**Analysis of Retail Sales Data**

**Abstract:**

This document presents a comprehensive analysis of the `PL\_RETAIL\_DB` database, which is designed to capture retail sales transactions. The study focuses on the structure of the database, data integrity checks, and various SQL queries that facilitate data retrieval and analysis. By utilising these queries, we aim to derive meaningful insights into customer behaviour, sales trends, and product performance within the retail sector.

**Keywords:**

Retail sales, SQL queries, data analysis, customer behaviour, database management.

**1. Introduction:**

The retail industry is a dynamic sector that requires efficient data management and analysis to understand consumer behaviour and optimise sales strategies. The `PL\_RETAIL\_DB` database serves as a repository for retail sales transactions, capturing essential information such as transaction details, customer demographics, and product categories. This paper outlines the structure of the database and provides SQL queries that can be employed to extract valuable insights from the data.

**2. Database Structure:**

**2.1 Database Creation**:

The `PL\_RETAIL\_DB` database is created using the following SQL commands:

```sql

CREATE DATABASE PL\_RETAIL\_DB;

USE PL\_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

);

This table captures critical information necessary for analysing retail sales.

**2.2 Data Integrity Checks:**

Data integrity is crucial for accurate analysis. The following checks are implemented:

**2.2.1 Record Count:**

To determine the total number of records in the dataset:

```sql

SELECT COUNT(\*) FROM retail\_sales;

Output:



**2.2.2 Customer Count:**

To find the number of unique customers:

```sql

SELECT COUNT(DISTINCT customer\_id) FROM retail\_sales;

Output :



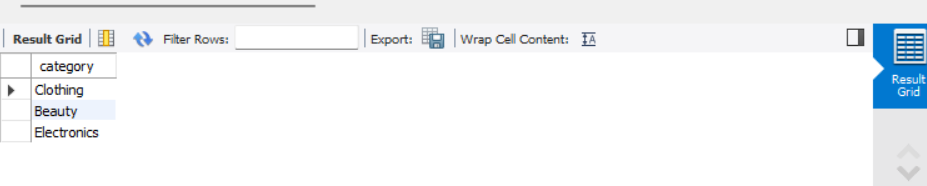
**2.2.3 Category Count:**

To identify all unique product categories:

```sql

SELECT DISTINCT category FROM retail\_sales;

Output:



**2.2.4 Null Value Check:**

To check for any null values in the dataset:

```sql

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

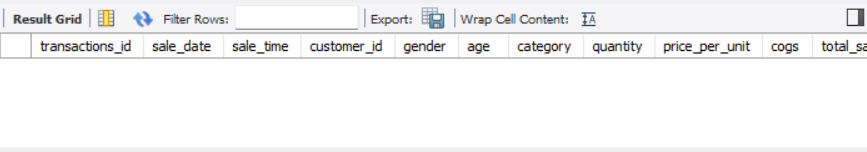
category IS NULL OR

quantity IS NULL OR

price\_per\_unit IS NULL OR

cogs IS NULL;

Output:



**2.2.5 Deleting Records with Missing Data:**

To maintain data integrity by deleting records with missing values:

```sql

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

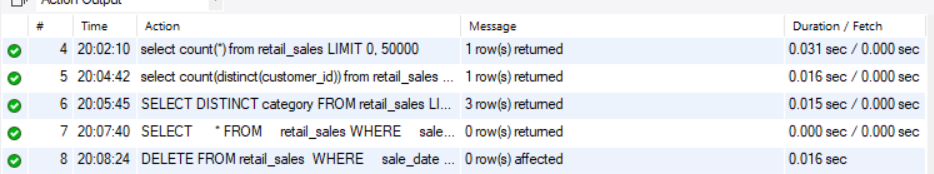
category IS NULL OR

quantity IS NULL OR

price\_per\_unit IS NULL OR

cogs IS NULL;

Output:



**3. Data Queries:**

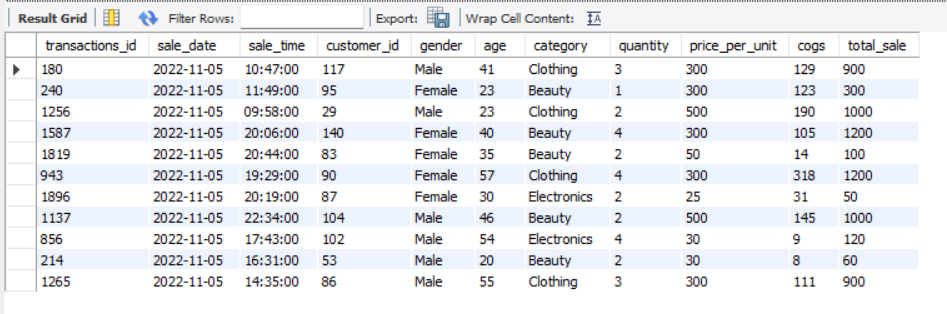
**3.1 Sales on Specific Dates:**

To retrieve all columns for sales made on '2022-11-05':

```sql

SELECT \* FROM retail\_sales WHERE sale\_date = '2022-11-05';

Output:



**3.2 Analysis of Clothing Sales:**

To analyze transactions where the category is 'Clothing' and quantity sold exceeds four in November 2022:

```sql

SELECT \* FROM retail\_sales WHERE

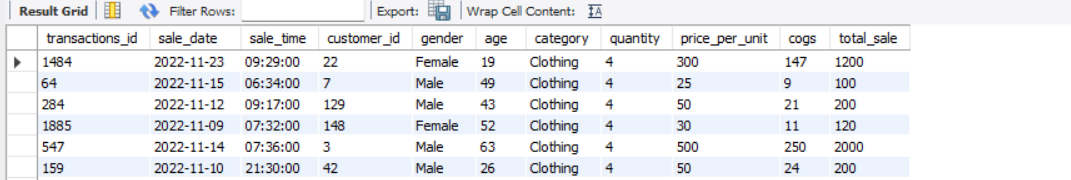
category = 'Clothing' AND

MONTH(sale\_date) = 11 AND

YEAR(sale\_date) = 2022 AND

quantity >= 4;

Output:



**3.3 Total Sales by Category:**

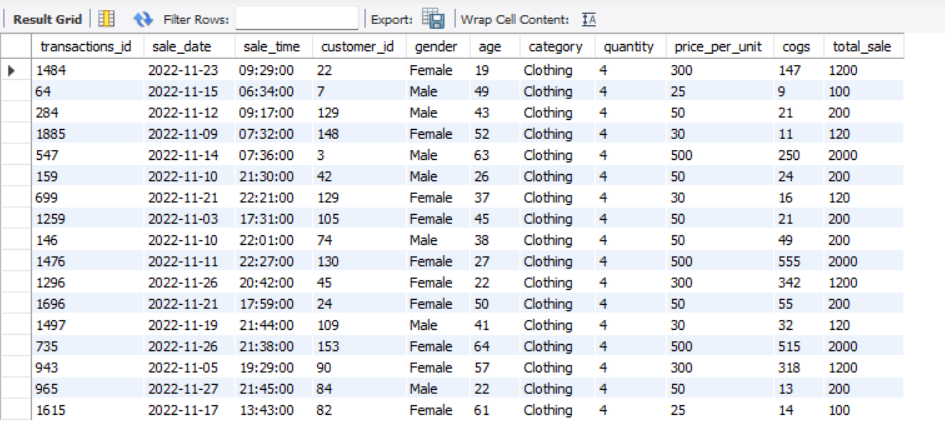
To calculate total sales for each product category:

```sql

SELECT category, SUM(total\_sale) AS net\_sale, COUNT(\*) AS total\_orders

FROM retail\_sales GROUP BY category;

Output:



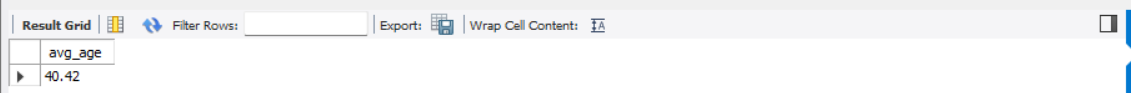
**3.4 Average Age of Customers in Beauty Category:**

To find the average age of customers purchasing items from the 'Beauty' category:

```sql

SELECT ROUND(AVG(age), 2) AS avg\_age FROM retail\_sales WHERE category = 'Beauty';

Output:



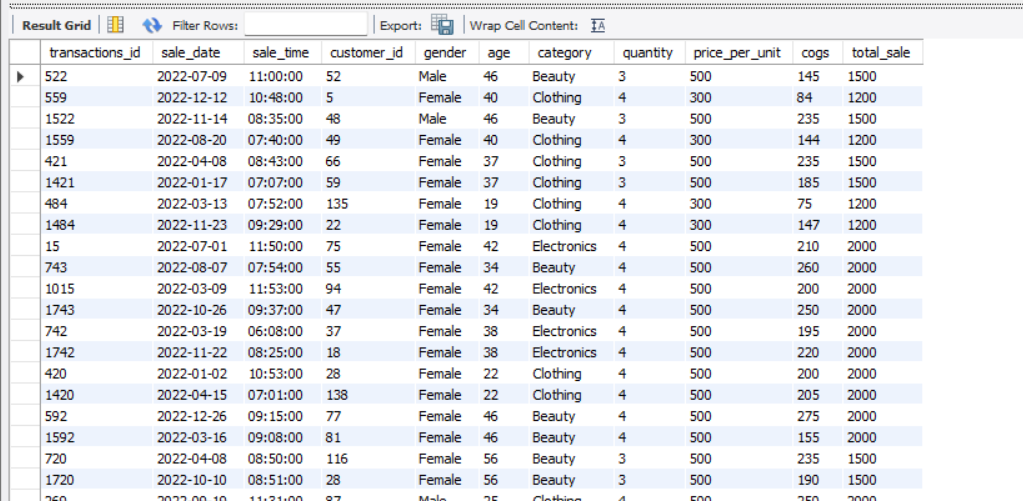
**3.5 High-Value Transactions Analysis:**

