

# RETAIL SALES ANALYSIS

**SQL-BASED DATA ANALYSIS  
PROJECT**



PRESENTED BY  
**RAJEEV KUMAR**



# INTRODUCTION

- **Objective:** Analyzing retail sales data to extract key business insights using SQL.
- **Dataset:** Retail sales transactions with attributes like sale date, time, customer demographics, category, quantity, price, and total sales.
- **Key Focus Areas:** Sales trends, customer behavior, and business performance.
- **Tools Used:** MySQL Workbench for querying and analysis.



# 1). RETRIEVE ALL SALES MADE ON '2022-11-05'.

SOLUTION

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

EXPLANATION

1. THE WHERE CLAUSE FILTERS RECORDS WHERE SALE\_DATE IS '2022-11-05'.
2. THIS HELPS IN ANALYZING SALES FOR A PARTICULAR DAY TO IDENTIFY TRENDS OR ANOMALIES.

RESULT

| transactions_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        |

## 2). RETRIEVE TRANSACTIONS WHERE CATEGORY IS 'CLOTHING' AND QUANTITY SOLD IS MORE THAN 4 IN NOV-2022.

### SOLUTION

```
SELECT
    transactions_id, sale_date, category, quantity
FROM
    retail_sales
WHERE
    category = 'Clothing' AND quantity >= 4
        AND sale_date BETWEEN '2022-11-01' AND '2022-11-30'
ORDER BY sale_date;
```

### RESULT

| transactions_id | sale_date  | category | quantity |
|-----------------|------------|----------|----------|
| 1259            | 2022-11-03 | Clothing | 4        |
| 943             | 2022-11-05 | Clothing | 4        |
| 1885            | 2022-11-09 | Clothing | 4        |
| 146             | 2022-11-10 | Clothing | 4        |
| 159             | 2022-11-10 | Clothing | 4        |
| 1476            | 2022-11-11 | Clothing | 4        |
| 284             | 2022-11-12 | Clothing | 4        |
| 547             | 2022-11-14 | Clothing | 4        |
| ...             | ...        | ...      | ...      |

### EXPLANATION

1. FILTERS TRANSACTIONS WHERE CATEGORY IS 'CLOTHING' AND QUANTITY IS GREATER THAN 4.
2. USES BETWEEN TO FOCUS ONLY ON NOVEMBER 2022 SALES.
3. ORDERS RESULTS BY SALE\_DATE TO OBSERVE SALES TRENDS OVER THE MONTH.

### 3). CALCULATE TOTAL SALES FOR EACH CATEGORY.

#### SOLUTION

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

#### RESULT

| category    | Total_sales |
|-------------|-------------|
| Electronics | 311445      |
| Clothing    | 309995      |
| Beauty      | 286790      |

#### EXPLANATION

1. GROUPS TRANSACTIONS BY CATEGORY.
2. USES **SUM(TOTAL\_SALE)** TO CALCULATE TOTAL REVENUE PER CATEGORY.
3. ORDERS THE RESULT IN DESCENDING ORDER TO IDENTIFY THE TOP-SELLING CATEGORIES.

## 4). FIND THE AVERAGE AGE OF CUSTOMERS WHO PURCHASED 'BEAUTY' PRODUCTS.

### SOLUTION

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

### RESULT

|   | avg_age |
|---|---------|
| ▶ | 40.42   |

### EXPLANATION

1. FILTERS TRANSACTIONS TO INCLUDE ONLY BEAUTY CATEGORY PURCHASES.
2. USES **AVG(AGE)** TO CALCULATE THE MEAN AGE OF CUSTOMERS.
3. **ROUND** ENSURES THE RESULT IS DISPLAYED WITH TWO DECIMAL PLACES.

## 5). RETRIEVE TRANSACTIONS WHERE TOTAL SALE IS GREATER THAN 1000.

### SOLUTION

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

### RESULT

| transactions_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       |

### EXPLANATION

1. THE **WHERE** CLAUSE FILTERS TRANSACTIONS WHERE **TOTAL\_SALE** EXCEEDS 1000.
2. HELPS IDENTIFY HIGH-VALUE SALES AND PREMIUM CUSTOMERS.

