Car Sales Analysis

Introduction

In today's competitive marketplace, businesses must analyze their **sales activities** carefully to maintain profitability and growth.
This project focuses on analyzing **car product sales** using a **MySQL relational database**, designed specifically for streamlined sales tracking.

The database structure is clean and centered around the core business components:

- **Products**: Items available for sale, such as different car models.
- **Product Lines**: Categories or families of products.
- **Orders**: Customer orders placed for various products.
- **Order Details**: Line items specifying quantities and prices for products in each order.

Unlike the broader ClassicModels example, this setup **excludes** extra layers like customer personal data, employee management, offices, and payments — focusing purely on **sales transactions and inventory**.

Database Structure Overview

Table Name	Description
`products`	Catalog of products (e.g., cars, motorcycles, trucks)
`productlines`	Product categories grouping related products
`orders`	Records of customer orders (date, status, order number)
`orderdetails`	Detailed breakdown of each order (product, quantity, price)

This focused structure allows efficient:

- **Sales performance analysis** 📈
- **Product inventory management** 🛒
- **Category-level (product line) analysis** 🏷️
- **Revenue and profitability reporting** 💰

Why MySQL for Car Sales Analysis?

- **Simplicity**: Relational structure is easy to navigate and maintain.

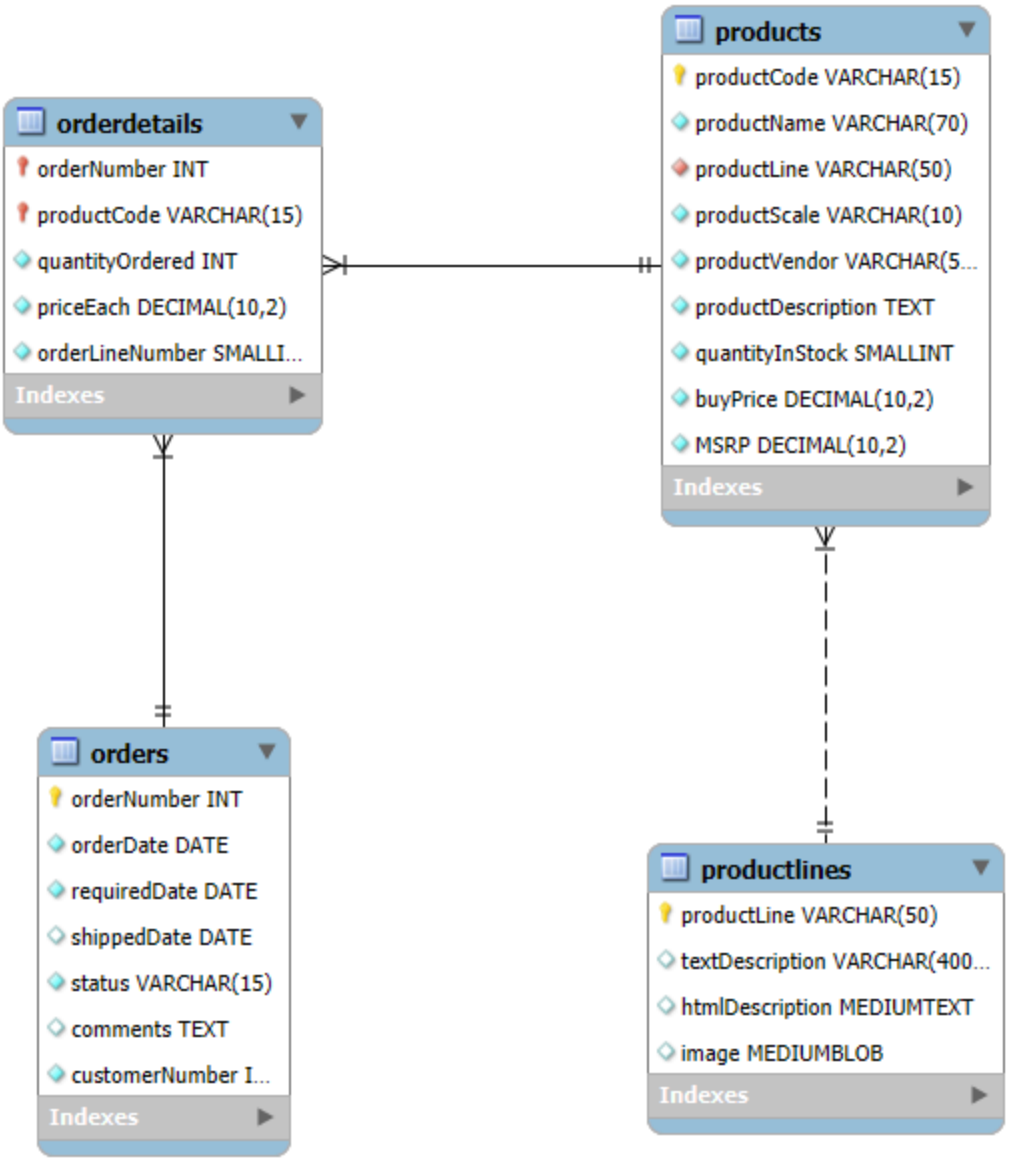
- **Speed**: Quick retrieval of important KPIs (Key Performance Indicators).
- **Expandability**: Easily add more entities (like customers or employees) later.
- **Real-World Simulation**: Ideal training for managing real automotive sales data.

