



# **RETAIL SALES ANALYSIS** **Using SQL**

PROJECT BY : PRIYANSHU KUMAR



# PROJECT OVERVIEW

The "Retail Sales Analysis" project aims to analyze retail sales data to uncover key insights and trends that can drive better business decisions. Utilizing a dataset consisting of various attributes such as Product ID, Sales, Date, Quantity, and Customer ID, the project involves cleaning the data and conducting a detailed analysis using SQL queries. The primary focus is to identify total sales, top-selling products, customer purchasing behavior, and inventory management insights. Through this analysis, the project highlights sales performance trends over time, segments customers based on purchasing patterns, and provides actionable recommendations to optimize sales strategies and improve overall business performance.



# DATASETS

**transactions\_id:** A unique identifier for each transaction.

**sale\_date:** The date on which each sale occurred.

**sale\_time:** The time when the sale was made.

**customer\_id:** A unique identifier for each customer.

**gender:** The gender of the customer.

**age:** The age of the customer.

**category:** The category of the product sold (e.g., Clothing, Beauty).

**quantity:** The quantity of the product sold in each transaction.

**price\_per\_unit:** The price of each unit of the product sold.

**cogs (Cost of Goods Sold):** The cost associated with the goods sold in each transaction.

**total\_sale:** The total sales value for each transaction.



# Key Objectives

**Analyze Sales Performance:** Evaluate overall sales performance by analyzing total sales, identifying trends over time, and understanding seasonality to help the business make data-driven decisions.

**Identify Top-Selling Products:** Determine which products are the top performers and which are underperforming. This analysis helps in inventory management and strategic stocking of popular products.

**Understand Customer Behavior:** Analyze customer demographics and purchasing patterns to identify different customer segments, such as high-value customers, and understand their preferences and buying behavior.

**Optimize Inventory Management:** Use sales data to provide insights into inventory turnover and stock levels, helping the business maintain optimal inventory levels and reduce overstock or stockout situations.

**Improve Profitability:** Assess pricing strategies by examining price per unit, cost of goods sold (COGS), and total sales. This helps identify opportunities for improving profit margins and adjusting pricing strategies.

**Support Marketing Strategies:** Identify trends and patterns in customer purchasing behavior that can inform targeted marketing campaigns, promotional strategies, and customer retention efforts.





```
SELECT
    *
FROM
    retail_sales
WHERE
    sale_date = '2022-10-6'
    OR sale_date = '2022-10-07';
```

[illegible]



Q2. Write a SQL Query to retrieve all transactions where the category is 'Electronics' and the quantity sold is more than 3 in the month of Oct-2022.

```
SELECT
    *
FROM
    retail_sales
WHERE
    category = 'Electronics'
    AND DATE_FORMAT(sale_date, '%Y-%m') = '2022-10'
    AND quantiy >= 3;
```

Result Grid		 Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:			
	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
	54	2022-10-20	10:17:00	142	Female	38	Electronics	3	500	200	1500
	329	2022-10-27	07:42:00	52	Female	46	Electronics	4	25	8	100
	333	2022-10-06	08:15:00	21	Female	54	Electronics	4	300	99	1200
	334	2022-10-16	20:37:00	103	Male	31	Electronics	3	300	360	900
	431	2022-10-29	22:02:00	99	Male	63	Electronics	4	300	345	1200
	513	2022-10-21	18:31:00	85	Male	24	Electronics	4	25	24.5	100

Q3. Write a SQL Query to calculate the total sales for each category.

```
SELECT
    category, SUM(total_sale) AS Total_Sales
FROM
    retail_sales
GROUP BY category;
```

Result Grid     Filter Rows:		
	category	Total_Sales
▶	Beauty	286935
	Clothing	311140
	Electronics	313810

Q4. Write a SQL Query to find the average age of customers who purchased items from the 'Electronics' category.

```
SELECT
    ROUND(AVG(age), 2) AS avg_age
FROM
    retail_sales
WHERE
    category = 'Electronics';
```

