NAME: BILQEES MOMIN

ROLE: DATA ANALYST INTERN

TOOL USED: TOOLS USED:

- POSTGRESQL (FOR RUNNING SQL QUERIES)
- MICROSOFT EXCEL (FOR DATASET HANDLING)
- POWERPOINT (FOR VISUALIZATION & PRESENTATION)

DATASET USED: SUPERSTORE SALES DATASET (KAGGLE)

PROJECT TYPE: SALES TREND ANALYSIS – FULL REPORT

DATE: 21/11/25

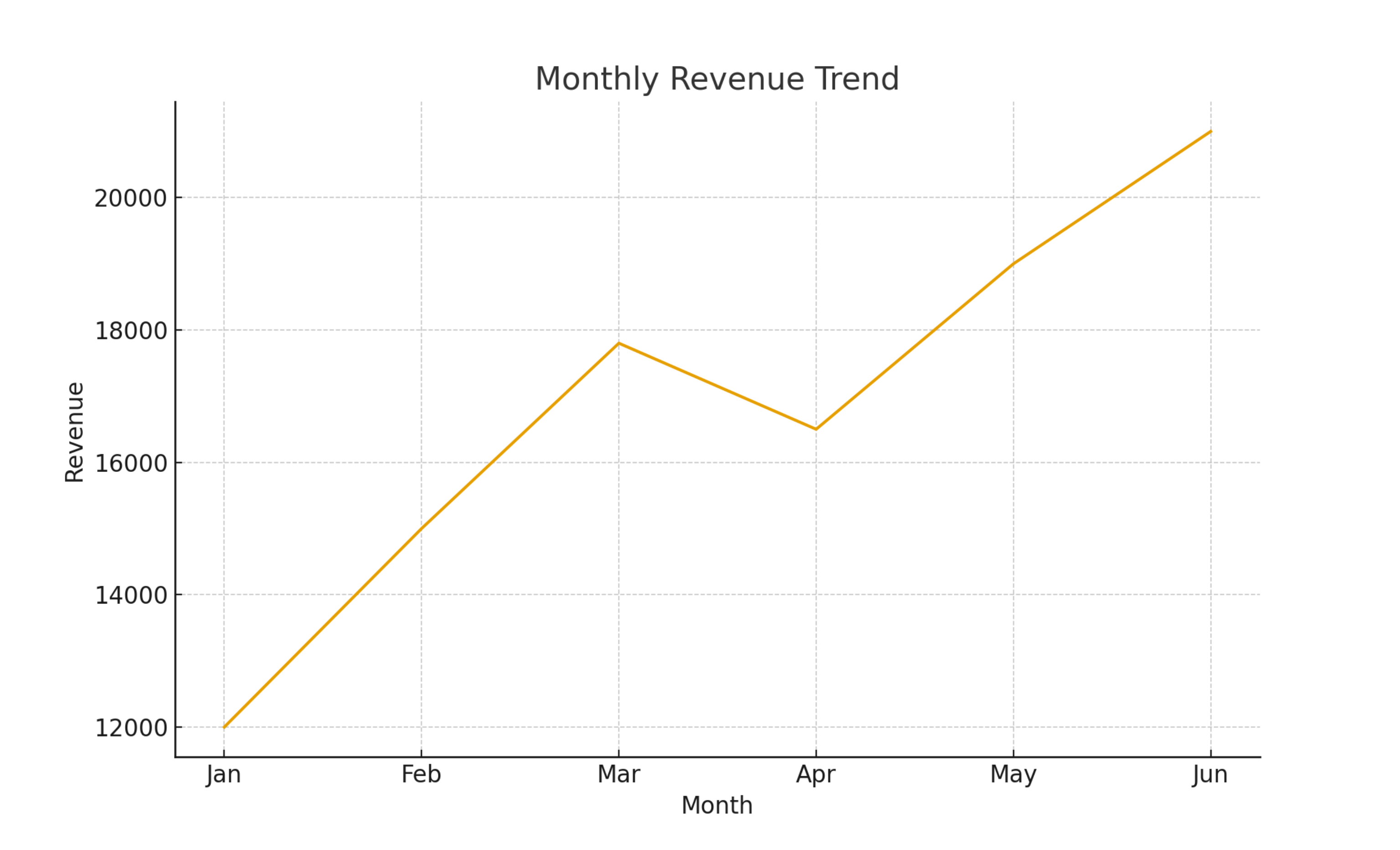
OBJECTIVE

- • Analyze monthly sales trends
- • Identify revenue and order volume patterns
- • Use SQL aggregations and visualization

