RETAIL-SALES ANALYSIS SQL PROJECT

Created by **Jawad Haq**

"NAVIGATING THE FUTURE OF SHOPPING"

Content

- Project Overview
- Objectives
- About the Dataset
- Database Setup
- Data Exploration & Cleaning
- Data Analysis
- Findings
- Reports
- Conclusion

Project Overview

This project is designed to demonstrate SQL skills and techniques typically used by data analysts to explore, clean, and analyze retail sales data. The project involves setting up a retail sales database, performing exploratory data analysis (EDA), and answering specific business questions through SQL queries. This project is ideal for those who are starting their journey in data analysis and want to build a solid foundation in SQL.

Objectives

Set up a retail sales database: Create and populate a retail sales database with the provided sales data.

Exploratory Data Analysis (EDA): Perform basic exploratory data analysis to understand the dataset.

Data Cleaning: Identify and remove any records with missing or null values.

Business Analysis: Use SQL to answer specific business questions and derive insights from the sales data.

About the Dataset

- **Dataset:** https://github.com/jawad-haque/Retail-Sales-Analysis-SQL-Project--P1.git
- **Records:** 2000 rows
- **Fields/Columns:**
  - transaction_id
  - sale_date
  - sale_time
  - customer_id
  - gender
  - age
  - category
  - quantity
  - price_per_unit
  - cogs
  - total_sale

Database Creation: The project starts by creating a database named retail_sales_p1.

```
create database retail_sales_p1;
```

Database Setup

Table Creation: A table named retail_sales is created to store the sales data.

```sql
CREATE TABLE retail_sales
            (
                transaction_id INT PRIMARY KEY,
                sale_date DATE,
                sale_time TIME,
                customer_id INT,
                gender  VARCHAR(15),
                age INT,
                category VARCHAR(15),
                quantity    INT,
                price_per_unit FLOAT,
                cogs    FLOAT,
                total_sale FLOAT
            );
```

Database Setup

Record Count: Determine the total number of records in the dataset.

```
SELECT
    COUNT(*)
FROM
    retail_sales;
```

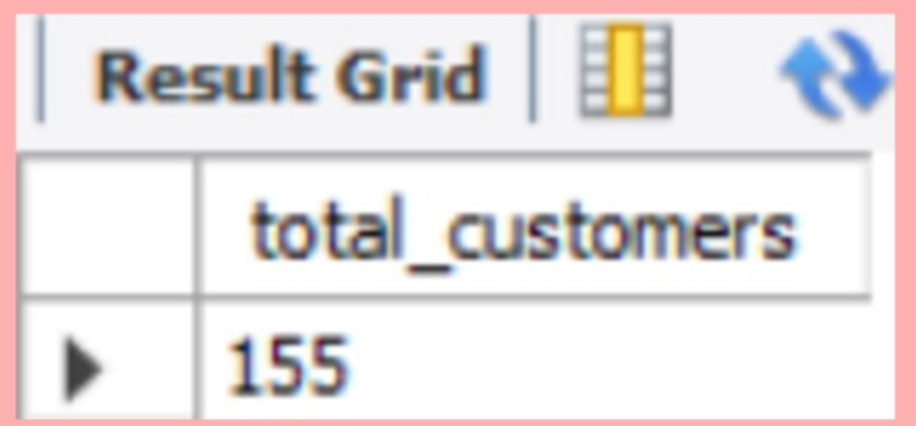
Result Grid

	COUNT(*)
▶	2000

Data Exploration & Cleaning

Customer Count: Find out how many unique customers are in the dataset.

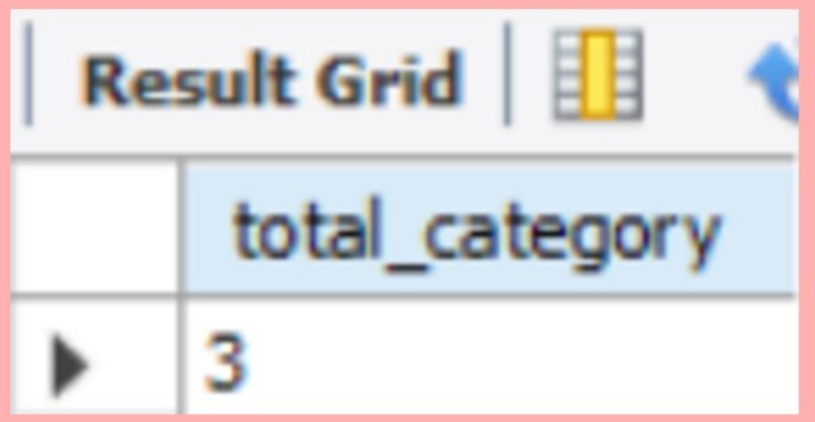
```
SELECT
    COUNT(DISTINCT customer_id)
FROM
    retail_sales;
```



Data Exploration & Cleaning

Category Count: Identify all unique product categories in the dataset.

