2023

OdinSchool

CAPSTONE PROJECT

DATA SCIENCE BOOTCAMP

NAME: T U DURGA PRASAD
STUDENT ID: S4168
COHORT: DS28C

Description

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

Approach

The objective of the capstone project is to create and execute a BI solution utilizing PowerBI and SQL that will assist the business in successfully managing and analyzing its sales data. The remedy needs to be able to:

Data from a MySQL database should be imported and integrated into PowerBI. Learning how to connect a MySQL database to PowerBI. To prepare the data for analysis, clean and convert it. Create dynamic dashboards and reports using PowerBI to aid in the management and sales team's understanding of the data.

Tech Used

Mysql

Power BI

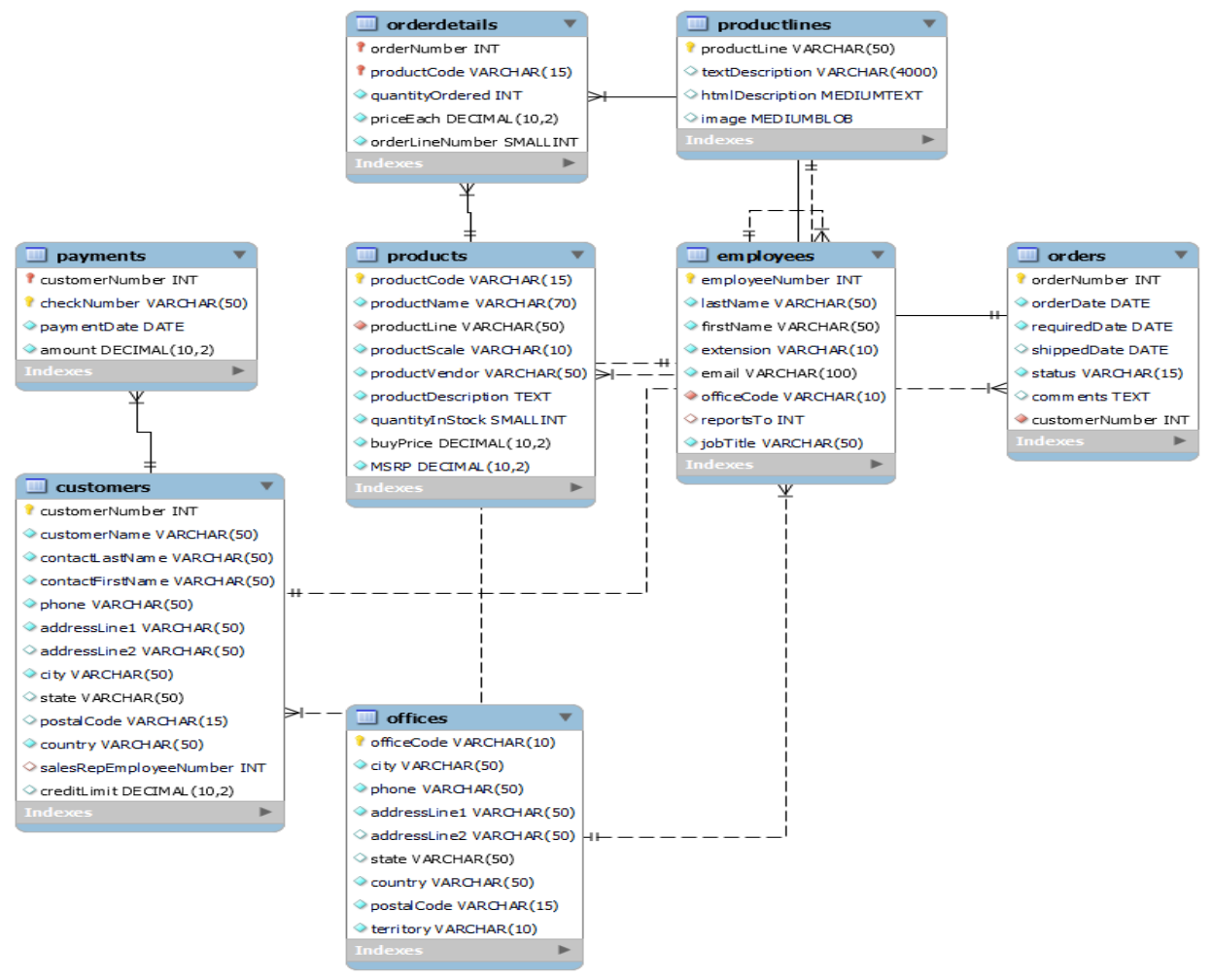
Outcome

The aim of the study is to arrange the database and gain the skills required to work with SQL and Power-BI. This includes details about clients, goods, sales orders, product lines, and various other often observed business-related specifics.

