DATA ANALYSIS AND EXPLORATION USING SQL (MYSQL WORKBENCH) AND POWER BI.

A small company Axon, which is a retailer selling classic cars, is facing issues in managing and analyzing their sales data. The sales team is struggling to make sense of the data and they do not have a centralized system to manage and analyze the data. The management is unable to get accurate and up-to-date sales reports, which is affecting the decision-making process.

To address this issue, the company has decided to implement a Business Intelligence (BI) tool that can help them manage and analyze their sales data effectively. They have shortlisted Microsoft PowerBI and SQL as the BI tools for this project.

The goal of the capstone project is to design and implement a BI solution using PowerBI and SQL that can help the company manage and analyze their sales data effectively.

DATA SOURCE

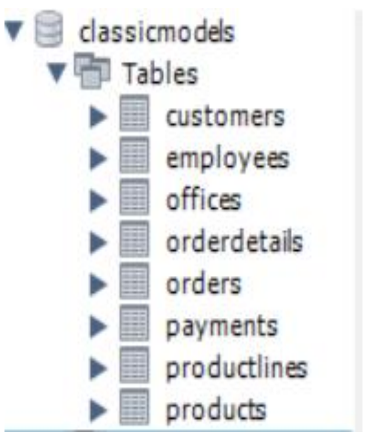
The database is gotten from the link: https://drive.google.com/file/d/1OB_iGw6vVS5KS7QwiwVChbeTfR4WvUy3/view?usp=share_link

and it contained several tables such as customers, employees, offices, orders, orderdetails, payments, products and productlines.

IMPORTING THE DATABASE INTO MYSQL WORKBENCH

I imported the database into MySQL workbench. After successfully importing the database, the tables were visible which means I can start exploring and analyzing the database.

CLASSICMODELS DATABASE SCHEMA

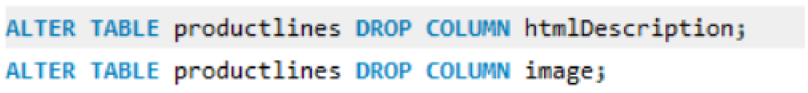


```
▼ classicmodels
  ▼ Tables
    ▶ customers
    ▶ employees
    ▶ offices
    ▶ orderdetails
    ▶ orders
    ▶ payments
    ▶ productlines
    ▶ products
```

- **Customers**: stores customer's data.
- **Products**: stores a list of scale model cars.
- **ProductLines**: stores a list of product line categories.
- **Orders**: stores sales orders placed by customers.
- **OrderDetails**: stores sales order line items for each sales order.
- **Payments**: stores payments made by customers based on their accounts.
- **Employees**: stores all employee information as well as the organization structure such as who reports to whom.
- **Offices**: stores sales office data.

DELETING UNWANTED COLUMNS IN THE *PRODUCTLINES* TABLE

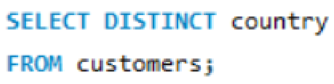
I noticed that there were two columns (“*html description”* and “*image”*) in the *productlines* tables Where no data is present so I deleted them using the **ALTER TABLE** and **DROP COLUMN** statement.



```
ALTER TABLE productlines DROP COLUMN htmlDescription;
ALTER TABLE productlines DROP COLUMN image;
```

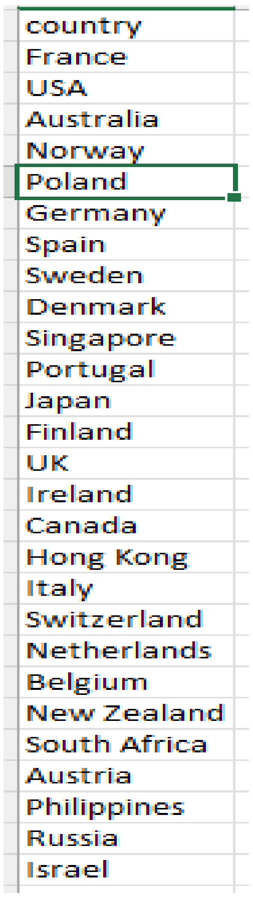
Let’s analyze the database!

→List of countries where customers are located



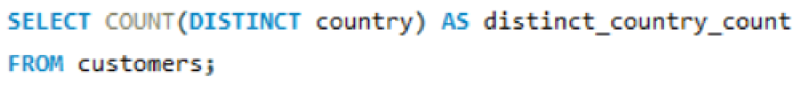
```
SELECT DISTINCT country
FROM customers;
```

Output:



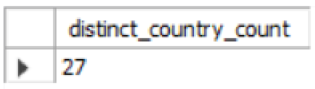
country
France
USA
Australia
Norway
Poland
Germany
Spain
Sweden
Denmark
Singapore
Portugal
Japan
Finland
UK
Ireland
Canada
Hong Kong
Italy
Switzerland
Netherlands
Belgium
New Zealand
South Africa
Austria
Philippines
Russia
Israel

→Total number of country where customers are located



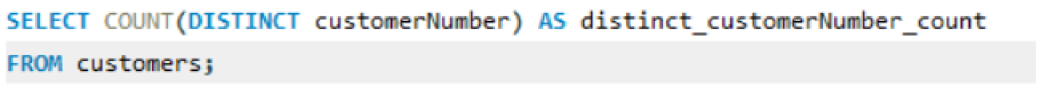
```
SELECT COUNT(DISTINCT country) AS distinct_country_count
FROM customers;
```

Output:



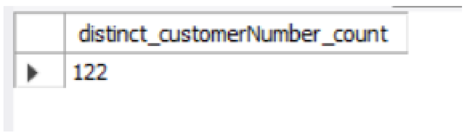
```
  distinct_country_count
▶ 27
```

→Total number of customers



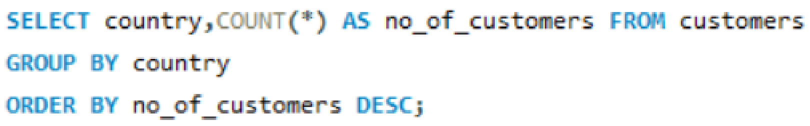
```
SELECT COUNT(DISTINCT customerNumber) AS distinct_customerNumber_count
FROM customers;
```

Output:



```
  distinct_customerNumber_count
▶ 122
```

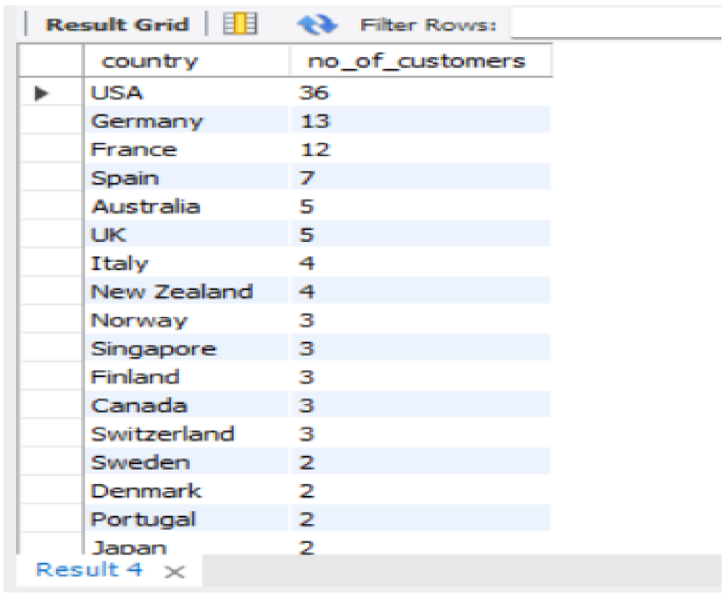
→Total number of customers for each country



```
SELECT country,COUNT(*) AS no_of_customers FROM customers
GROUP BY country
ORDER BY no_of_customers DESC;
```

Output:

We can see from the below result(figure) that the **highest number of customers** are located in the **USA**.


Result Grid | Filter Rows:

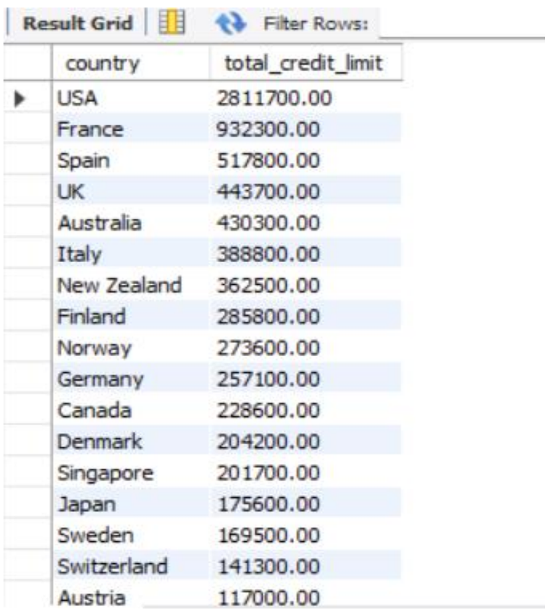
country	no_of_customers
USA	36
Germany	13
France	12
Spain	7
Australia	5
UK	5
Italy	4
New Zealand	4
Norway	3
Singapore	3
Finland	3
Canada	3
Switzerland	3
Sweden	2
Denmark	2
Portugal	2
Japan	2

Result 4

→Total credit limit for customers in each Country

```
SELECT country, SUM(creditLimit) AS total_credit_limit
FROM customers
GROUP BY country
ORDER BY total_credit_limit DESC;
```

Output:


Result Grid | Filter Rows:

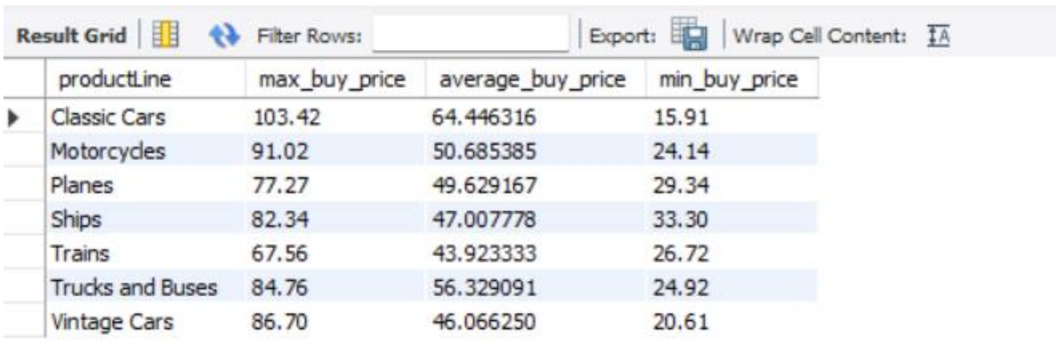
country	total_credit_limit
USA	2811700.00
France	932300.00
Spain	517800.00
UK	443700.00
Australia	430300.00
Italy	388800.00
New Zealand	362500.00
Finland	285800.00
Norway	273600.00
Germany	257100.00
Canada	228600.00
Denmark	204200.00
Singapore	201700.00
Japan	175600.00
Sweden	169500.00
Switzerland	141300.00
Austria	117000.00

USA had the **highest total credit limit**, while **six countries had zero credit limit** at all.

