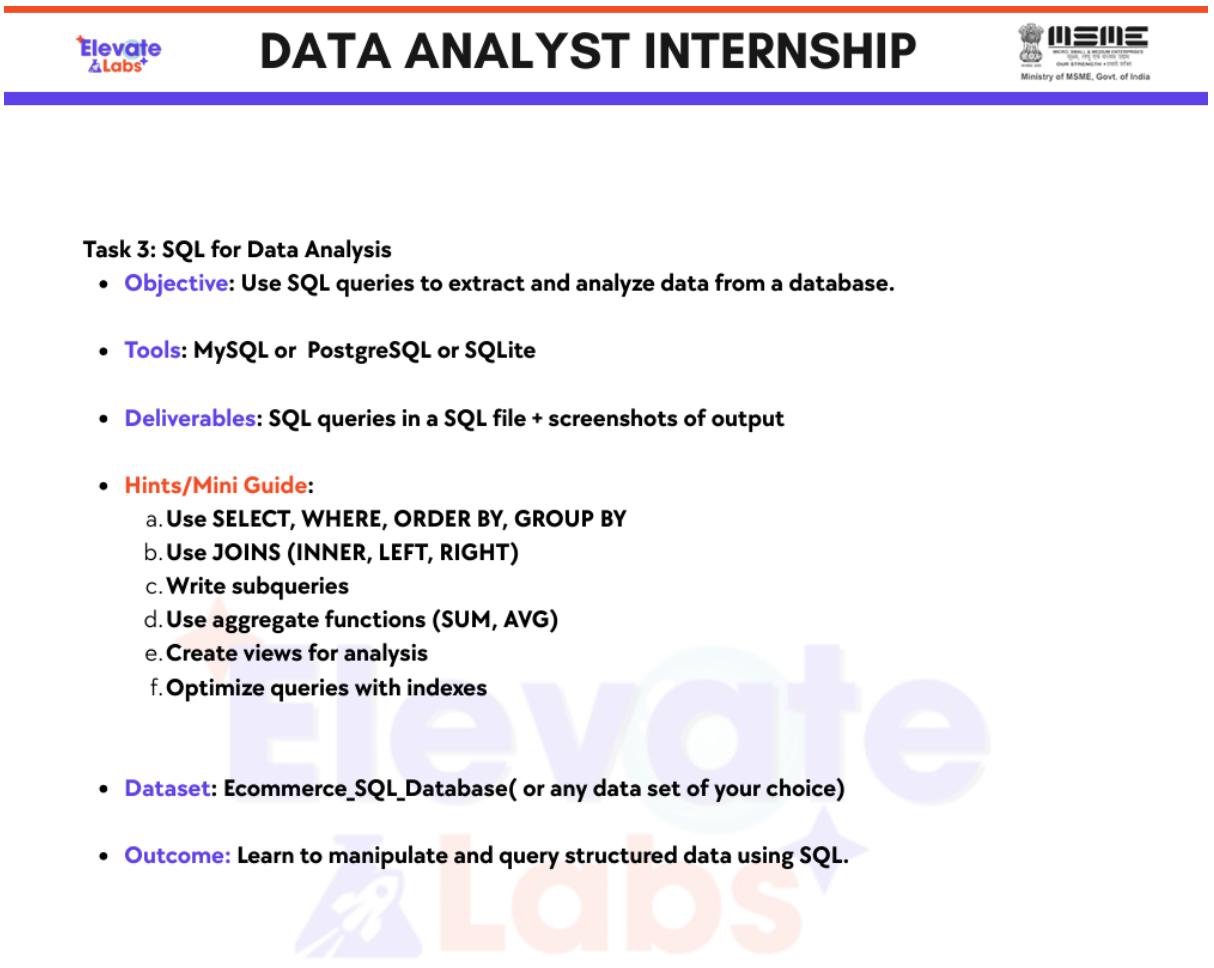
Name : Suhitha K

Task No : 04



DATA ANALYST INTERNSHIP

Task 3: SQL for Data Analysis

- **Objective: Use SQL queries to extract and analyze data from a database.**
- **Tools: MySQL or PostgreSQL or SQLite**
- **Deliverables: SQL queries in a SQL file + screenshots of output**
- **Hints/Mini Guide:**
  a. **Use SELECT, WHERE, ORDER BY, GROUP BY**
  b. **Use JOINS (INNER, LEFT, RIGHT)**
  c. **Write subqueries**
  d. **Use aggregate functions (SUM, AVG)**
  e. **Create views for analysis**
  f. **Optimize queries with indexes**
- **Dataset: Ecommerce_SQL_Database(or any data set of your choice)**
- **Outcome: Learn to manipulate and query structured data using SQL.**

Ecommerce SQL Analysis — My Step-by-Step Report

1. Setting Up the Environment

I started by setting up my workspace using **DB Browser for SQLite**, which is a free and easy-to-use tool for running SQL queries locally. I downloaded and installed it, then used the **Execute SQL** tab to write and run my queries.