SQL

The dataset provided here contains the transactions report from 2003 through 2005, and the provided SQL file is used for the analysis. The file consists of up of 8 tables that include information about customers, employees, orders, order details, payments, product lines, and office details that represents the operations of a business that sells vintage models of automobiles, railways, and other types of vehicles.

Let's first build a database called "classicmodels" and add the provided file to it. Let's add the provided file and then begin analyzing the information to gain some insightful knowledge.
Let's first use the ER Diagram to understand the relationship between the tables before we analyze the data.



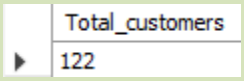
The relationship between the tables is clearly shown in the preceding graphic.

The MySQL classic model's database schema consists of the following tables:

- 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.

Now let's start analyzing the data for the information to gain some insightful knowledge.

1. Calculating the number of customers who made purchases from us between 2003 and 2005.
   SELECT COUNT (customerNumber) AS Total_customers FROM customers;

Total_customers
122

   There were 122 consumers in total who made purchases from the Axon company between 2003 and 2005.

2. Let's make a Report of the name and city of customers who don't have sales representatives?
   SELECT customerNumber,customerName,city FROM customers WHERE salesRepEmployeeNumber IS NULL;

customerNumber	customerName	city
125	Havel & Zbyszek Co	Warszawa
169	Porto Imports Co.	Lisboa
206	Asian Shopping Network, Co	Singapore
223	Natürlich Autos	Cunewalde
237	ANG Resellers	Madrid
247	Messner Shopping Network	Frankfurt
273	Franken Gifts, Co	München
293	BG&E Collectables	Fribourg
303	Schuyler Imports	Amsterdam
307	Der Hund Imports	Berlin
335	Cramer Spezialitäten, Ltd	Brandenburg
348	Asian Treasures, Inc.	Cork
356	SAR Distributors, Co	Hatfield
361	Kommission Auto	Münster
369	Lisboa Souveniers, Inc	Lisboa
409	Stuttgart Collectable Excha...	Stuttgart
443	Feuer Online Stores, Inc	Leipzig
459	Warburg Exchange	Aachen

There are 22 customers without sales representatives.

3. Preparing a list of offices sorted by country, state, city.
   SELECT country, state, city FROM offices
   ORDER BY country, state, city;

country	state	city
Australia	NULL	Sydney
France	NULL	Paris
Japan	Chiyoda-Ku	Tokyo
UK	NULL	London
USA	CA	San Francisco
USA	MA	Boston
USA	NY	NYC

There are 7 cities and 5 countries where the Axon offices are located.

4. How many employees work for the company?
   select count(employeeNumber)as Total_Employees from employees;

Total_Employees
23

There are 23 employees working for Axon company.

5. What is the total of payments received?
   select sum(amount)as Total_Payment from payments;

Total_Payment
8853839.23

   From 2003 - 2005, the customer made payments totaling $8853839.23

6. How many Product Lines are available in Axon?
   select productline from productlines;

productline
Classic Cars
Motorcycles
Planes
Ships
Trains
Trucks and Buses
Vintage Cars

7. Payments greater than $100,000
   SELECT * FROM payments WHERE amount > 100000;

customerNumber	checkNumber	paymentDate	amount
124	AE215433	2005-03-05	101244.59
124	KI131716	2003-08-15	111654.40
141	ID10962	2004-12-31	116208.40
141	JE105477	2005-03-18	120166.58
148	KM172879	2003-12-26	105743.00

   There are five payments made by customer which is greater than $100,000

8. Let us find how many products in each product line?
   SELECT productLine, count(productCode) as Total_No_products from products
   group by productLine order by Total_No_products desc;

productLine	Total_No_products
Classic Cars	38
Vintage Cars	24
Motorcycles	13
Planes	12
Trucks and Buses	11
Ships	9
Trains	3

The above query gives the total number of products available based on the product lines in which classic cars have more varieties and Train being the least with 3.

9. The average percentage markup of the MSRP on buy Price

```
SELECT ROUND(AVG(((MSRP - buyPrice) / buyPrice) * 100),2) AS
average_markup_percentage
FROM products;
```

average_markup_percentage
88.70

The result i.e., 88.70% of this query will be the average percentage markup of MSRP on Buy Price for all the products in your table

10. Let us know number of employees based on their job titles

```
select count(employeeNumber) as total_employees, jobTitle from
employees
group by jobTitle;
```

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

The Axon company has one President, one vice president for sales and one vice president marketing, four sales managers for different region and seventeen sales reps making a total of twenty-three employees.