Entity-Relationship Diagram (ERD) Overview

The ERD below shows the structure of the car sales database.
It includes four main tables: **Products**, **Product Lines**, **Orders**, and **Order Details**.

- Each **Product** belongs to a **Product Line**.
- Each **Order** can include multiple **Order Details**.
- Each **Order Detail** links an **Order** to a **Product**.

This design helps track product categories, sales transactions, and revenue efficiently.



Code

In [14]:
```
import mysql.connector
import pandas as pd

# ================================
# 🔍 Database Setup
# ================================

# Establish a connection to the MySQL database
connection = mysql.connector.connect(
    host='localhost',  # Replace with your host
    user='root',  # Replace with your username
    password='******',  # Replace with your password
    database='classicmodels_new'  # Switch to the desired database
)
```

In [16]:
```
# Create a cursor object to execute SQL queries
cursor = connection.cursor()

# Display the existing databases
cursor.execute("SHOW DATABASES;")
databases = cursor.fetchall()
print("Databases:")
for db in databases:
    print(db[0])

# Show all tables in the current database
cursor.execute("SHOW TABLES;")
tables = cursor.fetchall()
print("\nTables:")
for table in tables:
    print(table[0])

# Describe specific tables to understand their structure
tables_to_describe = ['orderdetails', 'orders', 'productlines', 'products']
for table in tables_to_describe:
    cursor.execute(f"DESCRIBE {table};")
    description = cursor.fetchall()

    # Create a DataFrame from the description
    description_df = pd.DataFrame(description, columns=['Field', 'Type', 'Null', 'K

    print(f"\nStructure of {table}:")
    print(description_df)

# ================================
# 🔧 STORED PROCEDURES
# ================================

# 1. Get Order Details by Order Number
def get_order_details(order_number):
    query = """
```

```
    SELECT
        o.orderNumber,
        o.orderDate,
        p.productName,
        od.quantityOrdered,
        od.priceEach,
        (od.quantityOrdered * od.priceEach) AS lineTotal
    FROM
        orders o
    JOIN
        orderdetails od ON o.orderNumber = od.orderNumber
    JOIN
        products p ON od.productCode = p.productCode
    WHERE
        o.orderNumber = %s;
    """
    cursor.execute(query, (order_number,))
    return cursor.fetchall()

# Call the stored procedure with an example order number
order_details = get_order_details(10165)
print("\nOrder Details for Order Number 10165:")
order_details_df = pd.DataFrame(order_details, columns=['Order Number', 'Order Date
print(order_details_df)

# 2. Get Products Below a Certain Stock Level
def get_low_stock_products(stock_threshold):
    query = """
    SELECT
        productCode,
        productName,
        quantityInStock
    FROM
        products
    WHERE
        quantityInStock < %s;
    """
    cursor.execute(query, (stock_threshold,))
    return cursor.fetchall()

# Call the function to see products with low stock
low_stock_products = get_low_stock_products(250)
print("\nLow Stock Products:")
low_stock_products_df = pd.DataFrame(low_stock_products, columns=['Product Code', '
print(low_stock_products_df)

# ================================
# 👁️ VIEWS
# ================================

# 3. Orders Revenue View
cursor.execute("""
CREATE OR REPLACE VIEW view_orders_revenue AS
SELECT
    o.orderNumber,
    o.orderDate,
```

```
    SUM(od.quantityOrdered * od.priceEach) AS totalOrderRevenue
FROM
    orders o
JOIN
    orderdetails od ON o.orderNumber = od.orderNumber
GROUP BY
    o.orderNumber;
""")

# 4. Product Inventory View
cursor.execute("""
CREATE OR REPLACE VIEW view_product_inventory AS
SELECT
    productCode,
    productName,
    quantityInStock,
    buyPrice,
    MSRP
FROM
    products;
""")

# Show all views
cursor.execute("SHOW FULL TABLES WHERE table_type = 'VIEW';")
views = cursor.fetchall()
print("\nViews:")
for view in views:
    print(view[0])

# Display the contents of the views
cursor.execute("SELECT * FROM view_orders_revenue;")
orders_revenue = cursor.fetchall()
print("\nOrders Revenue View:")
orders_revenue_df = pd.DataFrame(orders_revenue, columns=['Order Number', 'Order Da
print(orders_revenue_df)

cursor.execute("SELECT * FROM view_product_inventory;")
product_inventory = cursor.fetchall()
print("\nProduct Inventory View:")
product_inventory_df = pd.DataFrame(product_inventory, columns=['Product Code', 'Pr
print(product_inventory_df)

# ================================
#  💡 USEFUL MYSQL QUERIES
# ================================

# 6. Join Orders and Order Details
cursor.execute("""
