**AXON SALES REPORT**

**Briefing:**

This project aims to revolutionize Axon, a classic cars retailer, by implementing a robust Business Intelligence (BI) solution using Microsoft PowerBI and MySQL. Axon currently faces challenges in managing and analyzing their sales data efficiently, leading to a lack of accurate and timely sales reports. The goal is to design and implement a user-friendly BI solution that empowers the sales team and management to seamlessly manage and analyze their sales data.

The MySQL sample database includes essential tables such as Customers, Products, Orders, Payments, Employees, and more, providing a comprehensive dataset for analysis. By leveraging PowerBI's visualization capabilities and SQL's data management features, this solution intends to enhance Axon's decision-making process. Measures such as total revenue, total orders, average order value, and total profit will be crucial in providing actionable insights. Additionally, exploring customer demographics, top product lines, and yearly sales trends will further contribute to a successful BI implementation, ultimately facilitating informed decision-making for Axon's management.

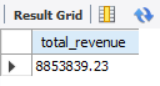
**MySQL Queries:**

**Primary KPIs**

Total revenue

select sum(amount) as total\_revenue

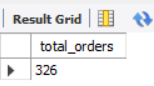
from payments;



Total orders

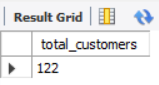
select count(distinct orderNumber) as total\_orders

from orders;



Total customers

select count(distinct customerNumber) as total\_customers from customers;



Total\_profit

SELECT

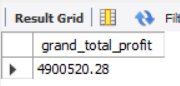
SUM((MSRP - buyPrice) \* quantityOrdered) AS grand\_total\_profit

FROM

products

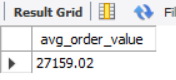
JOIN

orderdetails ON products.productCode = orderdetails.productCode;



Average order value

select round(sum(amount)/ (select count(distinct orderNumber) from orders),2) as avg\_order\_value from payments;



**Other Measures**

Daily Trend for Total Orders

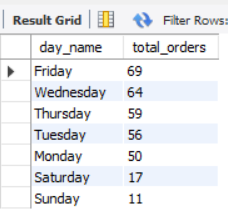
select dayname(orderDate) as day\_name, count(distinct o.orderNumber) as total\_orders

from orders o

join customers c on o.customerNumber = c.customerNumber

group by 1

order by 2 desc;



Monthly Trend for Total Orders

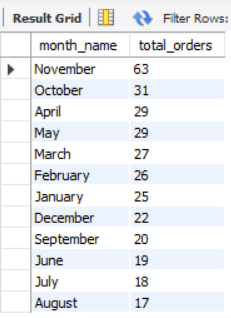
select monthname(orderDate) as month\_name, count(distinct o.orderNumber) as total\_orders

from orders o

join customers c on o.customerNumber = c.customerNumber

group by 1

order by 2 desc;



Percent Sales by Product Line

select round((sub.order\_count/sum(sub.order\_count) over())\*100,2) as percent\_of\_total

from

(SELECT

productLine,

ROUND((COUNT(DISTINCT orderNumber)\*100) / (SELECT COUNT(DISTINCT orderNumber) FROM orders), 2) as percentage\_sales,

count(distinct orderNumber) as order\_count

FROM

products p

JOIN

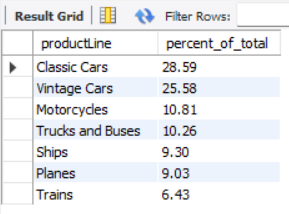
orderdetails od ON p.productCode = od.productCode

GROUP BY

productLine

ORDER BY

percentage\_sales DESC)sub;



Customer Distribution by Country

select country, count(distinct customerNumber) as customer\_count

from customers

group by country

order by count(distinct customerNumber) desc;



Percent profit by Product Line

select pl.productLine, SUM((MSRP - buyPrice) \* quantityOrdered) as profit,

round((SUM((MSRP - buyPrice) \* quantityOrdered)/

(SELECT SUM((MSRP - buyPrice) \* quantityOrdered)

FROM

products

JOIN

orderdetails ON products.productCode = orderdetails.productCode)\*100),2) as percent\_of\_total

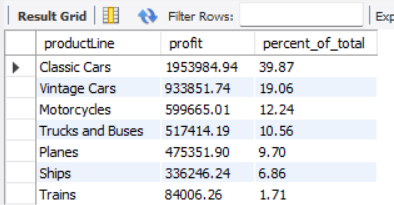
from products p

join orderdetails od on p.productCode = od.productCode

join productlines pl on p.productLine = pl.productLine

group by pl.productLine

order by 2 desc;



