Retail Sales Analysis

Overview

Project Title: Retail Sales Analysis

Field: Data Analytics

This is a project which is perfomed to explore, clean and analyze the retail sales data. The project involves creating a database, importing, performing exploratory data analysis (EDA) and answering specific business questions through SQL queries. This Provides basic insight on how queries help in solving the business problems.

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.

Details

1. Creating Database

- Database Creation: A database named `retailsales` is created.

- Table Creation: A table named `retailsales` is created to store the sales data. The table structure includes columns for transaction ID, sale date, sale time, customer ID, gender, age, product category, quantity sold, price per unit, cost of goods sold (COGS), and total sale amount.

```
CREATE DATABASE retailsales;


CREATE TABLE retailsales

(

   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
);
```

2. Data Exploration & Cleaning

Record Count: Determine the total number of records in the dataset.

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

Customer Count: Find out how many unique customers are in the dataset.

```
SELECT COUNT(DISTINCT customer_id) AS Customer_Count FROM retailsales;
```

Category Count: Identify all unique product categories in the dataset.

```
SELECT DISTINCT category AS Categories FROM retailsales;
```

Null Value Check: Check for any null values in the dataset and delete records with missing data.

```
SELECT * FROM retailsales

WHERE


transactions_id IS NULL

   OR    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

   quantiy IS NULL

   OR

   price_per_unit IS NULL

   OR
   cogs IS NULL

   OR

   total_sale IS NULL;


DELETE FROM retailsales

WHERE

            transactions_id IS NULL

   OR

   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
```

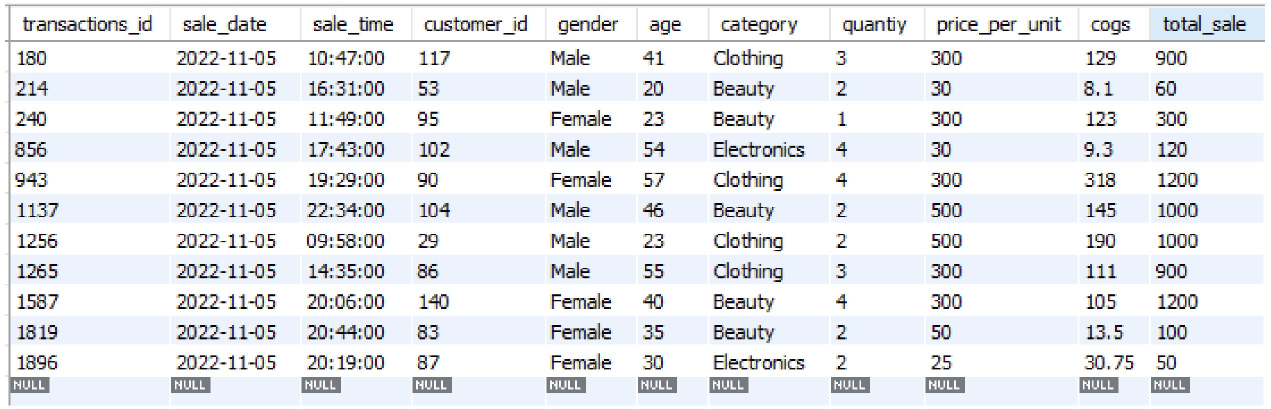
```
    quantiy IS NULL
    OR
    price_per_unit IS NULL
    OR
    cogs IS NULL
    OR
    total_sale IS NULL;
```

3. Data Analysis & Findings

The following SQL queries were developed to answer specific business questions:

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

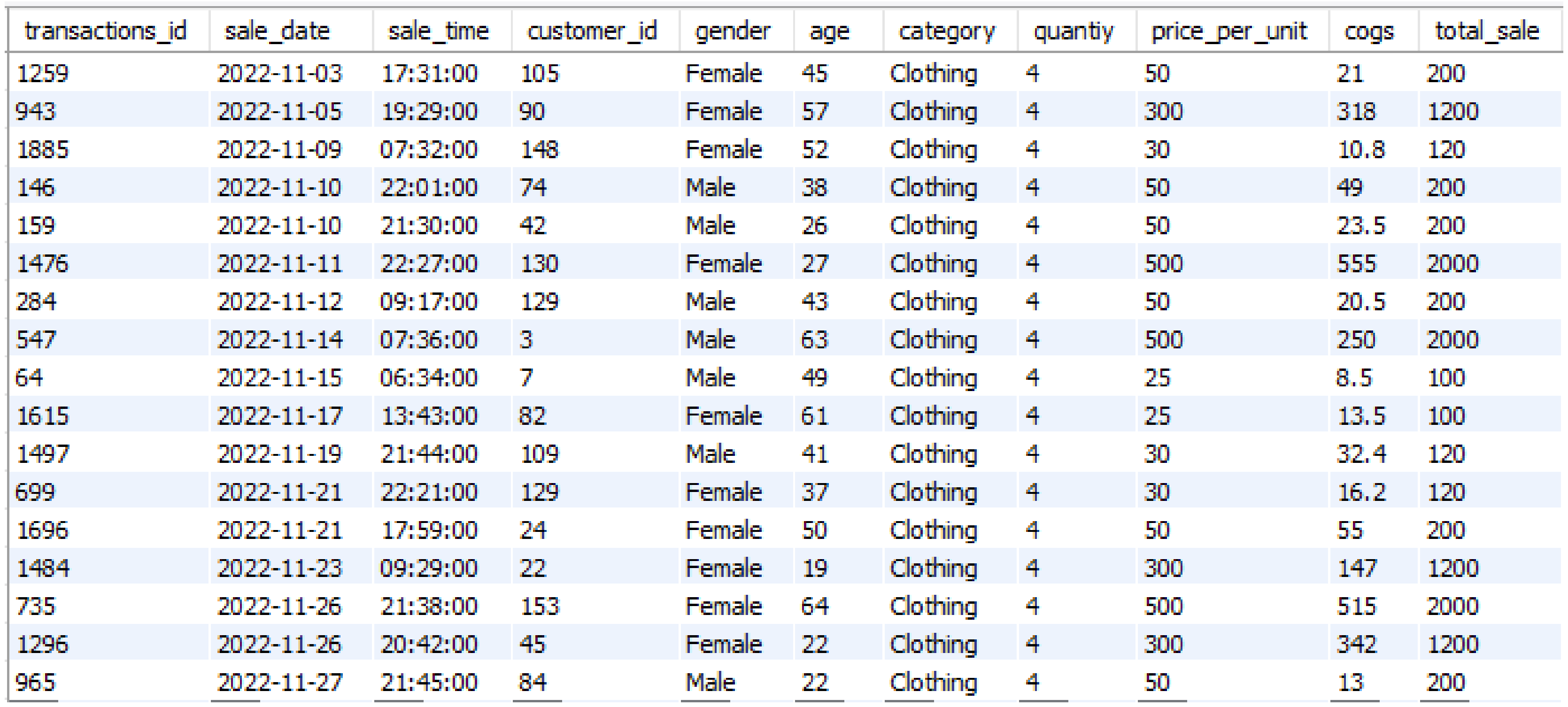
```
SELECT * FROM retailsales
WHERE sale_date = '2022-11-05';
```



transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	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
1587	2022-11-05	20:06:00	140	Female	40	Beauty	4	300	105	1200
1819	2022-11-05	20:44:00	83	Female	35	Beauty	2	50	13.5	100
1896	2022-11-05	20:19:00	87	Female	30	Electronics	2	25	30.75	50
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Write a SQL query to retrieve all transactions where category is 'clothing' and the quantity sold is more than 3 in month of nov 2022

```
SELECT * FROM retailsales

WHERE category = 'Clothing'

AND quantiy > 3

AND sale_date BETWEEN '2022-11-01' AND '2022-11-30'

ORDER BY sale_date;
```