→Minimum, Maximum, Average buy price for each product line

```sql
CREATE VIEW product_buyprice_view AS
SELECT
productLine,
MAX(buyPrice) AS max_buy_price,
AVG(buyPrice) AS average_buy_price,
MIN(buyPrice) AS min_buy_price
FROM products
GROUP BY productLine;
```

Output:

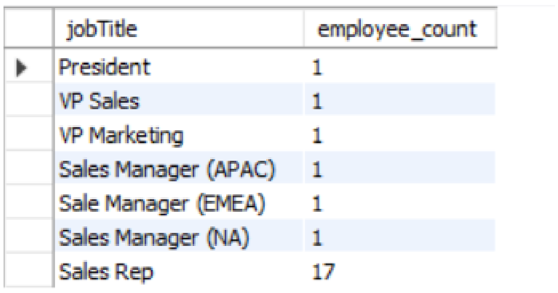


productLine	max_buy_price	average_buy_price	min_buy_price
Classic Cars	103.42	64.446316	15.91
Motorcycles	91.02	50.685385	24.14
Planes	77.27	49.629167	29.34
Ships	82.34	47.007778	33.30
Trains	67.56	43.923333	26.72
Trucks and Buses	84.76	56.329091	24.92
Vintage Cars	86.70	46.066250	20.61

→Total number of employees for each job title

```sql
SELECT jobTitle, COUNT(*) AS employee_count
FROM employees
GROUP BY jobTitle;
```

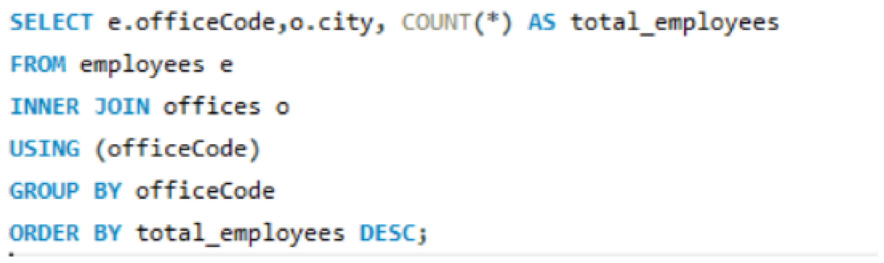
Output:



jobTitle	employee_count
President	1
VP Sales	1
VP Marketing	1
Sales Manager (APAC)	1
Sale Manager (EMEA)	1
Sales Manager (NA)	1
Sales Rep	17

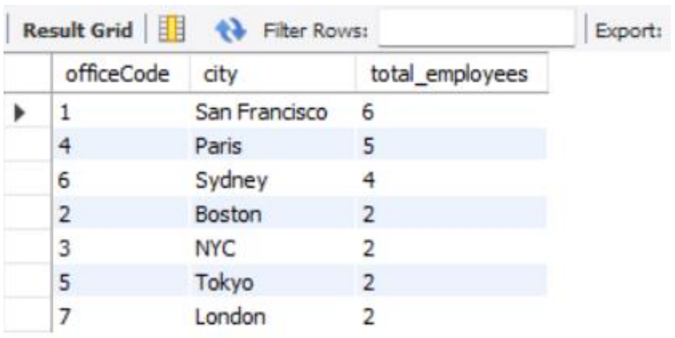
We can see that there were **17 employees who worked as Sales Rep** and only 1 employee per job title.

→Total number of employees count in each office



```sql
SELECT e.officeCode,o.city, COUNT(*) AS total_employees
FROM employees e
INNER JOIN offices o
USING (officeCode)
GROUP BY officeCode
ORDER BY total_employees DESC;
```

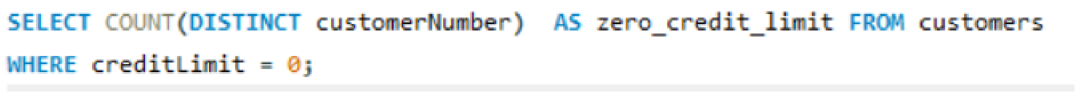
Output:



officeCode	city	total_employees
1	San Francisco	6
4	Paris	5
6	Sydney	4
2	Boston	2
3	NYC	2
5	Tokyo	2
7	London	2

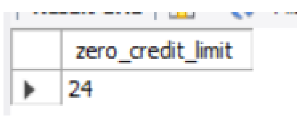
The **San Francisco office had the most employees (6)**, followed by **Paris (5)** and **Sydney (4)**.