2. Creating and Querying Views

I created a view called HighValuePurchases to filter orders where the "Final_Price(Rs.)" was greater than 10,000. When I first tried to create the view, I got an error saying it already existed. To fix this, I ran:

```
DROP VIEW IF EXISTS HighValuePurchases;

CREATE VIEW HighValuePurchases AS
```

```
SELECT "User_ID", "Category", "Final_Price(Rs.)", "Payment_Method"
FROM ecommerce_dataset_updated
WHERE "Final_Price(Rs.)" > 10000;
```

After creating the view, I queried it with:

```
SELECT * FROM HighValuePurchases;
```

But the result was empty, so I investigated further.

3. Data Investigation

To understand why the view was empty, I ran queries to check the data:

- I found the maximum final price in the dataset using:

```
SELECT MAX("Final_Price(Rs.)") FROM ecommerce_dataset_updated;
```

- I also checked the data type of the "Final_Price(Rs.)" column:

```
SELECT typeof("Final_Price(Rs.)") FROM ecommerce_dataset_updated LIMIT 5;
```

I realized that if the column was stored as text, it might affect numeric comparisons, so I tried casting it to an integer when needed.

4. Managing Indexes

To optimize queries involving the "Final_Price(Rs.)" column, I created an index:

```
CREATE INDEX idx_final_price ON ecommerce_dataset_updated("Final_Price(Rs.)");
```

When I tried to recreate the index later, I got an error because it already existed. To resolve this, I dropped the existing index first:

```
DROP INDEX IF EXISTS idx_final_price;
CREATE INDEX idx_final_price ON ecommerce_dataset_updated("Final_Price(Rs.)");
```

I also explored the existing indexes on my table using:

```
PRAGMA index_list('ecommerce_dataset_updated');
```

And checked details of my index with:

```
PRAGMA index_info('idx_final_price');
```

5. Summary of What I Learned

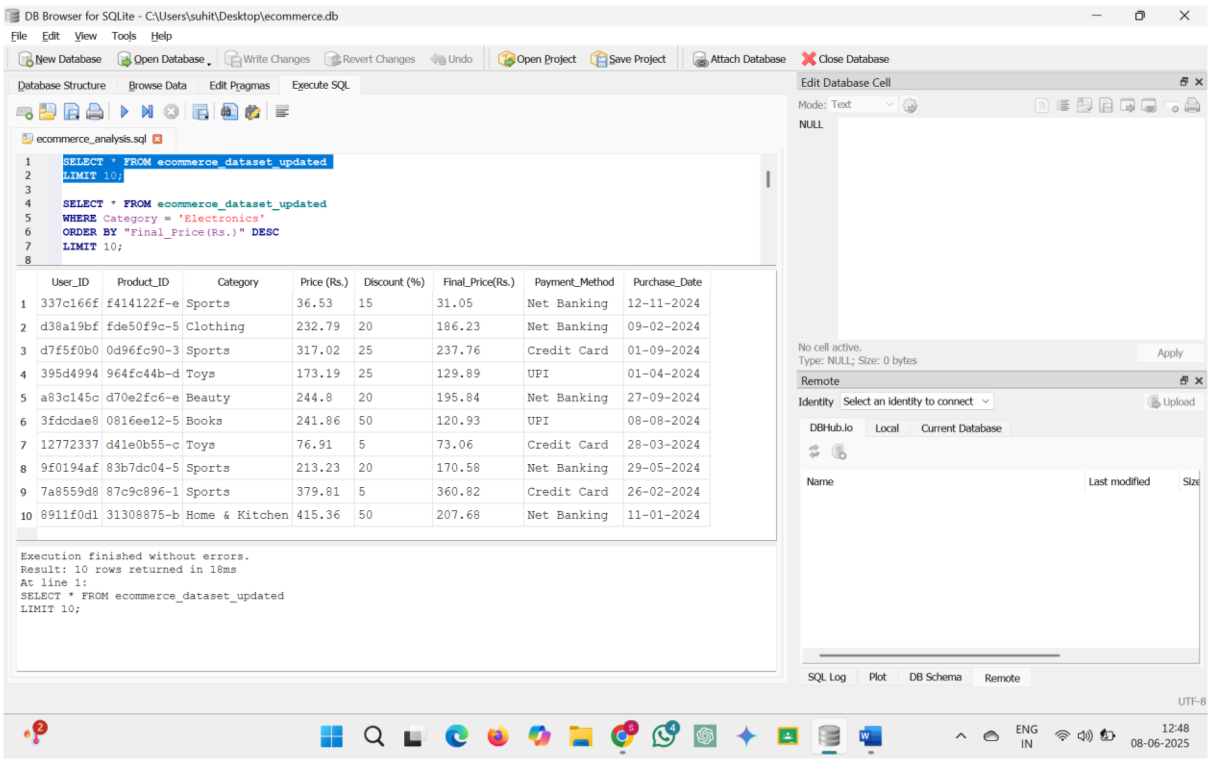
- How to create, drop, and query **views** in SQL.
- How to troubleshoot empty results by examining the data and data types.
- How to create and manage **indexes** to improve query performance.
- How to use SQLite pragmas to explore the database's metadata.
- How to use DB Browser for SQLite to write, run, and debug SQL queries.

