

# RETAIL SALES ANALYSIS

PREPARED BY PANKAJ YADAV

## SQL Project



# TABLE OF CONTENTS

1

**INTRODUCTION**

2

**OBJECTIVES**

3

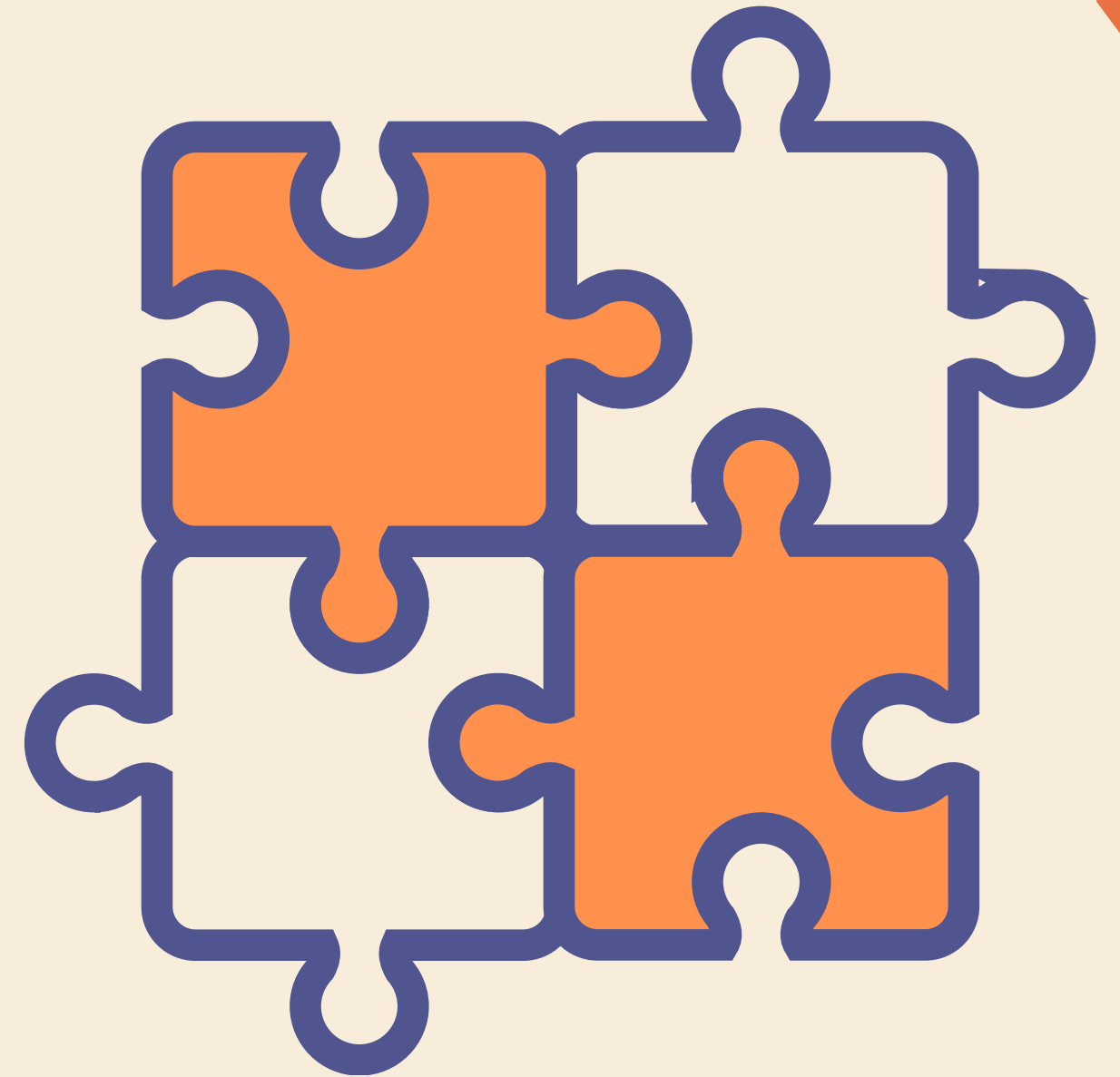
**PROJECT STRUCTURE**

4

**CONCLUSION**

# INTRODUCTION

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.



