SQL - Capstone Project

The Capstone Project aims to gain insights into Amazon sales data and understand the factors affecting sales across different branches.

by sanjay nayak

Product Analysis

Product Lines

Identify the different product lines and their performance.

Top Performers

Analyze the best-performing product lines.

Areas for Improvement

Identify product lines that need improvement.

Product Analysis

Explore Unique Product Lines:

Identify distinct product lines in the dataset.

```
SELECT DISTINCT product_line FROM amazon;
```

Sales Performance by Product Line:

Analyze total sales and profit for each product line.

```
SELECT product_line, SUM(total) AS total_sales,
SUM(gross_income) AS total_profit FROM amazon
GROUP BY product_line ORDER BY total_sales DESC;
```

Sales Analysis

1

Sales Trends

Analyze sales trends to measure strategy effectiveness.

2

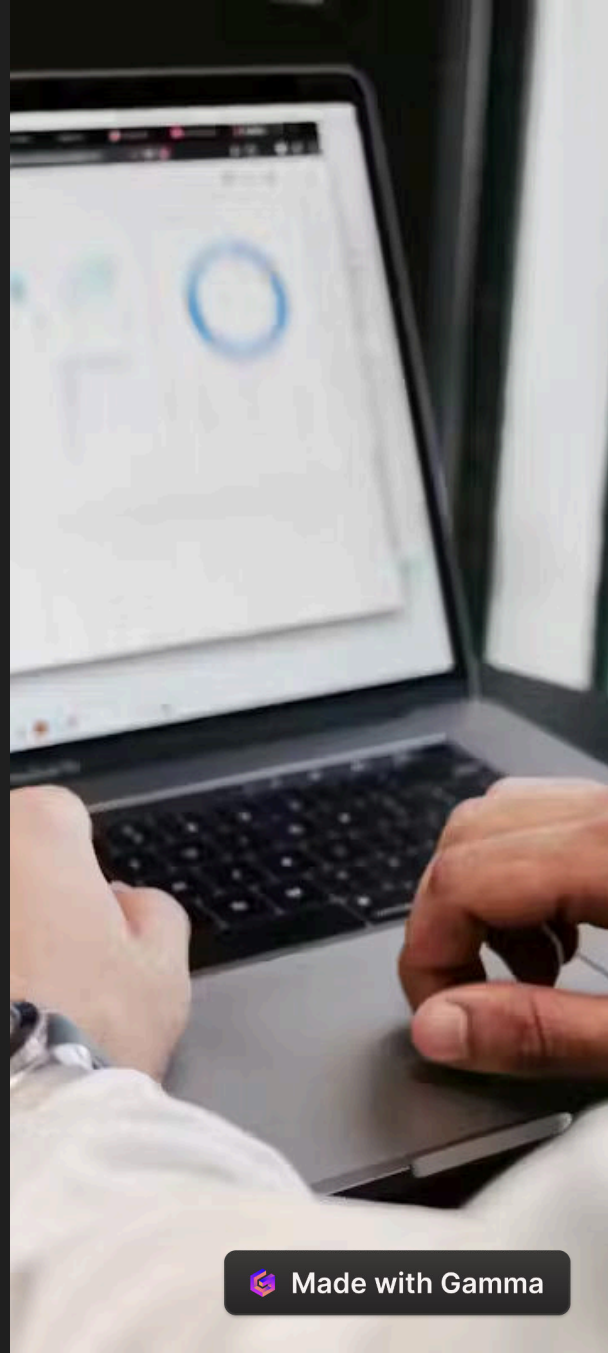
Strategy Effectiveness

Evaluate the effectiveness of each sales strategy.

3

Modifications

Identify necessary modifications for increased sales.



Sales Analysis

Monthly Sales Trends:

Analyze sales trends on a monthly basis.

```
SELECT
MONTHNAME(str_to_date(date,'
%d-%m-%y')) AS month,
SUM(total) AS monthly_sales
FROM amazon GROUP BY month
ORDER BY monthly_sales DESC;
```

Time of Day Sales Analysis:

Explore sales trends based on the time of day.

```
SELECT timeofday, COUNT(*) AS
sales_occurrences FROM
amazon GROUP BY timeofday
ORDER BY sales_occurrences
DESC;
```

Branch-wise Sales Comparison:

Compare sales performance across different branches.

```
SELECT branch, SUM(total) AS
branch_sales FROM amazon
GROUP BY branch ORDER BY
branch_sales DESC;
```

Customer Analysis

Customer Segments

Identify and profile different customer segments.

Purchase Trends

Analyze trends in customer purchases.

Profitability

Measure the profitability of each customer segment.