Screenshots of SQL queries along with their results :

Query 1 :

SELECT * FROM ecommerce_dataset_updated

LIMIT 10;


DB Browser for SQLite - C:\Users\suhit\Desktop\ecommerce.db
File Edit View Tools Help
New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database
Database Structure Browse Data Edit Pragmas Execute SQL
ecommerce_analysis.sql
Edit Database Cell
Remote

```
SELECT * FROM ecommerce_dataset_updated
LIMIT 10;

SELECT * FROM ecommerce_dataset_updated
WHERE Category = 'Electronics'
ORDER BY "Final_Price(Rs.)" DESC
LIMIT 10;
```

	User_ID	Product_ID	Category	Price (Rs.)	Discount (%)	Final_Price(Rs.)	Payment_Method	Purchase_Date
1	337c166f	f414122f-e	Sports	36.53	15	31.05	Net Banking	12-11-2024
2	d38a19bf	fde50f9c-5	Clothing	232.79	20	186.23	Net Banking	09-02-2024
3	d7f5f0b0	0d96fc90-3	Sports	317.02	25	237.76	Credit Card	01-09-2024
4	395d4994	964fc44b-d	Toys	173.19	25	129.89	UPI	01-04-2024
5	a83c145c	d70e2fc6-e	Beauty	244.8	20	195.84	Net Banking	27-09-2024
6	3fdcdae8	0816ee12-5	Books	241.86	50	120.93	UPI	08-08-2024
7	12772337	d41e0b55-c	Toys	76.91	5	73.06	Credit Card	28-03-2024
8	9f0194af	83b7dc04-5	Sports	213.23	20	170.58	Net Banking	29-05-2024
9	7a8559d8	87c9c896-1	Sports	379.81	5	360.82	Credit Card	26-02-2024
10	8911f0d1	31308875-b	Home & Kitchen	415.36	50	207.68	Net Banking	11-01-2024

```
Execution finished without errors.
Result: 10 rows returned in 18ms
At line 1:
SELECT * FROM ecommerce_dataset_updated
LIMIT 10;
```