SELECT
    orders.orderNumber,
    orders.orderDate,
    orderdetails.productCode,
    orderdetails.quantityOrdered,
    orderdetails.priceEach
FROM orders
JOIN orderdetails ON orders.orderNumber = orderdetails.orderNumber;
```

```
""")
joined_orders_details = cursor.fetchall()
print("\nJoined Orders and Order Details:")
joined_orders_details_df = pd.DataFrame(joined_orders_details, columns=['Order Numb
print(joined_orders_details_df)

# 7. Join Products with Product Lines
cursor.execute("""
SELECT p.productCode, p.productName, pl.productLine
FROM products p
JOIN productlines pl ON p.productLine = pl.productLine;
""")
joined_products_lines = cursor.fetchall()
print("\nJoined Products with Product Lines:")
joined_products_lines_df = pd.DataFrame(joined_products_lines, columns=['Product Co
print(joined_products_lines_df)

# 8. Total Revenue per Order
cursor.execute("""
SELECT
    orderdetails.orderNumber,
    SUM(orderdetails.quantityOrdered * orderdetails.priceEach) AS totalRevenue
FROM orderdetails
GROUP BY orderdetails.orderNumber;
""")
total_revenue_per_order = cursor.fetchall()
print("\nTotal Revenue per Order:")
total_revenue_per_order_df = pd.DataFrame(total_revenue_per_order, columns=['Order
print(total_revenue_per_order_df)

# 9. Orders That Have Not Been Shipped
cursor.execute("SELECT * FROM orders WHERE shippedDate IS NULL;")
unshipped_orders = cursor.fetchall()
print("\nOrders That Have Not Been Shipped:")

# Adjust the column names based on the actual structure of the orders table
unshipped_orders_df = pd.DataFrame(unshipped_orders, columns=['Order Number', 'Orde
print(unshipped_orders_df)

# 10. Orders Within a Specific Date Range
cursor.execute("""
SELECT orderNumber, orderDate, status
FROM orders
WHERE orderDate BETWEEN '2003-01-01' AND '2004-04-26';
""")
orders_in_date_range = cursor.fetchall()
print("\nOrders Within a Specific Date Range:")
orders_in_date_range_df = pd.DataFrame(orders_in_date_range, columns=['Order Number
print(orders_in_date_range_df)

# 11. Basic Product Information
cursor.execute("""
SELECT productCode, productName, quantityInStock, MSRP
FROM products;
""")
basic_product_info = cursor.fetchall()
```

```
print("\nBasic Product Information:")
basic_product_info_df = pd.DataFrame(basic_product_info, columns=['Product Code', '
print(basic_product_info_df)

# 12. Product Line Information
cursor.execute("""
SELECT productLine, textDescription
FROM productlines;
""")
product_line_info = cursor.fetchall()
print("\nProduct Line Information:")
product_line_info_df = pd.DataFrame(product_line_info, columns=['Product Line', 'De
print(product_line_info_df)

# 13. Top 5 Best-Selling Products
cursor.execute("""
SELECT
    p.productName,
    SUM(od.quantityOrdered) AS totalSold
FROM
    products p
JOIN
    orderdetails od ON p.productCode = od.productCode
GROUP BY
    p.productName
ORDER BY
    totalSold DESC
LIMIT 5;
""")
top_selling_products = cursor.fetchall()
print("\nTop 5 Best-Selling Products:")
top_selling_products_df = pd.DataFrame(top_selling_products, columns=['Product Name
print(top_selling_products_df)

# 14. Average Quantity Sold of Each Product
cursor.execute("""
SELECT p.productCode, p.productName, AVG(od.quantityOrdered) AS avgQuantity
FROM products p
JOIN orderdetails od ON p.productCode = od.productCode
GROUP BY p.productCode, p.productName;
""")
average_quantity_sold = cursor.fetchall()
print("\nAverage Quantity Sold of Each Product:")
average_quantity_sold_df = pd.DataFrame(average_quantity_sold, columns=['Product Co
print(average_quantity_sold_df)

# 15. Late Shipped Orders
cursor.execute("""
SELECT
    orderNumber,
    orderDate,
    requiredDate,
    shippedDate
FROM
    orders
WHERE
```

```
    shippedDate > requiredDate
ORDER BY
    shippedDate;
""")
late_shipped_orders = cursor.fetchall()
print("\nLate Shipped Orders:")
late_shipped_orders_df = pd.DataFrame(late_shipped_orders, columns=['Order Number',
print(late_shipped_orders_df)

# 16. Monthly Sales Revenue
cursor.execute("""
SELECT
    DATE_FORMAT(o.orderDate, '%Y-%m') AS orderMonth,
    SUM(od.quantityOrdered * od.priceEach) AS totalRevenue
FROM
    orders o
JOIN
    orderdetails od ON o.orderNumber = od.orderNumber
GROUP BY
    orderMonth
ORDER BY
    orderMonth;
""")
monthly_sales_revenue = cursor.fetchall()
print("\nMonthly Sales Revenue:")
monthly_sales_revenue_df = pd.DataFrame(monthly_sales_revenue, columns=['Order Mont
print(monthly_sales_revenue_df)

# 17. Product Line Info with Descriptions
cursor.execute("""
SELECT
    p.productCode,
    p.productName,
    pl.textDescription
FROM
    products p
LEFT JOIN
    productlines pl ON p.productLine = pl.productLine