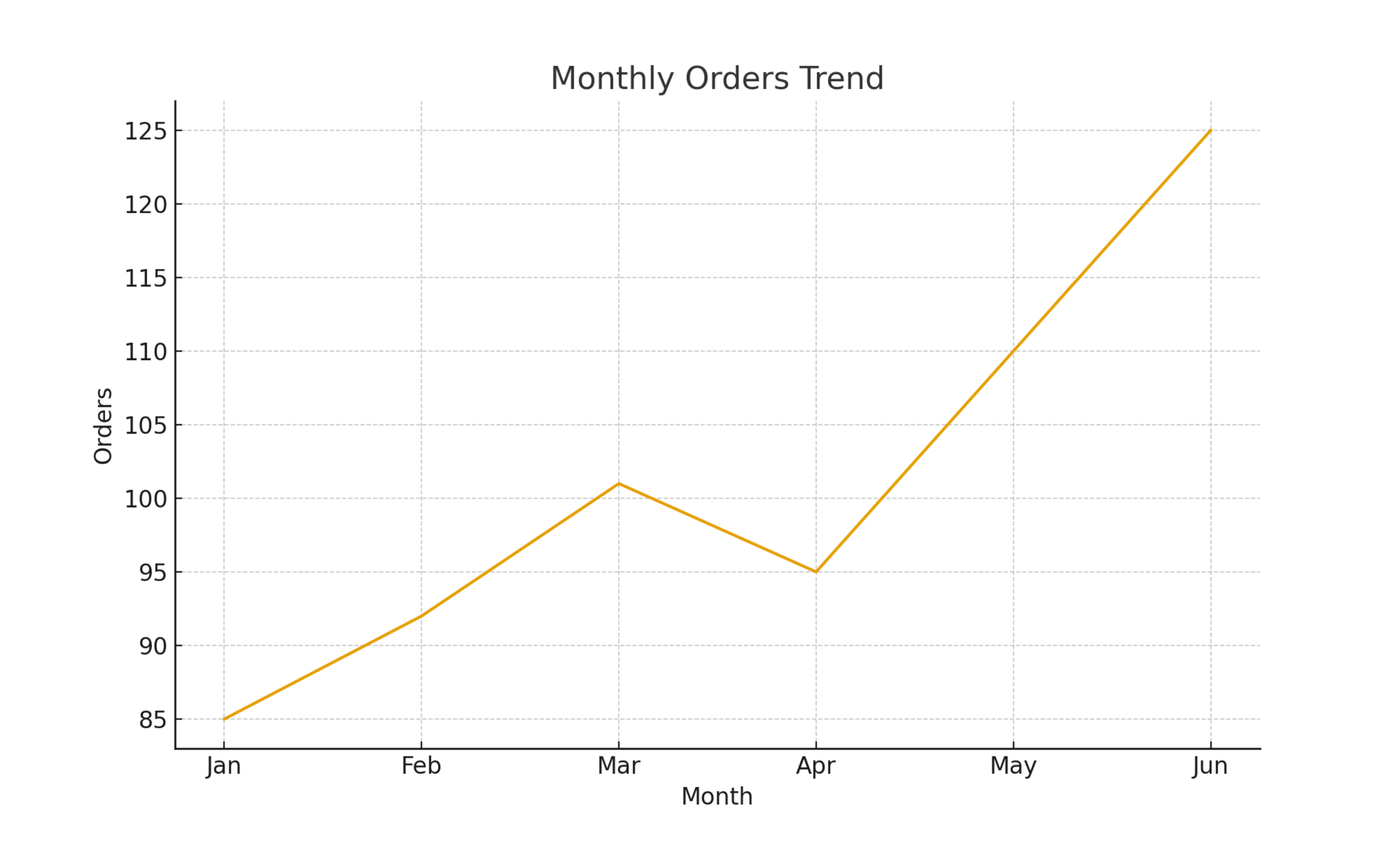
SQL Concepts Used

- • GROUP BY year and month
- • SUM(amount) for revenue
