RETAIL SALES ANALYSIS-SQL PROJECTS

RETAIL SALES ANALYSIS

This SQL-based project analyzes retail sales data to uncover actionable business insights. The project includes a comprehensive dataset with key attributes such as transaction details, customer demographics, product categories, and sales metrics. Through data cleaning, exploration, and targeted queries, the project addresses critical business questions to drive strategic decision-making.

KEY FEATURES:

1.DATABASE SETUP: Creates a retail sales table to store transaction data, including transaction_id, sale date, sale time, customer id, gender, age, category, quantity, price_per_unit, cogs, and total_sale.

2.DATA CLEANING: Identifies and removes records with missing values to ensure data integrity. Data Exploration: Provides an overview of total sales, unique customers, and product categories.

3.BUSINESS INSIGHTS: Answers key questions, including:
(i). Sales on a specific date (e.g., '2022-11-05’).
(ii). Transactions for specific categories with quantity filters (e.g., 'Clothing' with quantity ≥ 4 in Nov-2022).
(iii). Total sales and order counts by category.
(iv). Average customer age for 'Beauty' category purchases.
(v). High-value transactions (total_sale > 1000).
(vi). Transaction counts by gender and category.
(vii). Average sales per month and identification of the best-selling month per year.
(viii). Top 5 customers by total sales.

ix). Unique customers per category.
(x). Order distribution across shifts (Morning, Afternoon, Evening).

TECHNOLOGIES USED:

SQL (Database creation, querying, and analysis) Aggregation functions, window functions (RANK), and time-based extractions

PURPOSE:

This project demonstrates proficiency in SQL for data cleaning, exploration, and business analysis. It provides a framework for retail businesses to analyze sales trends, customer behavior, and operational performance, making it ideal for data analysts and business intelligence professionals.

CODE:

- CREATE DATABASE Retail_Sales_Project;

```
-- SQL Retail Sales Analysis - P1
CREATE DATABASE sql_project_p2;


-- Create TABLE
DROP TABLE IF EXISTS retail_sales;


CREATE TABLE retail_sales(
            transaction_id INT PRIMARY KEY, sale_date DATE,
             sale_time TIME, customer_id INT,
              gender     VARCHAR(15), age   INT,
             category VARCHAR(15), quantity INT,
             price_per_unit FLOAT, cogs FLOAT,
             total_sale FLOAT);
```

-- DATA CLEANING

```
SELECT * FROM retail_sales
WHERE transaction_id IS NULL
              OR sale_date IS NULL
              OR gender IS NULL
              OR customer_id IS NULL
              OR cogs IS NULL
              OR total_sale IS NULL;

DELETE FROM retail_sales
WHERE transaction_id IS NULL
              OR sale_date IS NULL
              OR gender IS NULL
              OR customer_id IS NULL
              OR cogs IS NULL
              OR total_sale IS NULL;
```

-- DATA EXPLORATION

1. How many sales we have

Code:

```
SELECT Count(*) as total_sale
FROM retail_sales;
```

2. How many Unique customer we have

Code:

```
 SELECT Count(Distinct customer_id) as unique_customer
FROM retail_sales;
```

3. How many Unique category we have

Code:

```
SELECT Distinct category as unique_category
FROM retail_sales;
```

-- DATA EXPLORATION

-- My Analysis & Findings

Q.1 Write a SQL query to retrieve all columns for sales made on '2022-11-05

Code:

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

Q.2 Write a SQL query to retrieve all transactions where the category is 'Clothing' and the quantity sold is more than 10 in the month of Nov-2022?

Code:

```
SELECT * FROM retail_sales
WHERE category = 'Clothing' AND quantity >=4
AND TO_CHAR(sale_date, 'YYYY-MM') = '2022-11';
```

Q.3 Write a SQL query to calculate the total sales (total_sale) for each category.

Code:

```
SELECT category,sum(total_sale) as total_sales,
COUNT(*) AS total_orders FROM retail_sales
GROUP BY category;
```

Q.4 Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.

Code:

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

Q.5 Write a SQL query to find all transactions where the total_sale is greater than 1000.

Code:

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

Q.6 Write a SQL query to find the total number of transactions (transaction_id) made by each gender in each category.

Code:

```
SELECT category,gender, COUNT(*) as total_trans
FROM retail_sales
GROUP BY  category,gender
ORDER BY category,gender;
```

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

Code:

```
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 rank
FROM retail_sales
GROUP BY 1,2
)
As t1
WHERE rank =1;
```

Q.8 Write a SQL query to find the top 5 customers based on the highest total sales

Code:

```
SELECT customer_id,
sum(total_sale) as highest_sale
FROM retail_sales
GROUP BY 1
ORDER BY 2 DESC LIMIT 5;
```

Q.9 Write a SQL query to find the number of unique customers who purchased items from each category.

Code:

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

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

Code:

```
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;
```

THANK YOU

Sangeeta Yadav-Data Analyst