Retail Sales Analysis with SQL

Divyanshi Nigam

Objective

This project is designed to demonstrate SQL skills and techniques typically used by data analysts to explore, clean, and analyze retail sales data. The project involves setting up a retail sales database, performing exploratory data analysis (EDA), and answering specific business questions through SQL queries.

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.

Date formating .csv file

As the .csv file contained date in “dd-mm-yyyy” format so, first change the format by either custom format to “yyyy-mm-dd” or adding a custom column then text(sale_date,”yyyy-mm-dd”) and copy and paste as values, or do by power query and then save the file as .csv utf-8

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	transactio	sale_time	customer_	gender	age	category	quantiy	price_per	cogs	total_sale	sale_date	age	
2	180	10:47:00	117	Male	41	Clothing	3	300	129	900	2022-11-05	41	
3	522	11:00:00	52	Male	46	Beauty	3	500	145	1500	2022-07-09	46	
4	559	10:48:00	5	Female	40	Clothing	4	300	84	1200	2022-12-12	40	
5	1180	08:53:00	85	Male	41	Clothing	3	300	129	900	2022-01-06	41	
6	1522	08:35:00	48	Male	46	Beauty	3	500	235	1500	2022-11-14	46	
7	1559	07:40:00	49	Female	40	Clothing	4	300	144	1200	2022-08-20	40	
8	163	09:38:00	144	Female	64	Clothing	3	50	23	150	2022-10-31	64	
9	303	11:09:00	54	Male	19	Electronic	3	30	14.7	90	2022-04-22	19	
10	421	08:43:00	66	Female	37	Clothing	3	500	235	1500	2022-04-08	37	
11	979	10:18:00	6	Female	19	Beauty	1	25	10.5	25	2022-05-18	19	
12	1163	10:52:00	120	Female	64	Clothing	3	50	27	150	2022-05-04	64	
13	1303	08:59:00	58	Male	19	Electronic	3	30	15	90	2022-03-19	19	
14	1421	07:07:00	59	Female	37	Clothing	3	500	185	1500	2022-01-17	37	
15	1979	11:34:00	102	Female	19	Beauty	1	25	7.75	25	2022-08-17	19	
16	610	06:56:00	137	Female	26	Beauty	2	300	93	600	2022-12-18	26	
17	1610	10:18:00	1	Female	26	Beauty	2	300	102	600	2022-11-23	26	
18	32	09:11:00	150	Male	30	Beauty	3	30	8.4	90	2022-07-16	30	
19	231	07:02:00	50	Female	23	Clothing	3	50	26.5	150	2022-07-09	23	
20	683	10:22:00	82	Male	38	Beauty	2	500	175	1000	2022-03-06	38	
21	1032	08:15:00	1	Male	30	Beauty	3	30	10.5	90	2022-04-01	30	
22	1231	07:05:00	12	Female	23	Clothing	3	50	23	150	2022-01-29	23	