ORDER BY
    p.productName;
""")
product_line_descriptions = cursor.fetchall()
print("\nProduct Line Info with Descriptions:")
product_line_descriptions_df = pd.DataFrame(product_line_descriptions, columns=['Pr
print(product_line_descriptions_df)

# 18. Products That Have Never Been Ordered
cursor.execute("""
SELECT
    p.productCode,
    p.productName
FROM
    products p
LEFT JOIN
    orderdetails od ON p.productCode = od.productCode
WHERE
```

```
    od.orderNumber IS NULL;
""")
never_ordered_products = cursor.fetchall()
print("\nProducts That Have Never Been Ordered:")
never_ordered_products_df = pd.DataFrame(never_ordered_products, columns=['Product
print(never_ordered_products_df)

# 19. Average Order Value Calculation
cursor.execute("""
SELECT
    AVG(orderRevenue) AS averageOrderValue
FROM (
    SELECT
        o.orderNumber,
        SUM(od.quantityOrdered * od.priceEach) AS orderRevenue
    FROM
        orders o
    JOIN
        orderdetails od ON o.orderNumber = od.orderNumber
    GROUP BY
        o.orderNumber
) AS revenue_per_order;
""")
average_order_value = cursor.fetchone()
print("\nAverage Order Value:")
average_order_value_df = pd.DataFrame([average_order_value], columns=['Average Orde
print(average_order_value_df)

# 20. Count Orders by Status
cursor.execute("""
SELECT
    status,
    COUNT(*) AS numberOfOrders
FROM
    orders
GROUP BY
    status;
""")
orders_count_by_status = cursor.fetchall()
print("\nCount of Orders by Status:")
orders_count_by_status_df = pd.DataFrame(orders_count_by_status, columns=['Status',
print(orders_count_by_status_df)

# 21. Identify Product Line with Most Products
cursor.execute("""
SELECT
    productLine,
    COUNT(*) AS totalProducts
FROM
    products
GROUP BY
    productLine
ORDER BY
    totalProducts DESC
LIMIT 1;
""")
```

```python
most_products_line = cursor.fetchone()
print("\nProduct Line with Most Products:")
most_products_line_df = pd.DataFrame([most_products_line], columns=['Product Line',
print(most_products_line_df)

# 22. Revenue by Product Line
cursor.execute("""
SELECT
    pl.productLine,
    SUM(od.quantityOrdered * od.priceEach) AS totalRevenue
FROM
    productlines pl
JOIN
    products p ON pl.productLine = p.productLine
JOIN
    orderdetails od ON p.productCode = od.productCode
GROUP BY
    pl.productLine
ORDER BY
    totalRevenue DESC;
""")
revenue_by_product_line = cursor.fetchall()
print("\nRevenue by Product Line:")
revenue_by_product_line_df = pd.DataFrame(revenue_by_product_line, columns=['Produc
print(revenue_by_product_line_df)

# 23. Products with Low Sales Volume
cursor.execute("""
SELECT
    p.productName,
    SUM(od.quantityOrdered) AS totalSold
FROM
    products p
LEFT JOIN
    orderdetails od ON p.productCode = od.productCode
GROUP BY
    p.productName
HAVING
    totalSold < 1500;
""")
low_sales_volume_products = cursor.fetchall()
print("\nProducts with Low Sales Volume:")
low_sales_volume_products_df = pd.DataFrame(low_sales_volume_products, columns=['Pr
print(low_sales_volume_products_df)

# 24. Year-Over-Year Sales Growth
cursor.execute("""
SELECT
    YEAR(o.orderDate) AS salesYear,
    SUM(od.quantityOrdered * od.priceEach) AS totalSales
FROM
    orders o
JOIN
    orderdetails od ON o.orderNumber = od.orderNumber
GROUP BY
    salesYear
```

```
ORDER BY
    salesYear;
""")
year_over_year_sales_growth = cursor.fetchall()
print("\nYear-Over-Year Sales Growth:")
year_over_year_sales_growth_df = pd.DataFrame(year_over_year_sales_growth, columns=
print(year_over_year_sales_growth_df)

# 25. Top Products by Revenue Contribution
cursor.execute("""
SELECT
    p.productName,
    SUM(od.quantityOrdered * od.priceEach) AS totalProductRevenue
FROM
    products p
JOIN
    orderdetails od ON p.productCode = od.productCode
GROUP BY
    p.productName
ORDER BY
    totalProductRevenue DESC
LIMIT 10;
""")
top_products_revenue = cursor.fetchall()
print("\nTop Products by Revenue Contribution:")
top_products_revenue_df = pd.DataFrame(top_products_revenue, columns=['Product Name
print(top_products_revenue_df)

# Close the cursor and connection
cursor.close()
connection.close()
```

```
Databases:
classicmodels
classicmodels_new
information_schema
mydb
mysql
mysql_python
performance_schema
sales
sql_intro
sql_iq
sql_joins
sys
test
triggers
world

Tables:
orderdetails
orders
productlines
products

Structure of orderdetails:
             Field           Type Null  Key Default Extra
0      orderNumber            int   NO  PRI    None
1      productCode    varchar(15)   NO  PRI    None
2  quantityOrdered            int   NO         None
3        priceEach  decimal(10,2)   NO         None