Customer Analysis:

Customer Segmentation:

Identify different customer segments based on the type of customer.

```
SELECT customer_type, COUNT(*)
AS customer_count FROM
amazon GROUP BY
customer_type;
```

Purchase Trends by Customer Type:

Analyze purchase trends and profitability for each customer segment.

```
SELECT customer_type, COUNT(*)
AS purchase_count, SUM(total)
AS total_purchase FROM amazon
GROUP BY customer_type
ORDER BY total_purchase DESC;
```

Customer Rating Analysis:

Explore average ratings given by different customer segments.

```
SELECT customer_type,
AVG(rating) AS average_rating
FROM amazon GROUP BY
customer_type;
```

Feature Engineering

1

Insight of Sales

Add a new column named timeofday to give insight of sales in the Morning, Afternoon and Evening. This will help answer the question on which part of the day most sales are made.

2

days of the week

Add a new column named dayname that contains the extracted days of the week on which the given transaction took place (Mon, Tue, Wed, Thur, Fri). This will help answer the question on which week of the day each branch is busiest.

3

months of the year

Add a new column named monthname that contains the extracted months of the year on which the given transaction took place (Jan, Feb, Mar).

Help determine which month of the year has the most sales and profit.

Feature Engineering Code

Add the column "timeofday" in existing table amazon.

```
ALTER TABLE amazon

ADD COLUMN timeofday VARCHAR(10);

SET SQL_SAFE_UPDATES = 0;  -- Disable safe update mode temporarily

UPDATE amazon

SET timeofday = CASE

WHEN HOUR(time) >= 5 AND HOUR(time) < 12 THEN 'Morning'

WHEN HOUR(time) >= 12 AND HOUR(time) < 17 THEN 'Afternoon'

WHEN HOUR(time) >= 17 AND HOUR(time) < 24 THEN 'Evening'

ELSE 'Night'

END

WHERE 1=1

LIMIT 1000;

SET SQL_SAFE_UPDATES = 1;  -- Enable safe update mode back
```

Add the column "dayname" in existing table amazon.

```
ALTER TABLE amazon ADD COLUMN dayname VARCHAR(10);
SET SQL_SAFE_UPDATES = 0;

UPDATE amazon SET dayname = CASE

 WHEN DAYNAME(STR_TO_DATE(date, '%d-%m-%Y')) =
'Monday' THEN 'mon'

WHEN DAYNAME(STR_TO_DATE(date, '%d-%m-%Y')) =
'Tuesday' THEN 'tue'

WHEN DAYNAME(STR_TO_DATE(date, '%d-%m-%Y')) =
'Wednesday' THEN 'wed'

WHEN DAYNAME(STR_TO_DATE(date, '%d-%m-%Y')) =
'Thursday' THEN 'thru'

WHEN DAYNAME(STR_TO_DATE(date, '%d-%m-%Y')) = 'Friday'
THEN 'fir'

WHEN DAYNAME(STR_TO_DATE(date, '%d-%m-%Y')) =
'Saturday' THEN 'sat' ELSE 'sunday' END WHERE 1=1 LIMIT
1000

SET SQL_SAFE_UPDATES = 1;
```

```
# Add the column "monthname" in existing table amazon.
ALTER TABLE amazon
ADD COLUMN monthname VARCHAR(10);

SET SQL_SAFE_UPDATES = 0;
UPDATE amazon
SET monthname = CASE
   WHEN MONTHNAME(STR_TO_DATE(date, '%d-%m-%Y')) = 'january' THEN 'Jan'
   WHEN MONTHNAME(STR_TO_DATE(date, '%d-%m-%Y')) = 'febuary' THEN 'Feb'
   ELSE 'Mar'
END
WHERE 1=1
LIMIT 1000
SET SQL_SAFE_UPDATES = 1;
```

Business Questions To Answer:

1) What is the count of distinct cities in the dataset?

```
SELECT count(DISTINCT city) AS distinct_city_count FROM amazon;

-- select distinct city, count(*) as count_is from amazon group by city;
```

2) For each branch, what is the corresponding city?

```
SELECT branch,city FROM amazon;
```

3) What is the count of distinct product lines in the dataset?

```
SELECT COUNT(DISTINCT product_line) AS distinct_product_line_count FROM amazon;
```

4) Which payment method occurs most frequently?

```
SELECT payment,count(*) AS payment_frequency FROM amazon GROUP BY payment ORDER BY payment_frequency DESC LIMIT 1;
```

5) Which product line has the highest sales?

```
SELECT product_line, SUM(total) AS highest_sales FROM amazon GROUP BY product_line ORDER BY highest_sales DESC LIMIT 1;
```

6) How much revenue is generated each month?

```
SELECT MONTHNAME(str_to_date(date,'%d-%m-%y')) AS month_name, SUM(total) AS monthly_revenue FROM amazon GROUP BY month_name ORDER BY monthly_revenue DESC;
```