Result Grid	
	avg_age
▶	41.49



Q5. Write a SQL Query to find all transactions where the total\_sale is greater than 1500.

```
SELECT
    *
FROM
    retail_sales
WHERE
    total_sale >= 1500;
```

Result Grid		Filter Rows:		Edit:	Export/Import:		Wrap Cell Content:				
	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
▶	13	2023-02-08	17:43:00	106	Male	22	Electronics	3	500	245	1500
	15	2022-07-01	11:50:00	75	Female	42	Electronics	4	500	210	2000
	16	2022-06-25	10:33:00	82	Male	19	Clothing	3	500	180	1500
	47	2022-10-22	17:22:00	96	Female	40	Beauty	3	500	600	1500
	54	2022-10-20	10:17:00	142	Female	38	Electronics	3	500	200	1500
	65	2022-12-11	20:03:00	84	Male	51	Electronics	4	500	160	2000
	72	2023-12-06	19:19:00	5	Female	20	Electronics	4	500	195	2000
	74	2023-10-05	19:50:00	56	Female	18	Beauty	4	500	205	2000
	78	2023-02-17	21:08:00	68	Female	47	Clothing	3	500	265	1500

Q6. Write a SQL Query to find the total number of transactions (transactions\_id) made by each gender in each category.

```
SELECT
    category,
    gender,
    COUNT(transactions_id) AS total_transactions
FROM
    retail_sales
GROUP BY category , gender
ORDER BY category;
```

Result Grid			
Filter Rows:			
	category	gender	total_transactions
▶	Beauty	Female	332
	Beauty	Male	282
	Clothing	Female	348
	Clothing	Male	354
	Electronics	Female	340
	Electronics	Male	344

Q7. Write a SQL Query to calculate the average sale for each month also find out the best selling month in each year.

```
select sale_year, sale_month, avg_sale
from (
select year(sale_date) as sale_year,
month(sale_date) as sale_month,
round(avg(total_sale),2) as avg_sale,
rank() over (partition by year(sale_date) order by avg(total_sale) desc) as rank1
from retail_sales
group by 1,2
) as t1
where rank1=1;
```

Result Grid			
Filter Rows:			
	sale_year	sale_month	avg_sale
▶	2022	7	530.12
	2023	2	535.53

Q8. Write a SQL Query to find the top 3 customers based on the highest total sales.

```
SELECT
    customer_id, SUM(total_sale) AS total_sale
FROM
    retail_sales
GROUP BY customer_id
ORDER BY total_sale DESC
LIMIT 3;
```

Result Grid			Filter Rows
	customer_id	total_sale	
▶	3	38440	
	1	30750	
	5	30405	

Q9. Write a SQL Query to find the number of unique customers who purchased items from each category.

```
SELECT
    category, COUNT(DISTINCT customer_id) AS total_customers
FROM
    retail_sales
GROUP BY category;
```

Result Grid			Filter Rows:
	category	total_customers	
▶	Beauty	141	
	Clothing	149	
	Electronics	144	

Q10. Write a SQL Query to create each shift and number of orders (Example Morning <= 12, Afternoon Between 12 and 17, Evening >=16).

```
with hourly_sale as
(select *, case
when hour(sale_time) < 12 then 'Morning'
when hour(sale_time) between 12 and 17 then 'Afternoon'
else 'Evening'
end as shift
from retail_sales)
select shift,
count(*) as total_orders from hourly_sale
group by shift order by total_orders desc;
```

Result Grid		Filter Rows:
	shift	total_orders
▶	Evening	1062
	Morning	561
	Afternoon	377



# CONCLUSION

the "Retail Sales Analysis" project provides valuable insights into the business's sales performance, customer behavior, and product trends. Through detailed analysis using SQL and data from the retail sales dataset, the project identifies key areas of strength, such as top-selling products and loyal customer segments, as well as opportunities for improvement, including inventory optimization and pricing strategies. By leveraging these findings, the business can enhance its decision-making processes, tailor marketing efforts more effectively, and ultimately improve profitability and customer satisfaction. The project underscores the importance of data-driven strategies in retail management to drive sustainable growth and success.