4  orderLineNumber       smallint   NO         None

Structure of orders:
            Field         Type Null  Key Default Extra
0     orderNumber          int   NO  PRI    None
1       orderDate         date   NO         None
2    requiredDate         date   NO         None
3     shippedDate         date  YES         None
4          status  varchar(15)   NO         None
5        comments         text  YES         None
6  customerNumber          int   NO         None

Structure of productlines:
             Field           Type Null  Key Default Extra
0      productLine    varchar(50)   NO  PRI    None
1  textDescription  varchar(4000)  YES         None
2  htmlDescription     mediumtext  YES         None
3            image     mediumblob  YES         None

Structure of products:
                Field           Type Null  Key Default Extra
0         productCode    varchar(15)   NO  PRI    None
1         productName    varchar(70)   NO         None
2         productLine    varchar(50)   NO  MUL    None
3        productScale    varchar(10)   NO         None
4       productVendor    varchar(50)   NO         None
5  productDescription           text   NO         None
```

```
6     quantityInStock       smallint   NO         None
7            buyPrice  decimal(10,2)   NO         None
8                MSRP  decimal(10,2)   NO         None

Order Details for Order Number 10165:
    Order Number  Order Date                           Product Name  \
0          10165  2003-10-22                       2001 Ferrari Enzo
1          10165  2003-10-22                      1969 Corvair Monza
2          10165  2003-10-22                        1969 Ford Falcon
3          10165  2003-10-22                       1957 Chevy Pickup
4          10165  2003-10-22          1998 Chrysler Plymouth Prowler
5          10165  2003-10-22                  1964 Mercedes Tour Bus
6          10165  2003-10-22                   1926 Ford Fire Engine
7          10165  2003-10-22             1992 Ferrari 360 Spider red
8          10165  2003-10-22                Collectable Wooden Train
9          10165  2003-10-22                   1970 Triumph Spitfire
10         10165  2003-10-22                      1970 Dodge Coronet
11         10165  2003-10-22                1962 Volkswagen Microbus
12         10165  2003-10-22     1958 Chevy Corvette Limited Edition
13         10165  2003-10-22       1992 Porsche Cayenne Turbo Silver
14         10165  2003-10-22             1954 Greyhound Scenicruiser
15         10165  2003-10-22  1950's Chicago Surface Lines Streetcar
16         10165  2003-10-22        Diamond T620 Semi-Skirted Tanker
17         10165  2003-10-22          1962 City of Detroit Streetcar

    Quantity Ordered Price Each Line Total
0                 44     168.32    7406.08
1                 34     123.89    4212.26
2                 27     152.26    4111.02
3                 48     109.02    5232.96
4                 29     134.26    3893.54
5                 46     120.28    5532.88
6                 31      60.77    1883.87
7                 47     154.10    7242.70
8                 50      84.71    4235.50
9                 28     123.51    3458.28
10                25      46.82    1170.50
11                32     117.57    3762.24
12                27      31.12     840.24
13                24     106.45    2554.80
14                48      50.86    2441.28
15                44      55.30    2433.20
16                48     106.49    5111.52
17                38      49.21    1869.98

Low Stock Products:
  Product Code              Product Name  Quantity In Stock
0     S12_1099         1968 Ford Mustang                 68
1     S24_2000  1960 BSA Gold Star DBD34                 15
2     S32_1374          1997 BMW F650 ST                178
3     S32_4289  1928 Ford Phaeton Deluxe                136

Views:
view_orders_revenue
view_product_inventory
```

```
Orders Revenue View:
     Order Number  Order Date Total Order Revenue
0           10100  2003-01-06            10223.83
1           10101  2003-01-09            10549.01
2           10102  2003-01-10             5494.78
3           10103  2003-01-29            50218.95
4           10104  2003-01-31            40206.20
..            ...         ...                 ...
321         10421  2005-05-29             7639.10
322         10422  2005-05-30             5849.44
323         10423  2005-05-30             8597.73
324         10424  2005-05-31            29310.30
325         10425  2005-05-31            41623.44

[326 rows x 3 columns]

Product Inventory View:
    Product Code                           Product Name  Quantity In Stock   \
