

Task 6 Report: Monthly Sales Trend Analysis

Objective

The goal of this task was to analyze sales trends by month, focusing on two key metrics:

- **Monthly Revenue:** Total income generated each month
- **Monthly Order Volume:** Number of transactions placed each month

Dataset Description

The dataset used for this analysis contained online sales records with the following key columns:

- **Transaction ID**
- **Date**
- **Product Category**
- **Product Name**
- **Units Sold**
- **Unit Price**
- **Total Revenue**
- **Region**
- **Payment Method**

Data was extracted from a CSV file and inserted into a structured SQL table named `online_sales`.

Tools Used

- **MySQL:** For database management and SQL execution
- **MySQL Workbench** (or similar GUI): To run queries and view results

Methodology

1. **Data Preparation:** A SQL table was created to match the structure of the CSV file, and the first few rows of data were manually inserted to test the process.
2. **Monthly Aggregation:**
 - o The Date column was used to extract the **year** and **month** of each transaction.
 - o Transactions were grouped by **year and month**.
 - o The **total number of orders** (count of transactions) and **total revenue** (sum of revenue) were calculated for each month.
3. **Result Organization:**
 - o The final results were sorted in chronological order by year and month to clearly visualize the trends.

Outcome

The output of the analysis was a summarized table showing:

- **Year**
- **Month**
- **Total Orders**
- **Total Revenue**

This revealed how the business performed on a monthly basis in terms of both volume and revenue. In the sample dataset provided, all transactions were in January 2024, resulting in a single-row summary.

Key Learnings

- How to group sales data by month using SQL.
- How to apply aggregation functions like **SUM()** and **COUNT()**.
- How to use SQL to derive business insights from raw sales data.

OUTPUT :

```
1 •  CREATE DATABASE sales_analysis_Data;
2 •  USE sales_analysis_Data;
3 •  CREATE TABLE online_sales_Data (
4      Transaction_ID INT PRIMARY KEY,
5      Date DATE,
6      Product_Category VARCHAR(100),
7      Product_Name VARCHAR(100),
8      Units_Sold INT,
9      Unit_Price DECIMAL(10,2),
10     Total_Revenue DECIMAL(10,2),
11     Region VARCHAR(100),
12     Payment_Method VARCHAR(100)
13 );
14 • INSERT INTO online_sales_Data
15     (Transaction_ID, Date, Product_Category, Product_Name, Units_Sold, Unit_Price, Total_Revenue, Region, Payment_Method)
16     VALUES
17     (10001, '2024-01-01', 'Electronics', 'iPhone 14 Pro', 2, 999.99, 1999.98, 'North America', 'Credit Card'),
18     (10002, '2024-01-02', 'Home Appliances', 'Dyson V11 Vacuum', 1, 499.99, 499.99, 'Europe', 'PayPal'),
19     (10003, '2024-01-03', 'Fashion', 'Nike Air Max 270', 3, 150.00, 450.00, 'Asia', 'Debit Card'),
20     (10004, '2024-01-04', 'Books', 'Atomic Habits', 5, 20.00, 100.00, 'North America', 'Credit Card'),
21     (10005, '2024-01-05', 'Gaming', 'PlayStation 5', 1, 499.99, 499.99, 'Europe', 'PayPal');
--
```

```
22 •   SELECT
23       EXTRACT(YEAR FROM Date) AS Year,
24       EXTRACT(MONTH FROM Date) AS Month,
25       COUNT(*) AS Total_Orders,
26       SUM(Total_Revenue) AS Monthly_Revenue
27   FROM
28       online_sales_Data
29   GROUP BY
30       Year, Month
31   ORDER BY
32       Year, Month;
33 -- SELECT
34 --     Product_Category,
35 --     EXTRACT(YEAR FROM Date) AS Year,
```

Result Grid				
	Year	Month	Total_Orders	Monthly_Revenue
▶	2024	1	5	3549.96

```
33 •   SELECT
34     Product_Category,
35     EXTRACT(YEAR FROM Date) AS Year,
36     EXTRACT(MONTH FROM Date) AS Month,
37     COUNT(*) AS Total_Orders,
38     SUM(Total_Revenue) AS Monthly_Revenue
39   FROM
40     online_sales_Data
41   GROUP BY
42     Product_Category, Year, Month
43   ORDER BY
44     Product_Category, Year, Month;
45 -- SELECT
```

Result Grid					
	Product_Category	Year	Month	Total_Orders	Monthly_Revenue
▶	Books	2024	1	1	100.00
	Electronics	2024	1	1	1999.98
	Fashion	2024	1	1	450.00
	Gaming	2024	1	1	499.99
	Home Appliances	2024	1	1	499.99

```
45 •   SELECT
46      EXTRACT(YEAR FROM Date) AS Year,
47      EXTRACT(MONTH FROM Date) AS Month,
48      SUM(Units_Sold) AS Total_Units_Sold
49  FROM
50      online_sales_Data
51  GROUP BY
52      Year, Month
53  ORDER BY
54      Year, Month;
55  -- SELECT
```

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

	Year	Month	Total_Units_Sold
▶	2024	1	12

```
55 •   SELECT
56      EXTRACT(YEAR FROM Date) AS Year,
57      EXTRACT(MONTH FROM Date) AS Month,
58      ROUND(AVG(Total_Revenue), 2) AS Average_Order_Value
59  FROM
60      online_sales_Data
61  GROUP BY
62      Year, Month
63  ORDER BY
64      Year, Month;
65  -- SELECT
```

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

	Year	Month	Average_Order_Value
▶	2024	1	709.99

```
65 •  SELECT
66      Product_Name,
67      SUM(Total_Revenue) AS Total_Revenue
68  FROM
69      online_sales_Data
70  GROUP BY
71      Product_Name
72  ORDER BY
73      Total_Revenue DESC
74  LIMIT 5;
75
```

Result Grid | Filter Rows: _____ | Export: Wrap Cell Content: Fetch rows:

Product_Name	Total_Revenue
iPhone 14 Pro	1999.98
Dyson V11 Vacuum	499.99
PlayStation 5	499.99
Nike Air Max 270	450.00
Atomic Habits	100.00