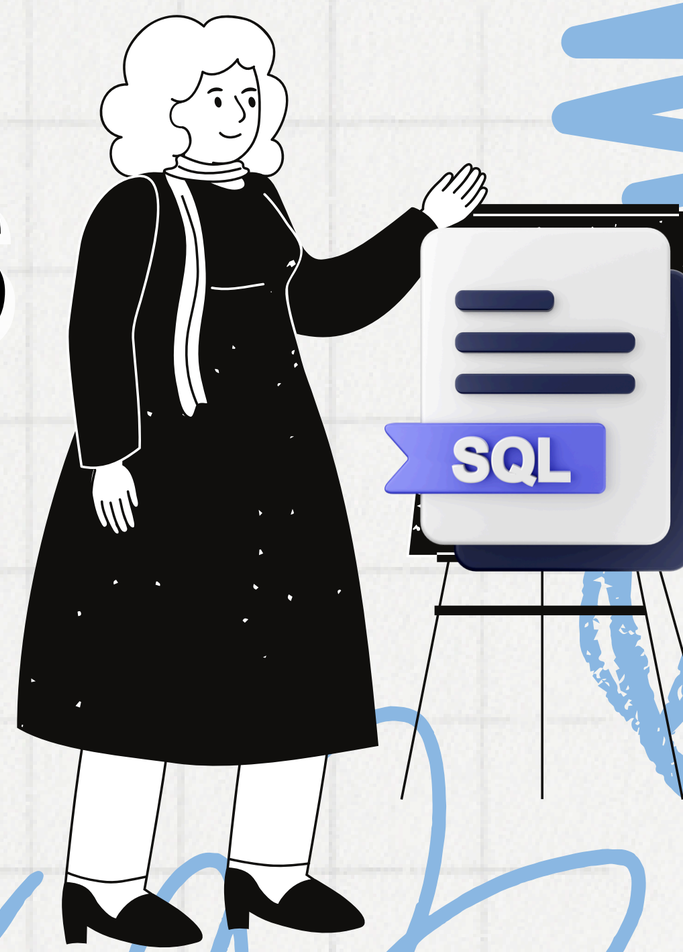
Tarini Prasad Das

RETAIL SALES ANALYSIS

using SQL

Presented by
Tarini Prasad Das



Project Overview

Project Title: Retail Sales Analysis
Level: Beginner
Database: `SQL_PROJECTS`
SCHEMA: 'retail_sales_analysis'

This project demonstrates SQL skills and techniques used by data analysts to explore, clean, and analyze retail sales data. It 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 beginning their journey in data analysis and looking to build a solid foundation in SQL.

Tarini Prasad Das

Objectives

01. **Set Up a Retail Sales Database:** Create and populate a retail sales database using the provided sales data.

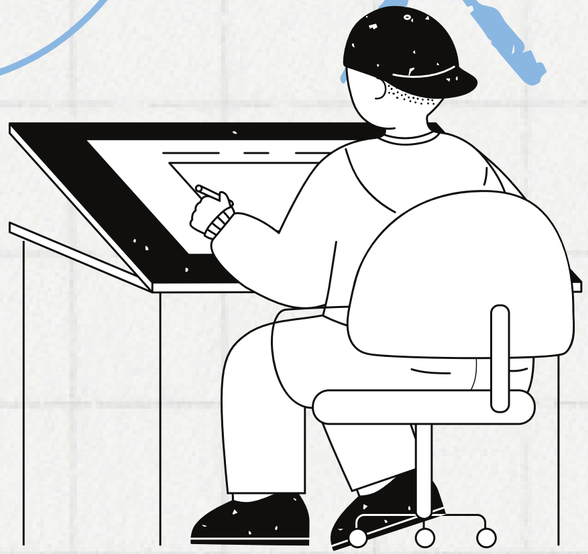
02. **Data Cleaning:** Identify and remove records with missing or null values to ensure data integrity.

03. **Exploratory Data Analysis (EDA)**: Understand the dataset through basic exploratory analysis.

04. **Business Analysis:** Answer specific business questions and generate insights using SQL queries.

