



RETAIL SALES ANALYSIS

"Conducted a comprehensive retail sales analysis using SQL to uncover key trends and insights for business optimization

BY JATIN SHARMA

OBJECTIVES



- 1. Set up a retail sales database:** Create and populate a retail sales database with the provided sales data.
- 2. Data Cleaning:** Identify and remove any records with missing or null values.
- 3. Exploratory Data Analysis (EDA):** Perform basic exploratory data analysis to understand the dataset.
- 4. Business Analysis:** Use SQL to answer specific business questions and derive insights from the sales data.



PROBLEM TO BE SOLVED

"The challenge is to identify key factors influencing retail sales performance, such as product demand, seasonal trends, and customer purchasing behavior. The goal is to uncover patterns in sales data that can help optimize inventory, reduce operational costs, and boost profitability. By solving this problem, the business can make data-driven decisions to improve stock management, streamline operations, and increase customer satisfaction by ensuring the availability of high-demand products at the right time."

DATA EXPLORATION

```
select count(transactions_id) from  
retail_sales;
```

	count(transactions_id)
▶	1987



RECORD COUNT

```
select count(distinct customer_id) from  
retail_sales;
```

	count(distinct customer_id)
▶	155



CUSTOMER COUNT

```
select count(distinct category) from  
retail_sales;
```

	count(distinct category)
▶	3



CATEGORY COUNT:

Q1 . Q1. Write a SQL query to retrieve all columns for sales made on '2022-11-05':

```
select * from retail_sales  
where sale_date = '2022-11-05';
```

	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
▶	180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129	900
	240	2022-11-05	11:49:00	95	Female	23	Beauty	1	300	123	300
	1256	2022-11-05	09:58:00	29	Male	23	Clothing	2	500	190	1000
	1587	2022-11-05	20:06:00	140	Female	40	Beauty	4	300	105	1200
	1819	2022-11-05	20:44:00	83	Female	35	Beauty	2	50	14	100
	943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
	1896	2022-11-05	20:19:00	87	Female	30	Electronics	2	25	31	50
	1137	2022-11-05	22:34:00	104	Male	46	Beauty	2	500	145	1000
	856	2022-11-05	17:43:00	102	Male	54	Electronics	4	30	9	120

Q2. Write a SQL query to retrieve all transactions where

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

	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
▶	1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200
	64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	9	100
	284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	21	200
	1885	2022-11-09	07:32:00	148	Female	52	Clothing	4	30	11	120
	547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000
	159	2022-11-10	21:30:00	42	Male	26	Clothing	4	50	24	200
	699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16	120
	1259	2022-11-03	17:31:00	105	Female	45	Clothing	4	50	21	200
	146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200

Q3. Write a SQL query to calculate the total sales (total_sale) for each category.:

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

	total_sales	category
▶	309995	Clothing
	286790	Beauty
	311445	Electronics

-- Q4. Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category

```
select round(avg(age),0) as avg_age , category  
from retail_sales where category = "beauty";
```

	avg_age	category
	40	Beauty

-- Q5. Write a SQL query to find all transactions where the total_sale is greater than 1000

```
select * from retail_sales  
where total_sale>1000;
```

	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
▶	522	2022-07-09	11:00:00	52	Male	46	Beauty	3	500	145	1500
	559	2022-12-12	10:48:00	5	Female	40	Clothing	4	300	84	1200
	1522	2022-11-14	08:35:00	48	Male	46	Beauty	3	500	235	1500
	1559	2022-08-20	07:40:00	49	Female	40	Clothing	4	300	144	1200
	421	2022-04-08	08:43:00	66	Female	37	Clothing	3	500	235	1500
	1421	2022-01-17	07:07:00	59	Female	37	Clothing	3	500	185	1500
	484	2022-03-13	07:52:00	135	Female	19	Clothing	4	300	75	1200
	1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200
	15	2022-07-01	11:50:00	75	Female	42	Electronics	4	500	210	2000

-- Q6. Write a SQL query to find the total number of transactions (transaction_id) made by each gender in each category

```
select count(transactions_id) as total_transactions , gender, category  
from retail_sales  
group by gender,category;
```

	total_transactions	gender	category
▶	351	Male	Clothing
	281	Male	Beauty
	347	Female	Clothing
	343	Male	Electronics
	330	Female	Beauty
	335	Female	Electronics

-- Q7. Write a SQL query to calculate the average sale for each month. Find out best selling month in each year:

```
select count(transactions_id) as total_transactions , gender, category  
from retail_sales  
group by gender,category;
```

	total_transactions	gender	category
▶	351	Male	Clothing
	281	Male	Beauty
	347	Female	Clothing
	343	Male	Electronics
	330	Female	Beauty
	335	Female	Electronics

Q8. 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 customer_id order by  
total_sales desc limit 5;
```

	CUSTOMER_ID	total_sales
▶	3	38440
	1	30750
	5	30405
	2	25295
	4	23580

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  
customer_id  
from retail_sales group by category;
```

	category	customer_id
▶	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 & 17, Evening >17)

```
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 count(*) as total_orders , shift from hourly_sale
group by shift;
```

	total_orders	shift
▶	548	morning
	1062	evening
	377	afternoon

Findings

- **Customer Demographics:** The dataset includes customers from various age groups, with sales distributed across different categories such as Clothing and Beauty.
- **High-Value Transactions:** Several transactions had a total sale amount greater than 1000, indicating premium purchases.
- **Sales Trends:** Monthly analysis shows variations in sales, helping identify peak seasons.
- **Customer Insights:** The analysis identifies the top-spending customers and the most popular product categories

GOALS AND TASKS



REPORT

- **SALES SUMMARY:** A DETAILED REPORT SUMMARIZING TOTAL SALES, CUSTOMER DEMOGRAPHICS, AND CATEGORY PERFORMANCE.
- **TREND ANALYSIS:** INSIGHTS INTO SALES TRENDS ACROSS DIFFERENT MONTHS AND SHIFTS.
- **CUSTOMER INSIGHTS:** REPORTS ON TOP CUSTOMERS AND UNIQUE CUSTOMER COUNTS PER CATEGORY.



CONCLUSION

JATIN SHARMA

DATA ANALYST

Conclusion

- This project serves as a comprehensive introduction to SQL for data analysts, covering database setup, data cleaning, exploratory data analysis, and business-driven SQL queries. The findings from this project can help drive business decisions by understanding sales patterns, customer behavior, and product performance.