11. Report the Sales representative for each customer.

```
select customers.customerNumber, employees.employeeNumber,
concat(employees.firstName, " ",employees.lastName)
as fullName from customers inner join employees on
customers.salesRepEmployeeNumber = employees.employeeNumber
order by customers.customerNumber;
```

customerNumber	employeeNumber	fullName
103	1370	Gerard Hernandez
112	1166	Leslie Thompson
114	1611	Andy Fixter
119	1370	Gerard Hernandez
121	1504	Barry Jones
124	1165	Leslie Jennings
128	1504	Barry Jones
129	1165	Leslie Jennings
131	1323	George Vanauf
141	1370	Gerard Hernandez
144	1504	Barry Jones
145	1401	Pamela Castillo
146	1337	Loui Bondur
148	1621	Mami Nishi
151	1286	Foon Yue Tseng
157	1216	Steve Patterson
161	1165	Leslie Jennings

The query gives us all the sales representative assigned for each customer.

12. List the amount paid by each customer.

```
select customerNumber ,
sum(amount) as Total_payment from payments
group by customerNumber
order by Total_payment desc;
```

customerNumber	Total_payment
475	43748.72
144	43680.65
487	42570.37
347	41506.19
328	38281.51
362	33533.47
173	32198.69
415	31310.09
489	29586.15
456	29230.43

13. List the value of 'On Hold' orders.

```
SELECT o.orderNumber, SUM(od.quantityOrdered * od.priceEach) AS
OrderValue, o.status
FROM orders o
INNER JOIN orderdetails od ON o.orderNumber = od.orderNumber
```

```
WHERE o.status = 'On Hold'
GROUP BY o.orderNumber;
```

orderNumber	OrderValue	status
10334	23014.17	On Hold
10401	43525.04	On Hold
10407	52229.55	On Hold
10414	50806.85	On Hold

And to know the number of orders 'On Hold' for each customer

```
SELECT c.customerNumber, c.customerName, COUNT(o.orderNumber) AS
OnHoldOrders
FROM customers c
LEFT JOIN orders o ON c.customerNumber = o.customerNumber AND
o.status = 'On Hold'
GROUP BY c.customerNumber
ORDER BY OnHoldOrders DESC;
```

customerNumber	customerName	OnHoldOrders
144	Volvo Model Replicas, Co	1
328	Tekni Collectables Inc.	1
362	Gifts4AllAges.com	1
450	The Sharp Gifts Warehouse	1
103	Atelier graphique	0
112	Signal Gift Stores	0
114	Australian Collectors, Co.	0
119	La Rochelle Gifts	0
121	Baane Mini Imports	0
124	Mini Gifts Distributors Ltd.	0

There are four orders 'On Hold' and their orderValue is as show in above screenshot.

14. Compute the total value ordered, total amount paid, and their difference for each customer for orders placed in 2004 and payments received in 2004

Let us create a View for the Total Ordered

```
CREATE VIEW TotalOrdered2004 AS
SELECT
   o.customerNumber,
   SUM(od.quantityOrdered * od.priceEach) AS TotalOrdered
```

```
FROM
   orders o
JOIN
   orderdetails od ON o.orderNumber = od.orderNumber
WHERE
   YEAR(o.orderDate) = 2004
GROUP BY
   o.customerNumber;
```

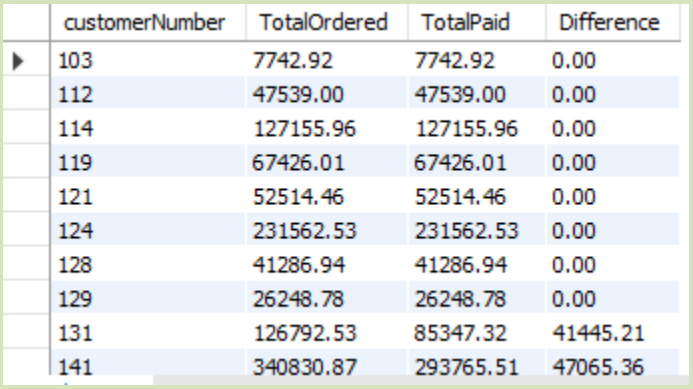
Now Let us create a View for the Total Paid:

```
CREATE VIEW TotalPaid2004 AS
SELECT
   c.customerNumber,
   SUM(p.amount) AS TotalPaid
FROM
   customers c
JOIN
   payments p ON c.customerNumber = p.customerNumber
WHERE
   YEAR(p.paymentDate) = 2004
GROUP BY
   c.customerNumber;
```

With the views in place, we can now compute the difference between the total value ordered and the total amount paid for each customer in 2004.

```
SELECT
   o.customerNumber,
   TotalOrdered,
   TotalPaid,
   TotalOrdered - TotalPaid AS Difference
FROM
   TotalOrdered2004 o
JOIN
```