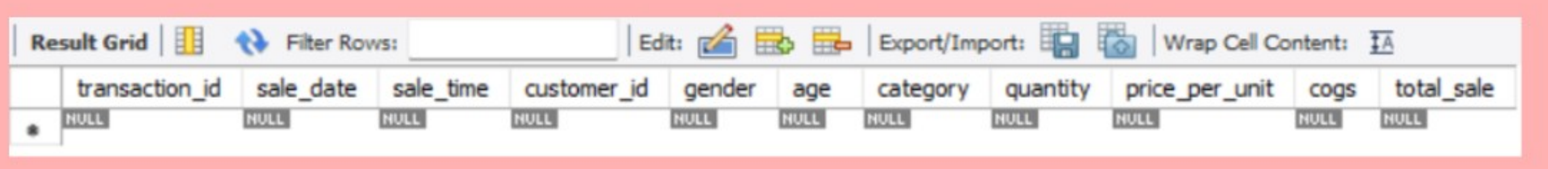
```
SELECT
    COUNT(DISTINCT category)
FROM
    retail_sales;
```



Data Exploration & Cleaning

Null Value Check: Check for any null values in the dataset and delete records with missing data.

```sql
SELECT
    *
FROM
    retail_sales
WHERE
        transaction_id IS NULL
        OR sale_time IS NULL
        OR sale_time IS NULL
        OR customer_id IS NULL
        OR gender IS NULL
        OR age IS NULL
        OR category IS NULL
        OR quantity IS NULL
        OR price_per_unit IS NULL
        OR cogs IS NULL
        OR total_sale IS NULL;
```



transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Data Exploration & Cleaning

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

```sql
SELECT
    *
FROM
    retail_sales
WHERE
    sale_date = '2022-11-05';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129	900
214	2022-11-05	16:31:00	53	Male	20	Beauty	2	30	8.1	60
240	2022-11-05	11:49:00	95	Female	23	Beauty	1	300	123	300
856	2022-11-05	17:43:00	102	Male	54	Electronics	4	30	9.3	120
943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
1137	2022-11-05	22:34:00	104	Male	46	Beauty	2	500	145	1000
1256	2022-11-05	09:58:00	29	Male	23	Clothing	2	500	190	1000
1265	2022-11-05	14:35:00	86	Male	55	Clothing	3	300	111	900
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	13.5	100
1896	2022-11-05	20:19:00	87	Female	30	Electronics	2	25	30.75	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Data Analysis

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.

```sql
SELECT
    *
FROM
    retail_sales
WHERE
    category = 'Clothing' AND quantity = 4
        AND sale_date >= '2022-11-01'
        AND sale_date < '2022-12-01';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

transaction_id	sale_date	sale_time	customer_id	gender	age	category	quantity	price_per_unit	cogs	total_sale
64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	8.5	100
146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200
159	2022-11-10	21:30:00	42	Male	26	Clothing	4	50	23.5	200
284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	20.5	200
547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000
699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16.2	120
735	2022-11-26	21:38:00	153	Female	64	Clothing	4	500	515	2000
943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
965	2022-11-27	21:45:00	84	Male	22	Clothing	4	50	13	200
1259	2022-11-03	17:31:00	105	Female	45	Clothing	4	50	21	200
1296	2022-11-26	20:42:00	45	Female	22	Clothing	4	300	342	1200
1476	2022-11-11	22:27:00	130	Female	27	Clothing	4	500	555	2000
1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200
1497	2022-11-19	21:44:00	109	Male	41	Clothing	4	30	32.4	120
1615	2022-11-17	13:43:00	82	Female	61	Clothing	4	25	13.5	100
1696	2022-11-21	17:59:00	24	Female	50	Clothing	4	50	55	200
1885	2022-11-09	07:32:00	148	Female	52	Clothing	4	30	10.8	120
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Data Analysis

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

```sql
SELECT
    category,
    SUM(total_sale) AS net_sale,
    COUNT(*) AS total_orders
FROM
    retail_sales
GROUP BY category;
```

