# Retail Sales Analysis SQL Project





### Project Overview

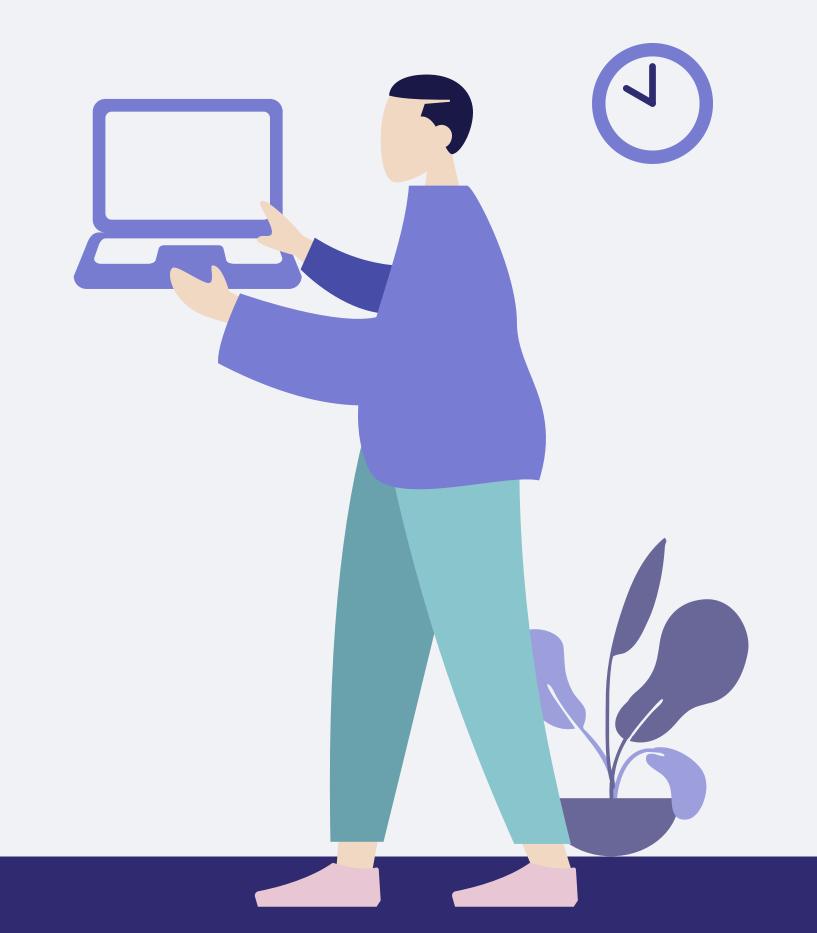
Project Title: Retail Sales Analysis

Database: sql\_p1

This project aims to showcase SQL competencies and techniques frequently utilized by data analysts to examine, cleanse, and interpret retail sales data. The project involves establishing a retail sales database, conducting exploratory data analysis (EDA), and resolving specific business queries through SQL statements. This project is particularly suitable for beginners looking to establish a strong foundation in SQL.

## Objectives

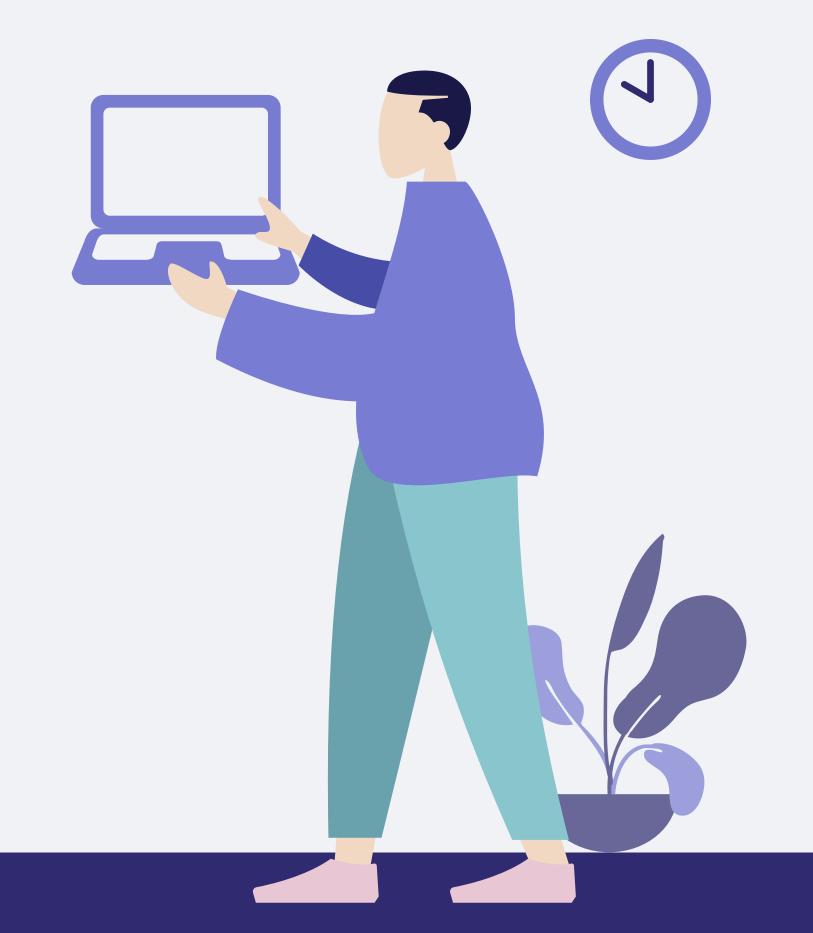
- 1. **Database Initialization:** Establish and populate a retail sales database with the provided sales data.
- 2. **Data Cleansing:** Detect and eliminate any records containing missing or null values.
- 3. Exploratory Data Analysis (EDA): Conduct basic exploratory analysis to gain insights into the dataset.
- 4. **Business Query Resolution:** Utilize SQL to address specific business queries and extract valuable insights from the sales data.