TotalPaid2004 p ON o.customerNumber = p.customerNumber;



customerNumber	TotalOrdered	TotalPaid	Difference
103	7742.92	7742.92	0.00
112	47539.00	47539.00	0.00
114	127155.96	127155.96	0.00
119	67426.01	67426.01	0.00
121	52514.46	52514.46	0.00
124	231562.53	231562.53	0.00
128	41286.94	41286.94	0.00
129	26248.78	26248.78	0.00
131	126792.53	85347.32	41445.21
141	340830.87	293765.51	47065.36

This query will give you the total value ordered, total amount paid, and the difference for each customer for orders placed and payments received in 2004.

Similarly we can calculate for the year 2003 and 2005 using the same steps.

15. Let us know what is the percentage value of each product in inventory sorted by the highest percentage first.

First let us start Creating a View for Total Inventory Value Create a view that calculates the total inventory value for each product. You'll need to multiply the quantity in stock by the buy price.

```
CREATE VIEW ProductInventoryValue AS
SELECT
   productCode,
   (quantityInStock * buyPrice) AS InventoryValue
FROM
   products;
```

With the view in place, we can now calculate the percentage value of each product in inventory. The percentage value is the ratio of the individual product's inventory value to the total inventory value.

	productCode	productName	PercentageValue
▶	S18_1984	1995 Honda Civic	3.004793
	S10_1949	1952 Alpine Renault 1300	2.358418
	S10_4962	1962 LanciaA Delta 16V	2.300118
	S12_3380	1968 Dodge Charger	2.245620
	S18_3482	1976 Ford Gran Torino	2.196687
	S700_2466	America West Airlines B757-200	2.175016
	S12_2823	2002 Suzuki XREO	2.169694
	S18_3232	1992 Ferrari 360 Spider red	2.129510
	S24_3856	1956 Porsche 356A Coupe	2.124757
	S18_2319	1964 Mercedes Tour Bus	2.024587

The product '1995 Honda Civic' of product code 'S18_1984' has the highest percentage value of 3.004% and the least is '1960 BSA Gold star DBD34' with 0.0018%

16. Let us see the profit generated by each customer based on their orders. Also, see each customer's profit as a percentage of total profit.

```
WITH CustomerProfit AS (
    SELECT
        c.customerNumber,
        c.customerName,
        SUM((od.priceEach - p.buyPrice) * od.quantityOrdered) AS Profit
    FROM
        customers c
    JOIN
        orders o ON c.customerNumber = o.customerNumber
    JOIN
        orderdetails od ON o.orderNumber = od.orderNumber
    JOIN
        products p ON od.productCode = p.productCode
    GROUP BY
        c.customerNumber, c.customerName
),
TotalProfit AS (
    SELECT SUM(Profit) AS TotalProfit FROM CustomerProfit
```

```
)
SELECT
   cp.customerNumber,
   cp.customerName,
   cp.Profit,
   (cp.Profit / tp.TotalProfit) * 100 AS ProfitPercentage
FROM
   CustomerProfit cp
CROSS JOIN
   TotalProfit tp
ORDER BY
   cp.Profit DESC;
```

customerNumber	customerName	Profit	ProfitPercentage
141	Euro+ Shopping Channel	326519.66	8.534498
124	Mini Gifts Distributors Ltd.	236769.39	6.188625
151	Muscle Machine Inc	72370.09	1.891593
114	Australian Collectors, Co.	70311.07	1.837775
119	La Rochelle Gifts	60875.30	1.591145
148	Dragon Souveniers, Ltd.	60477.38	1.580744
187	AV Stores, Co.	60095.86	1.570772
323	Down Under Souveniers, Inc	60013.99	1.568632
131	Land of Toys Inc.	58669.10	1.533480
450	The Sharp Gifts Warehouse	55931.37	1.461922
321	Corporate Gift Ideas Co.	55674.28	1.455202
382	Salzburg Collectables	54724.68	1.430381

17. The revenue generated by each sales representative based on the orders from the customers they serve

```
SELECT
   e.employeeNumber,
   CONCAT(e.firstName, ' ', e.lastName) AS SalesRepresentative,
   SUM(od.priceEach * od.quantityOrdered) AS Revenue
FROM
   employees e
JOIN
   customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN
```

```
    orders o ON c.customerNumber = o.customerNumber
JOIN
    orderdetails od ON o.orderNumber = od.orderNumber
GROUP BY
    e.employeeNumber, SalesRepresentative
ORDER BY
    Revenue DESC;
```

employeeNumber	SalesRepresentative	Revenue
1370	Gerard Hernandez	1258577.81
1165	Leslie Jennings	1081530.54
1401	Pamela Castillo	868220.55
1501	Larry Bott	732096.79
1504	Barry Jones	704853.91
1323	George Vanauf	669377.05
1612	Peter Marsh	584593.76
1337	Loui Bondur	569485.75
1611	Andy Fixter	562582.59
1216	Steve Patterson	505875.42
1286	Foon Yue Tseng	488212.67
1621	Mami Nishi	457110.07

We can see that 'Gerard Hernandez' has generated highest revenue that is '$ 1258577.81' among all the employees.