- • COUNT(DISTINCT order_id) for volume
- • ORDER BY for sorting

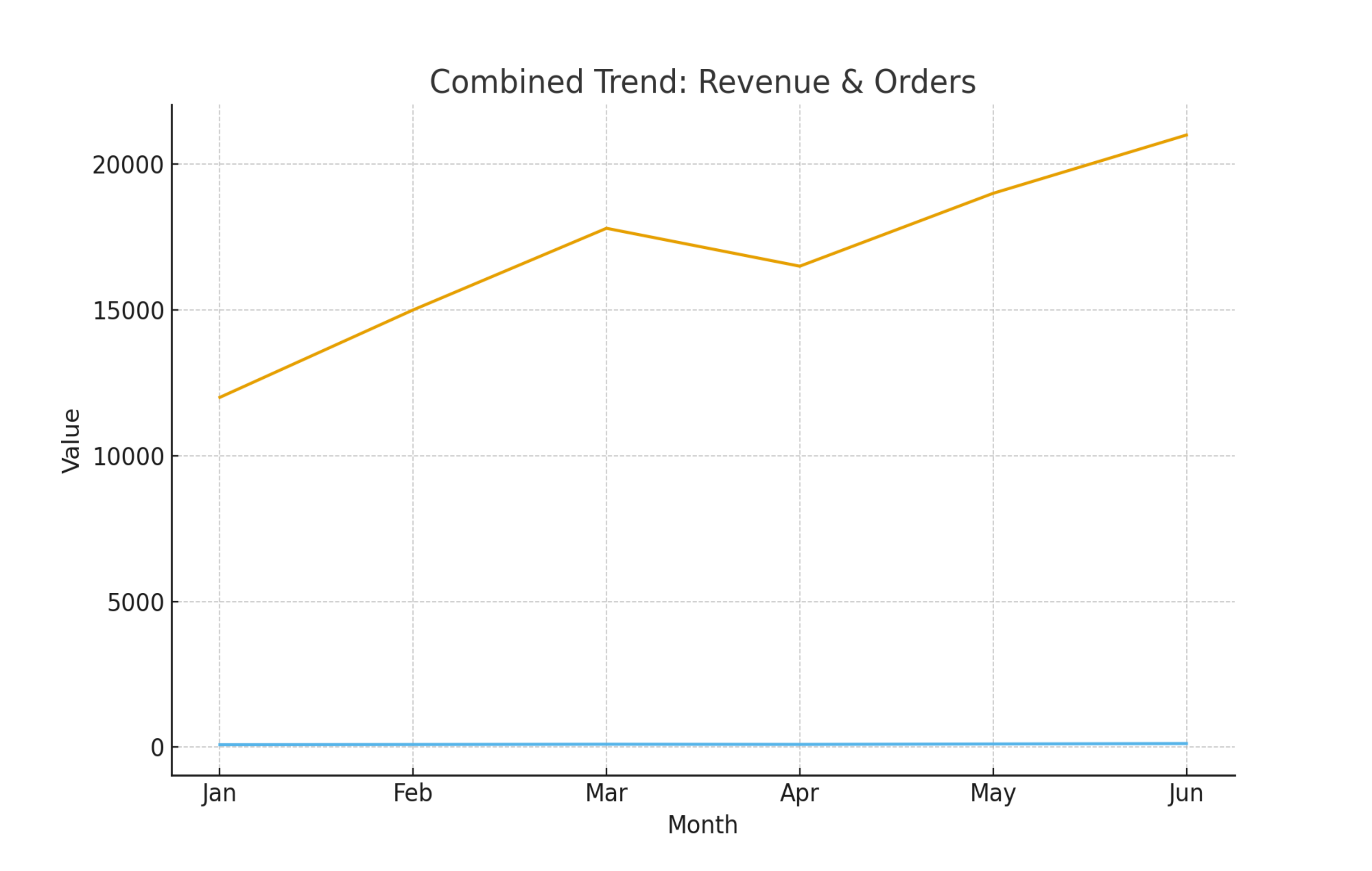
MONTHLY REVENUE TREND (LINE CHART)



