

## Project Title

# Retail Sales Data Analysis Using SQL

The Retail Sales Analysis project focuses on leveraging SQL to analyze transactional data and uncover insights into customer behavior, sales trends, and product performance. The dataset contains detailed information about transactions, including customer demographics, sales data, and product categories. This project aims to assist businesses in making informed decisions by identifying key business metrics and trends.

## Project Overview

In this project, SQL queries were used to analyze and process data stored in a retail sales dataset. The analysis includes data cleaning, exploration, and deriving key insights to understand customer purchasing patterns, sales performance, and product demand. By performing queries on the dataset, the project addresses various business questions, enabling better strategic planning and operational improvements.







# Key Features

- Data cleaning to handle missing and null values.
- Exploration of key metrics such as total sales, customer demographics, and product category performance.
- Detailed SQL queries to answer business-critical questions.
- Identification of trends and patterns to support decision-making.



## CREATE TABLE retail\_sales

```
(  
  "transactions_id" INT PRIMARY KEY,  
  sale_date DATE,  
  sale_time TIME,  
  customer_id INT,  
  gender VARCHAR(10),  
  age INT,  
  category VARCHAR (15),  
  quantiy INT,  
  price_per_unit FLOAT,  
  cogs FLOAT,  
  total_sale FLOAT  
);
```



## Data cleaning for null values

```
DELETE FROM retail_sales  
  WHERE  
    transactions_id IS NULL  
    OR  
    sale_date IS NULL  
    OR  
    sale_time IS NULL  
    OR  
    gender IS NULL  
    OR  
    category IS NULL  
    OR  
    quantiy IS NULL  
    OR  
    cogs IS NULL  
    OR  
    total_sale IS NULL;
```





## -- DATA EXPLORATION---

-- How many sales we have ?

```
select count (*) as total_sales from retail_sales;
```

-- how many unique customers we have?

```
select count( DISTINCT customer_id) as total_customer from retail_sales;
```

-- how many categories we have ?

```
select distinct category from retail_sales;
```



## -- Data Analysis & Business Key Problems & Answers

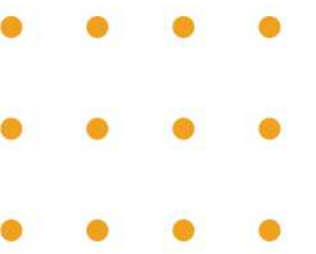
-- Q.1 Write a SQL query to retrieve all columns for sales made on '2023-05-15'.

```
select * from retail_sales  
where sale_date = '2023-05-15'
```





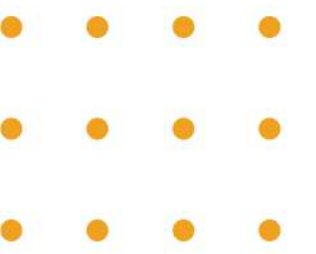
-- -- Q.2 Write a SQL query to retrieve all transactions where the category is  
-- 'Clothing' and the quantity sold is more than 4 in the month of Nov-2022



```
select *  
from retail_sales  
where category = 'Clothing'  
and quantity >= 4  
AND sale_date BETWEEN '2022-11-01' AND '2022-11-30';
```

	transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying (10)	age integer	category character varying (15)	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
1	1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200
2	64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	8.5	100
3	284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	20.5	200
4	1885	2022-11-09	07:32:00	148	Female	52	Clothing	4	30	10.8	120
5	547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000
6	159	2022-11-10	21:30:00	42	Male	26	Clothing	4	50	23.5	200
7	699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16.2	120
8	1259	2022-11-03	17:31:00	105	Female	45	Clothing	4	50	21	200
9	146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200
10	1476	2022-11-11	22:27:00	130	Female	27	Clothing	4	500	555	2000
11	1296	2022-11-26	20:42:00	45	Female	22	Clothing	4	300	342	1200
12	1696	2022-11-21	17:59:00	24	Female	50	Clothing	4	50	55	200
13	1497	2022-11-19	21:44:00	109	Male	41	Clothing	4	30	32.4	120
14	735	2022-11-26	21:38:00	153	Female	64	Clothing	4	500	515	2000
15	943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
16	965	2022-11-27	21:45:00	84	Male	22	Clothing	4	50	13	200

-- Q.3 Write a SQL query to calculate the total sales (total\_sale) for each category.



```
select category,  
sum(total_sale) as net_sales  
from retail_sales  
group by 1;
```

	category character varying (15) 🔒	net_sales double precision 🔒
1	Electronics	313810
2	Clothing	311070
3	Beauty	286840



