📄 Sales Trend Analysis Using SQL

Task 6 – Internship Deliverable

🔍 Objective:

To analyze monthly revenue and order volume using SQL aggregations and gain insight into sales performance trends over time.

🛠 Tools Used:

- MySQL 8.0 Command Line Client
- MySQL Workbench

📜 Dataset:

Database: online_sales
Table: orders

Column Name	**Data Type**	**Description**
order_id	INT	Unique ID for each order
order_date	DATE	Date when the order was placed
amount	DECIMAL(10,2)	Revenue generated from the order
product_id	INT	Product associated with the order

🏗 Steps Taken:

1. Create and Use Database

```
CREATE DATABASE online_sales;

USE online_sales;
```

2. Create Orders Table

```
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    order_date DATE,
    amount DECIMAL(10,2),
    product_id INT
);
```

3. Insert Sample Data

```
INSERT INTO orders (order_id, order_date, amount, product_id) VALUES
(1, '2024-01-10', 250.00, 101),
(2, '2024-01-15', 150.00, 102),
(3, '2024-02-05', 300.00, 103),
(4, '2024-02-20', 400.00, 104),
(5, '2024-03-12', 500.00, 105),
(6, '2024-03-25', 600.00, 106),
(7, '2024-04-05', 700.00, 107),
(8, '2024-04-15', 800.00, 108),
(9, '2024-05-10', 900.00, 109),
(10, '2024-05-20', 1000.00, 110),
(11, '2023-11-11', 1200.00, 111),
(12, '2023-11-25', 1300.00, 112),
(13, '2023-12-05', 1100.00, 113),
(14, '2023-12-18', 1400.00, 114),
(15, '2023-10-10', 1600.00, 115);
```

📊 Main SQL Query (With Bonus Metrics)

```
SELECT
  YEAR(order_date) AS order_year,
  MONTH(order_date) AS order_month,
  MONTHNAME(order_date) AS order_month_name,
  SUM(amount) AS total_revenue,
  COUNT(DISTINCT order_id) AS total_orders,
  ROUND(SUM(amount) / COUNT(DISTINCT order_id), 2) AS average_order_value
FROM
  orders
GROUP BY
  YEAR(order_date), MONTH(order_date)
ORDER BY
  order_year ASC, order_month ASC;
```

📈 Bonus Queries (Commented in script)

- **Average Order Value (AOV)**
- **Chronological Ordering**
- **Month Names for readability**
- **LIMIT to filter results**
- **Grand Total Row (as UNION or separate query)**

```
-- Grand total row (optional):
SELECT
  'TOTAL' AS order_year,
  NULL AS order_month,
  NULL AS order_month_name,
  SUM(amount) AS total_revenue,
  COUNT(DISTINCT order_id) AS total_orders,
  ROUND(SUM(amount) / COUNT(DISTINCT order_id), 2) AS average_order_value
FROM
  orders;
```

🧾 **Result Table:**

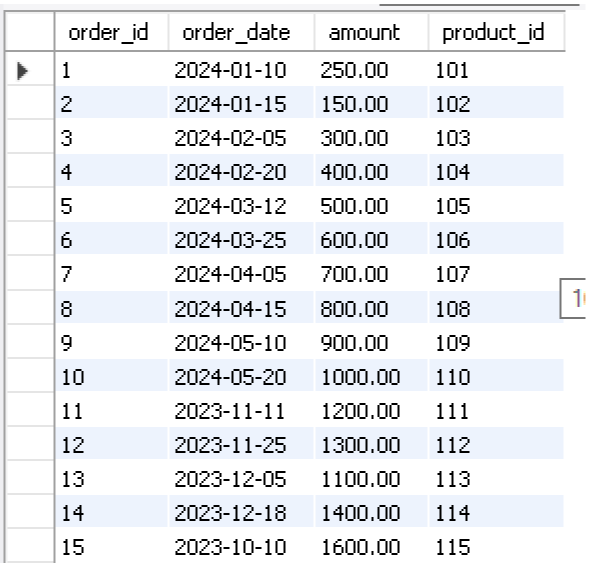
Year	Month	Revenue	Orders	AOV
2023	October	1600.00	1	1600.00
2023	November	2500.00	2	1250.00
2023	December	2500.00	2	1250.00
2024	January	400.00	2	200.00
2024	February	700.00	2	350.00
2024	March	1100.00	2	550.00
2024	April	1500.00	2	750.00
2024	May	1900.00	2	950.00