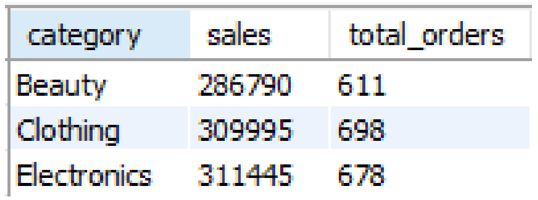


transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
1259	2022-11-03	17:31:00	105	Female	45	Clothing	4	50	21	200
943	2022-11-05	19:29:00	90	Female	57	Clothing	4	300	318	1200
1885	2022-11-09	07:32:00	148	Female	52	Clothing	4	30	10.8	120
146	2022-11-10	22:01:00	74	Male	38	Clothing	4	50	49	200
159	2022-11-10	21:30:00	42	Male	26	Clothing	4	50	23.5	200
1476	2022-11-11	22:27:00	130	Female	27	Clothing	4	500	555	2000
284	2022-11-12	09:17:00	129	Male	43	Clothing	4	50	20.5	200
547	2022-11-14	07:36:00	3	Male	63	Clothing	4	500	250	2000
64	2022-11-15	06:34:00	7	Male	49	Clothing	4	25	8.5	100
1615	2022-11-17	13:43:00	82	Female	61	Clothing	4	25	13.5	100
1497	2022-11-19	21:44:00	109	Male	41	Clothing	4	30	32.4	120
699	2022-11-21	22:21:00	129	Female	37	Clothing	4	30	16.2	120
1696	2022-11-21	17:59:00	24	Female	50	Clothing	4	50	55	200
1484	2022-11-23	09:29:00	22	Female	19	Clothing	4	300	147	1200
735	2022-11-26	21:38:00	153	Female	64	Clothing	4	500	515	2000
1296	2022-11-26	20:42:00	45	Female	22	Clothing	4	300	342	1200
965	2022-11-27	21:45:00	84	Male	22	Clothing	4	50	13	200

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

```
SELECT

        category,

        SUM(total_sale) AS sales,

        COUNT(*) AS total_orders

FROM retailsales
```

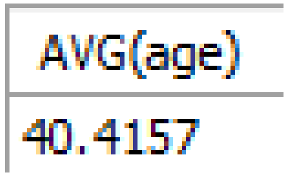
```sql
GROUP BY category;
```



category	sales	total_orders
Beauty	286790	611
Clothing	309995	698
Electronics	311445	678

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

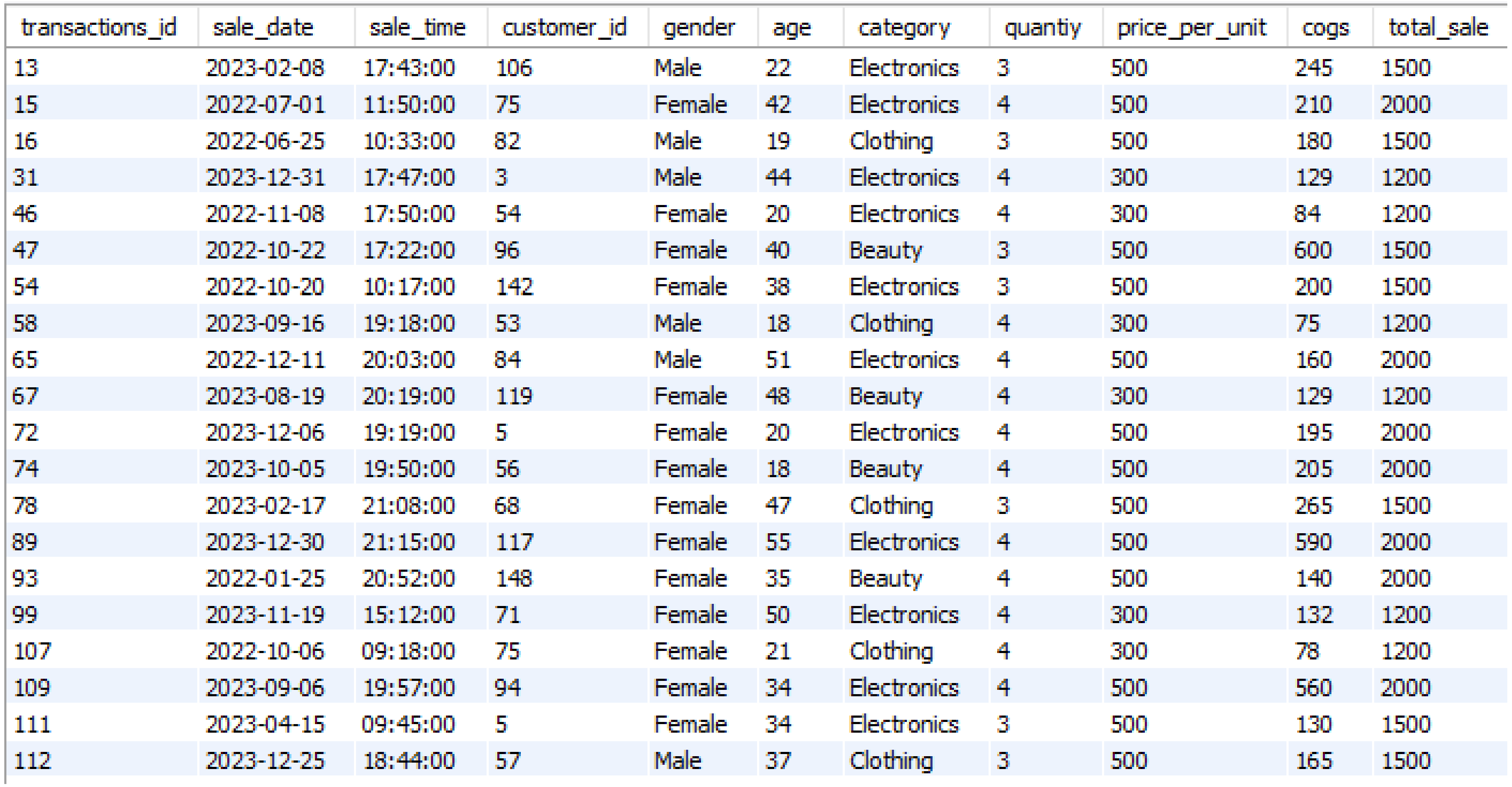
```sql
SELECT AVG(age) FROM retailsales

WHERE category = 'Beauty';
```