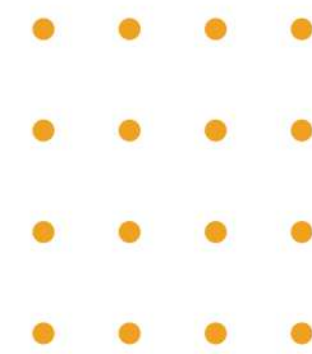
-- Q.4 Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.

```
select ROUND( AVG(age), 2) as AVERAGE_AGE  
      from retail_sales  
      WHERE category = 'Beauty';
```

	average_age numeric
1	40.42







-- Q.5 Write a SQL query to find all transactions where the total\_sale is greater than 1800

```
select * from retail_sales
where total_sale > 1800;
```

customer_id integer	gender character varying (10)	age integer	category character varying (15)	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
75	Female	42	Electronics	4	500	210	2000
55	Female	34	Beauty	4	500	260	2000
94	Female	42	Electronics	4	500	200	2000
47	Female	34	Beauty	4	500	250	2000
37	Female	38	Electronics	4	500	195	2000
18	Female	38	Electronics	4	500	220	2000
28	Female	22	Clothing	4	500	200	2000
138	Female	22	Clothing	4	500	205	2000
77	Female	46	Beauty	4	500	275	2000



-- Q.6 Write a SQL query to find the total number of transactions (transaction\_id) made by  
-- each gender in each category.

```
select
gender , category,
count (transactions_id ) as Total_trans
from retail_sales
GROUP BY gender, category
order by 1;
```

	gender character varying (10) 🔒	category character varying (15) 🔒	total_trans bigint 🔒
1	Female	Beauty	330
2	Female	Clothing	347
3	Female	Electronics	340
4	Male	Electronics	344
5	Male	Clothing	354
6	Male	Beauty	282



	year numeric 🔒	month numeric 🔒	avg_sale double precision 🔒
1	2022	7	541.3414634146342
2	2022	3	521.3829787234042
3	2022	4	486.52542372881356
4	2022	6	481.3953488372093
5	2022	5	480.38461538461536
6	2022	9	478.83720930232556
7	2022	11	472.02054794520546
8	2022	10	467.36301369863014
9	2022	12	464.20382165605093
10	2022	1	397.10526315789474
11	2022	8	385.3636363636364
12	2022	2	366.1363636363636
13	2023	2	535.531914893617
14	2023	8	495.96491228070175
15	2023	12	490.3900709219858
16	2023	4	466.48936170212767
17	2023	9	462.73972602739724
18	2023	11	453.45238095238096
19	2023	5	450.1666666666667
20	2023	6	438.48214285714283
21	2023	7	427.67857142857144
22	2023	10	399.17241379310343
23	2023	1	396.5
24	2023	3	394.8076923076923

-- -- Q.7 Write a SQL query to calculate the average sale for each month.  
-- Find out best selling month in each year

```
select
extract ( YEAR FROM sale_date ) as
year,
extract ( MONTH FROM sale_date) as
month,
avg (total_sale) as avg_sale
from retail_sales
group by 1,2
order by 1,3 desc;
```





-- Q.8 Write a SQL query to find the top 5 customers based on the highest total sales

```
select customer_id, sum (
total_sale ) as total_sales
from retail_sales
group by 1
order by 2 desc
limit 5;
```

customer_id	total_sales
integer	double precision
3	38440
1	30750
5	30405
2	25295
4	23580



-- Q.9 Write a SQL query to find the number of unique customers who purchased items from each category.



```
select category,  
count ( distinct customer_id)  
from retail_sales  
group by 1
```

	category character varying (15) 🔒	count bigint 🔒
1	Beauty	141
2	Clothing	149
3	Electronics	144



-- 10. Write a SQL query to calculate the total sales (total\_sale) for each product category.

```
select category,  
sum(total_sale) as total_sales  
FROM retail_sales  
group by 1  
order by total_sales DESC;
```

	category character varying (15) 🔒	total_sales double precision 🔒
1	Electronics	313810
2	Clothing	311070
3	Beauty	286840





-- 11. Write a SQL query to find the average age of customers who purchased items from each category.

```
SELECT category, AVG(age) AS average_age  
FROM retail_sales  
GROUP BY category  
ORDER BY average_age DESC;
```

	category character varying (15) 🔒	average_age numeric 🔒
1	Clothing	41.9297994269340974
2	Electronics	41.5988200589970501
3	Beauty	40.4157119476268412



-- 12. Write a SQL query to find the top 5 customers based on the highest total sales (total\_sale).

```
select
  customer_id,
  sum(total_sale) as total_sales
from retail_sales
group by customer_id
order by total_sales desc
limit 5;
```

	customer_id integer	total_sales double precision
1	3	38440
2	1	30750
3	5	30405
4	2	25295
5	4	23580



-- 13. Write a SQL query to calculate the total sales (total\_sale) made by male and female customers.

```
select gender ,  
sum (total_sale) AS total_sales  
from retail_sales  
group by gender;
```

	gender character varying (10) 🔒	total_sales double precision 🔒
1	Female	465400
2	Male	446320





-- 14. Write a SQL query to calculate the total sales (total\_sale) for each month of the year 2023.