Tarini Prasad Das

Project Structure



01

Database Setup

02

Data Exploration & Cleaning

03

Exploratory Data Analysis

04

Findings and Conclusion

Tarini Prasad Das

1. Database Setup

Database Creation: The project begins with the creation of a database named `SQL_PROJECTS`. Here's the SQL query to create a database named SQL_PROJECTS and a schema named ‘retail_sales_analysis’ within that database:

sql Copy code

```sql
-- Create the database
CREATE DATABASE SQL_PROJECTS;

-- Switch to the SQL_PROJECTS database
USE SQL_PROJECTS;

-- Create the schema
CREATE SCHEMA RETAIL_SALES_ANALYSIS;
```

Tarini Prasad Das

Table Creation:

A `retail_sales` table is established to store the sales data. The table structure includes columns for transaction ID, sale date, sale time, customer ID, gender, age, product category, quantity sold, price per unit, cost of goods sold (COGS), and total sale amount.

```sql
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(35),
    quantity INT,
    price_per_unit FLOAT,
    cogs FLOAT,
    total_sale FLOAT
);
```

Create a retail_sales table to organize the data with the following columns:

- **transactions_id:** Unique identifier for each transaction.
- **sale_date:** Date of the sale.
- **sale_time:** Time of the sale.
- **customer_id:** Unique identifier for each customer.
- **gender:** Gender of the customer.
- **age:** Age of the customer.
- **category:** Product category (e.g., Clothing, Beauty).
- **quantity:** Number of units sold.
- **price_per_unit:** Price per unit of the product.
- **cogs:** Cost of goods sold.
- **total_sale:** Total sale amount (price per unit * quantity).

Tarini Prasad Das

2. Data Exploration & Cleaning

Null Value Check:
Identify and delete records with any null values.

Customer Count:
Identify the number of unique customers in the dataset.

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

Category Count:
Identify all unique product categories present.

Tarini Prasad Das

1. Data Cleaning:

Before diving into analysis, it's crucial to clean the data by removing any rows with missing values that could potentially skew our results. Here's how I approached it:

- **Step 1: Identify rows with missing values.**

```sql
SELECT *
FROM retail_sales
WHERE
    transactions_id IS NULL
    OR sale_date IS NULL
    OR sale_time IS NULL
    OR customer_id IS NULL
    OR gender IS NULL
    OR category IS NULL
    OR quantity IS NULL
    OR cogs IS NULL
    OR total_sale IS NULL;
```

Output:

transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying	age integer	category character varying	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
679	2022-08-26	08:59:00	64	Female	18	Beauty	[null]	[null]	[null]	[null]
746	2022-07-05	11:33:00	42	Female	33	Clothing	[null]	[null]	[null]	[null]
1225	2022-02-02	09:51:00	137	Female	57	Beauty	[null]	[null]	[null]	[null]

- **Step 2: Remove rows with missing values.**

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

Output:(clened data)

transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying	age integer	category character varying	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
180	2022-11-05	10:47:00	117	Male	41	Clothing	3	300	129	900
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
1180	2022-01-06	08:53:00	85	Male	41	Clothing	3	300	129	900
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
163	2022-10-31	09:38:00	144	Female	64	Clothing	3	50	23	150
303	2022-04-22	11:09:00	54	Male	19	Electronics	3	30	14.7	90
421	2022-04-08	08:43:00	66	Female	37	Clothing	3	500	235	1500
979	2022-05-18	10:18:00	6	Female	19	Beauty	1	25	[illegible]	25

Tarini Prasad Das

2. Data Exploration :

Once the data is clean, we can explore the dataset to understand the distribution and key metrics:

- **Total Sales:**

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

Output :

count bigint
1997

- **Unique Customers:**

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

Output :

count bigint
155

- **Distinct Categories:**

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

Output :

count bigint
3

Tarini Prasad Das

3. Data Analysis & Business Key Problems:

To gain business insights, I conducted the following analyses:

- Explore the dataset to understand its structure, contents, and key characteristics.
- Identify patterns, trends, and anomalies within the data.
- Use SQL queries to answer specific business questions.
- Generate actionable insights from the sales data to support decision-making.

Tarini Prasad Das

1.Retrieve all sales made on '2022-11-05':

Find all transactions that occurred on this specific date.