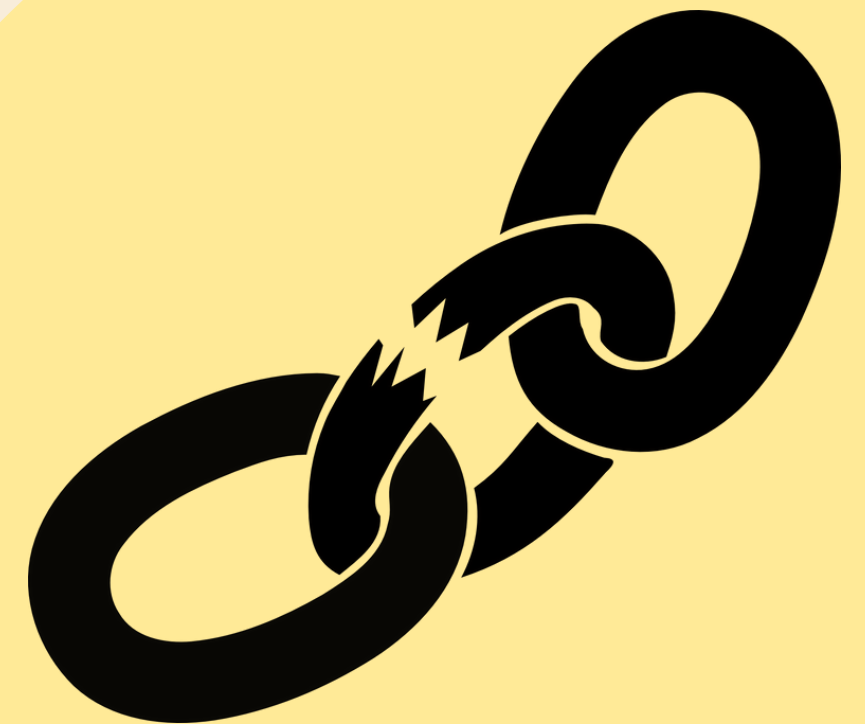
# OBJECTIVES



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.

# PROJECT STRUCTURE

1. Database Setup
2. Data Exploration & Cleaning
3. Data Analysis & Findings
4. Findings



# DATABASE AND TABLE CREATION

```
CREATE DATABASE p1_retail_db;
```

```
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  
);
```

# Count of:

**Total number of records:**

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

**Unique customers:**

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

**Unique product categories:**

```
SELECT DISTINCT category FROM retail_sales;
```

# Null Value Check:

```
SELECT * FROM retail_sales
```

```
WHERE
```

```
    sale_date 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;
```



# Deleting Null Value:

```
DELETE FROM retail_sales
```

```
WHERE
```

```
    sale_date 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;
```

**1. 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';
```

**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 YEAR(sale_date) = '2022'  
    AND MONTH(sale_date) = '11' AND quantity >= 4;
```

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

```
SELECT category, SUM(total_sale) AS Total_sales  
FROM retail_sales  
GROUP BY category;
```

**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 Avg_age_of_customers  
FROM retail_sales  
WHERE category = 'Beauty';
```

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

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

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

```
SELECT category, gender, COUNT(transaction_id) AS  
Num_of_transactions  
FROM retail_sales  
GROUP BY category, gender  
ORDER BY category;
```

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

```
WITH CTE AS(  
  SELECT  year(sale_date) AS Year, month(sale_date) AS  
  Month, ROUND(AVG(total_sale), 2) AS Avg_sale,  
  RANK () OVER(PARTITION BY year(sale_date) ORDER BY  
  AVG(total_sale) DESC) AS rnk  
  FROM retail_sales  
  GROUP BY month(sale_date), year(sale_date)  
)  
  
  SELECT Year, Month  
  FROM CTE  
  WHERE rnk = 1;
```

**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 customer_id  
ORDER BY Total_sales DESC  
LIMIT 5;
```

**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 Num_of_unique_customers
FROM retail_sales
GROUP BY category;
```

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

```
WITH CTE AS(
  SELECT *,
  CASE
    WHEN sale_time < '12:00:00' THEN 'Morning'
    WHEN sale_time BETWEEN '12:00:00' AND '17:00:00' THEN 'Afternoon'
    WHEN sale_time > '17:00:00' THEN 'Evening'
  END AS shift
FROM retail_sales
)

SELECT shift, COUNT(transactions_id) AS Num_of_orders
FROM CTE
GROUP BY shift
;
```



# 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.

# Conclusion:

This project 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!**