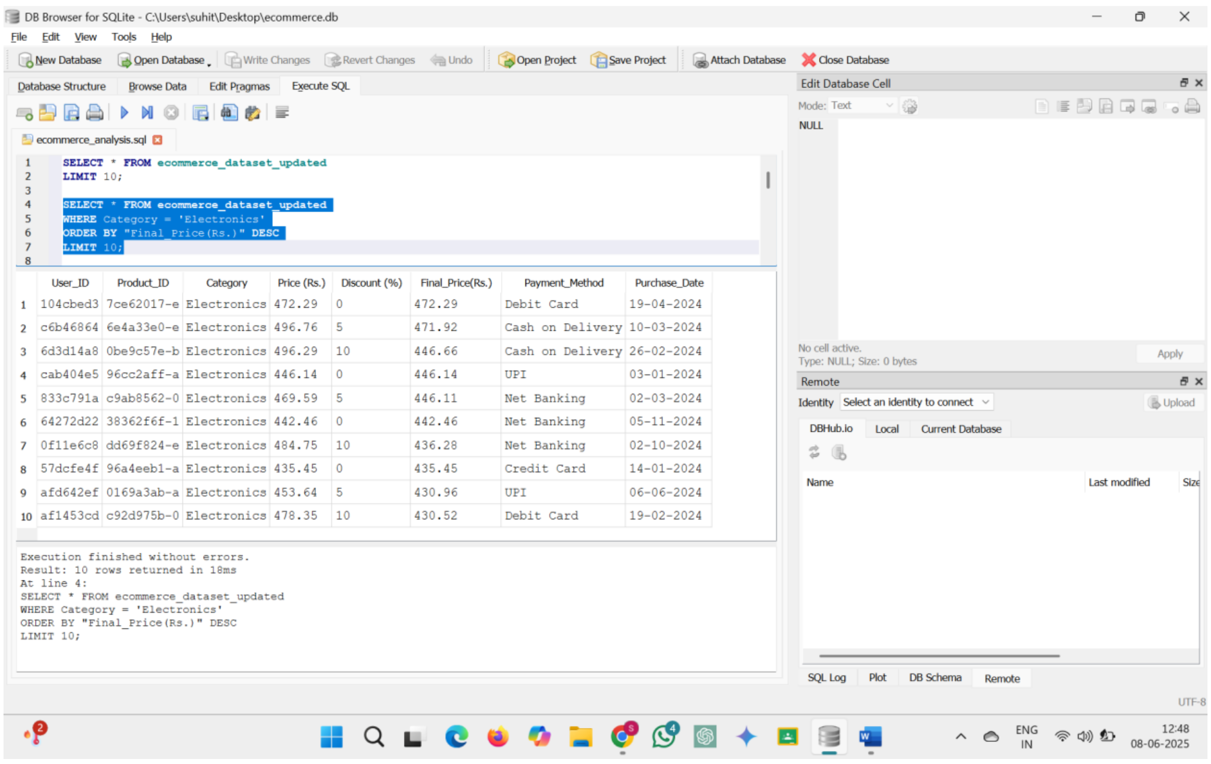
Query 2:

SELECT * FROM ecommerce_dataset_updated

WHERE Category = 'Electronics'

ORDER BY "Final_Price(Rs.)" DESC

LIMIT 10;


DB Browser for SQLite - C:\Users\suhit\Desktop\ecommerce.db
File Edit View Tools Help
New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database
Database Structure Browse Data Edit Pragmas Execute SQL
ecommerce_analysis.sql
Edit Database Cell
Mode: Text
NULL
No cell active.
Type: NULL; Size: 0 bytes
Apply
Remote
Identity Select an identity to connect
Upload
DBHub.io Local Current Database
Name Last modified Size
SQL Log Plot DB Schema Remote
UTF-8
ENG IN
12:48
08-06-2025

```sql
SELECT * FROM ecommerce_dataset_updated
LIMIT 10;

SELECT * FROM ecommerce_dataset_updated
WHERE Category = 'Electronics'
ORDER BY "Final_Price(Rs.)" DESC
LIMIT 10;
```

	User_ID	Product_ID	Category	Price (Rs.)	Discount (%)	Final_Price(Rs.)	Payment_Method	Purchase_Date
1	104cbed3	7ce62017-e	Electronics	472.29	0	472.29	Debit Card	19-04-2024
2	c6b46864	6e4a33e0-e	Electronics	496.76	5	471.92	Cash on Delivery	10-03-2024
3	6d3d14a8	0be9c57e-b	Electronics	496.29	10	446.66	Cash on Delivery	26-02-2024
4	cab404e5	96cc2aff-a	Electronics	446.14	0	446.14	UPI	03-01-2024
5	833c791a	c9ab8562-0	Electronics	469.59	5	446.11	Net Banking	02-03-2024
6	64272d22	38362f6f-1	Electronics	442.46	0	442.46	Net Banking	05-11-2024
7	0f11e6c8	dd69f824-e	Electronics	484.75	10	436.28	Net Banking	02-10-2024
8	57dcfe4f	96a4eeb1-a	Electronics	435.45	0	435.45	Credit Card	14-01-2024
9	afd642ef	0169a3ab-a	Electronics	453.64	5	430.96	UPI	06-06-2024
10	af1453cd	c92d975b-0	Electronics	478.35	10	430.52	Debit Card	19-02-2024

```
Execution finished without errors.
Result: 10 rows returned in 18ms
At line 4:
SELECT * FROM ecommerce_dataset_updated
WHERE Category = 'Electronics'
ORDER BY "Final_Price(Rs.)" DESC
LIMIT 10;
```

