion                      676050      482835   62920.0
Magic Motors                           602850      522630    8415.0
Glitz                                  595490      495502   30838.0
Capots Reluisants S.A.                 583115      495327   36600.0
King Leer Cars                         581580      490649   29951.0
SuperSport S.A.R.L.                    565630      429604   27760.0
Alexei Tolstoi                         560930      447927   58585.0
Screamin' Wheels                       540100      462000   22090.0
La Bagnole de Luxe                     525080      416882    3450.0
M. Pierre Dubois                       511675      436305    -575.0
Birmingham Executive Prestige Vehicles  469740      394152   68153.0

                                  Profit_Percentage  number_of_purchases
CustomerName
Honest Pete Motors                         0.047138                   12
Vive La Vitesse                            0.320988                   13
Laurent Saint Yves                         0.117610                   11
Antonio Maura                              0.231702                    9
Le Luxe en Motion                          0.129418                   11
Magic Motors                               0.064972                    8
Glitz                                      0.137429                    8
Capots Reluisants S.A.                     0.147560                    6
King Leer Cars                             0.089802                    9
SuperSport S.A.R.L.                        0.101505                    9
Alexei Tolstoi                             0.257945                    9
Screamin' Wheels                           0.082405                   15
La Bagnole de Luxe                         0.078924                    6
M. Pierre Dubois                          -0.163923                    8
Birmingham Executive Prestige Vehicles     0.084689                    6
```



Answer :

- **Top Customers by Total Sales:**

  Honest Pete Motors is the leading customer with $1.55 million in total sales, followed closely by Vive La Vitesse with $1.51 million. These customers are driving significant revenue.

Birmingham Executive Prestige Vehicles and La Bagnole de Luxe are at the lower end of sales, contributing only $469K and $525K, respectively. This suggests potential opportunities for increasing engagement or sales with these customers.

- **Top Customers by Profit**

  Honest Pete Motors generates the highest profit at $730 K, which aligns with their high profit margin This indicates efficient cost management and/or higher-priced sales for this customer.

  Antonio Maura also shows strong performance with a profit of $ 472 K.

  Vive La Vitesse also shows strong performance with a profit of $ 431 K.

- **Purchase Frequency**

  Customers like Screamin' Wheels Vive La Vitesse, and Honest Pete Motors make frequent purchases, with 15, 13 and 12 transactions, respectively. This indicates high customer loyalty or ongoing business relationships.

  Capots Reluisants S.A. and La Bagnole de Luxe, with only 6 purchases, are lower in terms of frequency, which may warrant further analysis on whether sales can be increased through personalized offers or incentives.

Opportunities for Improvement:

- **Focus on High-Value Customers (Vive La Vitesse, Antonio Maura and Vive La Vitesse)**

  Vive La Vitesse , Antonio Maura and Vive La Vitesse has high sales and profitability, making them a key target for further relationship-building and maximizing sales opportunities. Consider offering premium packages or additional services to further increase the lifetime value of such customers.

- **Increasing Sales from Lower-Performing Customers**

  Customers like Birmingham Executive Prestige Vehicles and La Bagnole de Luxe are among the lowest performers in terms of total sales. Exploring opportunities to increase engagement or tailor offerings could help drive more revenue from these clients.

- **Which regions or locations have the highest sales and profit margins?**

```
country_summary = df.groupby('CountryName').agg({
    'Total_Sales': 'sum',
    'Profit': 'sum'
}).sort_values(by='Total_Sales', ascending=False)

print(country_summary)  # Top 10 models
```

```
                Total_Sales    Profit
CountryName
United Kingdom     10019915  416905.0
France              6365400  406165.0
Spain               2089060  291096.0
United States       1995620  114498.0
Italy               1215280   27560.0
Switzerland          943065   27320.0
Germany              879040   45988.0
Belgium              376850   61995.0
```

Answer :

- ### Top Revenue Countries

United Kingdom is the top-performing market, generating $10.02 million in total sales and leading in profit with $ 2 million . This highlights it as a core strategic market.

France is the second-highest in revenue ($6.37M) and nearly matches the UK in profit with $1.4 million indicating excellent cost efficiency and healthy margins.

Spain follows as a mid-tier contributor with $2.09M in sales and a noteworthy profit of $0.6 million, suggesting strong returns relative to its size.

High Margin Efficiency

Belgium, despite having the lowest total sales ($0.4 million), yields a surprisingly high profit of($0.1 million) suggesting very efficient operations or premium sales in this market.

Spain and France also show strong profitability ratios, making them high-potential markets to reinforce.

- ### Low Profitability Markets

Italy and Switzerland both generate moderate sales ($1.2M and $943K respectively) but return very low profits ($0.2 million and $0.2 million).

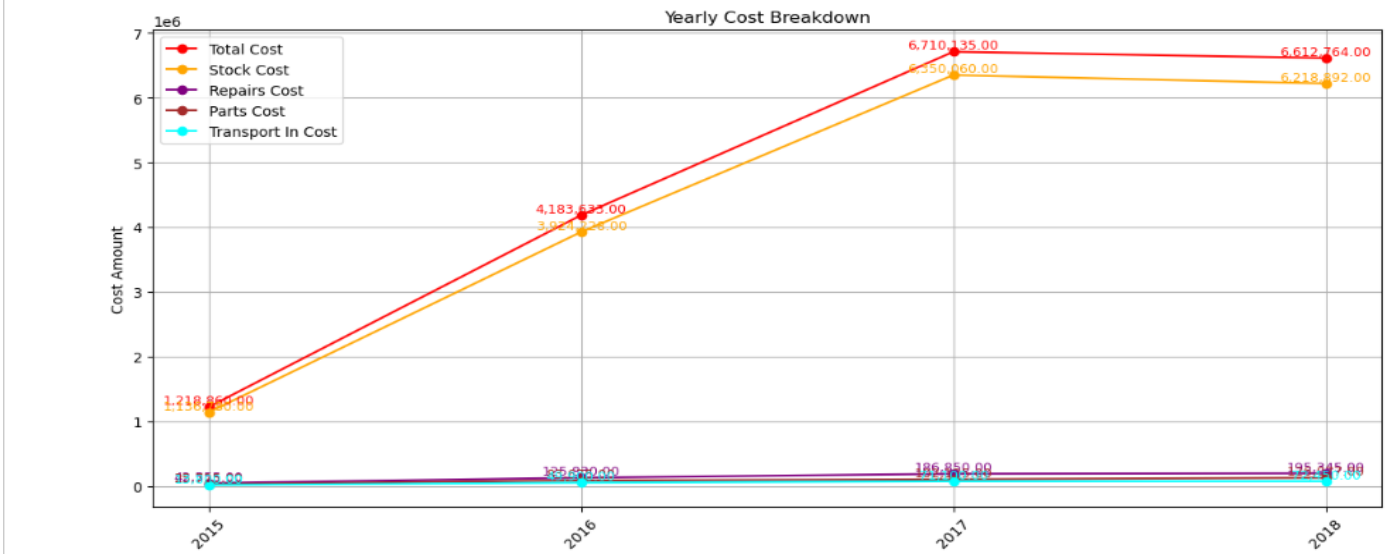
Recommendations:

- Strengthen high-value markets like the UK, France, and Spain by expanding offerings and reinforcing marketing efforts.
- Re-evaluate pricing and cost structures in Italy and Switzerland to improve profitability.
- Explore the Belgium model to replicate its high-margin approach in similar, smaller markets.
- Assess opportunities in the U.S. for expansion while managing costs to improve the profit-to-sales ratio.

- ## Diagnostic Questions (Why did it happen?)

- **Why did profit decrease in 2017?**