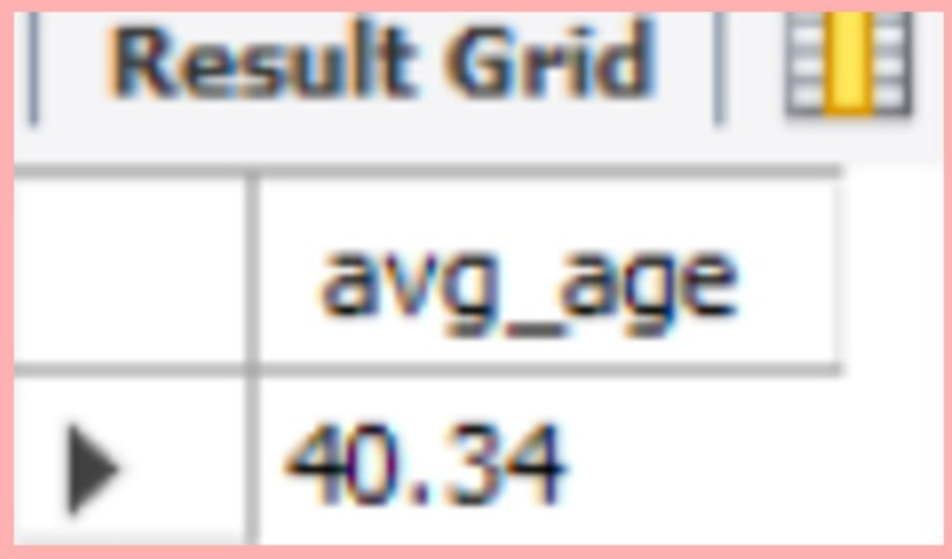
Result Grid | Filter Rows:

category	net_sale	total_orders
Beauty	286840	614
Clothing	311070	702
Electronics	313810	684

Data Analysis

4.Write a SQL query to find the average age of customers who purchased items from the ‘Beauty’ category.

```sql
SELECT
    round(AVG(age), 2) AS avg_age
FROM
    retail_sales
WHERE
    category = 'Beauty';
```



Data Analysis

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

```sql
SELECT
    *
FROM
    retail_sales
WHERE
    total_sale > 1000;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

transaction_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
31	2023-12-31	17:47:00	3	Male	44	Electronics	4	300	129	1200
46	2022-11-08	17:50:00	54	Female	20	Electronics	4	300	84	1200
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
58	2023-09-16	19:18:00	53	Male	18	Clothing	4	300	75	1200
65	2022-12-11	20:03:00	84	Male	51	Electronics	4	500	160	2000
67	2023-08-19	20:19:00	119	Female	48	Beauty	4	300	129	1200

Data Analysis

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

```sql
SELECT
    category,
    gender,
    COUNT(*) 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

Data Analysis

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

```sql
SELECT
        YEAR,
        MONTH,
        avg_sale
FROM
    (
        SELECT
                YEAR(sale_date) AS YEAR,
                MONTH(sale_date) AS MONTH,
                AVG(total_sale) AS avg_sale,
                RANK() OVER(PARTITION BY YEAR(sale_date)
                ORDER BY AVG(total_sale) DESC)
                AS rnk
                FROM retail_sales
                GROUP BY  YEAR, MONTH
    )
                AS t1
                WHERE rnk = 1;
```

Result Grid | Filter Rows:

YEAR	MONTH	avg_sale
2022	7	528.452380952381
2023	2	535.531914893617

Data Analysis

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

```sql
SELECT
    customer_id, SUM(total_sale) AS total_sales
FROM
    retail_sales
GROUP BY customer_id
ORDER BY total_sales DESC
LIMIT 5;
```

Result Grid | Filter Ro

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

Data Analysis

9.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_unique_customers
FROM
    retail_sales
GROUP BY category;
```

	category	total_unique_customers
▶	Beauty	141
	Clothing	149
	Electronics	144

Data Analysis

10.Write a SQL query to create each shift and number of orders (Example Morning <12, Afternoon Between 12 & 17, Evening >17).

```sql
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;
```

Result Grid | Filter Rows:

shift	total_orders
Evening	1062
Morning	561
Afternoon	377

Data Analysis

Findings

2022-2024

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.

Reports

1

Sales Summary:

A detailed report summarizing total sales, customer demographics, and category performance.

2

Trend Analysis:

Insights into sales trends across different months and shifts.

3

Customer Insights:

Reports on top customers and unique customer counts per category.

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.

Thank You

LIKE

SHARE