To identify all transactions where total sales exceed 1000:

```sql

SELECT \* FROM retail\_sales WHERE total\_sale > 1000;

Output:



**3.6 Transactions by Gender and Category:**

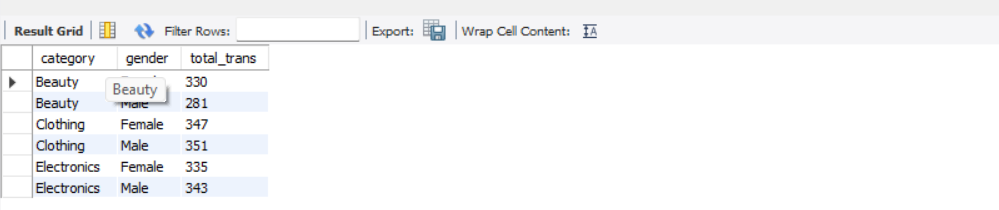
To analyze transactions made by each gender in different categories:

```sql

SELECT category, gender, COUNT(\*) AS total\_trans

FROM retail\_sales GROUP BY category, gender ORDER BY category;

Output:



**3.7 Best Selling Month Analysis:**

To calculate average sales per 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,

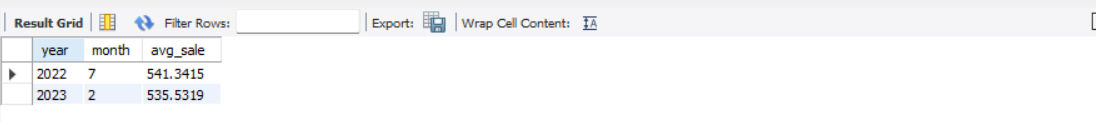
AVG(total\_sale) AS avg\_sale,

RANK() OVER(PARTITION BY EXTRACT(YEAR FROM sale\_date) ORDER BY AVG(total\_sale) DESC) AS ranks

FROM retail\_sales GROUP BY year, month

) AS t1 WHERE ranks = 1;

Output:



**3.8 Top Customers by Total Sales:**

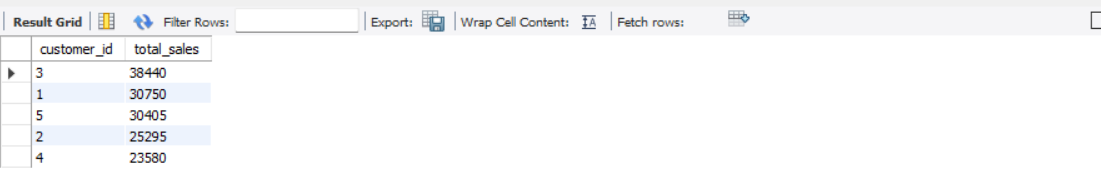
To find the top five customers based on total sales:

```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:



**3.9 Unique Customers per Category Analysis:**

To determine the number of unique customers purchasing items from each category:

```sql

SELECT category, COUNT(DISTINCT customer\_id) AS cnt\_unique\_cs FROM retail\_sales GROUP BY category;

Output:



**3.10 Shift Analysis of Orders:**

To categorize orders into shifts (Morning, Afternoon, Evening):

```sql

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'

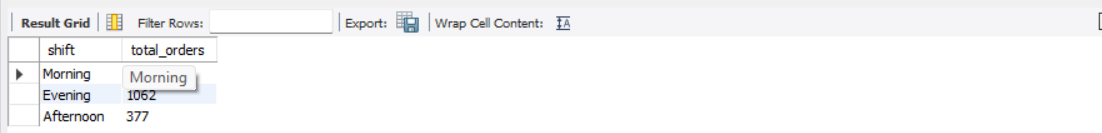
ELSE 'Evening'

END AS shift

FROM retail\_sales)

SELECT shift, COUNT(\*) AS total\_orders FROM hourly\_sale GROUP BY shift;

Output:



**4. Conclusion:**

The `PL\_RETAIL\_DB` database provides a robust framework for analyzing retail sales data. Through various SQL queries outlined in this paper, stakeholders can extract valuable insights regarding customer behaviour, sales performance across different categories, and overall market trends. Future work may involve enhancing data collection methods or integrating advanced analytics techniques to further optimize decision-making processes within the retail sector.