# SQL Internship Project Report – Task 6: Sales Trend Analysis

Name: SACHIN KUMAR SAHU

**Date: July 1, 2025**

## 1. Introduction

This report documents the completion of Task 6 from the SQL internship project, focused on Sales Trend Analysis. The objective was to create a table named 'online\_sales', insert sample data representing order transactions, and analyze monthly sales trends using aggregate functions and filtering queries. All work was executed through the MySQL command-line interface (CMD).

## 2. Table Creation

SQL statement used to create the table:

CREATE TABLE online\_sales (  
 order\_date DATE,  
 amount DECIMAL(10,2),  
 product\_id INT  
);

## 3. Data Insertion

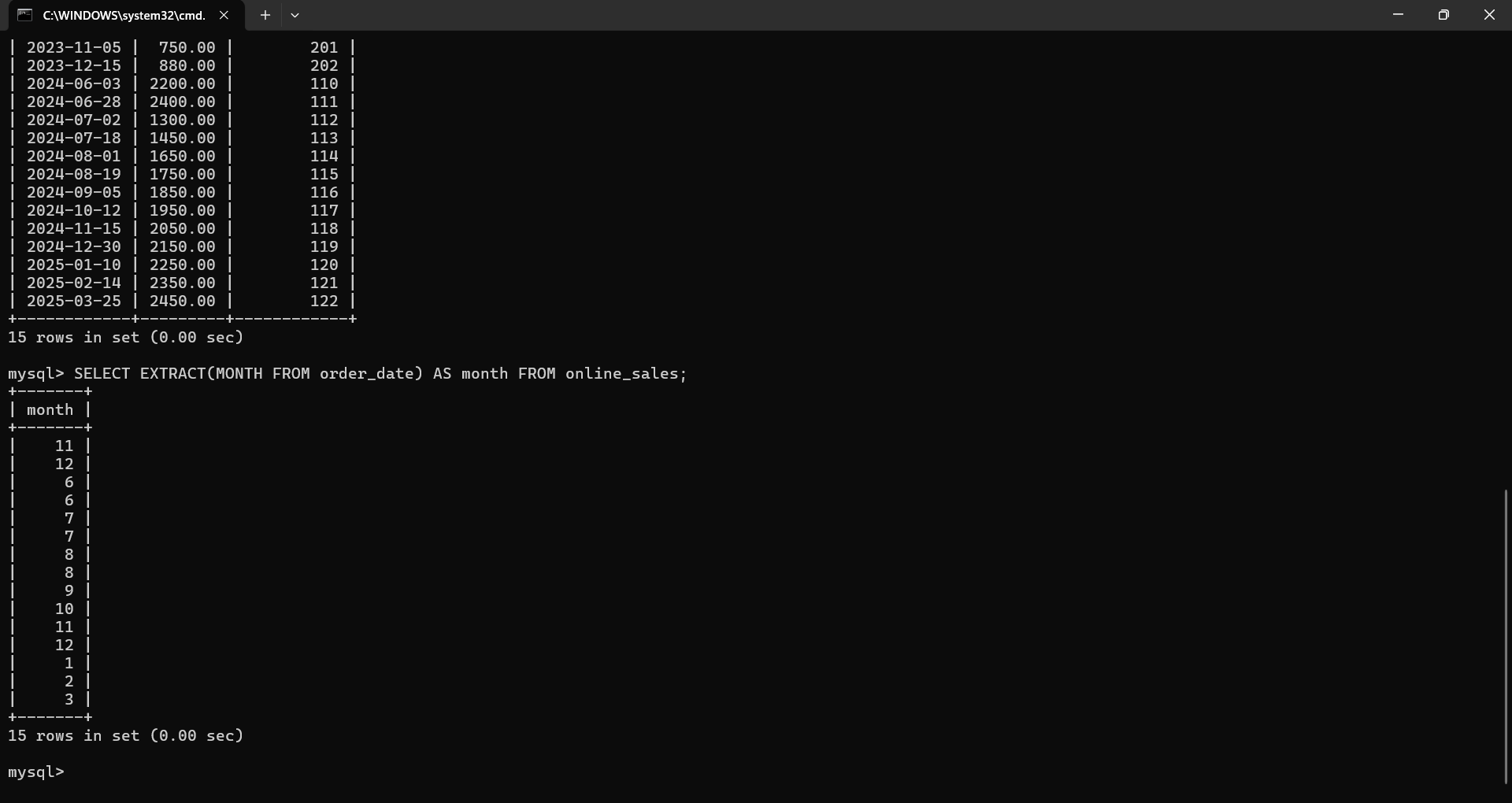
Sample data inserted into the 'online\_sales' table (25 rows spanning 2023 to 2025).