18. Let us calculate each salesperson's profit generated based on the purchases made by the customers they service.

```
SELECT
    e.employeeNumber,
    CONCAT(e.firstName, ' ', e.lastName) AS SalesRepresentative,
    SUM((od.priceEach - p.buyPrice) * od.quantityOrdered) AS Profit
FROM
    employees e
JOIN
    customers c ON e.employeeNumber = c.salesRepEmployeeNumber
JOIN
    orders o ON c.customerNumber = o.customerNumber
JOIN
    orderdetails od ON o.orderNumber = od.orderNumber
```

```
JOIN
    products p ON od.productCode = p.productCode
GROUP BY
    e.employeeNumber, SalesRepresentative
ORDER BY
    Profit DESC;
```

employeeNumber	SalesRepresentative	Profit
1370	Gerard Hernandez	504644.71
1165	Leslie Jennings	435208.35
1401	Pamela Castillo	340727.90
1501	Larry Bott	290203.59
1504	Barry Jones	276659.25
1323	George Vanauf	269596.09
1337	Loui Bondur	234891.07
1612	Peter Marsh	230811.75
1611	Andy Fixter	222207.18
1216	Steve Patterson	197879.23
1286	Foon Yue Tseng	194839.92
1621	Mami Nishi	181181.80

'Gerard Hernandez' made the most profit among all the sales representative i.e., $ 504644.71.

19. Let us see the revenue generated by each product and sort it by product name.

```
SELECT
    p.productCode,
    p.productName,
    SUM(od.priceEach * od.quantityOrdered) AS Revenue
FROM
    products p
JOIN
    orderdetails od ON p.productCode = od.productCode
GROUP BY
    p.productCode, p.productName
ORDER BY
    p.productName;
```

productCode	productName	Revenue
S24_2011	18th century schooner	112427.12
S18_3136	18th Century Vintage Horse Carriage	85328.57
S24_2841	1900s Vintage Bi-Plane	58434.07
S24_4278	1900s Vintage Tri-Plane	68276.35
S18_3140	1903 Ford Model A	111528.82
S18_4522	1904 Buick Runabout	77800.54
S18_2248	1911 Ford Town Car	45306.77
S24_3151	1912 Ford Model T Delivery Wagon	77239.92
S18_2949	1913 Ford Model T Speedster	97193.88
S18_1749	1917 Grand Touring Sedan	140535.60
S18_3320	1917 Maxwell Touring Car	87637.63
S18_2432	1926 Ford Fire Engine	55835.30

We can see from the above screenshot the highest revenue generated product is '18TH Century schooner'.

20. The profit generated by each product line.

```
SELECT
   p.productLine,
   SUM((od.priceEach - pr.buyPrice) * od.quantityOrdered) AS Profit
FROM
   products p
JOIN
   orderdetails od ON p.productCode = od.productCode
JOIN
   products pr ON od.productCode = pr.productCode
GROUP BY
   p.productLine
ORDER BY
   Profit DESC;
```

productLine	Profit
Classic Cars	1526212.20
Vintage Cars	737268.33
Motorcycles	469255.30
Trucks and Buses	400553.22
Planes	365960.71
Ships	261289.47
Trains	65341.02

The above image clearly states that ‘Classic cars’ stands at top generating a huge profit followed by other product line.

21. Let us calculate the ratio for each product of sales for 2003 versus 2004.

```
SELECT
   p.productCode,
   p.productName,
   COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2003 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0) AS Sales2003,
   COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2004 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0) AS Sales2004,
   CASE WHEN COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2003
THEN od.quantityOrdered * od.priceEach ELSE 0 END), 0) = 0 THEN NULL
   ELSE COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2004 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0) /
   COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2003 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0)
   END AS SALY_Ratio
FROM
   products p
LEFT JOIN
   orderdetails od ON p.productCode = od.productCode
LEFT JOIN
   orders o ON od.orderNumber = o.orderNumber
WHERE
   YEAR(o.orderDate) IN (2003, 2004)
GROUP BY
   p.productCode, p.productName
ORDER BY
   SALY_Ratio DESC;
```