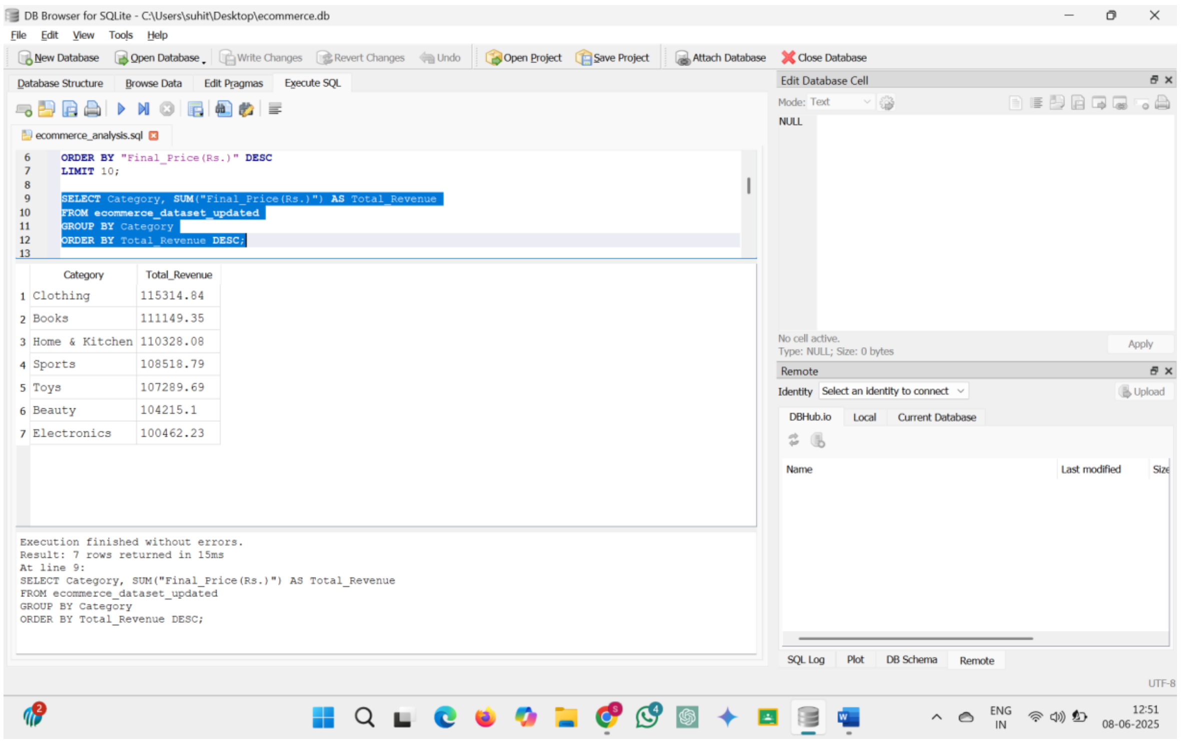
Query 3:

SELECT Category, SUM("Final_Price(Rs.)") AS Total_Revenue

FROM ecommerce_dataset_updated

GROUP BY Category

ORDER BY Total_Revenue DESC;


DB Browser for SQLite - C:\Users\suhit\Desktop\ecommerce.db
File Edit View Tools Help
New Database Open Database Write Changes Revert Changes Undo Open Project Save Project Attach Database Close Database
Database Structure Browse Data Edit Pragmas Execute SQL
ecommerce_analysis.sql
Edit Database Cell
Mode: Text
NULL
No cell active.
Type: NULL; Size: 0 bytes
Apply
Remote
Identity Select an identity to connect
Upload
DBHub.io Local Current Database
Name Last modified Size
SQL Log Plot DB Schema Remote
UTF-8
ENG IN
12:51
08-06-2025