→Total number of customers with zero credit limit



```sql
SELECT COUNT(DISTINCT customerNumber)  AS zero_credit_limit FROM customers
WHERE creditLimit = 0;
```

Output:



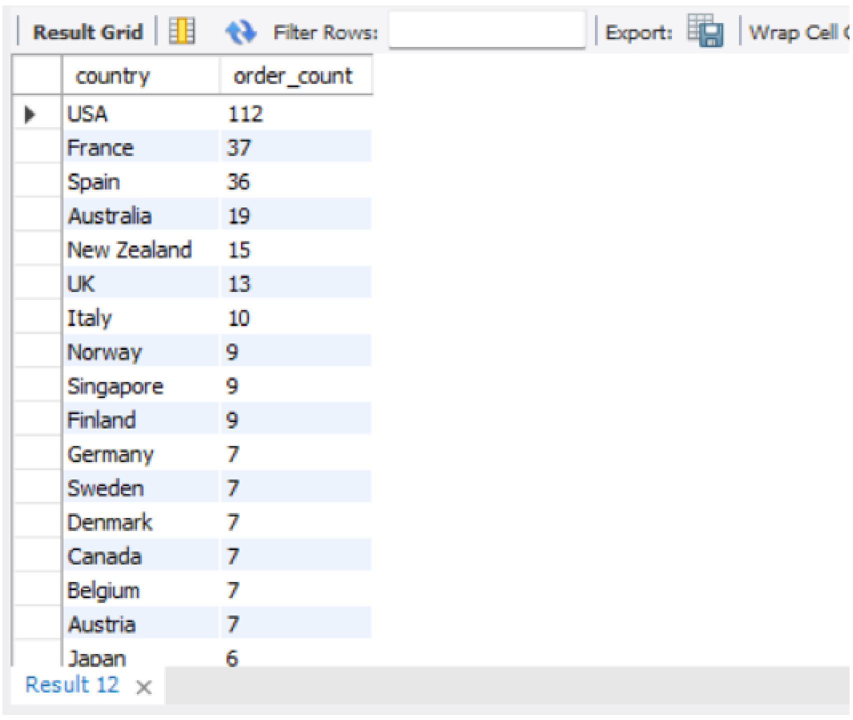
zero_credit_limit
24

We can see that there were **24 customers** who had **a credit limit of zero.**

→Highest number of orders for each country

```
SELECT country, COUNT(country) AS order_count
FROM orders o JOIN customers c
ON o.customerNumber = c.customerNumber
GROUP BY country
ORDER BY order_count DESC;
```

Output:



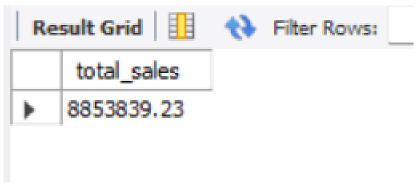
country	order_count
USA	112
France	37
Spain	36
Australia	19
New Zealand	15
UK	13
Italy	10
Norway	9
Singapore	9
Finland	9
Germany	7
Sweden	7
Denmark	7
Canada	7
Belgium	7
Austria	7
Japan	6

The **USA** has the **highest number of orders** on the list with a **total of 112**.

→Total sales

```
SELECT SUM(amount) AS total_sales FROM payments;
```

Output:

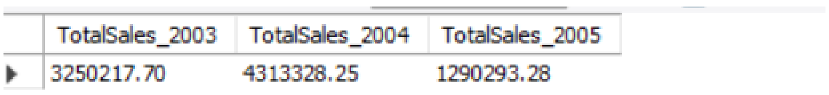


	total_sales
▶	8853839.23

→Total sales in 2003, 2004 and 2005

```
SELECT
    SUM(CASE WHEN YEAR(paymentDate) = 2003 THEN amount ELSE 0 END) AS TotalSales_2003,
    SUM(CASE WHEN YEAR(paymentDate) = 2004 THEN amount ELSE 0 END) AS TotalSales_2004,
    SUM(CASE WHEN YEAR(paymentDate) = 2005 THEN amount ELSE 0 END) AS TotalSales_2005
FROM
    payments;
```

Output:



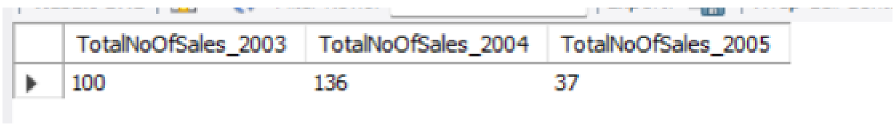
	TotalSales_2003	TotalSales_2004	TotalSales_2005
▶	3250217.70	4313328.25	1290293.28

The **highest sales** were in **2004**.

→Total number of sales in 2003, 2004 and 2005

```
SELECT
    COUNT(CASE WHEN YEAR(paymentDate) = 2003 THEN 1  END) AS TotalNoOfSales_2003,
    COUNT(CASE WHEN YEAR(paymentDate) = 2004 THEN 1  END) AS TotalNoOfSales_2004,
    COUNT(CASE WHEN YEAR(paymentDate) = 2005 THEN 1  END) AS TotalNoOfSales_2005
FROM
    payments;
```

Output:

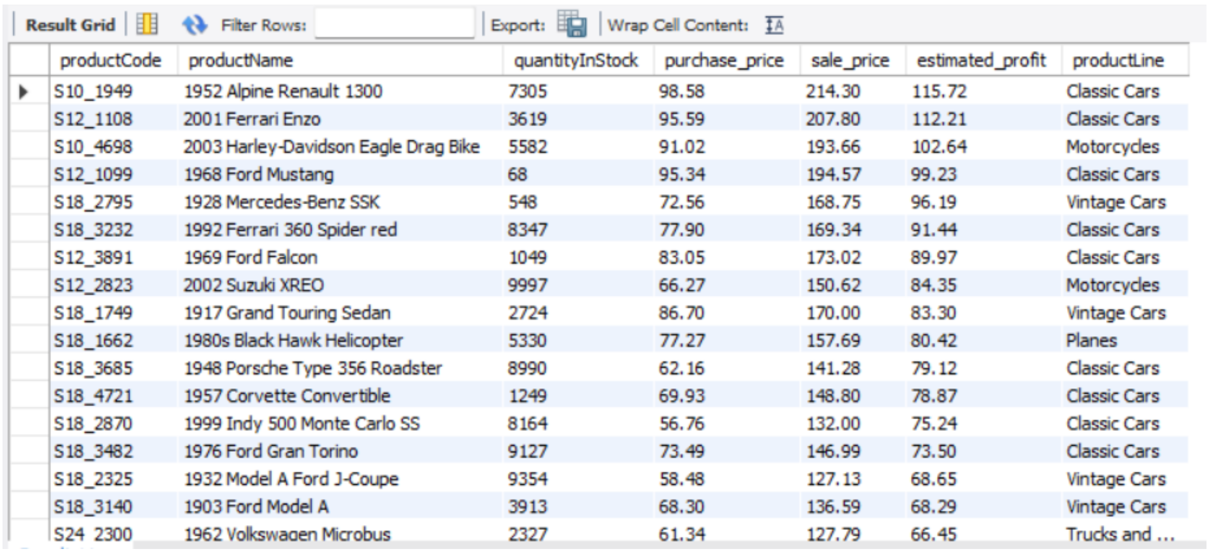


	TotalNoOfSales_2003	TotalNoOfSales_2004	TotalNoOfSales_2005
▶	100	136	37

→ Products which are currently in stock, purchase price, sale price and estimated profit

```
SELECT p.productCode,
       p.productName,
       p.quantityInStock,
       p.buyPrice AS purchase_price,
       p.MSRP AS sale_price,
       (p.MSRP - p.buyPrice) AS estimated_profit,
       pl.productLine
FROM products p
JOIN productlines pl
USING (productLine)
ORDER BY estimated_profit DESC;
```

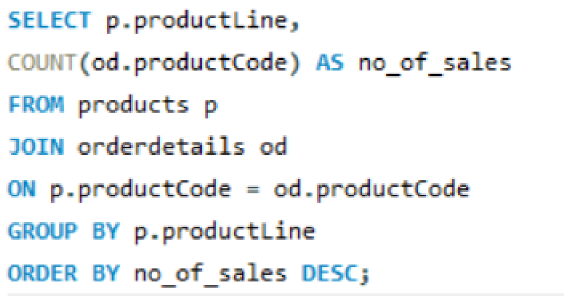
Output:


Result Grid
Filter Rows:
Export:
Wrap Cell Content:

	productCode	productName	quantityInStock	purchase_price	sale_price	estimated_profit	productLine
▶	S10_1949	1952 Alpine Renault 1300	7305	98.58	214.30	115.72	Classic Cars
	S12_1108	2001 Ferrari Enzo	3619	95.59	207.80	112.21	Classic Cars
	S10_4698	2003 Harley-Davidson Eagle Drag Bike	5582	91.02	193.66	102.64	Motorcycles
	S12_1099	1968 Ford Mustang	68	95.34	194.57	99.23	Classic Cars
	S18_2795	1928 Mercedes-Benz SSK	548	72.56	168.75	96.19	Vintage Cars
	S18_3232	1992 Ferrari 360 Spider red	8347	77.90	169.34	91.44	Classic Cars
	S12_3891	1969 Ford Falcon	1049	83.05	173.02	89.97	Classic Cars
	S12_2823	2002 Suzuki XREO	9997	66.27	150.62	84.35	Motorcycles
	S18_1749	1917 Grand Touring Sedan	2724	86.70	170.00	83.30	Vintage Cars
	S18_1662	1980s Black Hawk Helicopter	5330	77.27	157.69	80.42	Planes
	S18_3685	1948 Porsche Type 356 Roadster	8990	62.16	141.28	79.12	Classic Cars
	S18_4721	1957 Corvette Convertible	1249	69.93	148.80	78.87	Classic Cars
	S18_2870	1999 Indy 500 Monte Carlo SS	8164	56.76	132.00	75.24	Classic Cars
	S18_3482	1976 Ford Gran Torino	9127	73.49	146.99	73.50	Classic Cars
	S18_2325	1932 Model A Ford J-Coupe	9354	58.48	127.13	68.65	Vintage Cars
	S18_3140	1903 Ford Model A	3913	68.30	136.59	68.29	Vintage Cars
	S24_2300	1962 Volkswagen Microbus	2327	61.34	127.79	66.45	Trucks and ...

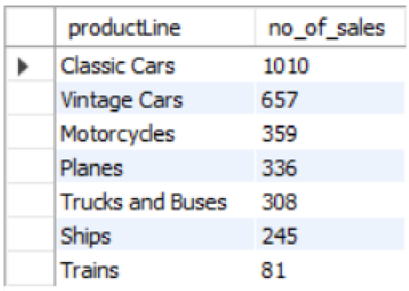
Based on the estimated profit, the product code "**S10_1949**" is expected to yield the **highest profit**.