AVG(age)
40.4157

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

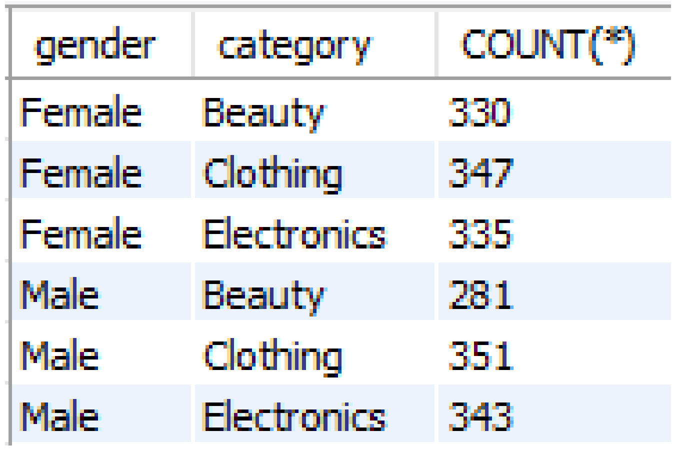
```sql
SELECT * FROM retailsales

WHERE total_sale > 1000;
```



transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	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
65	2022-12-11	20:03:00	84	Male	51	Electronics	4	500	160	2000
67	2023-08-19	20:19:00	119	Female	48	Beauty	4	300	129	1200
72	2023-12-06	19:19:00	5	Female	20	Electronics	4	500	195	2000
74	2023-10-05	19:50:00	56	Female	18	Beauty	4	500	205	2000
78	2023-02-17	21:08:00	68	Female	47	Clothing	3	500	265	1500
89	2023-12-30	21:15:00	117	Female	55	Electronics	4	500	590	2000
93	2022-01-25	20:52:00	148	Female	35	Beauty	4	500	140	2000
99	2023-11-19	15:12:00	71	Female	50	Electronics	4	300	132	1200
107	2022-10-06	09:18:00	75	Female	21	Clothing	4	300	78	1200
109	2023-09-06	19:57:00	94	Female	34	Electronics	4	500	560	2000
111	2023-04-15	09:45:00	5	Female	34	Electronics	3	500	130	1500
112	2023-12-25	18:44:00	57	Male	37	Clothing	3	500	165	1500