```
]: # Create a Yearly Cost Breakdown Graph
plt.figure(figsize=(12, 6))

# Plot the data
plt.plot(yearly_summary['year'], yearly_summary['Total_Cost'], marker='o', label='Total Cost', color='red')
plt.plot(yearly_summary['year'], yearly_summary['Stock.Cost'], marker='o', label='Stock Cost', color='orange')
plt.plot(yearly_summary['year'], yearly_summary['Stock.RepairsCost'], marker='o', label='Repairs Cost', color='purple')
plt.plot(yearly_summary['year'], yearly_summary['Stock.PartsCost'], marker='o', label='Parts Cost', color='brown')
plt.plot(yearly_summary['year'], yearly_summary['Stock.TransportInCost'], marker='o', label='Transport In Cost', color='cyan')

# Add data labels
for i in range(len(yearly_summary)):
    plt.text(yearly_summary['year'][i], yearly_summary['Total_Cost'][i], f'{yearly_summary["Total_Cost"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='red')
    plt.text(yearly_summary['year'][i], yearly_summary['Stock.Cost'][i], f'{yearly_summary["Stock.Cost"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='orange')
    plt.text(yearly_summary['year'][i], yearly_summary['Stock.RepairsCost'][i], f'{yearly_summary["Stock.RepairsCost"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='purple')
    plt.text(yearly_summary['year'][i], yearly_summary['Stock.PartsCost'][i], f'{yearly_summary["Stock.PartsCost"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='brown')
    plt.text(yearly_summary['year'][i], yearly_summary['Stock.TransportInCost'][i], f'{yearly_summary["Stock.TransportInCost"][i]:,.2f}',
             ha='center', va='bottom', fontsize=9, color='cyan')

# Customize the plot
plt.title('Yearly Cost Breakdown')
plt.xlabel('Year')
plt.ylabel('Cost Amount')
plt.legend()
plt.grid(True)
plt.xticks(yearly_summary['year'], rotation=45)
plt.tight_layout()

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



Answer :

Sharp Increase in Total Costs

Sharp increase in total cost from $4.18 million (2016) to $6.71 million (2017) due to the sharp increase in stock cost from $3.9 million (2016) to $6.3 million (2017) and decreased to 6.2 million (2018) due to cost reductions and improvements .

Lower Profit Margins

high-margin vehicles like GBR (Aston Martin , Bently , Rolls Royce and Jaguar) and Italy cars (Ferrari , Lamborghini and Alfa Romeo) which record total sales of 19.7 million and total profit of 4.2 million were sold in 2016 (87 cars with 5.3 million total sales and 1.5 million profit) , in 2017 (decreased to 72 cars with 6.4 million total sales and 1.1 million profit) and 2018 (increased to 96 cars with 6.6 total sales and 1.3 million profit)

The reduction in the number of high-margin units sold in 2017 had a notable impact on overall profitability.

Increased Operational or Marketing Expenses

Investment in expansion, advertising, or staffing may have inflated expenses without immediate ROI

Regional Performance Impact

If growth was driven in regions with lower profit contribution or higher costs (e.g., increased shipping/import costs), overall profit would suffer

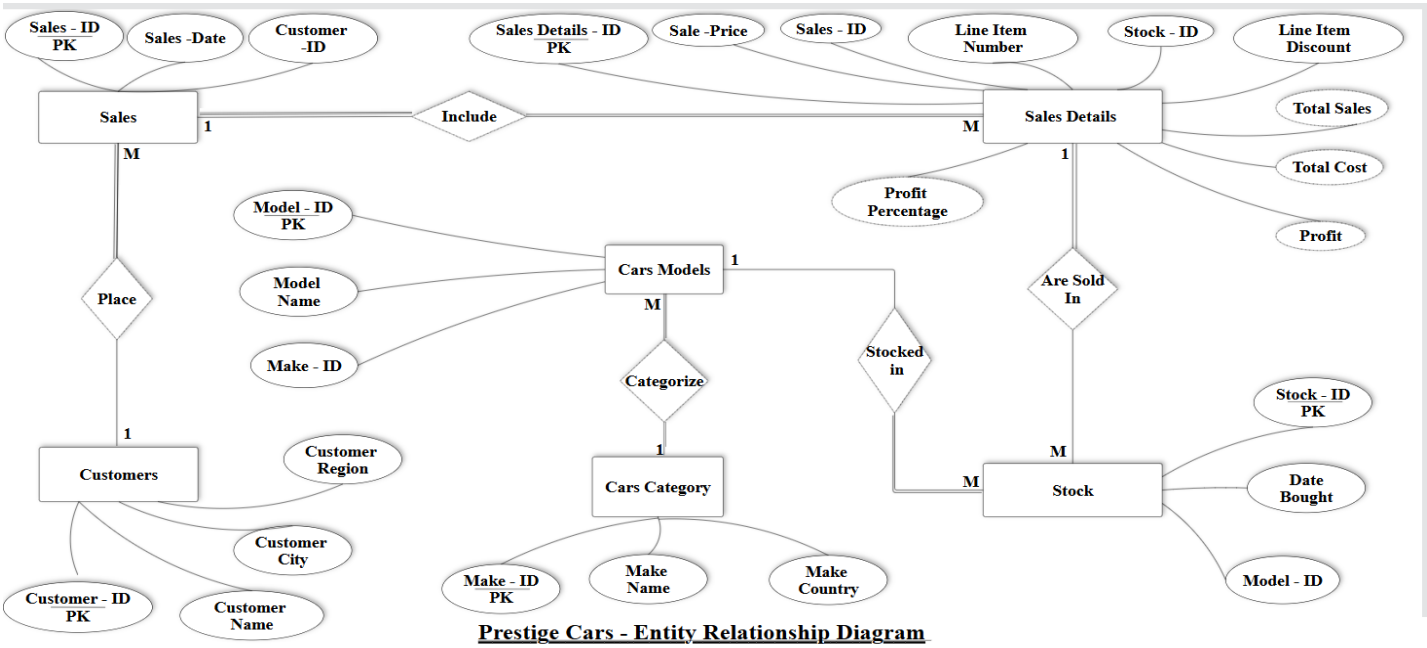
6- Exporting Excel File after analysis From Python

```
[182]: df.to_excel('Processed_Prestige_Cars.xlsx', index=False)
```

Phase 4: SQL Database Integration

Tools : Microsoft SQL Server Management Studio

- **Entity Relationship Diagram**



Prestige Cars - Entity Relationship Diagram

- **Entities and Relationships**

1- Entities:

- **Sales** (Sales-ID, Sale-Date, Customer-ID)
- **Sales-Details** (Sale-Details ID, Sales-ID, Line Item Number, Stock-ID, Sale-Price, Line Item Discount, Total Cost, Total Sales, Profit, Profit Percentages)
- **Customer** (Customer-ID, Customer Name, Customer City, Customer Region)
- **Stock** (Stock-ID, Date Bought, Model-ID)
- **Cars Category** (Make-ID, Make Name, Make Country)
- **Cars Model** (Model-ID, Model Name, Make-ID)

2- Relationships &Participation :

- **Sales to Customer**

  Relationship:

  One customer can have many sales (1:M relationship)

  Participation:

  Customer: Partial (a customer might exist without making a sale)

  Sales: Total (every sale must have a customer)

- **Sales to Sales Details**

  Relationship:

  One sale can have many sales details (1:M relationship)

  Participation:

  Sales: Total (every sale must have at least one detail)

  Sales Details: Total (every detail must belong to a sale)

- **Sales Details to Stock**

  Relationship:

  One stock item can appear in many sales details (1:M relationship)

  Participation:

  Stock: Partial (a stock item might exist without being sold yet)

  Sales Details: Total (every sales detail must reference a stock item)

- **Stock to Cars Model**

  Relationship:

  One car model can have many stock items (1: M relationship)

  Participation:

  Cars Model: Partial (a model might exist without current stock)

  Stock: Total (every stock item must reference a model)

- **Cars Model to Cars Category**

  Relationship:

  One make category can have many models (1: M relationship)

  Participation:

  Cars Category: Total (every make must have at least one model)

  Cars Model: Total (every model must belong to a make)

3- Mapping :

1- Customer

Customer-ID (PK)

Customer Name

Customer City

Customer Region

2- Sales

Sales-ID (PK)

Sale Date

Customer-ID (FK to Customer)

3- Cars Category

Make-ID (PK)

Make Name

Make Country

4- Cars Model

Model-ID (PK)

Model Name

Make-ID (FK to Cars Category)

5- Stock

Stock-ID (PK)

Date Bought

Model-ID (FK to Cars Model)

6- Sales Details

Sale Details-ID (PK)

Sales-ID (FK to Sales)

Line Item Number

Stock-ID (FK to Stock)

Sale Price

Line Item Discount

Total Cost

Total Sales

Profit

Profit Percentage

4- Creating Database :

Prestige Cars Dat...HOSSAM\abdel (52))

```
/* =============== 1) Customer =============== */
CREATE TABLE Customer (
    CustomerID      INT         NOT NULL,
    CustomerName    VARCHAR(100),
    CustomerCity    VARCHAR(100),
    CustomerRegion  VARCHAR(50),
    PRIMARY KEY (CustomerID)
);

-- INSERTS FOR Customers
INSERT INTO Customer (CustomerID, CustomerName, CustomerCity, CustomerRegion) VALUES (1, 'Magic Motors', 'Birmingham', 'EMEA');
INSERT INTO Customer (CustomerID, CustomerName, CustomerCity, CustomerRegion) VALUES (2, 'Snazzy Roadsters', 'Birmingham', 'EMEA');
INSERT INTO Customer (CustomerID, CustomerName, CustomerCity, CustomerRegion) VALUES (3, 'Birmingham Executive Prestige Vehicles', 'Birmingham', 'EMEA');
INSERT INTO Customer (CustomerID, CustomerName, CustomerCity, CustomerRegion) VALUES (4, 'WunderKar', 'Berlin', 'EMEA');
INSERT INTO Customer (CustomerID, CustomerName, CustomerCity, CustomerRegion) VALUES (5, 'Casseroles Chromes', 'Lyon', 'EMEA');
INSERT INTO Customer (CustomerID, CustomerName, CustomerCity, CustomerRegion) VALUES (7, 'Eat My Exhaust Ltd', 'Liverpool', 'EMEA');
```

Prestige Cars Dat...HOSSAM\abdel (52))

```
/* =============== 3) Cars_Model =============== */
CREATE TABLE Cars_Model (
    ModelID     INT         NOT NULL,
    ModelName   VARCHAR(100),
    MakeID      INT         NOT NULL,
    PRIMARY KEY (ModelID),
    FOREIGN KEY (MakeID) REFERENCES Cars_Category(MakeID)
);

-- INSERTS FOR Cars_Model
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (2, 'Testarossa', 1);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (3, '355', 1);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (11, '911', 2);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (12, '924', 2);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (13, '944', 2);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (22, 'DB4', 4);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (23, 'DB5', 4);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (24, 'DB6', 4);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (32, 'Flying Spur', 5);
INSERT INTO Cars_Model (ModelID, ModelName, MakeID) VALUES (45, '280SL', 8);
```

Prestige Cars Dat...HOSSAM\abdel (52))

```
/* =============== 2) Cars_Category =============== */
CREATE TABLE Cars_Category (
    MakeID      INT         NOT NULL,
    MakeName    VARCHAR(100),
    MakeCountry VARCHAR(50),
    PRIMARY KEY (MakeID)
);

-- INSERTS FOR Cars_Category
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (1, 'Ferrari', 'ITA');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (2, 'Porsche', 'GER');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (4, 'Aston Martin', 'GBR');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (5, 'Bentley', 'GBR');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (8, 'Mercedes', 'GER');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (9, 'Alfa Romeo', 'ITA');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (16, 'Jaguar', 'GBR');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (21, 'Triumph', 'GBR');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (3, 'Lamborghini', 'ITA');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (6, 'Rolls Royce', 'GBR');
INSERT INTO Cars_Category (MakeID, MakeName, MakeCountry) VALUES (23, 'Peugeot', 'GER');
```

Prestige Cars Dat...HOSSAM\abdel (52))

```
/* =============== 4) Stock =============== */
CREATE TABLE Stock (
    StockID     VARCHAR(50)   NOT NULL,
    DateBought  DATETIME      NULL,
    ModelID     INT           NOT NULL,
    PRIMARY KEY (StockID),
    FOREIGN KEY (ModelID) REFERENCES Cars_Model(ModelID)
);
-- INSERTS FOR Stock
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('B1C3B95E-3005-4840-8CE3-A7BC5F9CFB3F', '2015-01-01 00:00:00', 2);
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('A2C3B95E-3005-4840-8CE3-A7BC5F9CFB5F', '2015-01-10 00:00:00', 3);
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('558620F5-B9E8-4FFF-8F73-A83FA9559C41', '2015-01-29 00:00:00', 11);
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('72443561-FAC4-4C25-B8FF-0C47361DDE2D', '2015-02-14 00:00:00', 12);
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('2189D556-D1C4-4BC1-B0C8-4053319E8E9D', '2015-01-01 00:00:00', 13);
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('C1459308-7EA5-4A2D-82BC-38079BB4049B', '2015-03-05 00:00:00', 22);
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('E6E6270A-60B0-4817-AA57-17F26B2B8DAF', '2015-03-15 00:00:00', 23);
INSERT INTO Stock (StockID, DateBought, ModelID) VALUES ('CEDFB8D2-BD98-4A08-BC46-406D23940527', '2015-03-26 00:00:00', 24);
```

```sql
/* ================ 5) Sales ================ */
CREATE TABLE Sales (
    SalesID     INT          NOT NULL,
    SaleDate    DATETIME     NULL,
    CustomerID  INT          NOT NULL,
    PRIMARY KEY (SalesID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

-- INSERTS FOR Sales
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (1, '2015-01-02 08:00:00', 1);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (2, '2015-01-25 00:00:00', 2);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (3, '2015-02-03 10:00:00', 3);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (4, '2015-02-16 08:00:00', 4);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (5, '2015-01-02 10:33:00', 5);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (6, '2015-03-14 00:00:00', 1);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (7, '2015-03-24 00:00:00', 3);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (8, '2015-03-30 00:00:00', 7);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (9, '2015-04-06 00:00:00', 8);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (10, '2015-04-04 00:00:00', 9);
INSERT INTO Sales (SalesID, SaleDate, CustomerID) VALUES (11, '2015-04-30 00:00:00', 10);
```