0       S10_1678  1969 Harley Davidson Ultimate Chopper                7933
1       S10_1949               1952 Alpine Renault 1300                7305
2       S10_2016                  1996 Moto Guzzi 1100i                6625
3       S10_4698   2003 Harley-Davidson Eagle Drag Bike                5582
4       S10_4757                     1972 Alfa Romeo GTA                3252
..           ...                                     ...                 ...
105    S700_3505                            The Titanic                1956
106    S700_3962                        The Queen Mary                 5088
107    S700_4002              American Airlines: MD-11S                8820
108     S72_1253                       Boeing X-32A JSF                4857
109     S72_3212                             Pont Yacht                 414

    Buy Price    MSRP
0       48.81   95.70
1       98.58  214.30
2       68.99  118.94
3       91.02  193.66
4       85.68  136.00
..        ...     ...
105     51.09  100.17
106     53.63   99.31
107     36.27   74.03
108     32.77   49.66
109     33.30   54.60

[110 rows x 5 columns]

Joined Orders and Order Details:
      Order Number  Order Date Product Code  Quantity Ordered Price Each
0            10100  2003-01-06     S18_1749                30     136.00
1            10100  2003-01-06     S18_2248                50      55.09
2            10100  2003-01-06     S18_4409                22      75.46
3            10100  2003-01-06     S24_3969                49      35.29
4            10101  2003-01-09     S18_2325                25     108.06
...            ...         ...          ...               ...        ...
2991         10425  2005-05-31     S24_2300                49     127.79
2992         10425  2005-05-31     S24_2840                31      31.82
2993         10425  2005-05-31     S32_1268                41      83.79
```

```
2994         10425  2005-05-31     S32_2509                11      50.32
2995         10425  2005-05-31     S50_1392                18      94.92

[2996 rows x 5 columns]

Joined Products with Product Lines:
    Product Code                      Product Name  Product Line
0       S10_1949          1952 Alpine Renault 1300  Classic Cars
1       S10_4757               1972 Alfa Romeo GTA  Classic Cars
2       S10_4962            1962 LanciaA Delta 16V  Classic Cars
3       S12_1099                 1968 Ford Mustang  Classic Cars
4       S12_1108                 2001 Ferrari Enzo  Classic Cars
..           ...                               ...           ...
105     S24_3816          1940 Ford Delivery Sedan  Vintage Cars
106     S24_3969  1936 Mercedes Benz 500k Roadster  Vintage Cars
107     S24_4258             1936 Chrysler Airflow  Vintage Cars
108     S32_4289          1928 Ford Phaeton Deluxe  Vintage Cars
109     S50_1341      1930 Buick Marquette Phaeton  Vintage Cars

[110 rows x 3 columns]

Total Revenue per Order:
     Order Number Total Revenue
0           10100      10223.83
1           10101      10549.01
2           10102       5494.78
3           10103      50218.95
4           10104      40206.20
..            ...           ...
321         10421       7639.10
322         10422       5849.44
323         10423       8597.73
324         10424      29310.30
325         10425      41623.44

[326 rows x 2 columns]

Orders That Have Not Been Shipped:
    Order Number  Order Date Required Date Shipped Date      Status  \
0          10167  2003-10-23    2003-10-30         None   Cancelled
1          10248  2004-05-07    2004-05-14         None   Cancelled
2          10260  2004-06-16    2004-06-22         None   Cancelled
3          10262  2004-06-24    2004-07-01         None   Cancelled
4          10334  2004-11-19    2004-11-28         None     On Hold
5          10401  2005-04-03    2005-04-14         None     On Hold
6          10407  2005-04-22    2005-05-04         None     On Hold
7          10414  2005-05-06    2005-05-13         None     On Hold
8          10420  2005-05-29    2005-06-07         None  In Process
9          10421  2005-05-29    2005-06-06         None  In Process
10         10422  2005-05-30    2005-06-11         None  In Process
11         10423  2005-05-30    2005-06-05         None  In Process
12         10424  2005-05-31    2005-06-08         None  In Process
13         10425  2005-05-31    2005-06-07         None  In Process

                                     Customer Number  Other Column 1
0   Customer called to cancel. The warehouse was n...             448
```

```
1   Order was mistakenly placed. The warehouse not...            131
2   Customer heard complaints from their customers...            357
