

SQL Analysis – Retail Sales Project

Tool Used: DB Browser for SQLite

Dataset Used: Cleaned Retail Sales Data (from Excel)

1. Query

```
SELECT SUM(Sales) AS Total_Sales FROM sales;
```

👉 This query gives the total revenue from all sales.

The screenshot shows the DB Browser for SQLite interface. The toolbar at the top includes 'Database Structure', 'Browse Data', 'Edit Pragmas', and 'Execute SQL'. Below the toolbar is a toolbar with icons for file operations. A central window titled 'SQL 1*' contains the following SQL code:

```
5   --PRAGMA table_info(sales);
6
7   --ALTER TABLE sales ADD COLUMN Cleaned_Sales REAL;
8
9   UPDATE sales
10  SET Cleaned_Sales = REPLACE(sales, ',', '') * 1.0;
11
12  --SELECT SUM(Cleaned_Sales) FROM sales;
13
14  SELECT SUM(Cleaned_Sales) AS Total_Sales FROM sales;
```

Below the code, the results are displayed in a table:

Total_Sales
1 522504.72

2. Query

```
SELECT SUM(Profit) AS Total_Profit FROM sales;
```

👉 This shows total profit generated.

The screenshot shows the DB Browser for SQLite interface. The toolbar at the top includes 'Database Structure', 'Browse Data', 'Edit Pragmas', and 'Execute SQL'. Below the toolbar is a toolbar with icons for file operations. A central window titled 'SQL 1*' contains the following SQL code:

```
11
12  --SELECT SUM(Cleaned_Sales) FROM sales;
13
14  --SELECT SUM(Cleaned_Sales) AS Total_Sales FROM sales;
15
16  SELECT [Product Name], SUM(Cleaned_Sales) AS Total_Sales
17  FROM sales
18  GROUP BY [Product Name]
19  ORDER BY Total_Sales DESC
20  LIMIT 5;
```

Below the code, the results are displayed in a table:

Product Name	Total_Sales
1 Hon Deluxe Fabric Upholstered ...	3345.53
2 Fellowes Staxonsteel Drawer Files	3283.89
3 Hewlett Packard 610 Color Digital ...	2800.94
4 Bevis Oval Conference Table, Walnut	2714.19
5 GE 30524EE4	2705.67

3. Query

```
SELECT State, SUM(Sales) AS Total_Sales
```

```
FROM sales
```

```
GROUP BY State
```

```
ORDER BY Total_Sales DESC
```

```
LIMIT 5;
```

👉 Shows which 5 states have the highest total sales.

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```

16   --SELECT [Product Name], SUM(Cleaned_Sales) AS Total_Sales
17   --FROM sales
18   --GROUP BY [Product Name]
19   --ORDER BY Total_Sales DESC
20   --LIMIT 5;
21
22   SELECT Region, SUM(Cleaned_Sales) AS Regional_Sales
23   FROM sales
24   GROUP BY Region
25   ORDER BY Regional_Sales DESC;

```

Region	Regional_Sales
1 West	163944.23
2 East	152207.0
3 Central	126336.89
4 South	80016.6

4. Query

```

SELECT Category, SUM(Profit) AS Total_Profit
FROM sales
GROUP BY Category
ORDER BY Total_Profit DESC;

```

👉 Tells which category earned the most profit.

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```

21
22   --SELECT Region, SUM(Cleaned_Sales) AS Regional_Sales
23   --FROM sales
24   --GROUP BY Region
25   --ORDER BY Regional_Sales DESC;
26
27   SELECT Category, SUM(Cleaned_Sales) AS Category_Sales
28   FROM sales
29   GROUP BY Category
30   ORDER BY Category_Sales DESC;

```

Category	Category_Sales
1 Office Supplies	182317.7
2 Furniture	178090.55
3 Technology	162096.47

5. Query

```

SELECT substr([Order Date], 7, 4) AS Year, SUM(Sales) AS Total_Sales
FROM sales
GROUP BY Year
ORDER BY Year;

```

👉 Shows yearly sales trend (growth or drop).

Database Structure Browse Data Edit Pragmas Execute SQL

SQL 1*

```

31
32   SELECT substr([Order Date], 4, 7) AS Month_Year,
33   SUM(Cleaned_Sales) AS Monthly_Sales
34   FROM sales
35   GROUP BY Month_Year
36   ORDER BY Month_Year;
37
38
39
40

```

Month_Year	Monthly_Sales
1 01 2014	11349.51
2 01-2015	6395.01
3 01 2016	10026.78
4 01-2017	14107.44
5 02-2014	8991.7
6 02 2015	8174.14
7 02-2016	10078.78

Execution finished without errors.
Process: 45 queries, 0 rows returned in 1ms.
At line 32:
SELECT substr([Order Date], 4, 7) AS Month_Year,
 sum(cleaned_sales) AS Monthly_Sales
FROM sales
GROUP BY Month_Year
ORDER BY Month_Year;