SQL - Retail Sales Analysis_utf

Power Query

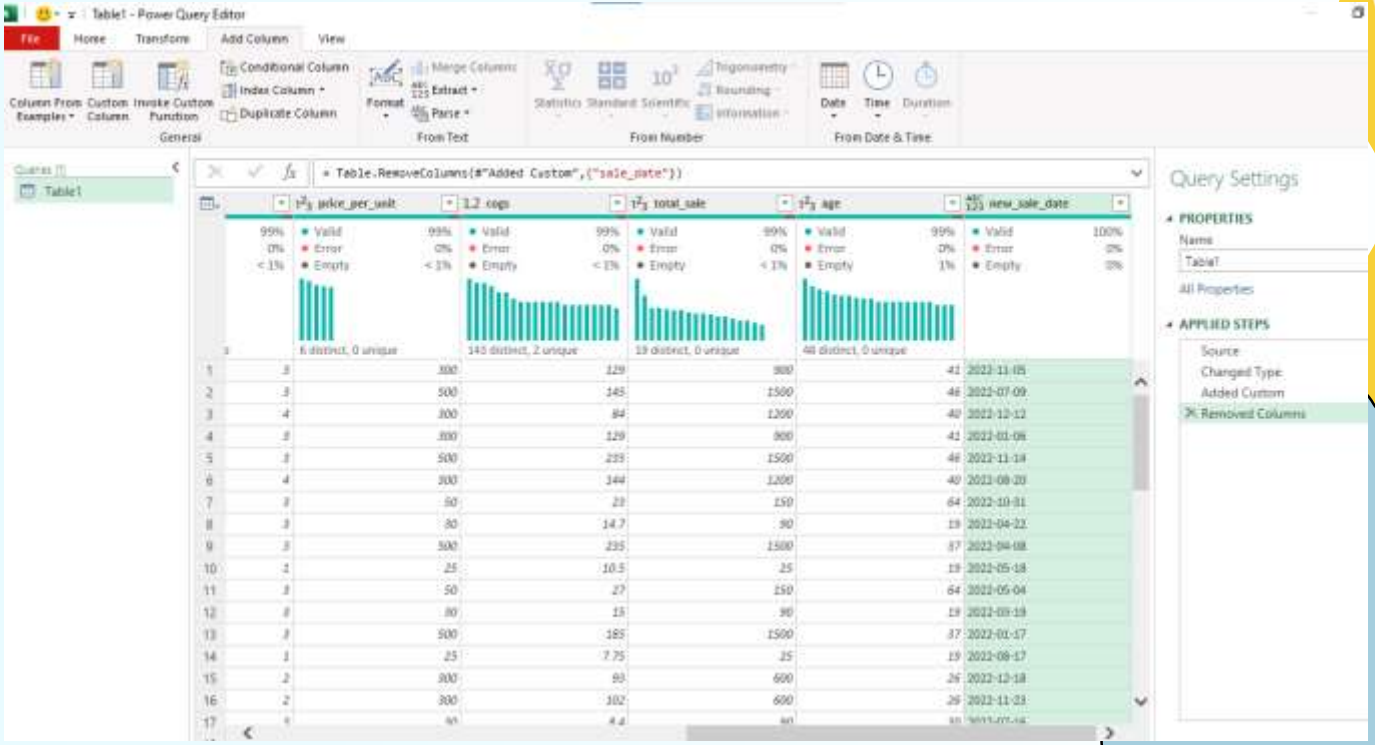
Table1 - Power Query Editor
File
Home
Transform
Add Column
View
Column From Examples
Custom Column
Invoke Custom Function
Conditional Column
Index Column
Duplicate Column
General
Format
Merge Columns
Extract
Parse
From Text
Statistics
Standard
Scientific
Trigonometry
Rounding
Information
From Number
Date
Time
Duration
From Date & Time
Queries [1]
Table1
= Table.RemoveColumns(#"Added Custom",{"sale_date"})
price_per_unit
cogs
total_sale
age
new_sale_date
Query Settings
PROPERTIES
Name
Table1
All Properties
APPLIED STEPS
Source
Changed Type
Added Custom
Removed Columns

Our Database and Tables

•**Database Creation**: The project starts by creating a database named retail_sales_analysis.
•**Table Creation**: A table named retail_sales is created to store the sales data. The table structure includes columns for transactions_id(primary key), sale)date, sale time, customer ID, gender, age, product category, quantity sold, price per unit, cost of goods sold (COGS), and total sale amount.

```sql
create database retail_sales_analysis;
use retail_sales_analysis;
/*table creation*/
create table retail_sales(
transactions_id int,
sale_time time,
customer_id int,
gender  varchar(20),
category varchar(20),
quantiy int,
price_per_unit  float,
cogs float,
total_sale float,
age int,
sale_date date
);
alter table retail_sales
change quantiy quantity int;
select * from retail_sales;
```

Data Cleaning

```
/*DATA CLEANING*/
/*to check for null values*/
select * from retail_sales
where
    transactions_id 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
    or age is null or sale_date is null;

SET SQL_SAFE_UPDATES = 0;

/*delete null values if exists*/
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;
```

Data Exploration

- **Record Count**: Determine the total number of records in the dataset.
- **Customer Count**: Find out how many unique customers are in the dataset.
- **Category Count**: Identify all unique product categories in the dataset.

```
/*DATA EXPLORATION*/
/*How many sale we have*/
select count(*) as total_sales from retail_Sales;
/*how unique many customers we have?*/
select count(distinct customer_id) as total_customers from retail_Sales;
/*how many unique categories we have?*/
select count(distinct category) as "total unique categories" from retail_sales;
select
```

	total_sales
▶	1987

	total_customers
▶	155

	total unique categories
▶	3

Data analysis a
findings

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

```
select *
from retail_sales
where sale_date="2022-11-05";
```