→Most sales by product line



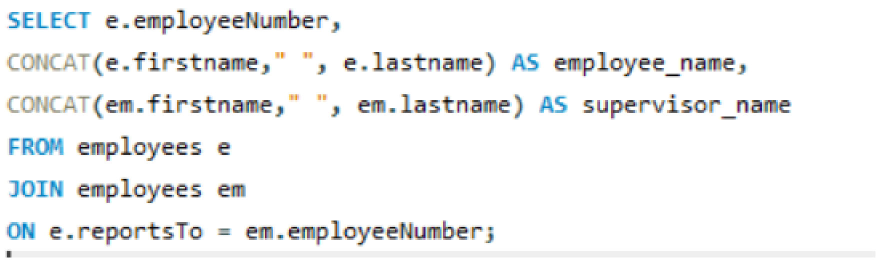
```sql
SELECT p.productLine,
COUNT(od.productCode) AS no_of_sales
FROM products p
JOIN orderdetails od
ON p.productCode = od.productCode
GROUP BY p.productLine
ORDER BY no_of_sales DESC;
```

Output:



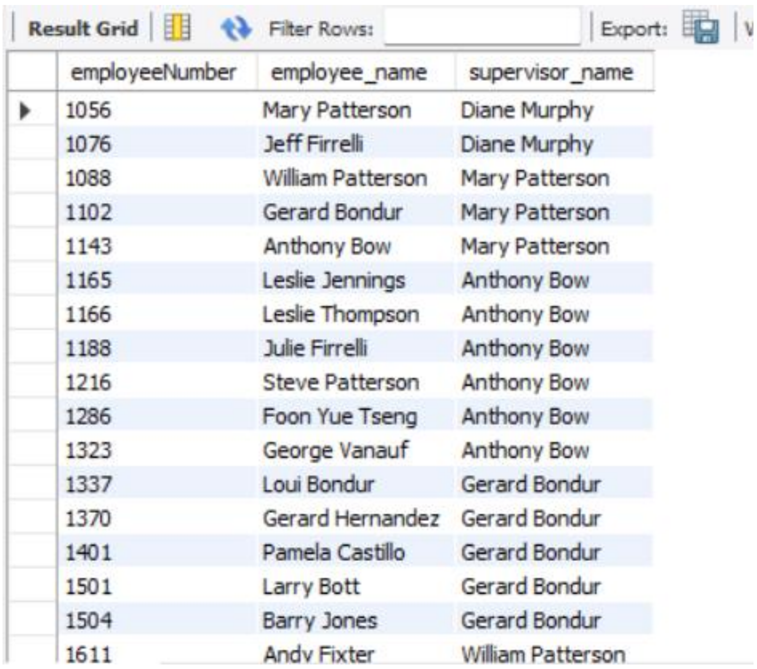
productLine	no_of_sales
Classic Cars	1010
Vintage Cars	657
Motorcycles	359
Planes	336
Trucks and Buses	308
Ships	245
Trains	81

→Hierarchy of the company's employees



```sql
SELECT e.employeeNumber,
CONCAT(e.firstname," ", e.lastname) AS employee_name,
CONCAT(em.firstname," ", em.lastname) AS supervisor_name
FROM employees e
JOIN employees em
ON e.reportsTo = em.employeeNumber;
```

Output:

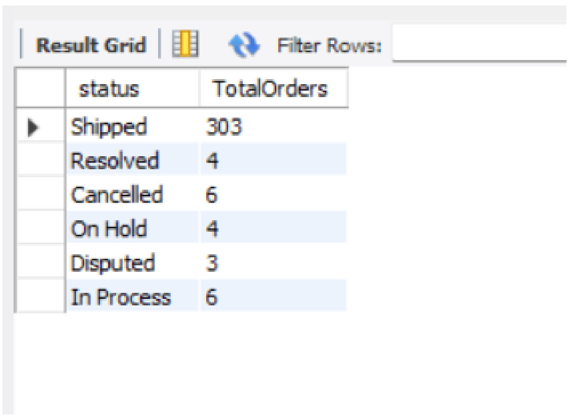

Result Grid | Filter Rows: | Export:

employeeNumber	employee_name	supervisor_name
1056	Mary Patterson	Diane Murphy
1076	Jeff Firrelli	Diane Murphy
1088	William Patterson	Mary Patterson
1102	Gerard Bondur	Mary Patterson
1143	Anthony Bow	Mary Patterson
1165	Leslie Jennings	Anthony Bow
1166	Leslie Thompson	Anthony Bow
1188	Julie Firrelli	Anthony Bow
1216	Steve Patterson	Anthony Bow
1286	Foon Yue Tseng	Anthony Bow
1323	George Vanauf	Anthony Bow
1337	Loui Bondur	Gerard Bondur
1370	Gerard Hernandez	Gerard Bondur
1401	Pamela Castillo	Gerard Bondur
1501	Larry Bott	Gerard Bondur
1504	Barry Jones	Gerard Bondur
1611	Andy Fixter	William Patterson

→Total number of orders per status

```
SELECT status, COUNT(*) AS TotalOrders
FROM orders
GROUP BY status;
```

Output:

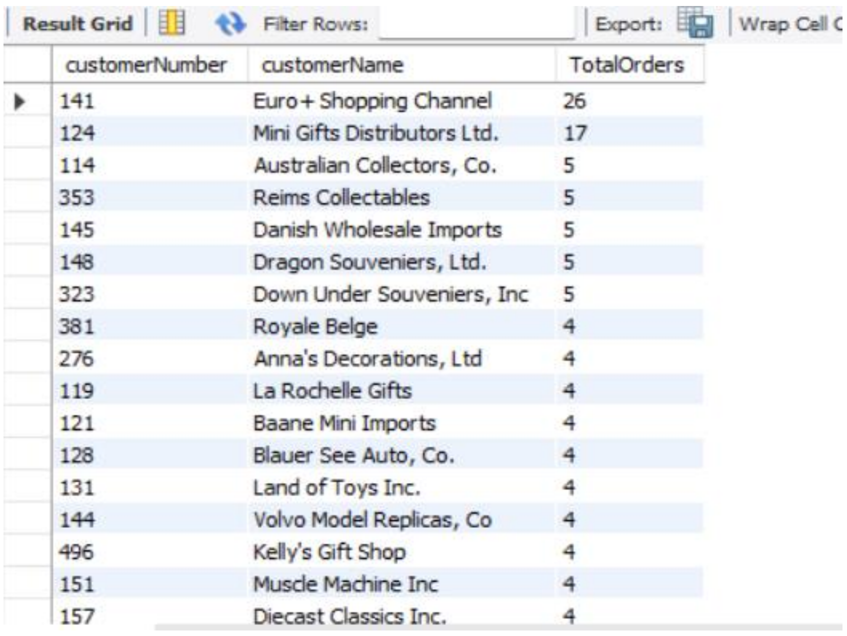

Result Grid | Filter Rows:

status	TotalOrders
Shipped	303
Resolved	4
Cancelled	6
On Hold	4
Disputed	3
In Process	6

→Total number of orders were placed by each customer

```
SELECT o.customerNumber,c.customerName, COUNT(*) AS TotalOrders
FROM orders o
LEFT JOIN customers c
USING(customerNumber)
GROUP BY o.customerNumber
ORDER BY TotalOrders DESC;
```

Output:

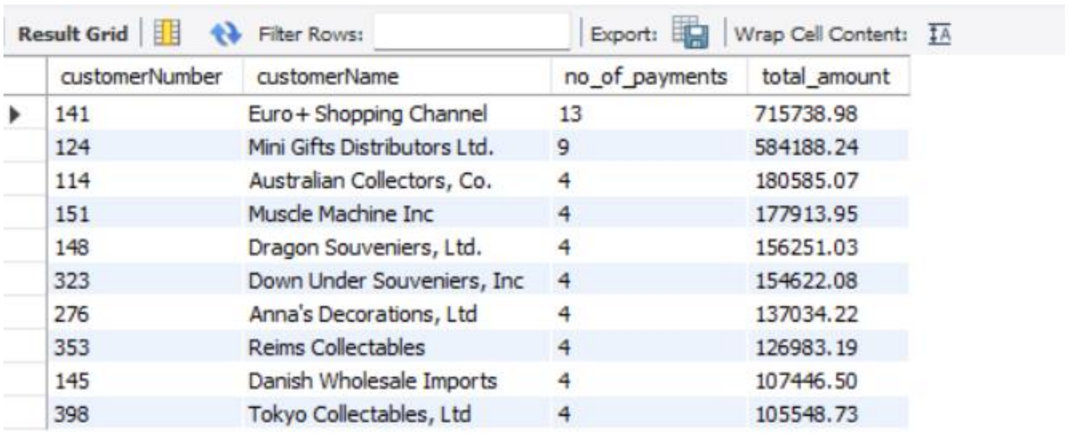


customerNumber	customerName	TotalOrders
141	Euro+ Shopping Channel	26
124	Mini Gifts Distributors Ltd.	17
114	Australian Collectors, Co.	5
353	Reims Collectables	5
145	Danish Wholesale Imports	5
148	Dragon Souveniers, Ltd.	5
323	Down Under Souveniers, Inc	5
381	Royale Belge	4
276	Anna's Decorations, Ltd	4
119	La Rochelle Gifts	4
121	Baane Mini Imports	4
128	Blauer See Auto, Co.	4
131	Land of Toys Inc.	4
144	Volvo Model Replicas, Co	4
496	Kelly's Gift Shop	4
151	Muscle Machine Inc	4
157	Diecast Classics Inc.	4

→Top 10 customers ranked by their payment amounts and total number of payments

```
SELECT p.customerNumber,c.customerName, COUNT(*) AS no_of_payments,SUM(amount) as total_amount
FROM payments p
INNER JOIN customers c
USING (customerNumber)
GROUP BY customerNumber
ORDER BY no_of_payments DESC ,total_amount DESC
LIMIT 10;
```

Output:



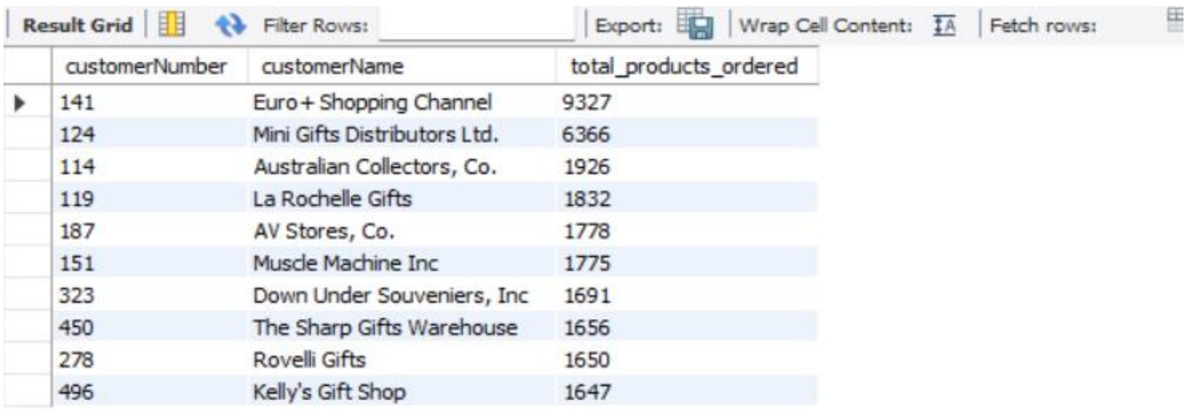
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customerNumber	customerName	no_of_payments	total_amount
141	Euro+ Shopping Channel	13	715738.98
124	Mini Gifts Distributors Ltd.	9	584188.24
114	Australian Collectors, Co.	4	180585.07
151	Muscle Machine Inc	4	177913.95
148	Dragon Souveniers, Ltd.	4	156251.03
323	Down Under Souveniers, Inc	4	154622.08
276	Anna's Decorations, Ltd	4	137034.22
353	Reims Collectables	4	126983.19
145	Danish Wholesale Imports	4	107446.50
398	Tokyo Collectables, Ltd	4	105548.73