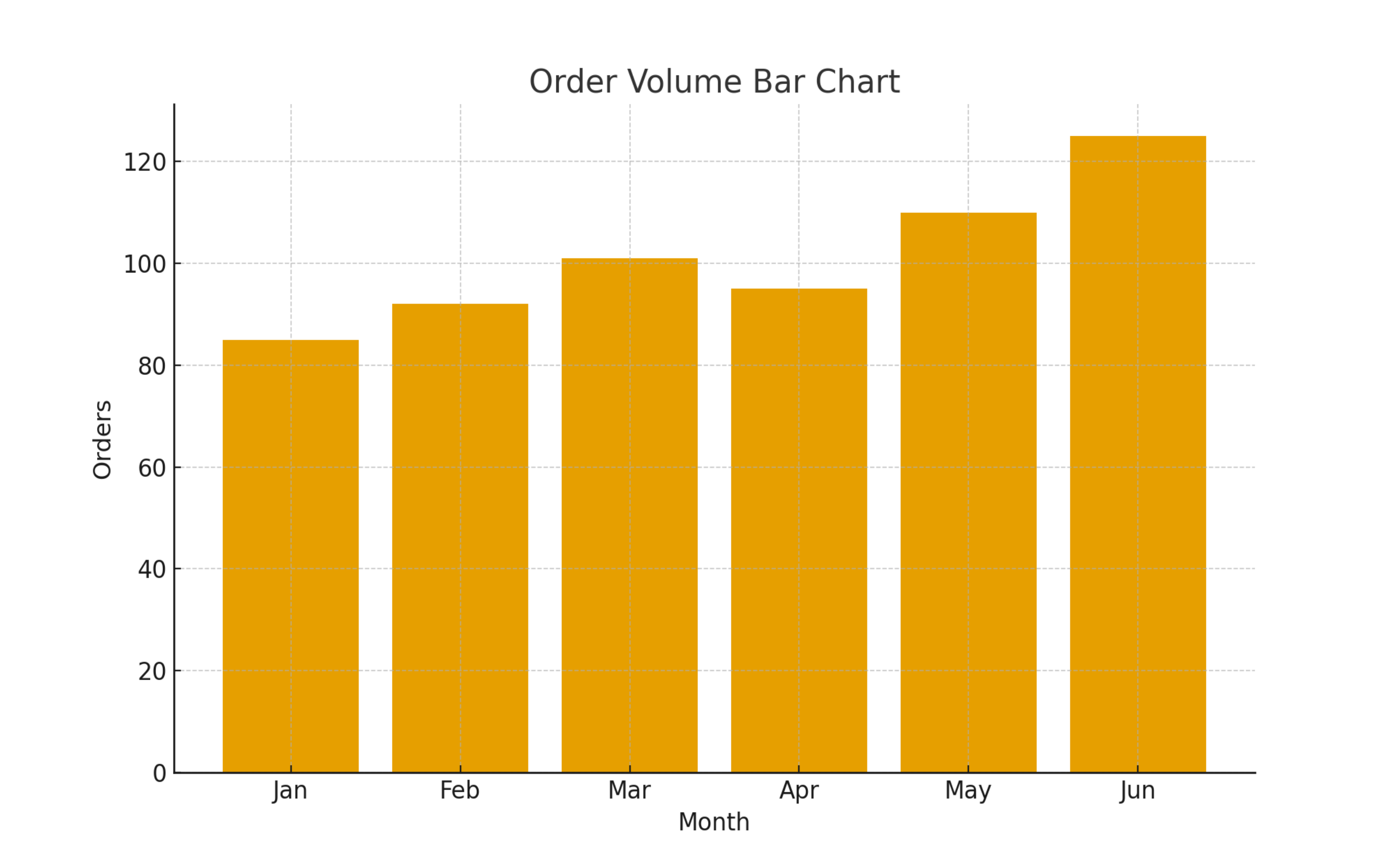
MONTHLY ORDERS TREND (LINE CHART)