Top 10 Employees by Total Sales

select sub.emp\_name, sub.total\_orders,

round(((sub.total\_orders/ sum(sub.total\_orders) over())\*100),2) as percent\_of\_total

from

(select e.employeeNumber, e.firstName as emp\_name, count(distinct od.orderNumber) as total\_orders

from customers c

join orders o on c.customerNumber = o.customerNumber

join orderdetails od on o.orderNumber = od.orderNumber

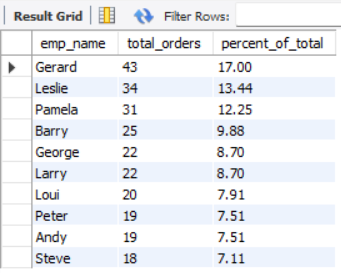
right join employees e on c.salesRepEmployeeNumber = e.employeeNumber

where e.jobTitle = 'Sales Rep'

group by 1,2

order by 3 desc

limit 10)sub;



Credit Limit VS Total Orders

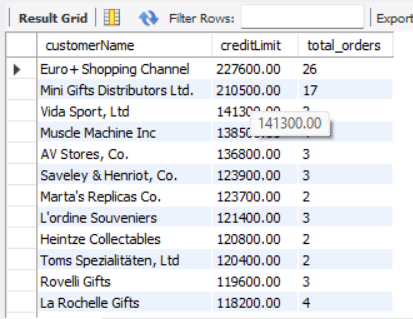
Select c.customerName, c.creditLimit, count(distinct o.orderNumber) as total\_orders

from customers c

left join orders o on c.customerNumber = o.customerNumber

group by 1,2

order by 2 desc;



Yearly Trend of Total Profit

SELECT year(orders.orderDate) as order\_year,

SUM((products.MSRP - products.buyPrice) \* orderdetails.quantityOrdered) as yearly\_profit

FROM

products

JOIN

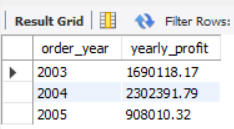
orderdetails ON products.productCode = orderdetails.productCode

JOIN

orders ON orders.orderNumber = orderdetails.orderNumber

group by 1

order by 1;



Top 5 Products by Total Profit

SELECT

products.productName,

SUM((products.MSRP - products.buyPrice) \* orderdetails.quantityOrdered) as total\_profit

FROM

products

JOIN

orderdetails ON products.productCode = orderdetails.productCode

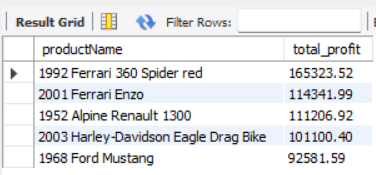
JOIN

orders ON orders.orderNumber = orderdetails.orderNumber

group by 1

order by 2 desc

limit 5;



Bottom 5 Products by Total Profit

SELECT

products.productName,

SUM((products.MSRP - products.buyPrice) \* orderdetails.quantityOrdered) as total\_profit

FROM

products

JOIN

orderdetails ON products.productCode = orderdetails.productCode

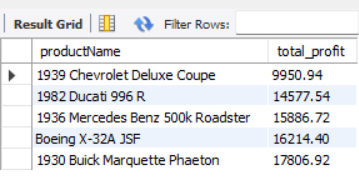
JOIN

orders ON orders.orderNumber = orderdetails.orderNumber

group by 1

order by 2

limit 5;



Top 5 Products Contribution in total Profit

select sum(sub.total\_profit) over() as total\_profit\_of\_top\_5,

round(((sum(sub.total\_profit) over()/

(SELECT

SUM((MSRP - buyPrice) \* quantityOrdered) AS grand\_total\_profit

FROM

products

JOIN

orderdetails ON products.productCode = orderdetails.productCode))\*100),2) as percent\_of\_total

from

(SELECT

products.productName,

SUM((products.MSRP - products.buyPrice) \* orderdetails.quantityOrdered) as total\_profit

FROM

products

JOIN

orderdetails ON products.productCode = orderdetails.productCode

JOIN

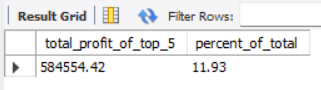
orders ON orders.orderNumber = orderdetails.orderNumber

group by 1

order by 2 desc

limit 5)sub

limit 1;



Bottom 5 Products Contribution in Total Profit

select sum(sub.total\_profit) over() as total\_profit\_of\_bottom\_5,

round(((sum(sub.total\_profit) over()/

(SELECT

SUM((MSRP - buyPrice) \* quantityOrdered) AS grand\_total\_profit

FROM

products

JOIN

orderdetails ON products.productCode = orderdetails.productCode))\*100),2) as percent\_of\_total

from

(SELECT

products.productName,

SUM((products.MSRP - products.buyPrice) \* orderdetails.quantityOrdered) as total\_profit

FROM

products

JOIN

orderdetails ON products.productCode = orderdetails.productCode

JOIN

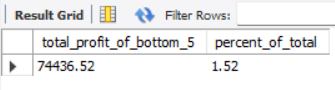
orders ON orders.orderNumber = orderdetails.orderNumber

group by 1

order by 2

limit 5)sub

limit 1;



**POWER BI:**

Data Cleaning & Processing

• Dropped a few columns (e.g. images, addressline 2) which were mostly null values & of no use to the report.