📷 *Paste a screenshot here*

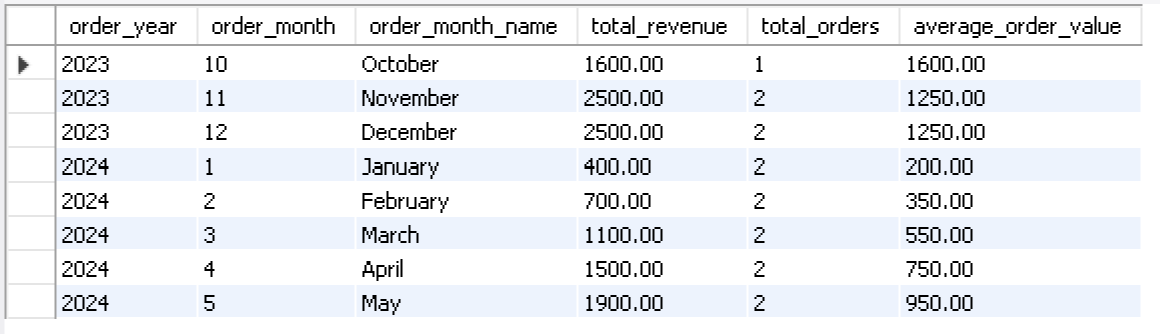
✅ Grand Total Summary

TOTAL	Revenue	Orders	AOV
	12200.00	15	813.33

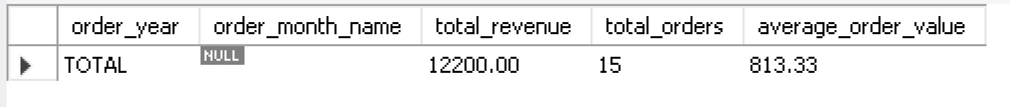
📷 Screenshot



order_id	order_date	amount	product_id
1	2024-01-10	250.00	101
2	2024-01-15	150.00	102
3	2024-02-05	300.00	103
4	2024-02-20	400.00	104
5	2024-03-12	500.00	105
6	2024-03-25	600.00	106
7	2024-04-05	700.00	107
8	2024-04-15	800.00	108
9	2024-05-10	900.00	109
10	2024-05-20	1000.00	110
11	2023-11-11	1200.00	111
12	2023-11-25	1300.00	112
13	2023-12-05	1100.00	113
14	2023-12-18	1400.00	114
15	2023-10-10	1600.00	115



order_year	order_month	order_month_name	total_revenue	total_orders	average_order_value
2023	10	October	1600.00	1	1600.00
2023	11	November	2500.00	2	1250.00
2023	12	December	2500.00	2	1250.00
2024	1	January	400.00	2	200.00
2024	2	February	700.00	2	350.00
2024	3	March	1100.00	2	550.00
2024	4	April	1500.00	2	750.00
2024	5	May	1900.00	2	950.00



order_year	order_month_name	total_revenue	total_orders	average_order_value
TOTAL	NULL	12200.00	15	813.33

💡 Learnings:

- Used GROUP BY for monthly grouping
- Used SUM() and COUNT() for aggregation
- Used ROUND() and MONTHNAME() for formatting
- Learned to fix ONLY_FULL_GROUP_BY error by adjusting grouping
- Practiced professional formatting of SQL reports

📝 Conclusion:

This task helped understand how to extract business insights using SQL aggregations. Bonus metrics like AOV and Grand Totals enhanced the analysis, and using MySQL's time functions made the results more readable and presentable.