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

Output :

transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying	age integer	category character varying	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
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	13.5	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	30.75	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	9.3	120
214	2022-11-05	16:31:00	53	Male	20	Beauty	2	30	8.1	60
1265	2022-11-05	14:35:00	86	Male	55	Clothing	3	300	111	900

Tarini Prasad Das

2.Retrieve transactions where the category is 'Clothing' and the quantity sold is more than 4 in November 2022:

```sql
SELECT * FROM retail_sales
WHERE category = 'Clothing'
  AND TO_CHAR(sale_date, 'YYYY-MM') = '2022-11'
  AND quantity >= 4;
```

Output :

transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying	age integer	category character varying	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
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	8.5	100
284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	20.5	200
1885	2022-11-09	07:32:00	148	Female	52	Clothing	4	30	10.8	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	23.5	200
699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16.2	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
1476	2022-11-11	22:27:00	130	Female	27	Clothing	4	500	555	2000
1296	2022-11-26	20:42:00	45	Female	22	Clothing	4	300	342	1200
1696	2022-11-21	17:59:00	24	Female	50	Clothing	4	50	55	200
1497	2022-11-19	21:44:00	109	Male	41	Clothing	4	30	32.4	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
1615	2022-11-17	13:43:00	82	Female	61	Clothing	4	25	13.5	100

Tarini Prasad Das

3.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;
```

Output :

category character varying	net_sale double precision	total_orders bigint
Electronics	313810	684
Clothing	311070	701
Beauty	286840	612

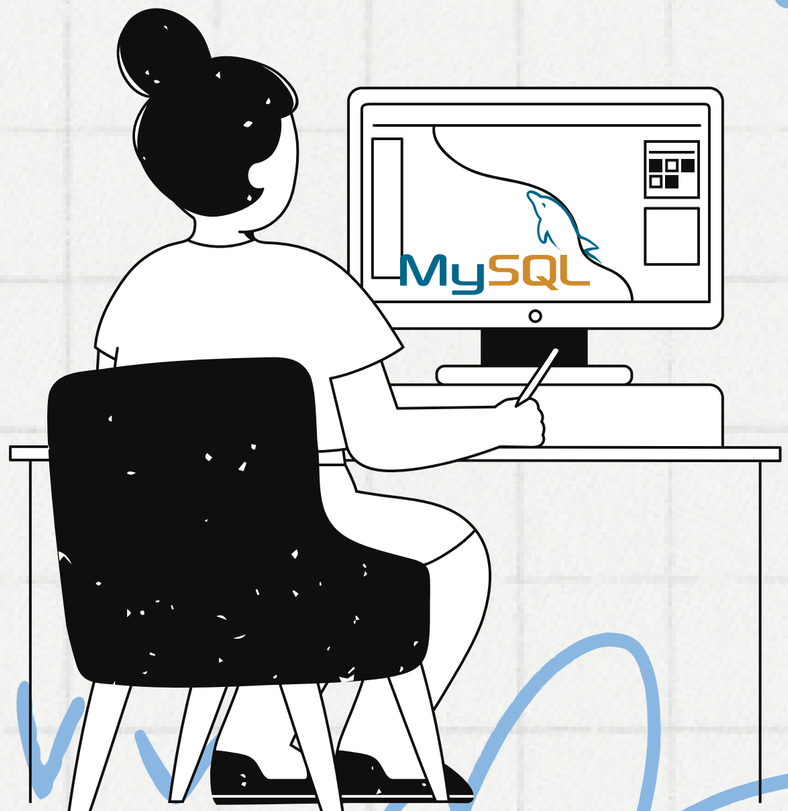
Tarini Prasad Das

4.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';
```

Output :

avg_age numeric
40.42



Tarini Prasad Das

5.Find transactions where the total_sale is greater than 1000:

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

Output :

transactions_id [PK] integer	sale_date date	sale_time time without time zone	customer_id integer	gender character varying	age integer	category character varying	quantity integer	price_per_unit double precision	cogs double precision	total_sale double precision
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
743	2022-08-07	07:54:00	55	Female	34	Beauty	4	500	260	2000

6.Calculate the total number of transactions made by each gender in each category:

```sql
SELECT category, gender, COUNT(*) as total_trans
FROM retail_sales
GROUP BY category, gender
ORDER BY category;
```