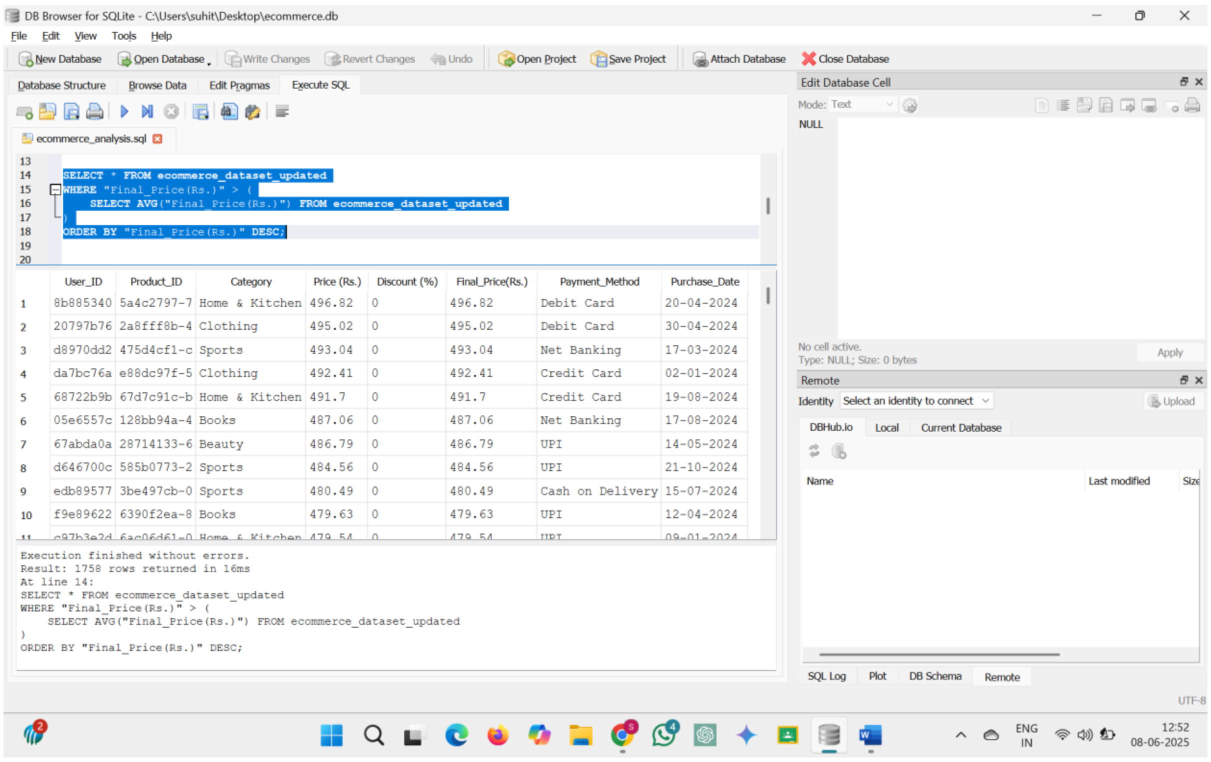
```sql
ORDER BY "Final_Price(Rs.)" DESC
LIMIT 10;

SELECT Category, SUM("Final_Price(Rs.)") AS Total_Revenue
FROM ecommerce_dataset_updated
GROUP BY Category
ORDER BY Total_Revenue DESC;
```

	Category	Total_Revenue
1	Clothing	115314.84
2	Books	111149.35
3	Home & Kitchen	110328.08
4	Sports	108518.79
5	Toys	107289.69
6	Beauty	104215.1
7	Electronics	100462.23

```
Execution finished without errors.
Result: 7 rows returned in 15ms
At line 9:
SELECT Category, SUM("Final_Price(Rs.)") AS Total_Revenue
FROM ecommerce_dataset_updated
GROUP BY Category
ORDER BY Total_Revenue DESC;
```

Query 4:

```
SELECT * FROM ecommerce_dataset_updated
WHERE "Final_Price(Rs.)" > (
    SELECT AVG("Final_Price(Rs.)") FROM ecommerce_dataset_updated
)
ORDER BY "Final_Price(Rs.)" DESC;
```



