**GPT + Excel + SQL + Power BI Project**

**End-to-End Sales Analysis**

This project showcases how I effectively utilized **GPT, Excel, PostgreSQL, and Power BI** in an integrated manner to perform end-to-end sales analysis.

**Tools and Technologies Used:**

**1)GPT**   
Generated synthetic sales data for 5 major countries — **Canada, China, India, the US, and the UK** — in separate .csv files using GPT. This simulated dataset was used for further analysis.

**2) Microsoft Excel:**  
Performed initial data exploration on all five datasets to understand the column structure and perform basic insights extraction.

**3)PostgreSQL:**  
Imported all Excel datasets into a PostgreSQL database. Conducted thorough data exploration, data cleaning, and answered **14 critical business questions** using advanced SQL queries.

**4)Database Integration with Power BI:**  
Established a live connection between the PostgreSQL database and Power BI for seamless data flow and visualization.

**5) Power BI:**  
Designed a visually appealing and interactive **Sales Dashboard** featuring key performance indicators (KPIs), trends, and country-wise sales comparisons etc.

**SQL DATA CLEANING, EXPLORATION AND BUSINESS QUESTIONS**

**Importing csv files from local machine into PostgreSQL**

select \*

from "Sales Canada";

select \*

from "Sales China";

select \*

from "Sales India";

select \*

from "Sales UK";

select \*

from "Sales US";

*NOW WILL UNION ALL THE 5 TABLES SO TO FORM 1 LARGER TABLE*

CREATE TABLE sales\_data as

select \* from "Sales Canada"

UNION ALL

select \* from "Sales China"

UNION ALL

select \* from "Sales India"

UNION ALL

select \* from "Sales UK"

UNION ALL

select \* from "Sales US";

*INITIAL DATA EXPLORATION*

SELECT \* FROM sales\_data;

TOP 5 ROWS

SELECT \* FROM sales\_data LIMIT 5

BOTTOM 5 ROWS

SELECT \*

FROM sales\_data

ORDER BY "Transaction ID" DESC

LIMIT 5

CHECK FOR DATA TYPES

SELECT column\_name, data\_type

FROM information\_schema.columns

WHERE table\_name = 'sales\_data';

*DATA CLEANING*

Checking for missing values

select \*

from sales\_data

where

"Country" is null

or "Cost Price" is null

or " Discount Applied" is null

or " Quantity Purchased" is null

or "Price Per Unit" is null;

There's only one missing value in "Price Per Unit" Column will update it.

Updating "Price Per Unit"

UPDATE sales\_data

SET "Price Per Unit" = (SELECT AVG("Price Per Unit")

FROM sales\_data)

WHERE "Transaction ID" = '001898f7-b696-4356-91dc-8f2b73d09c63';

Checking for duplicates records

SELECT "Transaction ID", COUNT(\*)

FROM sales\_data

GROUP BY 1

HAVING COUNT(\*) > 1;

No duplicates in the dataset

Removing the inconsistencies present in the dataset

SELECT

column\_name,

LENGTH(column\_name) AS actual\_length,

LENGTH(TRIM(column\_name)) AS trimmed\_length

FROM information\_schema.columns

WHERE table\_name = 'sales\_data';

From here it is clear only two columns has leading spaces i.e.

" Quantity Purchased" and " Discount Applied"

ALTER TABLE sales\_data RENAME COLUMN "quantity\_purchased" TO "Quantity Purchased";

ALTER TABLE sales\_data RENAME COLUMN "discount\_applied" TO "Discount Applied";

Converting all columns to lowercases to avoid writing column names in quotes

again and again.

ALTER TABLE sales\_data RENAME COLUMN "Transaction ID" TO transaction\_id;

ALTER TABLE sales\_data RENAME COLUMN "Date" TO date;

ALTER TABLE sales\_data RENAME COLUMN "Country" TO country;

ALTER TABLE sales\_data RENAME COLUMN "Product ID" TO product\_id;

ALTER TABLE sales\_data RENAME COLUMN "Product Name" TO product\_name;

ALTER TABLE sales\_data RENAME COLUMN "Category" TO category;

ALTER TABLE sales\_data RENAME COLUMN "Price Per Unit" TO price\_per\_unit;

ALTER TABLE sales\_data RENAME COLUMN "Quantity Purchased" TO quantity\_purchased;

ALTER TABLE sales\_data RENAME COLUMN "Cost Price" TO cost\_price;

ALTER TABLE sales\_data RENAME COLUMN "Discount Applied" TO discount\_applied;

ALTER TABLE sales\_data RENAME COLUMN "Payment Method" TO payment\_method;

ALTER TABLE sales\_data RENAME COLUMN "Customer Age Group" TO customer\_age\_group;

ALTER TABLE sales\_data RENAME COLUMN "Customer Gender" TO customer\_gender;

ALTER TABLE sales\_data RENAME COLUMN "Store Location" TO store\_location;

ALTER TABLE sales\_data RENAME COLUMN "Sales Representative" TO sales\_representative;

SELECT \* FROM sales\_data

Adding a 'total\_amount' column

ALTER TABLE sales\_data ADD total\_amount numeric

Updating total\_amount column

UPDATE sales\_data SET total\_amount = (price\_per\_unit \* quantity\_purchased) - discount\_applied;

Adding a 'profit' column

ALTER TABLE sales\_data ADD profit numeric

Updating profit column

UPDATE sales\_data SET profit = total\_amount - (cost\_price\*quantity\_purchased);

**Business Questions**

Q1.What is the total revenue generated?