	productCode	productName	Sales2003	Sales2004	SALY_Ratio
▶	S700_2834	ATA: B757-300	29987.69	56357.28	1.879347
	S700_2047	HMS Bounty	20872.61	38656.18	1.852005
	S32_4289	1928 Ford Phaeton Deluxe	17150.96	31678.84	1.847059
	S24_3420	1937 Horch 930V Limousine	15180.11	27828.49	1.833221
	S700_3962	The Queen Mary	23702.57	43444.51	1.832903
	S10_4962	1962 LanciaA Delta 16V	35454.67	64642.73	1.823250
	S700_1138	The Schooner Bluenose	17213.36	30427.75	1.767682
	S12_1108	2001 Ferrari Enzo	59852.24	105487.46	1.762465
	S700_2466	America West Airlines B757-200	25581.95	44854.25	1.753355
	S18_1662	1980s Black Hawk Helicopter	45774.48	80076.41	1.749368
	S24_4620	1961 Chevrolet Impala	20867.64	35876.68	1.719250
	S32_4485	1974 Ducati 350 Mk3 Desmo	23852.10	40851.72	1.712710

For year 2004- 2005

```
SELECT
   p.productCode,
   p.productName,
   COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2004 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0) AS Sales2004,
   COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2005 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0) AS Sales2005,
   CASE WHEN COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2004
THEN od.quantityOrdered * od.priceEach ELSE 0 END), 0) = 0 THEN NULL
   ELSE COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2005 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0) /
   COALESCE(SUM(CASE WHEN YEAR(o.orderDate) = 2004 THEN
od.quantityOrdered * od.priceEach ELSE 0 END), 0)
   END AS SALY_Ratio
FROM
   products p
LEFT JOIN
   orderdetails od ON p.productCode = od.productCode
LEFT JOIN
   orders o ON od.orderNumber = o.orderNumber
WHERE
   YEAR(o.orderDate) IN (2004, 2005)
```

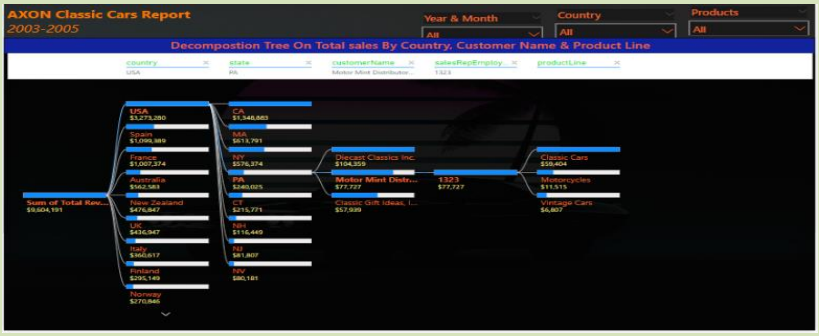
```
GROUP BY
   p.productCode, p.productName
ORDER BY
   SALY_Ratio DESC;
```

productCode	productName	Sales2004	Sales2005	SALY_Ratio
S12_4675	1969 Dodge Charger	39826.80	29567.27	0.742396
S24_1578	1997 BMW R 1100 S	42111.85	28747.69	0.682651
S18_4933	1957 Ford Thunderbird	18340.95	11251.68	0.613473
S24_3856	1956 Porsche 356A Coupe	52464.70	31432.14	0.599110
S32_1374	1997 BMW F650 ST	42155.35	22469.91	0.533026
S24_2000	1960 BSA Gold Star DBD34	30213.36	16049.47	0.531204
S24_2887	1952 Citroen-15CV	39957.80	20627.17	0.516224
S24_1785	1928 British Royal Navy Airplane	39254.87	20214.53	0.514956
S24_4278	1900s Vintage Tri-Plane	31073.11	15940.74	0.513008
S18_1589	1965 Aston Martin DB5	38822.78	19812.95	0.510343
S10_2016	1996 Moto Guzzi 1100i	51645.70	26139.34	0.506128
S18_1984	1995 Honda Civic	47971.69	23935.48	0.498950

Power BI

The next step after evaluating the data in MySQL is to use PowerBI to create interactive dashboards and reports that can aid management and the sales team in understanding the information. First, we must retrieve the data from the MySQL database, therefore we utilize Power BI to do so by providing the server and database names. After importing the data, we must transform it to remove any blank entries and process it so that we can use PowerBI to create interactive dashboards and reports.

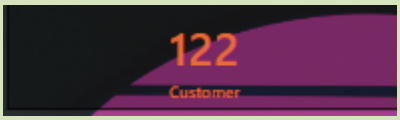
Let's now examine the constructed dashboard using PowerBI.