7) In which month did the cost of goods sold reach its peak?

```
SELECT MONTHNAME(str_to_date(date,'%d-%m-%y')) AS peak_month, MAX(cogs) AS peak_cogs FROM amazon GROUP BY peak_month LIMIT 1;
```

8) Which product line generated the highest revenue?

```
SELECT product_line, SUM(total) AS total_revenue FROM amazon GROUP BY product_line ORDER BY total_revenue DESC LIMIT 1;
```

9) In which city was the highest revenue recorded?

```
SELECT city, SUM(total) AS total_revenue FROM amazon GROUP BY city ORDER BY total_revenue DESC LIMIT 1;
```

10) Which product line incurred the highest Value Added Tax?

```
SELECT product_line, MAX(VAT) AS highest_VAT FROM amazon GROUP BY product_line ORDER BY highest_VAT desc;
```

11) For each product line, add a column indicating "Good" if its sales are above average, otherwise "Bad."

```
SELECT product_line,total, CASE WHEN total>(SELECT AVG(total) from amazon) THEN 'GOOD' ELSE 'BAD' END AS sales_status from amazon;
```

12) I**dentify the branch that exceeded the average number of products sold**

```
SELECT branch FROM amazon GROUP BY branch HAVING AVG(Quantity)>(select AVG(quantity) FROM amazon);
```

13) Which product line is most frequently associated with each gender?

```
SELECT product_line,gender, COUNT(*) AS frequency FROM amazon GROUP BY gender,product_line ORDER BY frequency DESC;
```

14) Calculate the average rating for each product line.

```
SELECT product_line,AVG(rating) AS average_rating From amazon GROUP BY product_line
```

15) Count the sales occurrences for each time of day on every weekday.

```
SELECT DAYNAME(str_to_date(date,'%d-%m-%y')) AS day, timeofday,count(*) AS sales_occurrences FROM amazon GROUP BY day,timeofday ORDER BY day, sales_occurrences DESC;
```

16) Identify the customer type contributing the highest revenue.

```
SELECT customer_type, SUM(total) AS highest_revenue FROM amazon GROUP BY customer_type ORDER BY highest_revenue DESC LIMIT 1;
```

17) Determine the city with the highest VAT percentage.

```
SELECT city, AVG(VAT/total)*100 AS VAT_percentage FROM amazon GROUP BY city ORDER BY VAT_percentage DESC LIMIT 1;
```

18) Identify the customer type with the highest VAT payments.

```
SELECT customer_type, SUM(VAT) AS total_VAT_payments FROM amazon GROUP BY customer_type ORDER BY total_VAT_payments DESC LIMIT 1;
```

19) What is the count of distinct customer types in the dataset?

```
SELECT COUNT(DISTINCT customer_type) AS distinct_customer_type_count FROM amazon;
```

20) What is the count of distinct payment methods in the dataset?

```
SELECT count(DISTINCT payment) AS distinct_payment FROM amazon;
```

21) Which customer type occurs most frequently?

```
SELECT customer_type, COUNT(*) AS customer_type_count FROM amazon GROUP BY customer_type ORDER BY customer_type_count DESC LIMIT 1;
```

22) Identify the customer type with the highest purchase frequency.

```
SELECT customer_type, COUNT(*) AS purchase_frequency FROM amazon GROUP BY customer_type ORDER BY purchase_frequency DESC LIMIT 1;
```

23) Determine the predominant gender among customers.

```
SELECT gender, COUNT(*) AS gender_count FROM amazon GROUP BY gender ORDER BY gender_count DESC LIMIT 1;
```

24) Examine the distribution of genders within each branch.

```
SELECT branch, gender, COUNT(*) AS gender_count FROM amazon GROUP BY branch, gender ORDER BY branch, gender_count DESC;
```

25) Identify the time of day when customers provide the most ratings.

```
SELECT timeofday, COUNT(*) AS rating_count FROM amazon GROUP BY timeofday ORDER BY rating_count DESC LIMIT 1;
```

26) Determine the time of day with the highest customer ratings for each branch.

```
SELECT branch, timeofday, AVG(rating) AS average_rating FROM amazon GROUP BY branch, timeofday ORDER BY branch, average_rating DESC;
```

27) Identify the day of the week with the highest average ratings.

```
SELECT DAYNAME(str_to_date(date,'%d-%m-%y')) AS day, AVG(rating) AS average_rating FROM amazon GROUP BY day ORDER BY average_rating DESC LIMIT 1;
```

28) Determine the day of the week with the highest average ratings for each branch.

```
SELECT branch,DAYNAME(str_to_date(date,'%d-%m-%y')) AS day, AVG(rating) AS average_rating FROM amazon GROUP BY day,branch ORDER BY average_rating DESC LIMIT 3;
```

Thank You