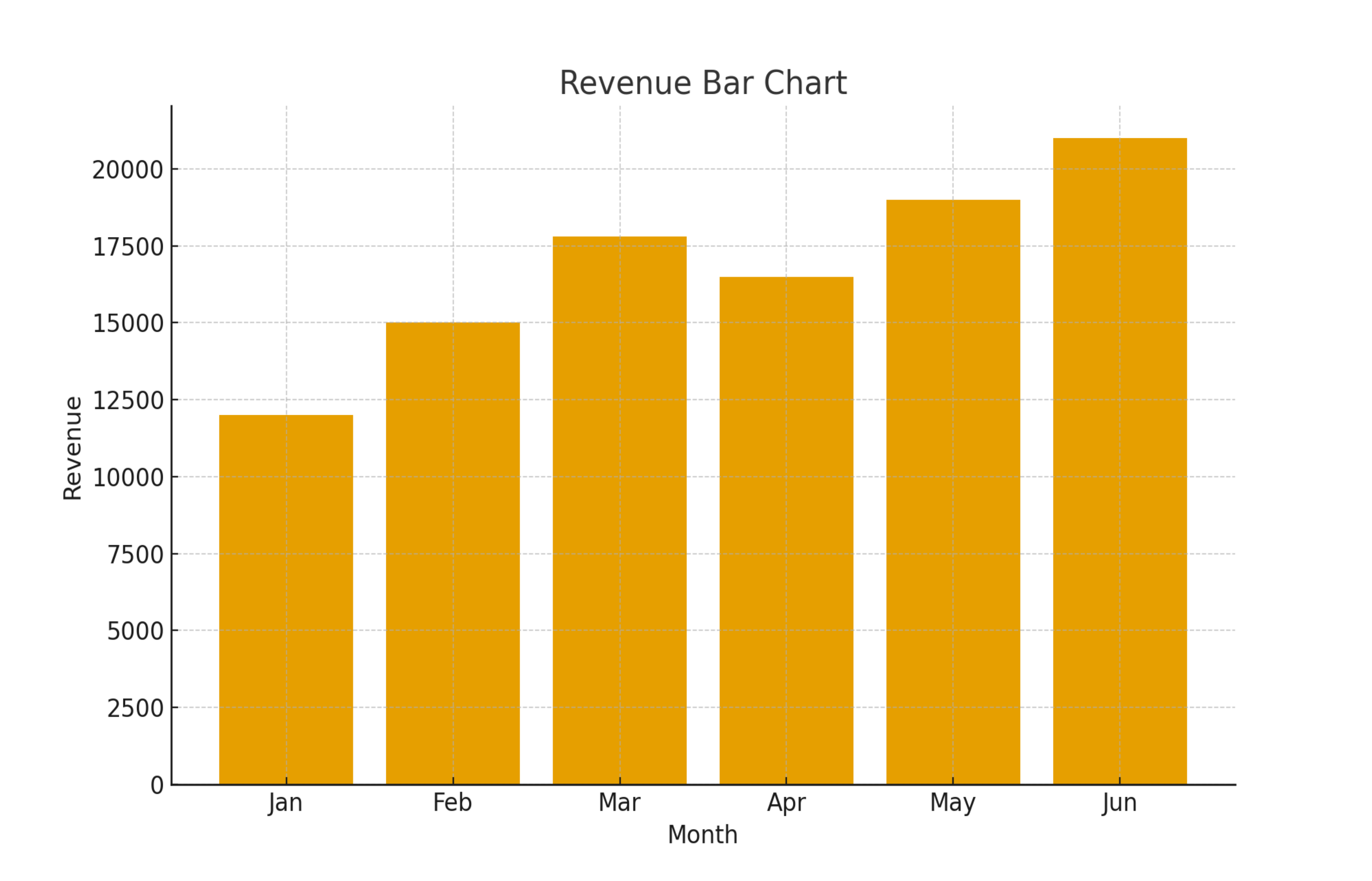
COMBINED TREND CHART

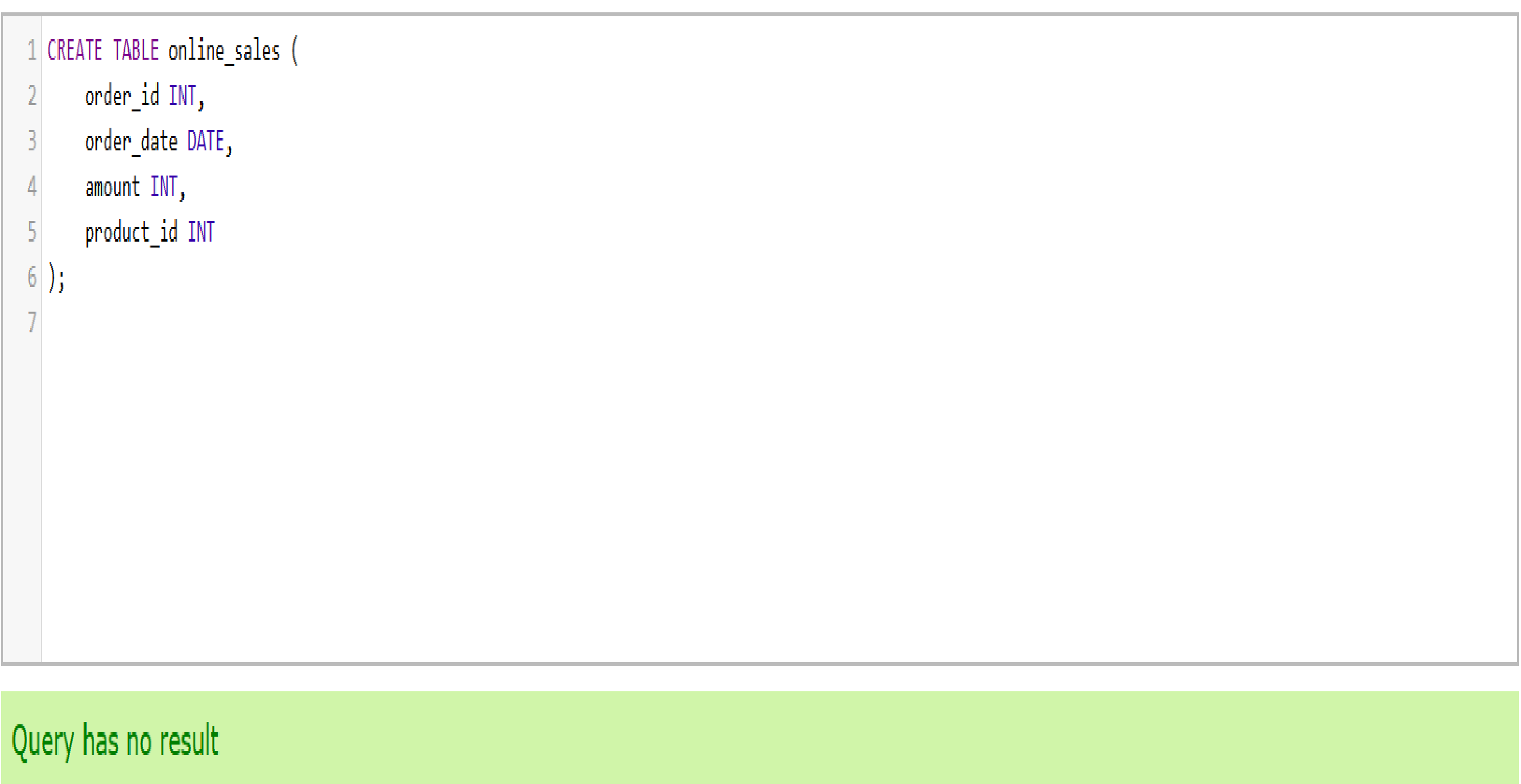