transactions_id	sale_time	customer_id	gender	category				al_sale	age	sale_date
180	10:47:00	117	Male	Clothing	[illegible]	[illegible]	[illegible]	[illegible]	41	2022-11-05
240	11:49:00	95	Female	Beauty	1	300	123	300	23	2022-11-05
1256	09:58:00	29	Male	Clothing	2	500	190	1000	23	2022-11-05
1587	20:06:00	140	Female	Beauty	4	300	105	1200	40	2022-11-05
1819	20:44:00	83	Female	Beauty	2	50	13.5	100	35	2022-11-05
943	19:29:00	90	Female	Clothing	4	300	318	1200	57	2022-11-05
1896	20:19:00	87	Female	Electronics	2	25	30.75	50	30	2022-11-05
1137	22:34:00	104	Male	Beauty	2	500	145	1000	46	2022-11-05
856	17:43:00	102	Male	Electronics	4	30	9.3	120	54	2022-11-05
214	16:31:00	53	Male	Beauty	2	30	8.1	60	20	2022-11-05
1265	14:35:00	86	Male	Clothing	3	300	111	900	55	2022-11-05

```
select *
from retail_sales
where sale_date="2022
```

Write a SQL query to retrieve all transactions where the categ
'Clothing' and the quantity sold is more than 4 in the month of
2022:

```
select *
from retail_sales
where category="Clothing" and quantity >=4 and year(sale_date)=2022 and month(sale_date)= 11;
/* DATE_FORMAT(sale_date, '%Y-%m') = '2022-11'*/
```

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

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

```sql
select category,
sum(total_sale) as "net_sales",
count(*) as "total_orders"
from retail_sales
group by category
order by net_sales desc;
```

Result Grid | Filter Rows:

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

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

```sql
select category,
round(avg(age),2) as "Average age of customers"
from retail_sales
where category="Beauty";
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

	category	Average age of customers
▶	Beauty	40.42

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

```sql
select *
from retail_sales
where total_sale>1000
order by total_sale desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

transactions_id	sale_time	customer_id	gender	category	quantity	price_per_unit	cogs	total_sale	age	sale_date
15	11:50:00	75	Female	Electronics	4	500	210	2000	42	2022-07-01
743	07:54:00	55	Female	Beauty	4	500	260	2000	34	2022-08-07
1015	11:53:00	94	Female	Electronics	4	500	200	2000	42	2022-03-09
1743	09:37:00	47	Female	Beauty	4	500	250	2000	34	2022-10-26
742	06:08:00	37	Female	Electronics	4	500	195	2000	38	2022-03-19
1742	08:25:00	18	Female	Electronics	4	500	220	2000	38	2022-11-22
420	10:53:00	28	Female	Clothing	4	500	200	2000	22	2022-01-02
1420	07:01:00	138	Female	Clothing	4	500	205	2000	22	2022-04-15
592	09:15:00	77	Female	Beauty	4	500	275	2000	46	2022-12-26
1592	09:08:00	81	Female	Beauty	4	500	155	2000	46	2022-03-16
269	11:31:00	87	Male	Clothing	4	500	250	2000	25	2022-09-19
1269	08:09:00	71	Male	Clothing	4	500	145	2000	25	2022-01-01
577	11:55:00	45	Male	Beauty	4	500	215	2000	21	2022-04-21
1577	06:22:00	145	Male	Beauty	4	500	160	2000	21	2022-09-11

retail_sales 23

Output

Action Output

#	Time	Action	Message
47	14:42:38	select * from retail_sales where total_sale>1000 LIMIT 0, 1000	306 row(s) returned
48	14:43:36	select * from retail_sales where total_sale>1000 order by total_sale desc LIMIT 0, 1000	306 row(s) returned

transactions_id	sale_time	customer_id	gender	category	quantity	price_per_unit	cogs	total_s
1199	17:46:00	110	Male	Beauty	3	500	190	1500
580	14:44:00	104	Female	Clothing	3	500	200	1500
1580	15:47:00	105	Female	Clothing	3	500	250	1500
805	13:55:00	59	Female	Beauty	3	500	155	1500
1805	13:35:00	79	Female	Beauty	3	500	225	1500
211	14:02:00	54	Male	Beauty	3	500	235	1500
1211	14:59:00	82	Male	Beauty	3	500	235	1500
559	10:48:00	5	Female	Clothing	4	300	84	1200
1559	07:40:00	49	Female	Clothing	4	300	144	1200
484	07:52:00	135	Female	Clothing	4	300	75	1200
1484	09:29:00	22	Female	Clothing	4	300	147	1200
320	08:35:00	57	Female	Electronics	4	300	159	1200
1320	11:55:00	102	Female	Electronics	4	300	84	1200
142	10:05:00	61	Male	Electronics	4	300	138	1200