## 6). FIND THE TOTAL NUMBER OF TRANSACTIONS MADE BY EACH GENDER IN EACH CATEGORY.

### SOLUTION

```
SELECT  
    COUNT(transactions_id) AS transactions, gender, category  
FROM  
    retail_sales  
GROUP BY category , gender  
ORDER BY transactions , category;
```

### RESULT

|     | transactions | gender      | category |
|-----|--------------|-------------|----------|
| 281 | Male         | Beauty      |          |
| 330 | Female       | Beauty      |          |
| 335 | Female       | Electronics |          |
| 343 | Male         | Electronics |          |
| 347 | Female       | Clothing    |          |
| 351 | Male         | Clothing    |          |

### EXPLANATION

1. GROUPS TRANSACTIONS BY GENDER AND CATEGORY.
2. USES COUNT(TRANSACTIONS\_ID) TO CALCULATE TOTAL TRANSACTIONS PER GROUP.
3. ORDERS RESULTS TO HIGHLIGHT THE MOST ACTIVE CUSTOMER SEGMENTS.

## 7). CALCULATE THE AVERAGE SALE FOR EACH MONTH AND FIND THE BEST-SELLING MONTH PER YEAR.

### SOLUTION

```
select *  
from  
(  
    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 round(avg(total_sale),2) desc) as ranks  
    from retail_sales  
    group by 1,2  
)as t1  
where ranks = 1;
```

### RESULT

| year | month | avg_sale | ranks |
|------|-------|----------|-------|
| 2022 | 7     | 541.34   | 1     |
| 2023 | 2     | 535.53   | 1     |

### EXPLANATION

1. **GROUPS** DATA BY YEAR AND MONTH, CALCULATING **AVG(TOTAL\_SALE)**.
2. USES **RANK()** TO FIND THE BEST-SELLING MONTH WITHIN EACH YEAR.
3. FILTERS ONLY THE TOP-RANKED MONTH PER YEAR.

## 8). IDENTIFY THE TOP 5 CUSTOMERS BASED ON TOTAL SALES.

### SOLUTION

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

### RESULT

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

### EXPLANATION

1. **GROUPS** TRANSACTIONS BY CUSTOMER\_ID.
2. USES **SUM(TOTAL\_SALE)** TO FIND TOTAL SPENDING PER CUSTOMER.
3. **LIMIT 5** ENSURES ONLY THE TOP 5 CUSTOMERS ARE DISPLAYED.

## 9). FIND THE NUMBER OF UNIQUE CUSTOMERS PER CATEGORY.

### SOLUTION

```
SELECT  
    COUNT(DISTINCT customer_id) AS customer, category  
FROM  
    retail_sales  
GROUP BY category;
```

### EXPLANATION

1. USES COUNT(DISTINCT CUSTOMER\_ID) TO COUNT UNIQUE CUSTOMERS PER CATEGORY.
2. HELPS UNDERSTAND CUSTOMER DISTRIBUTION ACROSS PRODUCT CATEGORIES.

### RESULT

| customer | category    |
|----------|-------------|
| 141      | Beauty      |
| 149      | Clothing    |
| 144      | Electronics |

# 10). CATEGORIZE SALES INTO SHIFTS AND COUNT TRANSACTIONS.

## SOLUTION

```
WITH hourly_sale
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(transactions_id) as total_orders
FROM hourly_sale
GROUP BY shift;
```

## EXPLANATION

1. CLASSIFIES SALES INTO MORNING, AFTERNOON, AND EVENING SHIFTS.
2. COUNTS TOTAL TRANSACTIONS PER SHIFT TO ANALYZE PEAK SHOPPING HOURS.

## RESULT

| shift     | total_orders |
|-----------|--------------|
| Evening   | 1062         |
| Morning   | 548          |
| Afternoon | 377          |



RAJEEV KUMAR

# KEY LEARNINGS & ACHIEVEMENTS

- IMPROVED SQL QUERYING SKILLS.
- GAINED INSIGHTS INTO SALES TRENDS, CUSTOMER BEHAVIOR, AND PEAK SELLING PERIODS.
- LEARNED HOW TO USE RANKING FUNCTIONS AND TIME-BASED ANALYSIS.

*Thank  
You*