3   This customer found a better offer from one of...            141
4   The outstaniding balance for this customer exc...            144
5   Customer credit limit exceeded. Will ship when...            328
6   Customer credit limit exceeded. Will ship when...            450
7   Customer credit limit exceeded. Will ship when...            362
8                                                 None            282
9   Custom shipping instructions were sent to ware...            124
10                                                None            157
11                                                None            314
12                                                None            141
13                                                None            119

Orders Within a Specific Date Range:
     Order Number  Order Date   Status
0           10100  2003-01-06  Shipped
1           10101  2003-01-09  Shipped
2           10102  2003-01-10  Shipped
3           10103  2003-01-29  Shipped
4           10104  2003-01-31  Shipped
..            ...         ...      ...
142         10242  2004-04-20  Shipped
143         10243  2004-04-26  Shipped
144         10300  2003-10-04  Shipped
145         10301  2003-10-05  Shipped
146         10302  2003-10-06  Shipped

[147 rows x 3 columns]

Basic Product Information:
    Product Code                          Product Name  Quantity In Stock  \
0       S10_1678  1969 Harley Davidson Ultimate Chopper               7933
1       S10_1949               1952 Alpine Renault 1300               7305
2       S10_2016                  1996 Moto Guzzi 1100i               6625
3       S10_4698   2003 Harley-Davidson Eagle Drag Bike               5582
4       S10_4757                     1972 Alfa Romeo GTA               3252
..           ...                                     ...                ...
105    S700_3505                           The Titanic               1956
106    S700_3962                        The Queen Mary               5088
107    S700_4002              American Airlines: MD-11S               8820
108     S72_1253                       Boeing X-32A JSF               4857
109     S72_3212                             Pont Yacht                414

       MSRP
0     95.70
1    214.30
2    118.94
3    193.66
4    136.00
..      ...
105  100.17
106   99.31
107   74.03
108   49.66
109   54.60
```

```
[110 rows x 4 columns]

Product Line Information:
       Product Line                                        Description
0      Classic Cars  Attention car enthusiasts: Make your wildest c...
1       Motorcycles  Our motorcycles are state of the art replicas ...
2            Planes  Unique, diecast airplane and helicopter replic...
3             Ships  The perfect holiday or anniversary gift for ex...
4            Trains  Model trains are a rewarding hobby for enthusi...
5  Trucks and Buses  The Truck and Bus models are realistic replica...
6      Vintage Cars  Our Vintage Car models realistically portray a...

Top 5 Best-Selling Products:
                                Product Name Total Sold
0              1992 Ferrari 360 Spider red       1808
1                     1937 Lincoln Berline       1111
2                American Airlines: MD-11S       1085
3  1941 Chevrolet Special Deluxe Cabriolet       1076
4             1930 Buick Marquette Phaeton       1074

Average Quantity Sold of Each Product:
    Product Code                           Product Name Average Quantity
0       S10_1678  1969 Harley Davidson Ultimate Chopper          37.7500
1       S10_1949               1952 Alpine Renault 1300          34.3214
2       S10_2016                  1996 Moto Guzzi 1100i          35.6786
3       S10_4698   2003 Harley-Davidson Eagle Drag Bike          35.1786
4       S10_4757                     1972 Alfa Romeo GTA          36.7857
..           ...                                     ...              ...
104    S700_3505                            The Titanic          35.2593
105    S700_3962                        The Queen Mary          33.1852
106    S700_4002              American Airlines: MD-11S          38.7500
107     S72_1253                       Boeing X-32A JSF          34.2857
108     S72_3212                             Pont Yacht          35.4815

[109 rows x 3 columns]

Late Shipped Orders:
   Order Number  Order Date Required Date Shipped Date
0         10165  2003-10-22    2003-10-31   2003-12-26

Monthly Sales Revenue:
   Order Month Total Revenue
0      2003-01     116692.77
1      2003-02     128403.64
2      2003-03     160517.14
3      2004-04     185848.59
4      2003-05     179435.55