Prestige Cars Dat...HOSSAM\abdel (52))

```sql
/* ============== 6) SalesDetails ============== */
CREATE TABLE SalesDetails (
    SalesDetailsID     INT           NOT NULL,
    SalesID            INT           NOT NULL,
    LineItemNumber     INT           NULL,
    StockID            VARCHAR(50)   NOT NULL,
    SalePrice          DECIMAL(10,2) NULL,
    LineItemDiscount   DECIMAL(10,2) NULL,
    Total_Cost         DECIMAL(10,2) NULL,
    Total_Sales        DECIMAL(10,2) NULL,
    Profit             DECIMAL(10,2) NULL,
    Profit_Percentage  DECIMAL(5,4)  NULL,
    PRIMARY KEY (SalesDetailsID),
    FOREIGN KEY (SalesID) REFERENCES Sales(SalesID),
    FOREIGN KEY (StockID) REFERENCES Stock(StockID)
);
-- INSERTS FOR SalesDetails
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (1,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (2,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (3,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (4,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (5,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (6,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (7,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (8,
INSERT INTO SalesDetails (SalesDetailsID, SalesID, LineItemNumber, StockID, SalePrice, LineItemDiscount, Total_Cost, Total_Sales, Profit, Profit_Percentage) VALUES (9,
```

5- Validation

```sql
SELECT
    YEAR(s.SaleDate) AS SaleYear,
    SUM(sd.Total_Sales) AS [Total Sales]
FROM SalesDetails sd
JOIN Sales s ON sd.SalesID = s.SalesID
GROUP BY YEAR(s.SaleDate)
ORDER BY SaleYear DESC;
```

90 %

Results | Messages

	SaleYear	Total Sales
1	2018	8280755.00
2	2017	8180775.00
3	2016	5865225.00
4	2015	1557475.00

Prestige Cars Dat...HOSSAM\abdel (52))

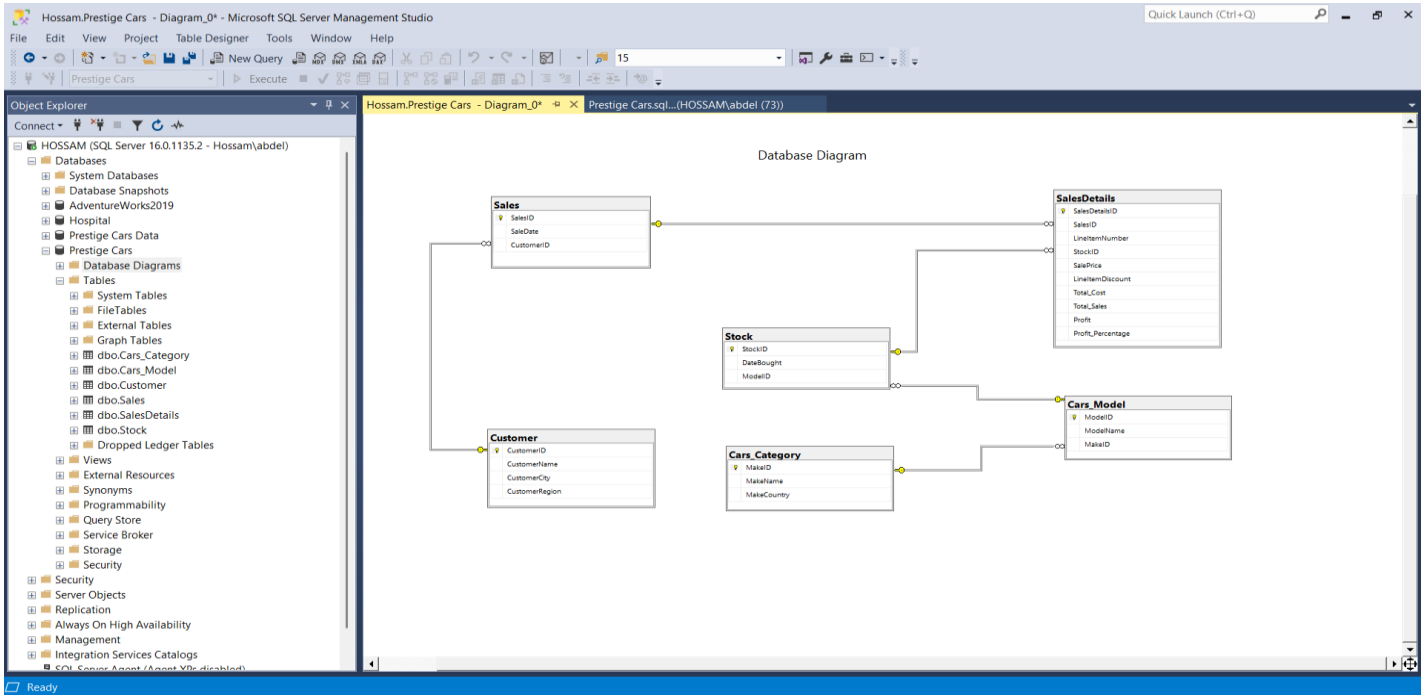
```sql
SELECT TOP 5
    c.CustomerID,
    c.CustomerName,
    SUM(sd.Total_Sales) AS TotalSales
FROM Customer c
JOIN Sales s ON c.CustomerID = s.CustomerID
JOIN SalesDetails sd ON s.SalesID = sd.SalesID
GROUP BY c.CustomerID, c.CustomerName
ORDER BY TotalSales DESC;
```