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

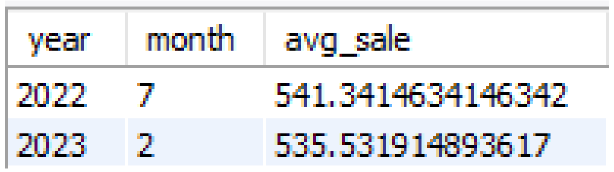
```
SELECT
        gender,
        category,
        COUNT(*)
FROM retailsales
GROUP BY gender,category
ORDER BY gender;
```



```
gender   category     COUNT(*)
Female   Beauty       330
Female   Clothing     347
Female   Electronics  335
Male     Beauty       281
Male     Clothing     351
Male     Electronics  343
```

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

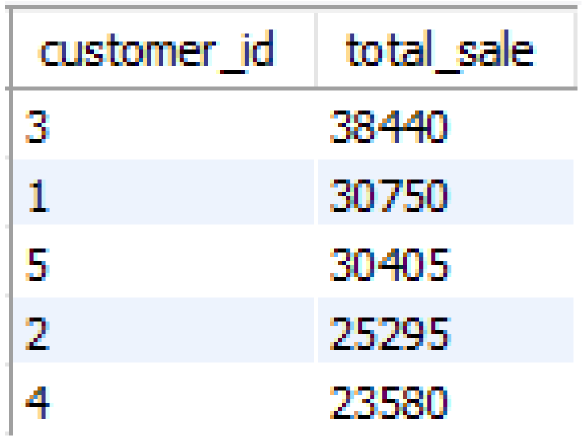
```
SELECT
        year,
        month,
        avg_sale
FROM
(
SELECT
        YEAR(sale_date) AS year,
        MONTH(sale_date) AS month,
        AVG(total_sale) AS avg_sale,
        RANK() OVER(PARTITION BY YEAR(sale_date) ORDER BY AVG(total_sale)DESC) AS pos
FROM retailsales
GROUP BY year, month
) AS T1
WHERE pos = 1;
```



year	month	avg_sale
2022	7	541.3414634146342
2023	2	535.531914893617

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

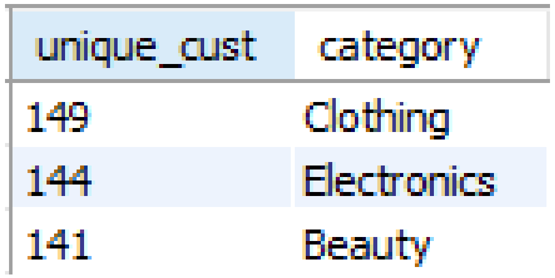
```
SELECT
        customer_id,
        SUM(total_sale) AS total_sale
FROM retailsales
GROUP BY customer_id
ORDER BY total_sale DESC
LIMIT 5;
```



customer_id	total_sale
3	38440
1	30750
5	30405
2	25295
4	23580

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

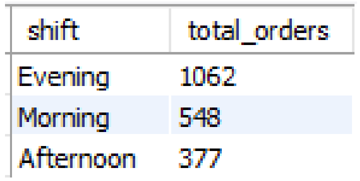
```
SELECT
        COUNT(DISTINCT customer_id) AS unique_cust,
        category
FROM retailsales
GROUP BY category
ORDER BY unique_cust DESC;
```



unique_cust	category
149	Clothing
144	Electronics
141	Beauty

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

```
WITH hourly_sales
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 retailsales
)
SELECT
        shift,
         COUNT(*) AS total_orders
FROM hourly_sales
GROUP BY shift;
```



```
shift       total_orders
Evening     1062
Morning     548
Afternoon   377
```

4. 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 is a basic SQL Data Analysis comprising of database setup, data cleaning, exploratory data analytics(EDA) and business-driven SQL queries. Findings from this projects will help solve the business problems and data driven decision making.

How to use.

- Download the Folder named 'Retail Sales Analysis'.

- Unzip the file.

- Open the file named 'myfile.sql'.