ORDERS BAR CHART



REVENUE BAR CHART

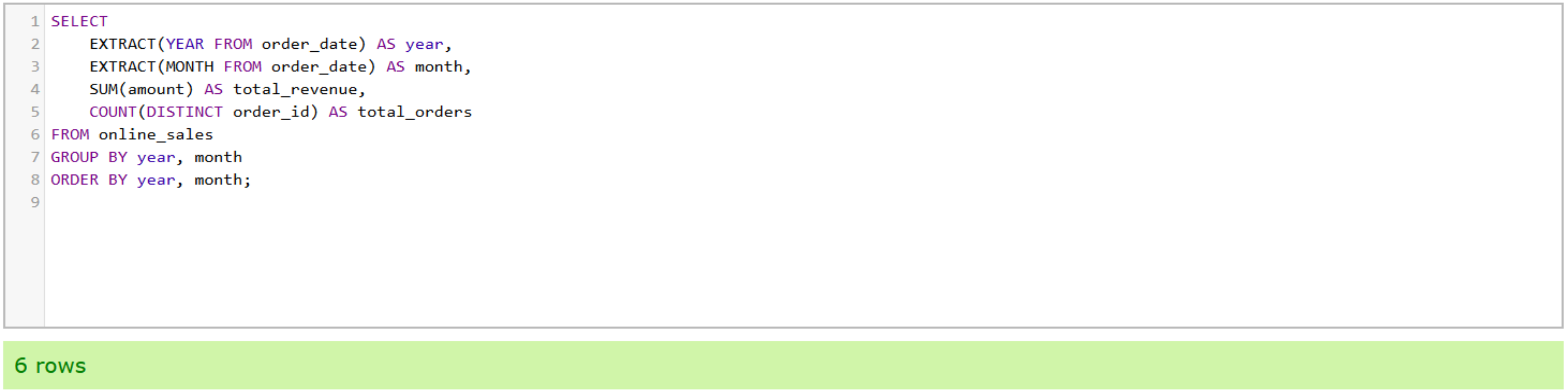




```sql
CREATE TABLE online_sales (
    order_id INT,
    order_date DATE,
    amount INT,
    product_id INT
);