Output :

category character varying	gender character varying	total_trans bigint
Beauty	Female	330
Beauty	Male	282
Clothing	Female	347
Clothing	Male	354
Electronics	Male	344
Electronics	Female	340

Tarini Prasad Das

7.Calculate the average sale for each month and identify the best-selling month in each year:

```sql
SELECT year, month, avg_sale
FROM (
    SELECT EXTRACT(YEAR FROM sale_date) as year,
           EXTRACT(MONTH FROM sale_date) as month,
           AVG(total_sale) as avg_sale,
           RANK() OVER(PARTITION BY EXTRACT(YEAR FROM sale_date)
                       ORDER BY AVG(total_sale) DESC) as rank
    FROM retail_sales
    GROUP BY year, month
) as monthly_sales
WHERE rank = 1;
```

Output :

year numeric	month numeric	avg_sale double precision
2022	7	541.3414634146342
2023	2	535.531914893617

Tarini Prasad Das

8.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;
```

Output :

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

Tarini Prasad Das

9.Count the number of unique customers who purchased items from each category:

```sql
SELECT category, COUNT(DISTINCT customer_id) as cnt_unique_cs
FROM retail_sales
GROUP BY category;
```

Output :

category character varying	cnt_unique_cs bigint
Beauty	141
Clothing	149
Electronics	144

Tarini Prasad Das

10. Determine the number of orders per shift (Morning, Afternoon, Evening):

```sql
WITH hourly_sales AS (
    SELECT *,
           CASE
               WHEN EXTRACT(HOUR FROM sale_time) < 12 THEN 'Morning'
               WHEN EXTRACT(HOUR FROM sale_time) BETWEEN 12 AND 17 THEN 'Afternoon
               ELSE 'Evening'
           END as shift
    FROM retail_sales
)
SELECT shift, COUNT(*) as total_orders
FROM hourly_sales
GROUP BY shift;
```

Output :

shift text	total_orders bigint
Afternoon	377
Evening	1062
Morning	558

Tarini Prasad Das

4. Findings & Conclusion:

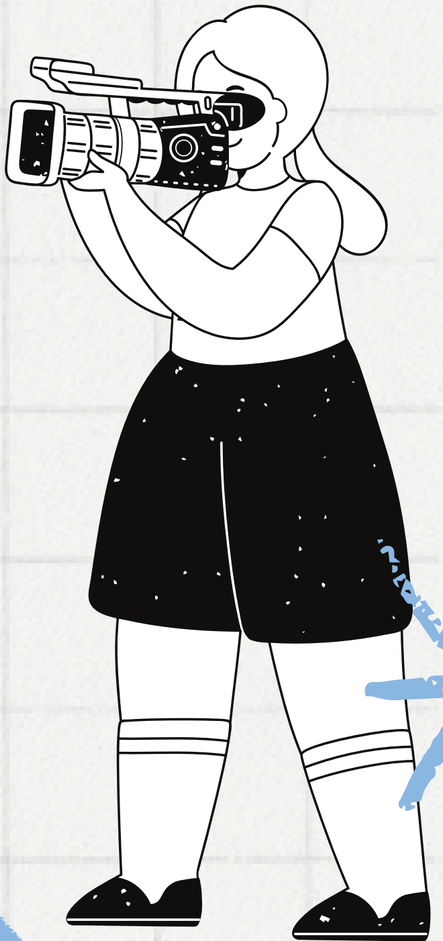
Findings :

Customer Demographics: The dataset includes diverse customer age groups and covers various product categories such as Clothing and Beauty.

High-Value Transactions: Several transactions exceed a total sale amount of 1000, indicating premium purchases.

Sales Trends: The monthly analysis highlights sales fluctuations, helping identify peak seasons.

Customer Insights: The top-spending customers and popular product categories were identified, offering valuable business insights.



Reports

- **Sales Summary:** Summarizes total sales, customer demographics, and category performance.
- **Trend Analysis:** Provides insights into sales trends across different months and shifts.
- **Customer Insights:** Includes 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 can inform business decisions by highlighting sales patterns, customer behavior, and product performance.

Tapini Prasad Das

Thank you very much!

Tarini Prasad Das