	month numeric	total_sales double precision
1	1	23790
2	2	25170
3	3	20530
4	4	21925
5	5	27010
6	6	24555
7	7	35925
8	8	28270
9	9	67560
10	10	57880
11	11	57135
12	12	69145

```
SELECT EXTRACT(MONTH FROM  
sale_date) AS month,  
SUM(total_sale) AS total_sales  
FROM retail_sales  
WHERE EXTRACT(YEAR FROM  
sale_date) = 2023  
GROUP BY month  
ORDER BY month;
```



-- 15. Write a SQL query to find the month with the highest total sales (total\_sale) in the year 2023.



	month numeric	total_sales double precision
1	12	69145

```
SELECT EXTRACT(MONTH FROM sale_date)
AS month, SUM(total_sale) AS total_sales
FROM retail_sales
WHERE EXTRACT(YEAR FROM sale_date) =
2023
GROUP BY month
ORDER BY total_sales DESC
LIMIT 1;
```



	sale_date date	total_sales double precision
1	2022-01-01	2150
2	2022-01-02	2075
3	2022-01-03	200
4	2022-01-04	1450
5	2022-01-05	290
6	2022-01-06	1200
7	2022-01-07	30
8	2022-01-08	945
9	2022-01-09	2500
10	2022-01-10	1600
11	2022-01-12	30
12	2022-01-14	140
13	2022-01-15	25
14	2022-01-16	225
15	2022-01-17	1530
16	2022-01-19	120
17	2022-01-20	125
18	2022-01-21	1560
19	2022-01-23	1250
20	2022-01-24	1225
21	2022-01-25	2100
22	2022-01-26	30
23	2022-01-27	250
24	2022-01-28	160

-- 16. Write a SQL query to calculate the total sales (total\_sale) for each day.

```
select sale_date ,
sum (total_sale) AS total_sales
from retail_sales
group by sale_date
order by sale_date;
```

and many more....





-- 17. Write a SQL query to find all transactions in the 'Electronics' category for  
-- customers between the ages of 39 and 40.

```
select * from retail_sales
where category = 'Electronics' and age
between 39 and 40;
```

transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying (10)	age integer	category character varying (15)
659	2023-06-09	06:08:00	75	Female	39	Electronics
1659	2023-07-27	09:29:00	22	Female	39	Electronics
357	2023-05-18	18:57:00	4	Female	40	Electronics
1357	2023-03-20	19:14:00	3	Female	40	Electronics
744	2023-05-23	18:02:00	1	Male	40	Electronics
1744	2023-07-07	21:49:00	4	Male	40	Electronics
883	2023-08-20	20:43:00	5	Male	40	Electronics
1883	2023-09-20	17:54:00	3	Male	40	Electronics
501	2023-06-28	20:21:00	2	Male	39	Electronics
1501	2023-01-31	18:00:00	1	Male	39	Electronics
123	2023-08-11	19:00:00	3	Female	40	Electronics
1123	2023-12-11	21:04:00	1	Female	40	Electronics
81	2023-06-07	22:48:00	4	Male	40	Electronics
1081	2023-06-27	22:23:00	5	Male	40	Electronics
694	2023-06-30	17:47:00	3	Female	39	Electronics
1694	2023-01-10	19:22:00	4	Female	39	Electronics
297	2023-09-12	21:09:00	70	Female	40	Electronics
1297	2023-10-20	22:32:00	21	Female	40	Electronics

-- 18. Write a SQL query to calculate the total sales (total\_sale) for each gender in each product category.

```
select gender , category,  
sum (total_sale) as total_sales  
from retail_sales  
group by gender, category  
order by 3;
```

	gender character varying (10) 🔒	category character varying (15) 🔒	total_sales double precision 🔒
1	Male	Beauty	137370
2	Male	Clothing	148610
3	Female	Beauty	149470
4	Female	Electronics	153470
5	Male	Electronics	160340
6	Female	Clothing	162460

-- 19. Write a SQL query to find the transaction with the highest total\_sale.



```
select * from retail_sales
order by total_sale desc
limit 1;
```



transactions_id [Pk] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying (10)	age integer	category character varying (15)	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
15	2022-07-01	11:50:00	75	Female	42	Electronics	4	500	210	



-- 20. Write a SQL query to find the total number of transactions made in each month of the year 2023.



```
select extract ( month from sale_date) as  
       month , count (transactions_id) as  
       total_transaction  
       from retail_sales  
where  extract (year from sale_date) = 2023  
       group by month  
       order by month;
```

	month numeric 	total_transaction bigint 
2	2	47
3	3	52
4	4	47
5	5	60
6	6	56
7	7	84
8	8	57
9	9	146
10	10	145
11	11	126
12	12	141



# Conclusion

The Retail Sales Data Analysis Using SQL project provided valuable insights into customer behavior, sales trends, and product performance. By analyzing the data through SQL, the project answered key business questions and highlighted opportunities for growth.

The analysis identified top-performing categories like 'Clothing' and 'Electronics,' peak sales periods, and customer purchasing patterns based on demographics.

These insights can help businesses focus on popular products, target the right customers, and plan promotions during high-demand months.





# Thank You

"Aspiring Business Analyst"



[github.com/RAVI4458](https://github.com/RAVI4458)