• There were many columns that was of no use & could have been dropped without an issue.

However I decided to keep those intact in order to preserve data integrity as much as possible.

• Converted Null values of PostalCode column to N/A.

• Added a few columns in power query, such as dayname, monthname, daynumber, monthnumber, orderyear.

• Add few calculated columns, such as orderday, ordermonth etc.

• Created a separate table for my measures.

Data Modeling

Post importing the data base it was already in snow flake schema. In order to make the relationships direct I decided to convert the snow flake schema into a star schema. I achieved the star schema by merging the sub-dimension tables with the dimension tables. Star schema in turn helped me build the visualizations easily due to more straight forward relationships between the tables.

DAX

Total Revenue = SUM('classicmodels payments'[amount])

Total Orders = DISTINCTCOUNT('classicmodels orders'[orderNumber])

Total Customers = DISTINCTCOUNT('classicmodels customers'[customerNumber])

Average Order Value = ([Total Revenue]/[Total Orders])

TotalProfit =

SUMX(

'classicmodels orders',

(('classicmodels orders'[classicmodels products.MSRP]) - ('classicmodels orders'[classicmodels products.buyPrice])) \* 'classicmodels orders'[classicmodels orderdetails.quantityOrdered]

)

Top 5 Profitable Product = RANKX(ALL('classicmodels orders'[classicmodels products.productName]), [TotalProfit], ,DESC, Dense)

Least 5 Profitable Product = RANKX(ALL('classicmodels orders'[classicmodels products.productName]), [TotalProfit], ,ASC, Dense)

SUM of Top 5 Profitable Product Total Profit =

CALCULATE(

SUMX(

TOPN(5, ALL('classicmodels orders'[classicmodels products.productName]), [TotalProfit], DESC),

[TotalProfit]

)

)

SUM of Bottom 5 Products total profit =

CALCULATE(

SUMX(

TOPN(5, ALL('classicmodels orders'[classicmodels products.productName]), [TotalProfit], ASC),

[TotalProfit]

)

)

Visualizations

1. Card: All Primary KPIs
2. Bar Chart: Daily Trend for Total Orders
3. Area Chart: Monthly Trend for Total Orders
4. Donut Chart: Percentage Sales by Product Line
5. Treemap: Distribution of Customers by Country
6. Donut Chart: Percentage Profit by Product Line
7. Scatter Plot(Bubble): Credit Limit VS Total Orders
8. Donut Chart: Top 10 Employees by Total Sales
9. Waterfall Chart: Yearly Trend of Total Profit
10. Horizontal Bar Chart: Top 5 Products by Total Profit
11. Donut Chart: Top 5 Products Contribution in Total Profit
12. Horizontal Bar Chart: Bottom 5 Products by Total Profit
13. Donut Chart: Bottom 5 Products Contribution in Total Profit

Inferences:

1. Orders are highest on Wednesday and Friday.
2. Orders are lowest on Saturday and Sunday.
3. The monthly trend of sales is quite linear from January to September with a slight dip from May to August, then there is quite a substantial surge from October to November followed by a dip in December.
4. From the monthly trend we can tell that people are buying a great amount during the holiday season, Axon should consider releasing some discount and offers during that period going forward.
5. Among the product lines Classic Cars is the most profitable followed by Vintage cars and the least profitable is Trains.
6. From the scatter plot we can tell that there is no linear relationship between Credit Limit and Total Orders. Most Customers are in the credit limit range of 80K to 130K with total orders 2 to 5. Euro+ Shopping Channel highest credit limit 227k with total orders 26.
7. Top Performing sales representative is Gerard with 43 sales followed by Leslie 34 sales and Pamela 31 sales. Steven being the last of the top 10 employees has 18 sales.
8. Sales of Axon is increasing each year, in 2003 it was 1.7M, followed by 2004 2.3M and in just 6 moths of 2005 they made a profit of 0.9M and from monthly trend we can tell it will cross 2004 profit mark as the surge months (October-November) are yet to come.
9. Top 5 products contributes 10.66% of total profit, highest contributing product being 1992 Ferrari 360 Spider Red (165K).
10. Bottom 5 products contributes 1.5% of total profit, lowest contributing product being 1939 Chevrolet Deluxe Coupe (10K).

Note: (All monetary values are in US dollars.)

Thank you for taking the time to review this project. Your attention and interest in my effort is truly appreciated. If you have any questions or require further clarification, feel free to reach out. Your engagement adds value to this endeavor, and your time is valued.

Thanks again for your consideration.

Sourasatya Roy

www.linkedin.com/in/sourasatya-roy-b5a852296