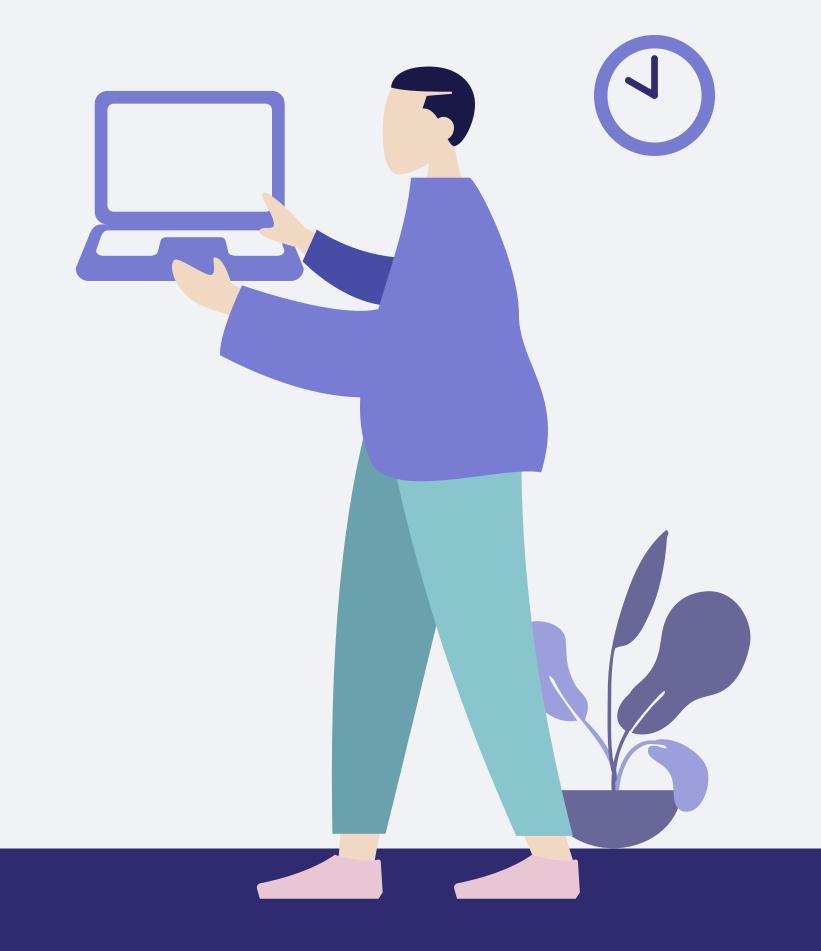
### Project Structure

### 1. Database Setup

- **Database Creation:** The project begins by creating a database named sql\_p1.
- Table Creation: A table named retail\_sales is established to store the sales data. The table 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.



```
CREATE DATABASE sql_p1;
Create table retail_sales
               transactions_id INT PRIMARY KEY,
               sale_date
                              DATE,
               sale_time
                              TIME,
               customer_id
                              INT,
               gender VARCHAR(10),
                       INT,
               age
               category VARCHAR(15),
               quantiy INT,
               price_per_unit FLOAT,
               cogs FLOAT,
               total_sale FLOAT
               );
```