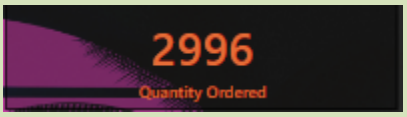
Cards

This card details the seven product lines we now provide that is

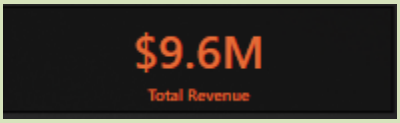
- Classic cars
- Motorcycles
- Planes
- Ships
- Trains
- Trucks and Buses
- Vintage Cars



We have total of 122 customers who have purchased from Axon company



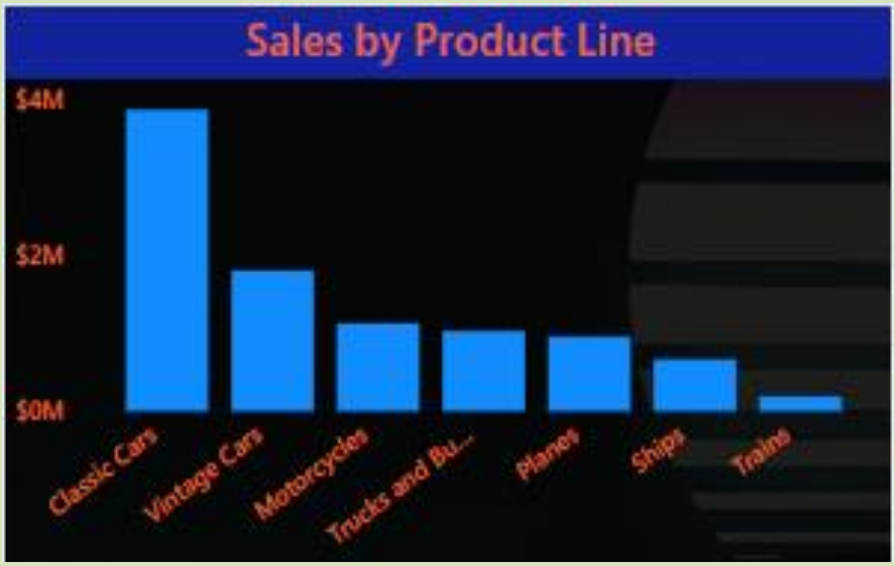
The total quantity ordered from Axon company from 2003-2005 is 2996.



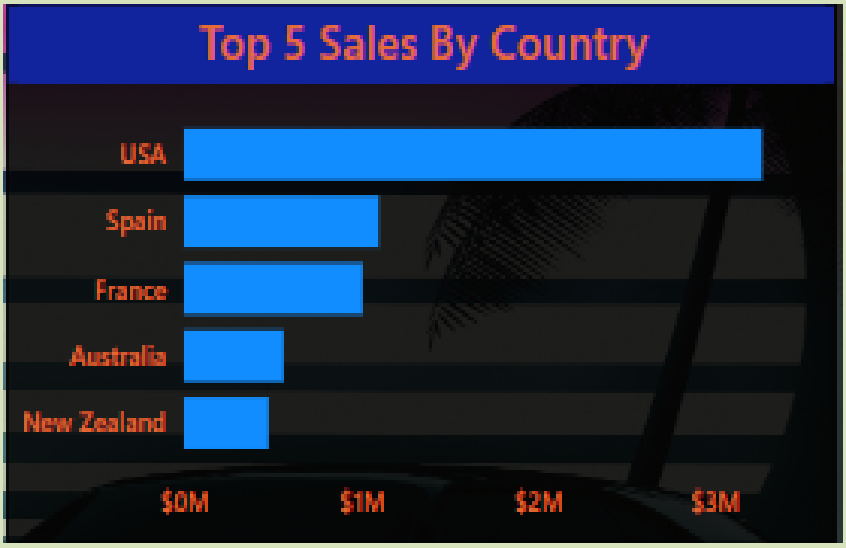
The revenue made by the company from year 2003 to 2005 is 9.6 million dollars.

The Average days taken to deliver the Product is 4 days.