SELECT SUM(total\_amount) as total\_revenue

FROM sales\_data;

Q2.Which product sold the most units?

SELECT product\_name , sum(quantity\_purchased) as most\_units\_sold

FROM sales\_data

GROUP BY 1

ORDER BY 2 DESC

LIMIT 1;

Q3.What are the top 3 countries by total revenue and profit?

SELECT country, sum(total\_amount) as total\_revenue , sum(profit) as net\_profit

FROM sales\_data

GROUP BY 1

ORDER BY 2 DESC, 3

LIMIT 3;

Q4.Which customer gender contributes more to total revenue?

SELECT customer\_age\_group, customer\_gender, sum(total\_amount) as total\_revenue

FROM sales\_data

GROUP BY 1,2

ORDER BY 3 DESC;

Q5.Which payment method was used most frequently store location wise?

SELECT COUNT(\*) as usage\_count, payment\_method, store\_location

FROM sales\_data

GROUP BY 2,3

ORDER BY 1 DESC;

SELECT \* FROM sales\_data

Q6.What is the average discount offered per product category?

SELECT category , ROUND(AVG(discount\_applied),2) as discount\_offered

FROM sales\_data

GROUP BY 1

ORDER BY 2 DESC

Q7.What is the monthly sales trend?

SELECT TO\_CHAR(date, 'Month') AS month\_name, sum(total\_amount) as sales

FROM sales\_data

GROUP BY 1

ORDER BY 2 DESC

Q8.What are the top 3 products by revenue within each category?

WITH CTE AS

(SELECT category,

product\_name,

sum(total\_amount),

rank()

over(partition by category order by sum(total\_amount) desc ) as rnk

FROM sales\_data

GROUP BY 1,2)

SELECT \*

FROM CTE

WHERE rnk<=3

Q9.What is month-on-month revenue growth?

WITH monthly\_revenue AS

(SELECT EXTRACT(MONTH FROM date) as month ,

sum(total\_amount) as revenue

FROM sales\_data

GROUP BY 1)

SELECT month,

revenue,

LAG(revenue) OVER(ORDER BY month) as prev\_month ,

(revenue - LAG(revenue) OVER(ORDER BY month))\*100.0 /

(LAG(revenue) OVER(ORDER BY month)) as growth\_percent

FROM monthly\_revenue;

Q10.Identify top 5 customers who bought most frequently

SELECT customer\_name, total\_orders

FROM (

SELECT customer\_gender || ' - ' || customer\_age\_group AS customer\_name,

COUNT(\*) AS total\_orders

FROM sales\_data

GROUP BY 1

) sub

ORDER BY total\_orders DESC

LIMIT 5;

Q11.Find products with increasing revenue trend over 3 months

SELECT \* FROM sales\_data

WITH monthly\_prod\_sales AS

(SELECT product\_name, EXTRACT(MONTH FROM date) as month ,

sum(total\_amount) as revenue

FROM sales\_data

GROUP BY 1,2

ORDER BY 2 ),

prev\_sales AS

( SELECT \*,

LAG(revenue, 1) OVER (PARTITION BY product\_name ORDER BY month) AS prev1,

LAG(revenue, 2) OVER (PARTITION BY product\_name ORDER BY month) AS prev2

FROM monthly\_prod\_sales

)

SELECT product\_name, month, revenue

FROM prev\_sales

WHERE revenue > prev1 AND prev1 < prev2;

Q12.Calculate revenue contribution % by store location

SELECT \* FROM sales\_data

WITH location\_revenue AS

(SELECT DISTINCT store\_location , sum(total\_amount) as revenue

FROM sales\_data

GROUP BY 1)

SELECT store\_location , revenue, revenue \*100.0/ sum(revenue) over() AS contribution\_percent

FROM location\_revenue

ORDER BY 3 desc;

Q13.List down all the over-discounted products (above category average)

SELECT product\_name , category, discount\_applied

FROM sales\_data

WHERE discount\_applied > (SELECT AVG(discount\_applied) as discount\_on\_category

FROM sales\_data

)

Q14.Find products with declining sales over the last 3 consecutive months

SELECT \* FROM sales\_data;

WITH monthly\_prod\_sales AS

(SELECT product\_name, EXTRACT(MONTH FROM date) as month ,

sum(total\_amount) as revenue

FROM sales\_data

GROUP BY 1,2

),

prev\_sales AS

( SELECT \*,

LAG(revenue, 1) OVER (PARTITION BY product\_name ORDER BY month) AS prev1,

LAG(revenue, 2) OVER (PARTITION BY product\_name ORDER BY month) AS prev2

FROM monthly\_prod\_sales

)

SELECT product\_name, month, revenue

FROM prev\_sales

WHERE revenue < prev1 AND prev1 < prev2;

**Power BI Dashboard:**