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

```sql
select category,
gender,
count(transactions_id) as "total_transactions"
from retail_sales
group by category,gender
order by total_transactions desc;
```

Result Grid | Filter Rows: | Export: | Wrap

category	gender	total_transactions
Clothing	Male	351
Clothing	Female	347
Electronics	Male	343
Electronics	Female	335
Beauty	Female	330
Beauty	Male	281

```sql

select category,
count(case when gender="Male" then transactions_id end)
count(Case when gender="Female" then transactions_id end
from retail_sales
group by category;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	Male	Female
Clothing	351	347
Beauty	281	330
Electronics	343	335

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

```sql
select
year(sale_date) as "Year",
month(sale_date) as "Month_number",
monthname(sale_date) as "Month",
round(avg(total_sale),2) as "Average sale"
from retail_sales
group by Year,Month_number,Month
order by Year, Month_number;
```

Year	Month_number	Month	Average sale
2022	1	January	397.11
2022	2	February	366.14
2022	3	March	521.22
2022	4	April	500.61
2022	5	May	400
2022	6	June	481.4
2022	7	July	541.34
2022	8	August	390.28
2022	9	September	485.2
2022	10	October	467.14
2022	11	November	472.02
2022	12	December	460.77
2023	1	January	396.5
2023	2	February	535.53
2023	3	March	394.81
2023	4	April	466.49
2023	5	May	450.17
2023	6	June	438.48
2023	7	July	427.68
2023	8	August	495.96
2023	9	September	462.71
2023	10	October	399.17
2023	11	November	453.45
2023	12	December	490.39

```sql
/*Write a SQL query to calculate the average sale for each month. Find out best selling
select * from
(select
year(sale_date) as "Year",
month(sale_date) as "Month_number",
monthname(sale_date) as "Month",
round(avg(total_sale),2) as "Average sale",
rank() over (partition by year(sale_date) order by round(avg(total_sale),2) Desc) as mon
from retail_sales
group by Year,Month_number,Month
) as t1
where month_rank=1;
```

Year	Month_number	Month	Average sale	mo
2022	7	July	541.34	1
2023	2	February	535.53	1

Write a SQL query to find the top 5 customers based on the highes 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;
```

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

Write a SQL query to find the number of unique customers wh purchased items from each category

```sql
select count(distinct customer_id) as "unique_customers",
category
from retail_sales
group by category
order by unique_customers desc;
```

unique_customers	category
149	Clothing
144	Electronics
141	Beauty

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

```sql
with hourly_sale as
(select *,
case
when hour(sale_time)<12 then "Morning"
when hour(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;
```

	Shift	total_orders
▶	Morning	548
	Evening	1062
	Afternoon	377

Key Findings

1. Customer Demographics:

•The dataset spans **a wide range of age groups**, with the **average customer age being 40**.
•A significant portion of transactions in the **Beauty** category came from customers in this age group, accounting for **306 high-value transactions** (out of 1,987) where total sale > 1,000.

2. High-Value Transactions

•Several transactions exceeded **₹1,000 in total sales**, indicating a presence of **premium buyers**.
•**Customer ID 3** is the top spender, with a total purchase value of **₹38,440**.

3. Sales by Category
•**Electronics** leads with the **highest net sales** of **₹311,445** from **678 orders**.
•**Clothing** has the **highest number of orders** (698), generating **₹309,995** in net sales.
•**Beauty** ranks lowest in both metrics with **611 orders** and **₹286,790** in net sales.

Key Findings

4. Gender-Based Insights
•**Clothing** is the most inclusive category, with **351 male** and **347 female** buyers.
•**Electronics** has slightly more male buyers (**343**) than female (**335**).
•**Beauty** sees lower engagement from both genders (**281 male**, **330 female**), suggesting a need for targeted marketing and promotional efforts.

5. Monthly Sales Trends
•Sales vary significantly by month:

- **July 2022** was the best-selling month with an average sale of **₹541.4**.
- **February 2023** followed closely with an average sale of **₹535**.

•This helps identify **seasonal peaks** and plan inventory and marketing accordingly.

6. Unique Customer Distribution
•**Clothing** has the highest number of **unique customers (149)**.
•Followed by **Electronics (144)** and **Beauty (141)**.

Key Findings

7. Order Time Analysis

•Most orders are placed in the **evening (1,062 orders)**.

•Followed by the **morning**, while the **afternoon** sees the least activity (**377 orders**).

•This insight supports optimizing promotions and ad timing for higher conversion.

Thank you!

Divyanshi Nigam