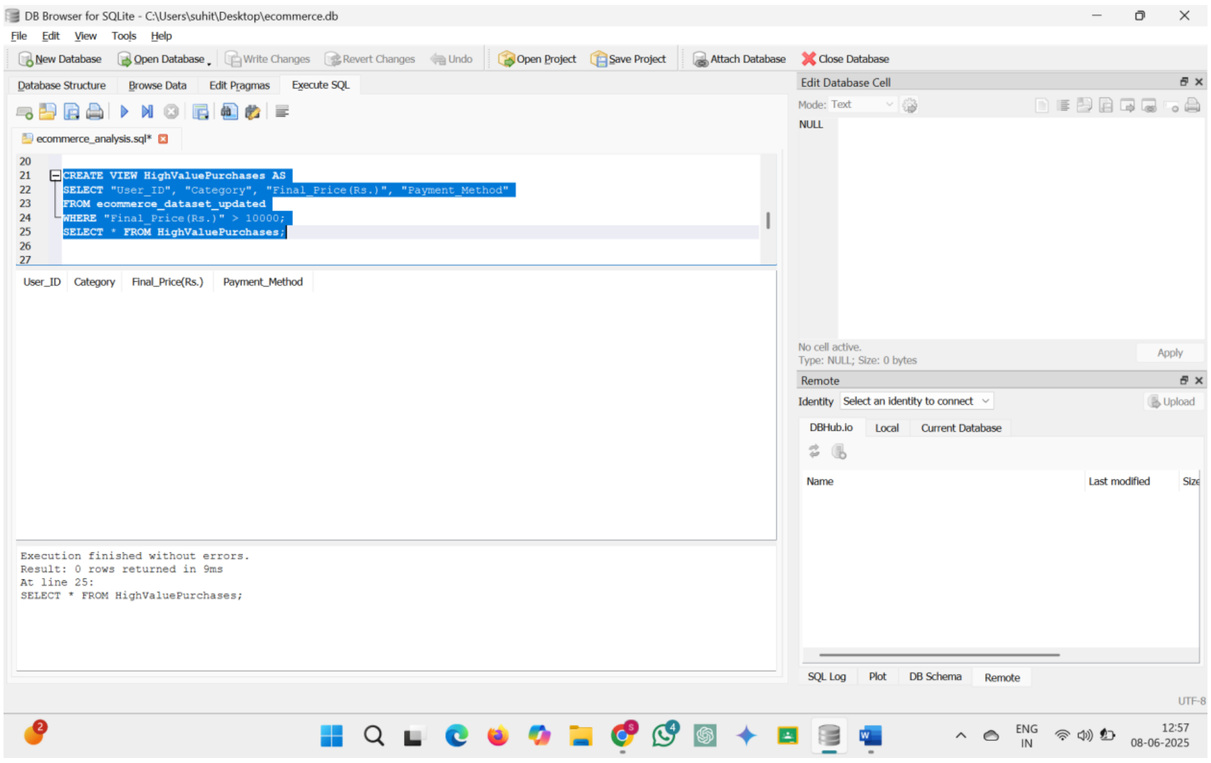
```
SELECT * FROM ecommerce_dataset_updated
WHERE "Final_Price(Rs.)" > (
    SELECT AVG("Final_Price(Rs.)") FROM ecommerce_dataset_updated
)
ORDER BY "Final_Price(Rs.)" DESC;
```

	User_ID	Product_ID	Category	Price (Rs.)	Discount (%)	Final_Price(Rs.)	Payment_Method	Purchase_Date
1	8b885340	5a4c2797-7	Home & Kitchen	496.82	0	496.82	Debit Card	20-04-2024
2	20797b76	2a8fff8b-4	Clothing	495.02	0	495.02	Debit Card	30-04-2024
3	d8970dd2	475d4cf1-c	Sports	493.04	0	493.04	Net Banking	17-03-2024
4	da7bc76a	e88dc97f-5	Clothing	492.41	0	492.41	Credit Card	02-01-2024
5	68722b9b	67d7c91c-b	Home & Kitchen	491.7	0	491.7	Credit Card	19-08-2024
6	05e6557c	128bb94a-4	Books	487.06	0	487.06	Net Banking	17-08-2024
7	67abda0a	28714133-6	Beauty	486.79	0	486.79	UPI	14-05-2024
8	d646700c	585b0773-2	Sports	484.56	0	484.56	UPI	21-10-2024
9	edb89577	3be497cb-0	Sports	480.49	0	480.49	Cash on Delivery	15-07-2024
10	f9e89622	6390f2ea-8	Books	479.63	0	479.63	UPI	12-04-2024
11	c97b3e2d	6ac06d61-0	Home & Kitchen	479.54	0	479.54	UPI	09-01-2024

```
Execution finished without errors.
Result: 1758 rows returned in 16ms
At line 14:
SELECT * FROM ecommerce_dataset_updated
WHERE "Final_Price(Rs.)" > (
    SELECT AVG("Final_Price(Rs.)") FROM ecommerce_dataset_updated
)
ORDER BY "Final_Price(Rs.)" DESC;
```

