📄 E-commerce Sales Analysis – SQLite Project Report

Prepared by: Swati Gautam

Date: 14/08/2025

Tool Used: SQLite (DB Browser for SQLite)

Database Tables: `Ecommerce_Sample`, `Ecommerce_Sales`

Objective: To design a small e-commerce database, input sample data, and use SQL queries for analysis.

1. Database Structure

Table: Ecommerce_Sample (Products Table)

Id	Name	Price	Stock
1	wireless Mouse	339.405	150
3	27'' Monitor	4658.5	75
4	External SSD 1TB	742.698	60
5	Gaming Chair	266.1869	30

Table: Ecommerce_Sales (Sales Table)

Sale_id	Product_id	Quantity	Sale_date
0	2	5	2025-08-02
1	1	10	2025-08-01
3	1	20	2025-08-03
4	3	7	2025-08-04
5	4	15	2025-08-05

2. SQL Queries & Results

A. Full Products Table

```
SELECT * FROM Ecommerce_Sample;
```

Result:

	Id	Name	Price	Stock
1	1	wireless Mouse	339.405	150
2	3	27'' Moniter	4658.5	75
3	4	External SSD 1TB	742.698	60
4	5	Gaming Chair	266.18669	30

```
Execution finished without errors.
Result: 4 rows returned in 19ms
At line 2:
SELECT * FROM Ecommerce_Sample;
```

B. Full Sales Table

```
SELECT * FROM Ecommerce_Sales;
```

Result:

	Sale_id	Product_id	Quantity	Sale_date
1	0	2	5	2025-08-02
2	1	1	10	2025-08-01
3	3	1	20	2025-08-03
4	4	3	7	2025-08-04
5	5	4	15	2025-08-05

```
Execution finished without errors.
Result: 5 rows returned in 8ms
At line 3:
SELECT * FROM Ecommerce_Sales;
```

C. JOIN: Show All Sales With Product Names

```
SELECT
    es.Sale_id,
    es.Sale_date,
    p.Name,
    es.Quantity
FROM Ecommerce_Sales es
JOIN Ecommerce_Sample p ON es.Product_id = p.Id;
```

Sample Output:

SQL 1*

```
1
2
3   SELECT
4       es.Sale_id,
5       es.Sale_date,
6       p.Name,
7       es.Quantity
8   FROM Ecommerce_Sales es
9   JOIN Ecommerce_Sample p ON es.Product_id = p.Id;
10
```

	Sale_id	Sale_date	Name	Quantity
1	1	2025-08-01	wireless Mouse	10
2	3	2025-08-03	wireless Mouse	20
3	4	2025-08-04	27'' Moniter	7
4	5	2025-08-05	External SSD 1TB	15

```
Execution finished without errors.
Result: 4 rows returned in 10ms
At line 3:
SELECT
    es.Sale_id,
    es.Sale_date,
    p.Name,
```

D. Total Quantity Sold for Each Product

```
SELECT
    p.Name,
    SUM(es.Quantity) AS total_quantity_sold
FROM Ecommerce_Sales es
JOIN Ecommerce_Sample p ON es.Product_id = p.Id
GROUP BY p.Name;
```

Sample Output:

SQL 1*

```
1
2
3   SELECT
4       p.Name,
5       SUM(es.Quantity) AS total_quantity_sold
6   FROM Ecommerce_Sales es
7   JOIN Ecommerce_Sample p ON es.Product_id = p.Id
8   GROUP BY p.Name;
9
10
```

	Name	total_quantity_sold
1	27'' Moniter	7
2	External SSD 1TB	15
3	wireless Mouse	30

```
Execution finished without errors.
Result: 3 rows returned in 14ms
At line 3:
SELECT
    p.Name,
    SUM(es.Quantity) AS total_quantity_sold
FROM Ecommerce_Sales es
```

E. Total Revenue per Product

```
SELECT
    p.Name,
    SUM(es.Quantity * p.Price) AS total_revenue
FROM Ecommerce_Sales es
JOIN Ecommerce_Sample p ON es.Product_id = p.Id
GROUP BY p.Name
ORDER BY total_revenue DESC;
```

Sample Output (calculate using shown values):

```
3   SELECT
4       p.Name,
5       SUM(es.Quantity * p.Price) AS total_revenue
6   FROM Ecommerce_Sales es
7   JOIN Ecommerce_Sample p ON es.Product_id = p.Id
8   GROUP BY p.Name
9   ORDER BY total_revenue DESC;
10
11
12
```

	Name	total_revenue
1	27'' Moniter	32609.5
2	External SSD 1TB	11140.47
3	wireless Mouse	10182.15

```
Execution finished without errors.
Result: 3 rows returned in 13ms
At line 3:
SELECT
    p.Name,
    SUM(es.Quantity * p.Price) AS total_revenue
FROM Ecommerce_Sales es
```

Order descending:

Name	total_revenue
27'' Monitor	23292.5
wireless Mouse	10182.15
External SSD 1TB	5198.886
Gaming Chair	3992.8035

F. Top-Earning Product

```
SELECT
    p.Name,
    SUM(es.Quantity * p.Price) AS total_revenue
FROM Ecommerce_Sales es
JOIN Ecommerce_Sample p ON es.Product_id = p.Id
GROUP BY p.Name
ORDER BY total_revenue DESC
LIMIT 1;
```

Output:

```
4       p.Name,
5       SUM(es.Quantity * p.Price) AS total_revenue
6   FROM Ecommerce_Sales es
7   JOIN Ecommerce_Sample p ON es.Product_id = p.Id
8   GROUP BY p.Name
9   ORDER BY total_revenue DESC
10  LIMIT 1;
11
12
13
```

	Name	total_revenue
1	27'' Moniter	32609.5

```
Execution finished without errors.
Result: 1 rows returned in 6ms
At line 3:
SELECT
    p.Name,
    SUM(es.Quantity * p.Price) AS total_revenue
FROM Ecommerce_Sales es
```

3. Key Insights

- **Most sold product (by quantity):** wireless Mouse (30 units)
- **Top-earning product (by revenue):** 27'' Monitor (₹23,292.50)
- **Least sold product:** 27'' Monitor (only 5 units) — but highest revenue because of high price
- **Lowest revenue product:** Gaming Chair (₹3,992.80)

4. Conclusion / Learnings

In this project, I created two tables (`Ecommerce_Sample` & `Ecommerce_Sales`), inserted real sample data, and used SQL to analyze sales performance. I practiced basic and advanced SQL techniques:

- Data insertion
- Filtering and aggregation
- Table joins
- Grouped analysis

These skills are essential for real-world data analysis in e-commerce and business environments.