90 %

Results | Messages

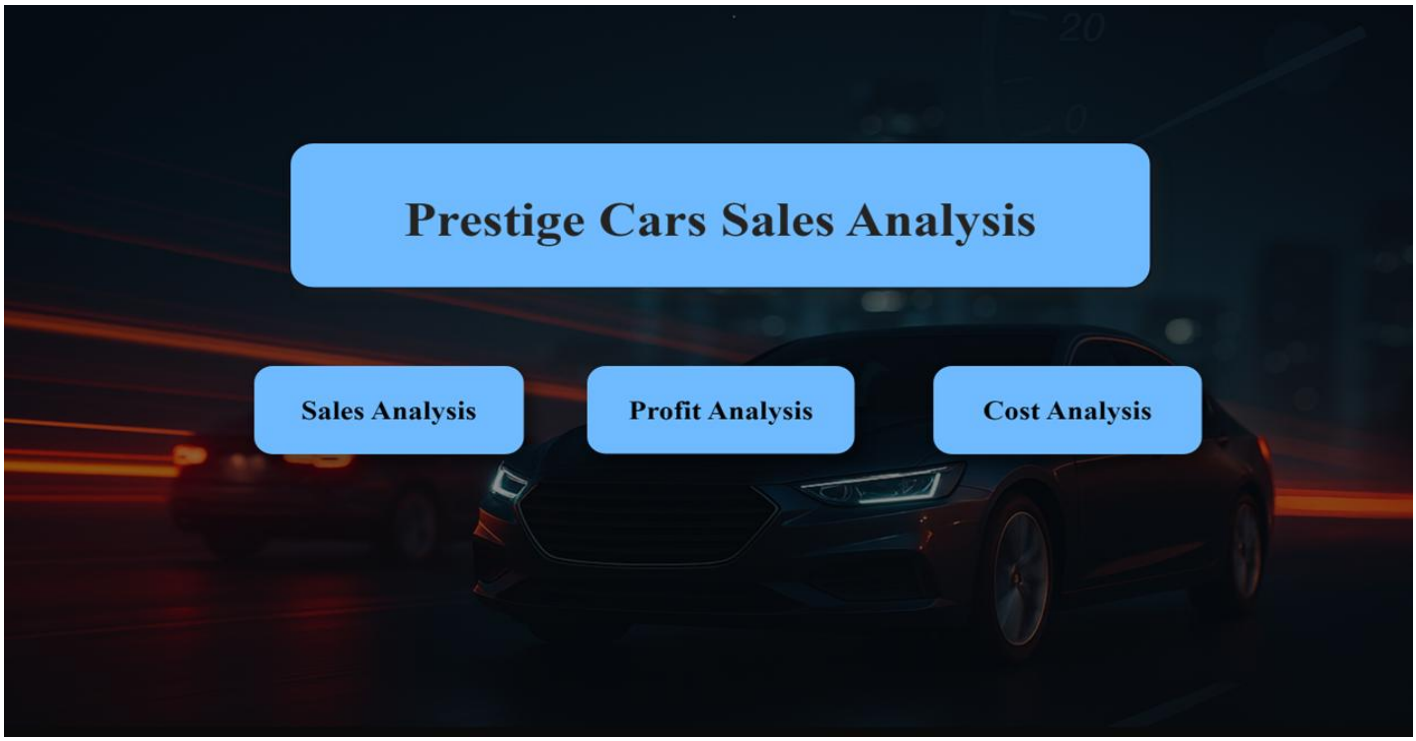
	CustomerID	CustomerName	TotalSales
1	26	Honest Pete Motors	1545640.00
2	28	Vive La Vitesse	1514100.00
3	35	Laurent Saint Yves	1421450.00
4	48	Antonio Maura	946330.00
5	1	Magic Motors	602850.00