5      2003-06     150470.77
6      2003-07     201940.36
7      2003-08     178257.11
8      2003-09     236697.85
9      2003-10     514336.21
10     2003-11     988025.15
11     2003-12     276723.25
12     2004-01     292385.21
```

```
13     2004-02     289502.84
14     2004-03     217691.26
15     2004-04     187575.77
16     2004-05     248325.30
17     2004-06     343370.74
18     2004-07     325563.49
19     2004-08     419327.09
20     2004-09     283799.80
21     2004-10     500233.86
22     2004-11     979291.98
23     2004-12     428838.17
24     2005-01     307737.02
25     2005-02     317192.17
26     2005-03     359711.96
27     2005-04     344820.62
28     2005-05     441474.94

Product Line Info with Descriptions:
    Product Code                          Product Name  \
0       S24_2011                18th century schooner
1       S18_3136  18th Century Vintage Horse Carriage
2       S24_2841               1900s Vintage Bi-Plane
3       S24_4278              1900s Vintage Tri-Plane
4       S18_3140                    1903 Ford Model A
..           ...                                   ...
105    S700_1938                       The Mayflower
106    S700_3962                      The Queen Mary
107    S700_1138                The Schooner Bluenose
108    S700_3505                          The Titanic
109    S700_2610            The USS Constitution Ship

                                           Description
0    The perfect holiday or anniversary gift for ex...
1    Our Vintage Car models realistically portray a...
2    Unique, diecast airplane and helicopter replic...
3    Unique, diecast airplane and helicopter replic...
4    Our Vintage Car models realistically portray a...
..                                                ...
105  The perfect holiday or anniversary gift for ex...
106  The perfect holiday or anniversary gift for ex...
107  The perfect holiday or anniversary gift for ex...
108  The perfect holiday or anniversary gift for ex...
109  The perfect holiday or anniversary gift for ex...

[110 rows x 3 columns]

Products That Have Never Been Ordered:
  Product Code       Product Name
0     S18_3233  1985 Toyota Supra

Average Order Value:
  Average Order Value
0        29460.707393

Count of Orders by Status:
       Status  Number of Orders
```

```
0     Shipped               303
1    Resolved                 4
2   Cancelled                 6
3     On Hold                 4
4    Disputed                 3
5  In Process                 6

Product Line with Most Products:
   Product Line  Total Products
0  Classic Cars              38

Revenue by Product Line:
       Product Line Total Revenue
0      Classic Cars    3853922.49
1      Vintage Cars    1797559.63
2       Motorcycles    1121426.12
3  Trucks and Buses    1024113.57
4            Planes     954637.54
5             Ships     663998.34
6            Trains     188532.92

Products with Low Sales Volume:
                              Product Name Total Sold
0    1969 Harley Davidson Ultimate Chopper       1057
1                 1952 Alpine Renault 1300        961
2                    1996 Moto Guzzi 1100i        999
3     2003 Harley-Davidson Eagle Drag Bike        985
4                      1972 Alfa Romeo GTA       1030
..                                     ...        ...
103                          The Titanic        952
104                        The Queen Mary        896
105              American Airlines: MD-11S       1085
106                       Boeing X-32A JSF        960
107                             Pont Yacht        958

[108 rows x 2 columns]

Year-Over-Year Sales Growth:
   Sales Year Total Sales
0        2003  3317348.39
1        2004  4515905.51
2        2005  1770936.71

Top Products by Revenue Contribution:
                           Product Name Total Product Revenue
0           1992 Ferrari 360 Spider red             276839.98
1                     2001 Ferrari Enzo             190755.86
2              1952 Alpine Renault 1300             190017.96
3  2003 Harley-Davidson Eagle Drag Bike             170686.00
4                     1968 Ford Mustang             161531.48
5                      1969 Ford Falcon             152543.02
6           1980s Black Hawk Helicopter             144959.91
7        1998 Chrysler Plymouth Prowler             142530.63
8              1917 Grand Touring Sedan             140535.60
9                      2002 Suzuki XREO             135767.03
```

Reference

1. prof3ssorSt3v3. (n.d.). *ClassicModels Database Schema* [SQL script]. GitHub Gist. Retrieved July 15, 2024, from https://gist.github.com/prof3ssorSt3v3/796ebc82fd8eeb0b697effaa1e86c3a6