Query 5:

```
CREATE VIEW HighValuePurchases AS
SELECT "User_ID", "Category", "Final_Price(Rs.)", "Payment_Method"
FROM ecommerce_dataset_updated
WHERE "Final_Price(Rs.)" > 10000;
SELECT * FROM HighValuePurchases;
```



```
CREATE VIEW HighValuePurchases AS
SELECT "User_ID", "Category", "Final_Price(Rs.)", "Payment_Method"
FROM ecommerce_dataset_updated
WHERE "Final_Price(Rs.)" > 10000;
SELECT * FROM HighValuePurchases;
```

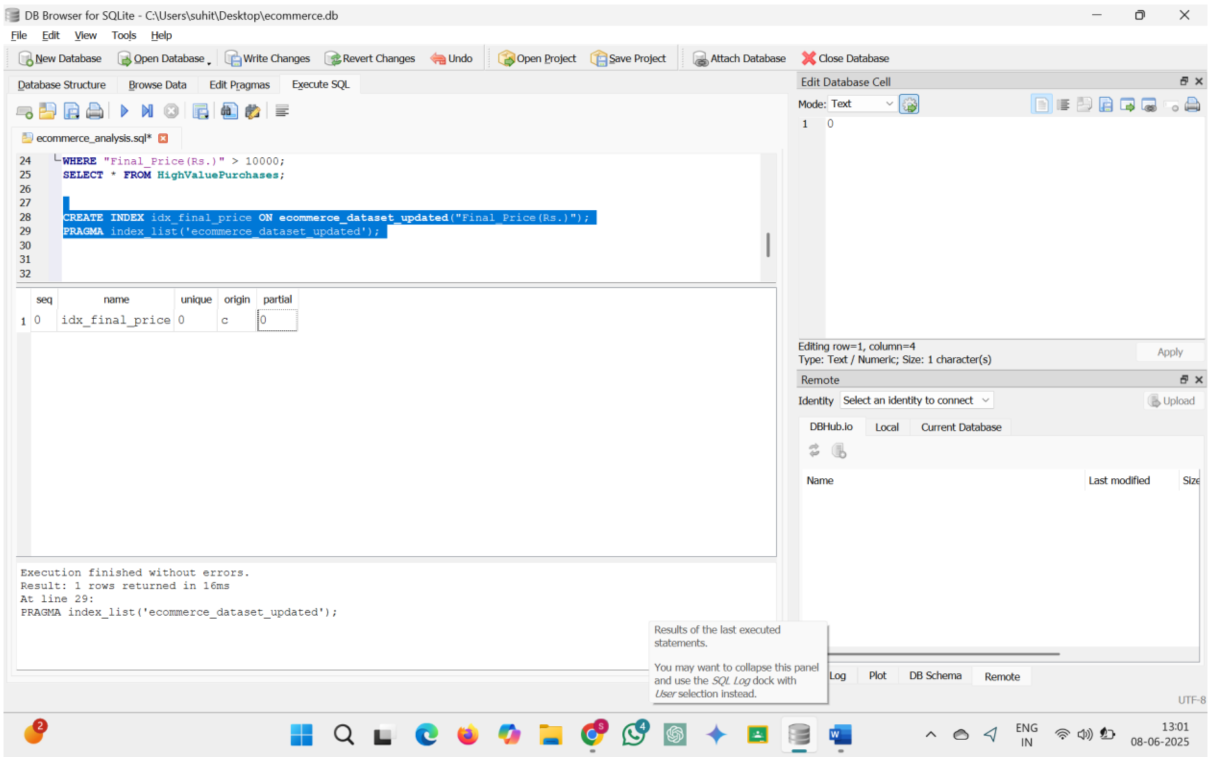
User_ID	Category	Final_Price(Rs.)	Payment_Method

```
Execution finished without errors.
Result: 0 rows returned in 9ms
At line 25:
SELECT * FROM HighValuePurchases;
```

Query 6 :

CREATE INDEX idx_final_price ON ecommerce_dataset_updated("Final_Price(Rs.)");

PRAGMA index_list('ecommerce_dataset_updated');



```
WHERE "Final_Price(Rs.)" > 10000;
SELECT * FROM HighValuePurchases;


CREATE INDEX idx_final_price ON ecommerce_dataset_updated("Final_Price(Rs.)");
PRAGMA index_list('ecommerce_dataset_updated');
```

	seq	name	unique	origin	partial
1	0	idx_final_price	0	c	0

```
Execution finished without errors.
Result: 1 rows returned in 16ms
At line 29:
PRAGMA index_list('ecommerce_dataset_updated');
```