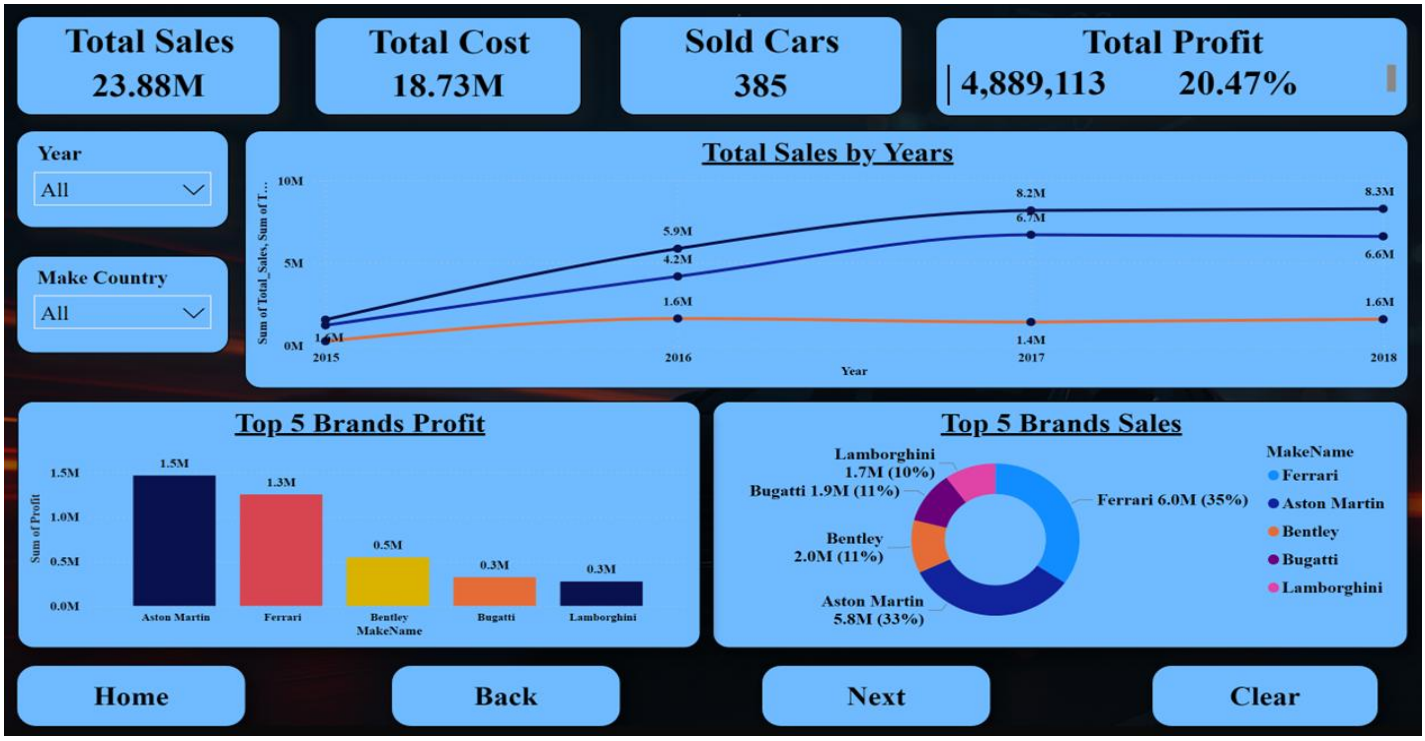
6- Database Diagram

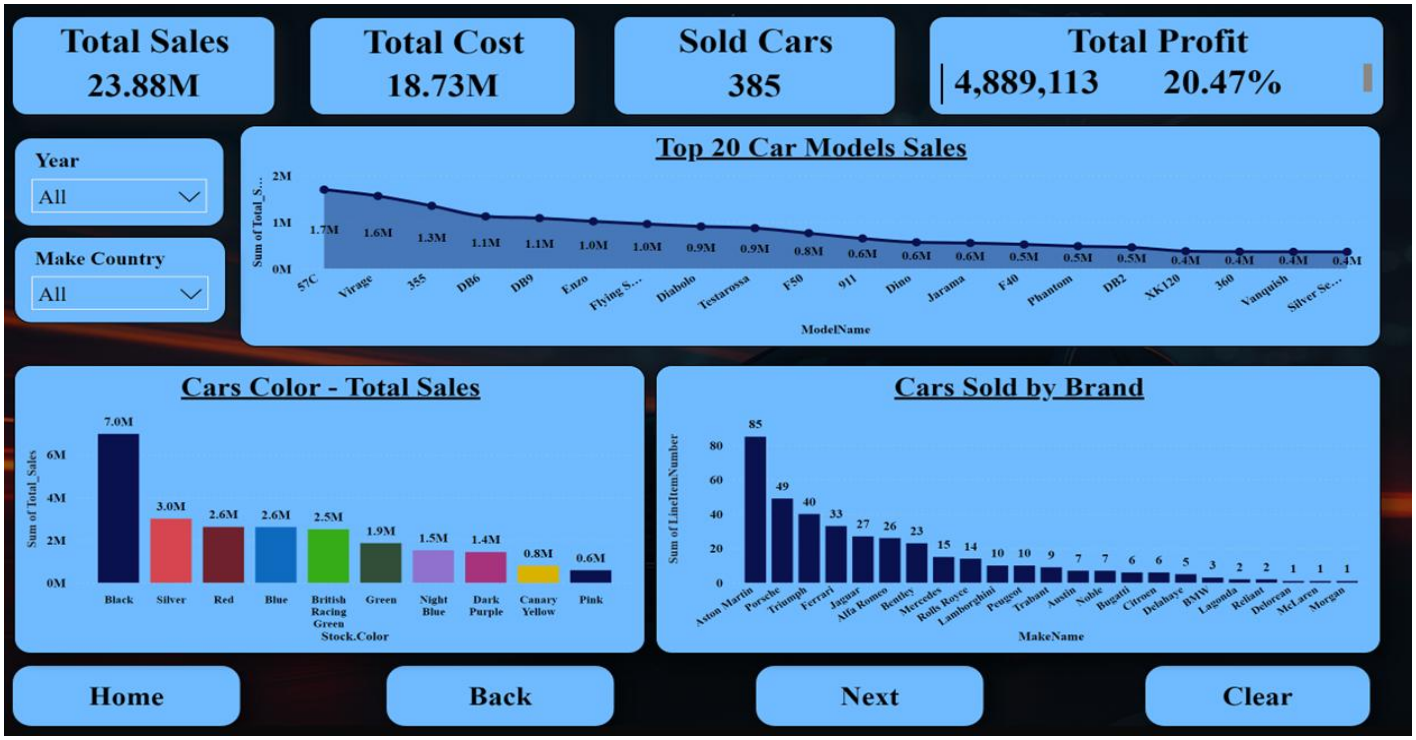


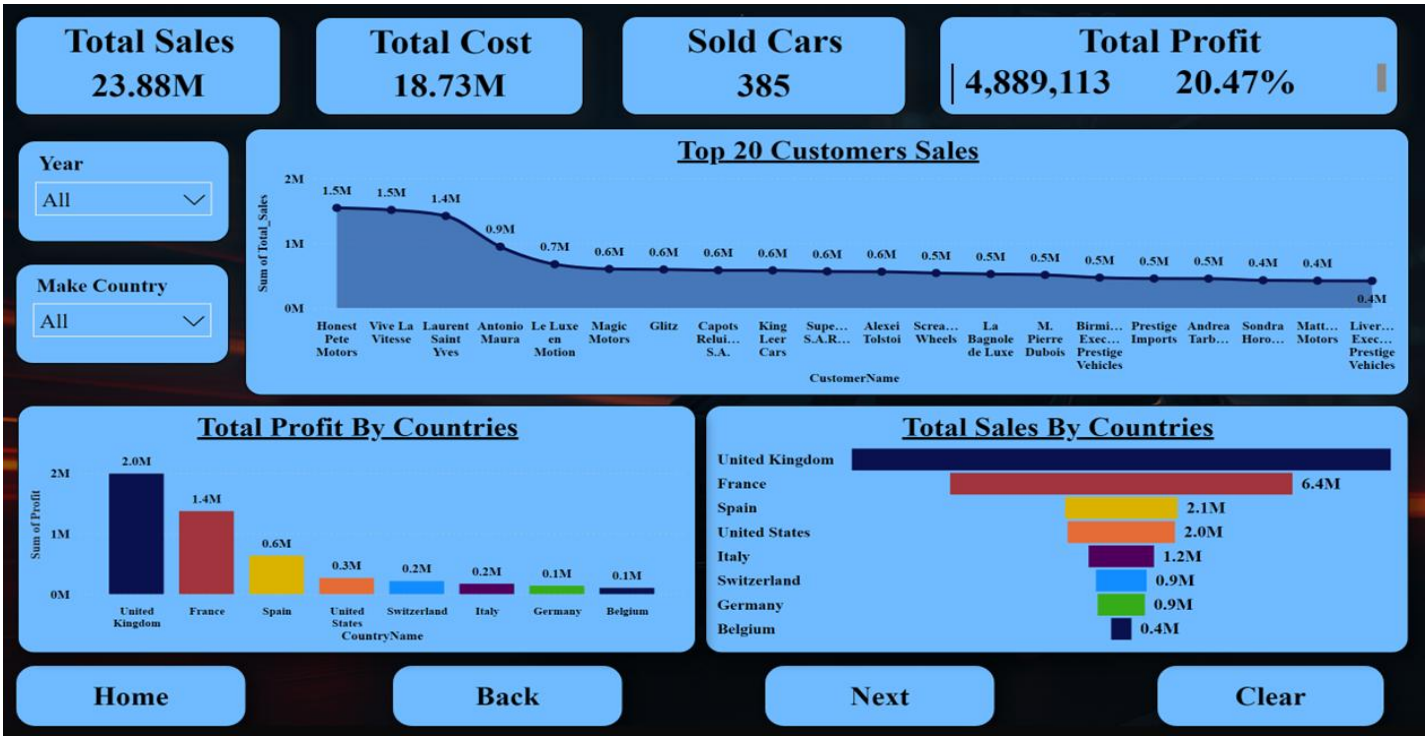
Phase 5: Visualization and Reporting

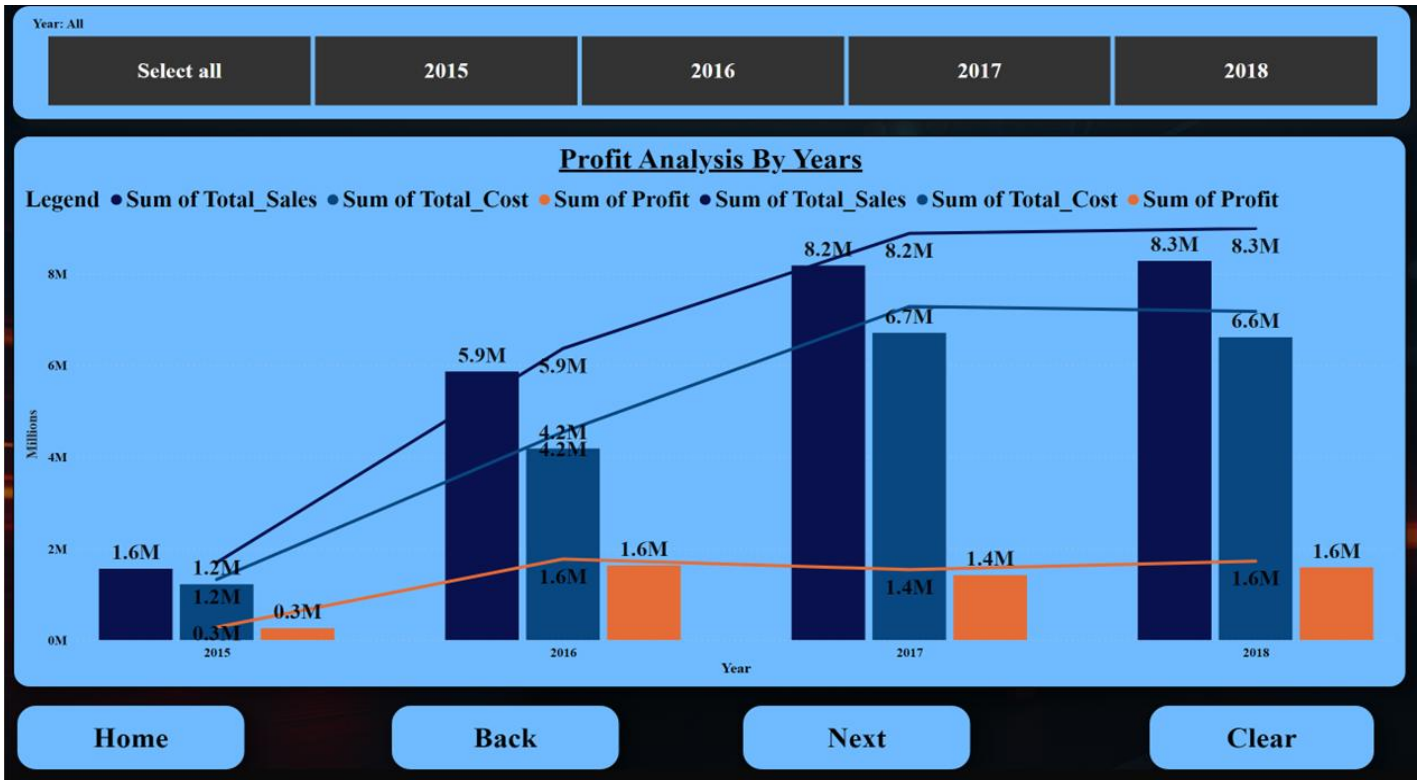
Tools : Power BI

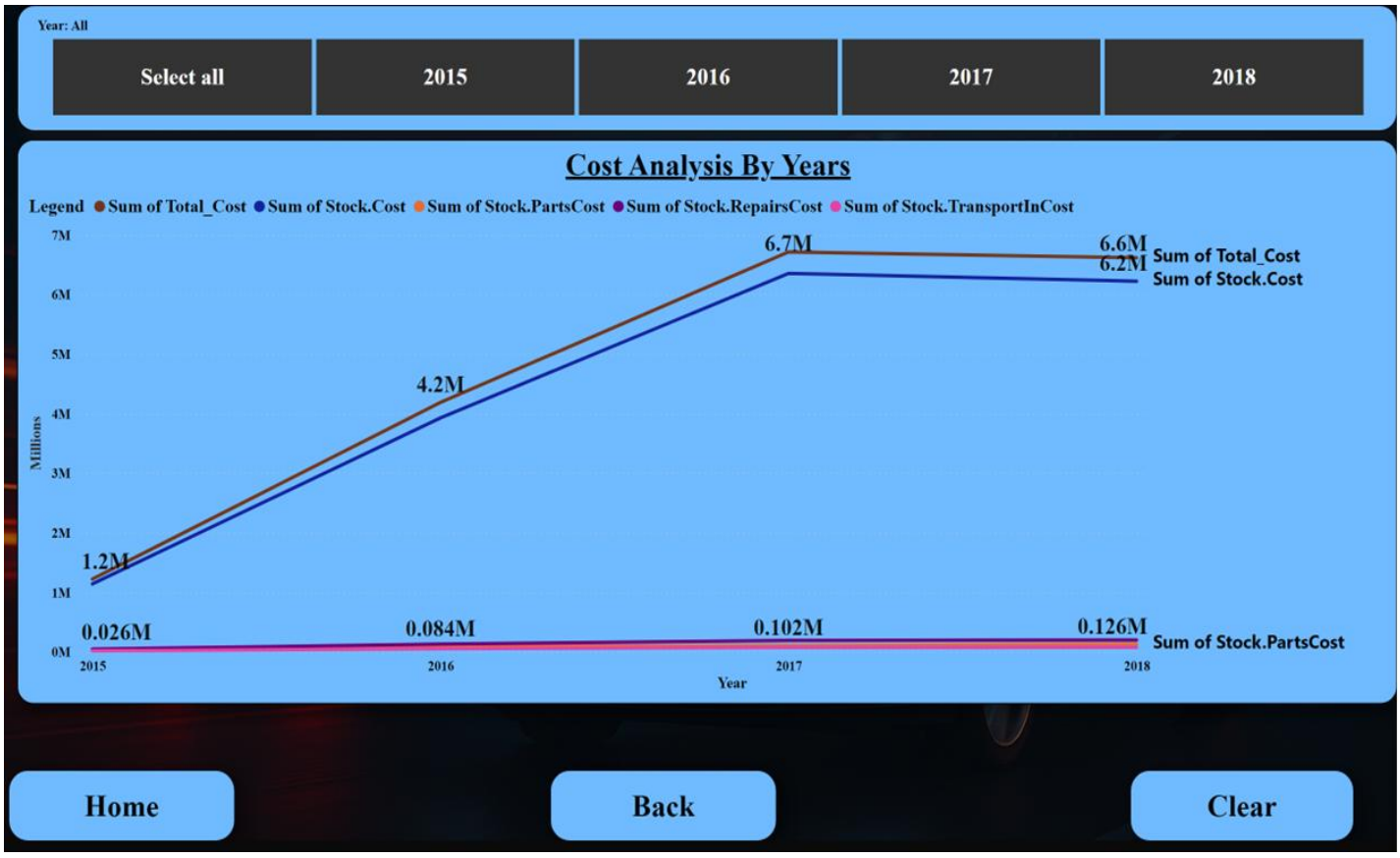





Total Sales
23.88M
Total Cost
18.73M
Sold Cars
385
Total Profit
4,889,113
20.47%
Year
All
Make Country
All
Top 20 Car Models Sales
Sum of Total_S...
2M
1M
0M
1.7M
1.6M
1.3M
1.1M
1.1M
1.0M
1.0M
0.9M
0.9M
0.8M
0.6M
0.6M
0.6M
0.5M
0.5M
0.5M
0.4M
0.4M
0.4M
0.4M
57C
Virage
355
DB6
DB9
Enzo
Flying S...
Diabolo
Testarossa
F50
911
Dino
Jarama
F40
Phantom
DB2
XK120
360
Vanquish
Silver Se...
ModelName
Cars Color - Total Sales
Sum of Total_Sales
7.0M
3.0M
2.6M
2.6M
2.5M
1.9M
1.5M
1.4M
0.8M
0.6M
Black
Silver
Red
Blue
British Racing Green
Green
Night Blue
Dark Purple
Canary Yellow
Pink
Stock.Color
Cars Sold by Brand
Sum of LineItemNumber
85
49
40
33
27
26
23
15
14
10
10
9
7
7
6
6
5
3
2
2
1
1
1
Aston Martin
Porsche
Triumph
Ferrari
Jaguar
Alfa Romeo
Bentley
Mercedes
Rolls Royce
Lamborghini
Peugeot
Trabant
Austin
Noble
Bugatti
Citroen
Delahaye
BMW
Lagonda
Reliant
Delorean
McLaren
Morgan
MakeName
Home
Back
Next
Clear