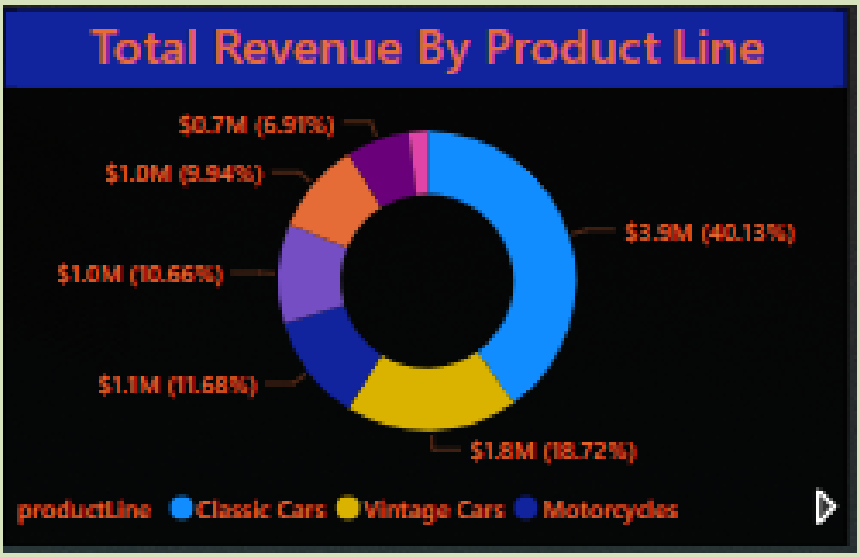
Let's now analyze the sales report and the volume of products sold between 2003 and 2005.



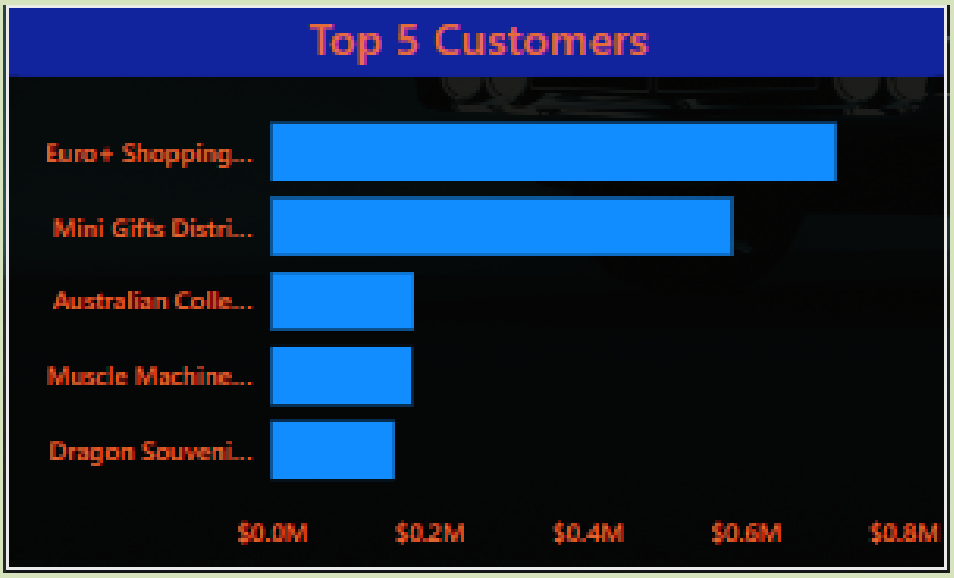
The relationship between Sales and Product line is depicted in the graph above. The sales of classic cars are higher than those of other product lines, which have a $ 3853922 sales figure. Vintage cars are second in sales, while Motorcycles are third. At $3,853,922, Classic Cars had the highest Sum of Total Revenue and was 1,944.16% higher than Trains, which had the lowest Sum of Total Revenue at $188,533.



The top 5 countries by sales are represented in the graph above, with the United States leading Spain, France, Australia, and New Zealand. The total sales happened in USA from 2003- 2004 is $ 3,2732,80.



The pie chart clearly illustrates the overall revenue earned by each product line for Axon. The percentage of each product line's overall revenue is displayed, with Classic Cars occupying the greater portion of the space on the graph followed by other Product lines and Train being the least. Classic Cars accounted for 40.13% of Sum of Total Revenue.



The top 5 customers who have bought products from Axon are represented in the graph above. This enables us to focus on our most valuable clients by making additional offers and benefits available to them.

The relationship between the quantity ordered by country and product line is shown on the map above. As far as we can tell, most orders come from the USA.

The total quantity ordered by product line is shown on the graph. The Trains are in the lowest order while Classic cars are at the top spot.

The sales pattern by year and month may be clearly seen in the line graph. Compared to 2005, the sales figures for the years 2003 and 2004 were strong.

The decomposition tree visual in Power BI lets you visualize data across multiple dimensions. It automatically aggregates data and enables drilling down into your dimensions in any order. We can see the overall revenue produced by the various categories, such as country, product line, and state, in the decomposition tree shown above.

Conclusion

In conclusion, this project provides a comprehensive analysis of the sales performance of Axon Company for the years 2003 to 2005, offering valuable insights into the trends, patterns, and opportunities within this critical period.