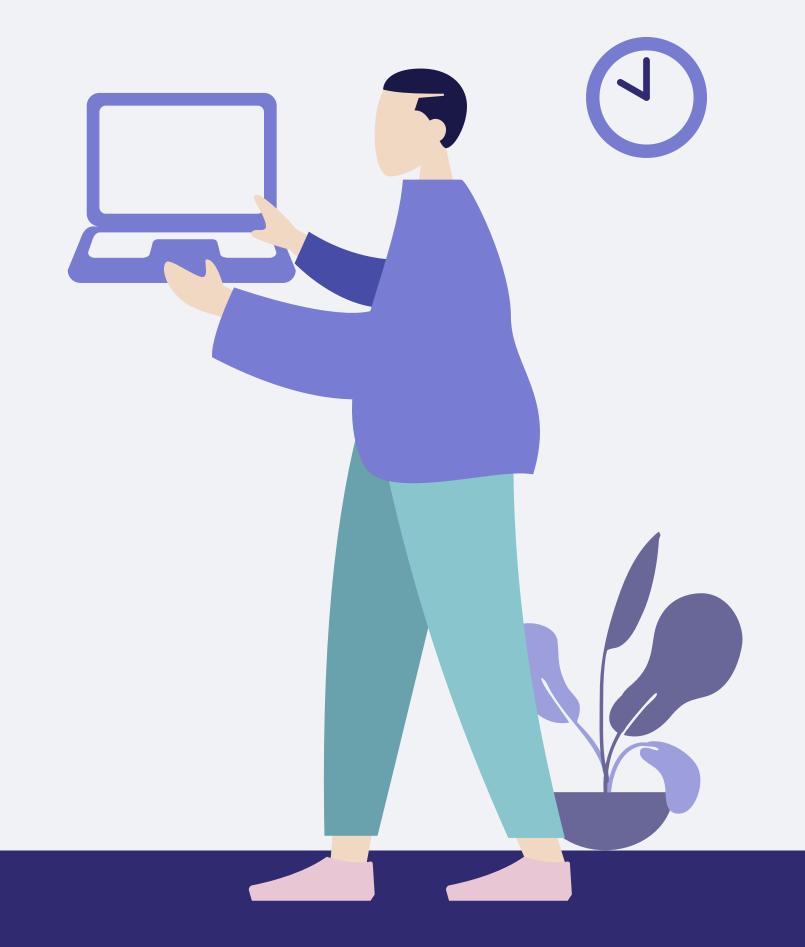


### 2. Data Exploration & Cleaning

- **Record Count:** Ascertain the total number of records within the dataset.
- Customer Count: Determine the number of unique customers present in the dataset.
- Category Count: Identify all distinct product categories within the dataset.
- Missing Data: Identify and remove records with missing or null values to ensure the dataset is clean.
- Basic Descriptive Statistics: Compute basic descriptive statistics such as average quantity sold, average price, and total sales per category.



```
SELECT COUNT(*) FROM retail_sales;
SELECT COUNT(DISTINCT customer_id) FROM retail_sales;
SELECT DISTINCT category FROM retail_sales;
select * from retail_sales
where transactions_id IS NULL
               sale_date IS NULL
       OR
               sale_time IS NULL
       OR
               customer_id
       OR
                              IS NULL
               gender IS NULL
       OR
       OR
                      IS NULL
       OR
               category IS NULL
               quantiy IS NULL
       OR
               price_per_unit IS NULL
       OR
               cogs IS NULL
       OR
               total_sale IS NULL;
       OR
DELETE FROM retail_sales
where transactions_id IS NULL
               sale_date IS NULL
       OR
               sale_time IS NULL
       OR
       OR
               customer_id
                              IS NULL
               gender IS NULL
       OR
       OR
                      IS NULL
               age
               category IS NULL
       OR
       OR
               quantiy IS NULL
               price_per_unit IS NULL
       OR
               cogs IS NULL
       OR
               total_sale IS NULL;
       OR
```



### 01

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

### 02

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:

### 03

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

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

```
select * from retail_sales
where category = 'Clothing'
and quantiy >= 4
and TO_CHAR(sale_date, 'YYYY-MM') = '2022-11';
```



### 04

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

### 05

Write a SQL query to find all transactions where the total\_sale is greater than 1000.

### 06

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

# select Round(Avg(age),2) as AVG\_Age\_Under\_beauty from retail\_sales where Category ='Beauty';

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

```
select
category,
gender,
count(transactions_id) as Transactions
from retail_sales
group by category,gender
order by 1;
```



### 07

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

```
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 1, 2
) as t1

WHERE rank = 1
```

### 08

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

```
customer_id,
SUM(total_sale) as totalsales
from retail_sales
group by 1
order by 2 desc limit 5;
```



### 09

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

### 10

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

```
select
category,
COUNT(DISTINCT customer_id) as unique_cs
from retail_sales
group by 1;
```

```
WITH hourly_sale
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_sale
GROUP BY shift
Order by 2 desc;
```



# Findings



 Customer Demographics: The dataset includes customers from various age groups, with sales distributed across different categories such as Clothing and Beauty.

### 02

 High-Value Transactions: Several transactions had a total sale amount greater than 1000, indicating premium purchases.

### 03

- 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.



### Conclusion

This project provides a thorough introduction to SQL for data analysts, covering database setup, data cleaning, exploratory data analysis, and business-oriented SQL queries. The insights derived from this project can inform business decisions by revealing sales trends, customer behavior, and product performance.

Check out the project on GitHub:

[https://github.com/diwakar18d/Retail sales analysis p1 sql basic]

er.diwakark1@gmail.com