Total Sales
23.88M
Total Cost
18.73M
Sold Cars
385
Total Profit
4,889,113
20.47%
Year
All
Make Country
All
Top 20 Customers Sales
Sum of Total_Sales
2M
1M
0M
1.5M
1.5M
1.4M
0.9M
0.7M
0.6M
0.6M
0.6M
0.6M
0.6M
0.6M
0.5M
0.5M
0.5M
0.5M
0.5M
0.5M
0.4M
0.4M
0.4M
Honest Pete Motors
Vive La Vitesse
Laurent Saint Yves
Antonio Maura
Le Luxe en Motion
Magic Motors
Glitz
Capots Relui... S.A.
King Leer Cars
Supe... S.A.R...
Alexei Tolstoi
Screa... Wheels
La Bagnole de Luxe
M. Pierre Dubois
Birmi... Exec... Prestige Vehicles
Prestige Imports
Andrea Tarb...
Sondra Horo...
Matt... Motors
Liver... Exec... Prestige Vehicles
CustomerName
Total Profit By Countries
Sum of Profit
2M
1M
0M
2.0M
1.4M
0.6M
0.3M
0.2M
0.2M
0.1M
0.1M
United Kingdom
France
Spain
United States
Switzerland
Italy
Germany
Belgium
CountryName
Total Sales By Countries
United Kingdom
France
6.4M
Spain
2.1M
United States
2.0M
Italy
1.2M
Switzerland
0.9M
Germany
0.9M
Belgium
0.4M
Home
Back
Next
Clear


Year: All
Select all
2015
2016
2017
2018
Profit Analysis By Years
Legend Sum of Total_Sales Sum of Total_Cost Sum of Profit Sum of Total_Sales Sum of Total_Cost Sum of Profit
8M
6M
4M
2M
0M
Millions
1.6M
1.2M
1.2M
0.3M
0.3M
5.9M
5.9M
4.2M
4.2M
1.6M
1.6M
8.2M
8.2M
6.7M
1.4M
1.4M
8.3M
8.3M
6.6M
1.6M
1.6M
2015
2016
2017
2018
Year
Home
Back
Next
Clear


Year: All
Select all
2015
2016
2017
2018
Cost Analysis By Years
Legend Sum of Total_Cost Sum of Stock.Cost Sum of Stock.PartsCost Sum of Stock.RepairsCost Sum of Stock.TransportInCost
7M
6M
5M
4M
3M
2M
1M
0M
Millions
1.2M
4.2M
6.7M
6.6M
6.2M
Sum of Total_Cost
Sum of Stock.Cost
0.026M
0.084M
0.102M
0.126M
Sum of Stock.PartsCost
2015
2016
2017
2018
Year
Home
Back
Clear

Phase 6: Final Presentation & Submission

- Create a structured presentation.
- Rehearse and refine the delivery.
- Submit the final report and supporting materials.

1.5 Task Assignment& Roles

Each team member had a defined role in the project.

- **Data Collection:** Responsible for acquiring and validating data sources.
- **Data Cleaning & Processing:** Ensuring data accuracy and consistency.
- **Analysis & Visualization:** Applying statistical methods and creating dashboards.
- **Report Generation:** Documenting insights and creating presentations.

1.6 Risk Assessment & Mitigation

- **Data Quality Issues:** Addressed by implementing validation techniques.
- **Time Constraints:** Managed by setting milestones and review checkpoints.
- **Technical Challenges:** Resolved through collaboration and research.

1.7 KPIs (Key Performance Indicators)

- **Sales & Revenue KPIs** (Total Sales Revenue , Sales Growth Rate, Sales by Region, Top Selling Models)
- **Profitability KPIs** (Total Profit , Profit Margin , Model Profitability)
- **Inventory/Stock KPIs** (Stock Turnover Rate , Days in Inventory , Unsold Stock Value)
- **Customer KPIs** (Customer Lifetime Value (CLV) , Repeat Purchase Rate , Top Customers by Profit)

2. Literature Review

2.1 Research Background

Store data analysis is widely used in retail to improve decision-making. Previous studies highlight the importance of sales forecasting, inventory management, and customer behavior analysis.

Various methodologies such as machine learning and statistical analysis have been applied to similar projects.

2.2 Feedback & Evaluation

We regularly consulted with our lecturer to get feedback on our approach and methodology. This will help us refine our techniques and ensure our project meets academic and industry standards.

2.3 Suggested Improvements

Based on evaluations, we made necessary adjustments to improve our data processing techniques, visualizations, and overall analysis.

3. Requirements Gathering

3.1 Stakeholder Analysis

- **Sales Manager :** Oversees the sales team and performance
- **Inventory Manager :** Manages stock levels and car availability
- **Finance Department :** Handles budgets, profits, and cost analysis
- **Marketing Team :** Designs campaigns and promotions
- **Management (CEO/COO) :** Strategic decision-makers
- **Customers :** Buyers of prestige cars
- **IT/Data Team :** Maintains the database and reporting systems
- **Car Suppliers / Manufacturers :** Provide vehicles for resale
- **Customer Service :** Supports post-sale queries

3.2 User Stories & Use Cases

- **Sales Agent**

Record a new sale for a customer so that the transaction is tracked in the

Add multiple car items to a single sale so that I can handle bulk purchases.

Apply discounts to line items so I can offer promotional deals.

See the total cost, total sale price, and profit of each sale to quickly assess the transaction value.

- **Sales Manager**

Generate reports showing total sales and profit by car make, model, and country so I can identify high-performing brands.

Track sales by customer region and city to understand market distribution.

Analyze profit percentages over time to identify trends and business growth.

Evaluate which customers made the most purchases to offer them loyalty rewards.

- **Inventory Staff**

Add new stock items by selecting a car model and date bought so I can track inventory accurately.

Update the stock status once a car is sold so that the system reflects availability.

Categorize each car by category (e.g., luxury, sports) so I can organize the catalog.

- **Customer**

View a catalog of available cars by model and make so that I can select the one I want to purchase.

Purchase history recorded so that I can refer back to past transactions.

- **System Admin / Data Analyst**

Calculate the profit and profit percentage for each sale so I can perform ROI analysis.

Ensure that each stock item links correctly to a car model and that each sale is tied to a customer.

3.3 Functional Requirements

The system includes features such as:

- Data import and export capabilities
- Interactive dashboards for visualization
- Automated trend detection and report generation

3.4 Non-functional Requirements

The System:

- **Efficient :** Process large datasets quickly
- **Secure:** Protect data from unauthorized access
- **User-friendly:** Provide an intuitive interface
- **Reliable**: Ensure minimal downtime and high availability

4. References

- https://github.com/Hossame-Abdelhady/Final-Project-Files
- **Data Excel File**
- **Python File**
- **SQL File**
- **Power BI File**
- **Project Documentation**
- **Presentation**