→Top 10 customers based on the total number of products they have ordered

```
SELECT o.customerNumber,c.customerName,
SUM(od.quantityOrdered) AS total_products_ordered
FROM orderdetails od
INNER JOIN orders o
USING(orderNumber)
INNER JOIN customers c
USING(customerNumber)
GROUP BY customerNumber
ORDER BY total_products_ordered DESC
LIMIT 10;
```

Output:



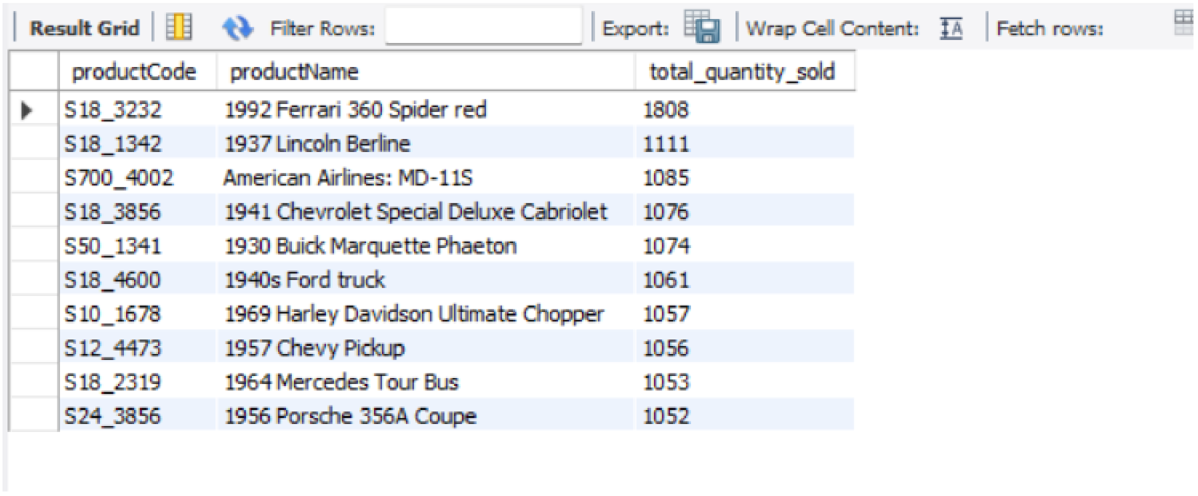
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

customerNumber	customerName	total_products_ordered
141	Euro+ Shopping Channel	9327
124	Mini Gifts Distributors Ltd.	6366
114	Australian Collectors, Co.	1926
119	La Rochelle Gifts	1832
187	AV Stores, Co.	1778
151	Muscle Machine Inc	1775
323	Down Under Souveniers, Inc	1691
450	The Sharp Gifts Warehouse	1656
278	Rovelli Gifts	1650
496	Kelly's Gift Shop	1647

→Top 10 products based on total quantity sold

```
SELECT od.productCode,p.productName,
SUM(od.quantityOrdered) AS total_quantity_sold
FROM orderdetails od
INNER JOIN products P
USING (productCode)
GROUP BY productCode
ORDER BY total_quantity_sold DESC
LIMIT 10;
```

Output:



	productCode	productName	total_quantity_sold
▶	S18_3232	1992 Ferrari 360 Spider red	1808
	S18_1342	1937 Lincoln Berline	1111
	S700_4002	American Airlines: MD-11S	1085
	S18_3856	1941 Chevrolet Special Deluxe Cabriolet	1076
	S50_1341	1930 Buick Marquette Phaeton	1074
	S18_4600	1940s Ford truck	1061
	S10_1678	1969 Harley Davidson Ultimate Chopper	1057
	S12_4473	1957 Chevy Pickup	1056
	S18_2319	1964 Mercedes Tour Bus	1053
	S24_3856	1956 Porsche 356A Coupe	1052

Summary:

Based on our data analysis, we have come to the conclusion that

- The **total sales** of Axon Company between 2003 and 2005 amounted to **$8853839.23.**
- **Most sales were made in 2004** with **$4313328.25** and 136 by count, while **the least sales** were recorded **in 2005** with only **$1290293.28** and 37 by count.
- List of **303 orders** that were **shipped between 2003 and 2005**.
- The Axon company has **customers in 27 countries**, with the **highest number** in the **USA**.
- In total, **the number of customers** for Axon Company was **122**.

- The **product line with the highest number of sales** is the **classic cars**, which sold a total of **1010**.
- The **USA** has the **highest number of orders: 112**, followed by **France with 37** and **Spain with 36**.
- The customer with **customer number 141 placed the highest number of orders**.
- The product with **the product code S18_3232 had the highest quantity of sales**.

The visualization aspect has been completed using Power BI Desktop.