

Retail Sales SQL Analysis

This project contains a collection of SQL queries designed to analyze and clean a retail sales dataset using PostgreSQL. Below is a step-by-step breakdown of each query along with its purpose.

Data Cleaning

Q1. Fix column encoding issues:

```
ALTER TABLE retail_sales
RENAME COLUMN "ï»¿transactions_id" TO transaction_id;
```

Q2. View raw data:

```
SELECT * FROM retail_sales;
```

Q3. Identify NULLs in key columns:

```
SELECT * FROM retail_sales
WHERE
    "transaction_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
    "price_per_unit" IS NULL OR
    "cogs" IS NULL OR
    "total_sale" IS NULL;
```

Q4. Delete rows with NULLs:

```
DELETE FROM retail_sales
WHERE
    "transaction_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
    "price_per_unit" IS NULL OR
```

```
"cogs" IS NULL OR  
"total_sale" IS NULL;
```

Data Exploration

Q5. Total sales count:

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

Q6. Unique customers:

```
SELECT COUNT(DISTINCT "customer_id") AS Total_Customer FROM retail_sales;
```

Q7. Unique categories:

```
SELECT COUNT(DISTINCT "category") AS Total_Category FROM retail_sales;
```

Advanced Analysis

Q8. Get all Clothing transactions where quantity > 10 in Nov 2022:

```
SELECT * FROM retail_sales  
WHERE  
    "category" = 'Clothing' AND  
    "quantity" > 10 AND  
    TO_DATE("sale_date", 'DD-MM-YYYY')  
    BETWEEN '2022-11-01' AND '2022-11-30';
```

Q9. Average sale per month:

```
SELECT  
    TO_CHAR(TO_DATE(sale_date, 'DD-MM-YYYY'), 'Month') AS sale_month,  
    ROUND(AVG(total_sale), 2) AS avg_monthly_sale  
FROM retail_sales  
GROUP BY TO_CHAR(TO_DATE(sale_date, 'DD-MM-YYYY'), 'Month'),  
         EXTRACT(MONTH FROM TO_DATE(sale_date, 'DD-MM-YYYY'))  
ORDER BY EXTRACT(MONTH FROM TO_DATE(sale_date, 'DD-MM-YYYY'));
```

Q10. Best-selling month in each year:

```
WITH monthly_sales AS (  
    SELECT  
        EXTRACT(YEAR FROM TO_DATE(sale_date, 'DD-MM-YYYY')) AS year,  
        TO_CHAR(TO_DATE(sale_date, 'DD-MM-YYYY'), 'Month') AS month_name,  
        EXTRACT(MONTH FROM TO_DATE(sale_date, 'DD-MM-YYYY')) AS month_num,  
        SUM(total_sale) AS total_sales  
    FROM retail_sales  
    GROUP BY year, month_name, month_num  
)  
SELECT DISTINCT ON (year)  
    year, month_name, ROUND(total_sales, 2) AS total_sales  
FROM monthly_sales  
ORDER BY year, total_sales DESC;
```

Q11. Add time of day (shift) label:

```
SELECT  
    "sale_time",  
    CASE  
        WHEN EXTRACT(HOUR FROM "sale_time"::TIME) < 12 THEN 'Morning'  
        WHEN EXTRACT(HOUR FROM "sale_time"::TIME) BETWEEN 12 AND 17 THEN 'Afternoon'  
        ELSE 'Evening'  
    END AS time_of_day  
FROM retail_sales;
```

Q12. Count number of orders by shift:

```
SELECT time_of_day, COUNT(*) AS number_of_orders FROM (  
    SELECT  
        sale_time,  
        CASE  
            WHEN EXTRACT(HOUR FROM "sale_time"::TIME) < 12 THEN 'Morning'  
            WHEN EXTRACT(HOUR FROM "sale_time"::TIME) BETWEEN 12 AND 17 THEN 'Afternoon'  
            ELSE 'Evening'  
        END AS time_of_day  
    FROM retail_sales  
) AS shifts  
GROUP BY time_of_day  
ORDER BY number_of_orders DESC;
```

File Structure

- **SQL Query.sql**: Contains all the queries listed above

Key Findings

- The majority of orders occurred in the **Afternoon**, followed by **Evening**.
 - **Clothing** category had high-value bulk orders, especially in **November 2022**.
 - The **best-selling months** vary by year, suggesting seasonality in customer behavior.
 - There are noticeable data quality issues (e.g., encoding errors, missing values) that were cleaned.
-

Reports & Insights

- **Monthly Sales Trend:** Identified average sales for each month, useful for forecasting.
 - **Shift-wise Sales Volume:** Helps allocate resources efficiently based on time-of-day performance.
 - **Category Performance:** Quick filter on high-performing categories like Clothing.
-

Conclusion

This SQL analysis provides a structured approach to clean, explore, and extract insights from retail sales data. By organizing transactions into shifts and summarizing monthly trends, the business can make better decisions around marketing campaigns, workforce planning, and inventory stocking based on actual sales patterns.