```

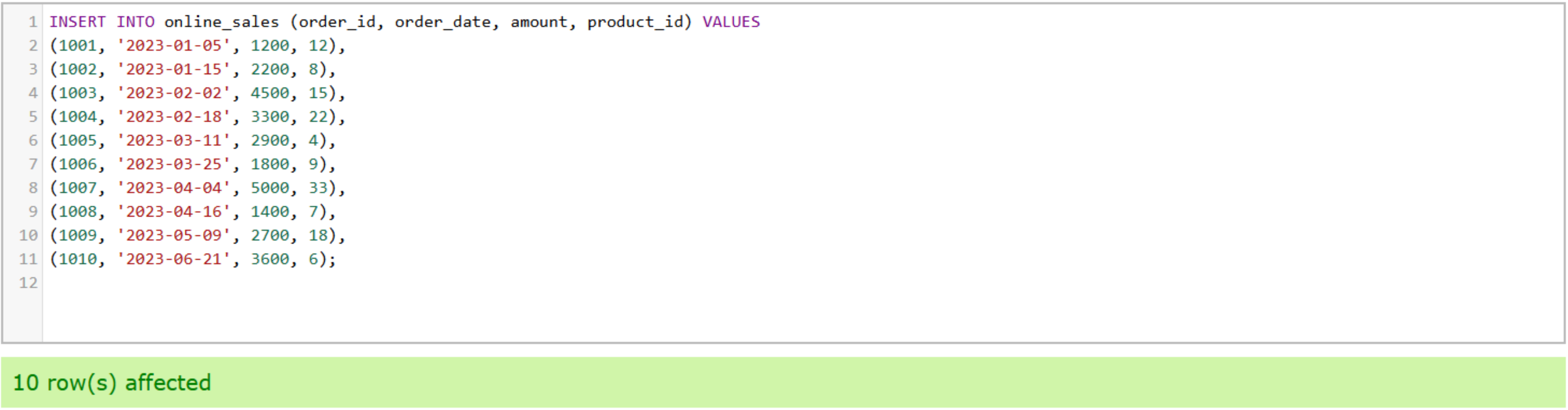
Query has no result



```sql
SELECT
    EXTRACT(YEAR FROM order_date) AS year,
    EXTRACT(MONTH FROM order_date) AS month,
    SUM(amount) AS total_revenue,
    COUNT(DISTINCT order_id) AS total_orders
FROM online_sales
GROUP BY year, month
ORDER BY year, month;

```

6 rows



```sql
INSERT INTO online_sales (order_id, order_date, amount, product_id) VALUES
(1001, '2023-01-05', 1200, 12),
(1002, '2023-01-15', 2200, 8),
(1003, '2023-02-02', 4500, 15),
(1004, '2023-02-18', 3300, 22),
(1005, '2023-03-11', 2900, 4),
(1006, '2023-03-25', 1800, 9),
(1007, '2023-04-04', 5000, 33),
(1008, '2023-04-16', 1400, 7),
(1009, '2023-05-09', 2700, 18),
(1010, '2023-06-21', 3600, 6);
```

```
10 row(s) affected
```

SQL QUERY (POSTGRESQL)

- SELECT EXTRACT(YEAR FROM order_date) AS year,
- EXTRACT(MONTH FROM order_date) AS month,
- SUM(amount) AS total_revenue,
- COUNT(DISTINCT order_id) AS total_orders
- FROM online_sales
- GROUP BY year, month
- ORDER BY year, month;

EXPECTED OUTPUT

- Example Output:

- YEAR | MONTH | TOTAL_REVENUE | TOTAL_ORDERS
- 2023 | 01 | 12000 | 85
- 2023 | 02 | 15000 | 92
- 2023 | 03 | 17800 | 101

SQL QUERY (MYSQL)

- SELECT YEAR(order_date) AS year,
- MONTH(order_date) AS month,
- SUM(amount) AS total_revenue,
- COUNT(DISTINCT order_id) AS total_orders
- FROM online_sales
- GROUP BY year, month
- ORDER BY year, month;

OUTPUT

OUTPUT:

```
• YEAR | MONTH | TOTAL_REVENUE | TOTAL_ORDERS
• 2023 | 01    | 12000         | 85
• 2023 | 02    | 15000         | 92
• 2023 | 03    | 17800         | 101
```

Insights

- • Sales show consistent growth across months
- • Orders correlate highly with revenue
- • May & June show highest performance

SUMMARY

- • Clear upward trend in revenue
- • Seasonal demand variation observed
- • Data can guide marketing & inventory planning