Paste Screenshot of successful inserts or SELECT \* queries:  
[Insert Screenshot Here]

## 4. Hint-Based SQL Queries

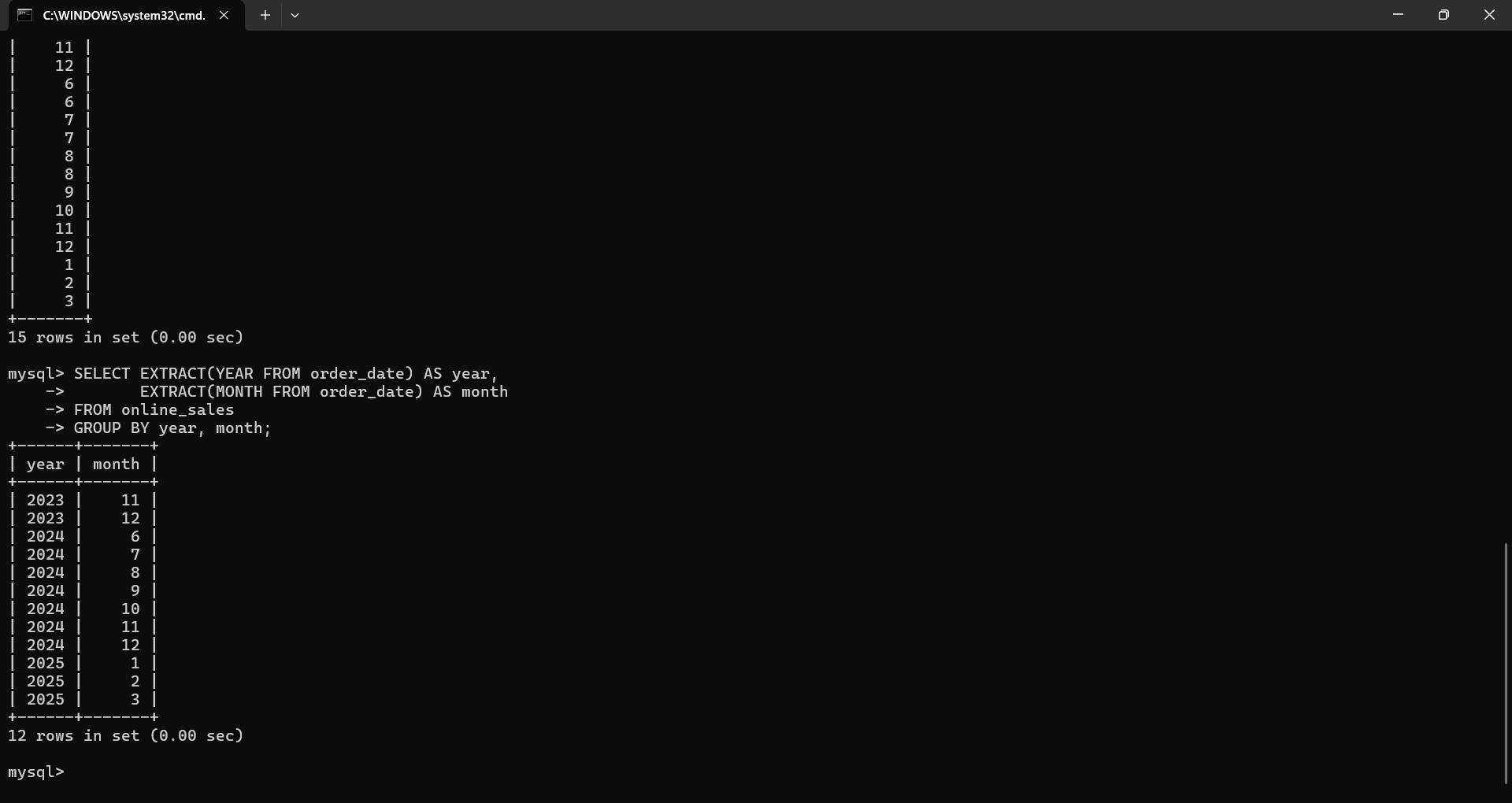
**Hint 1: Use EXTRACT(MONTH FROM order\_date)**

SQL: SELECT EXTRACT(MONTH FROM order\_date) AS month FROM online\_sales;



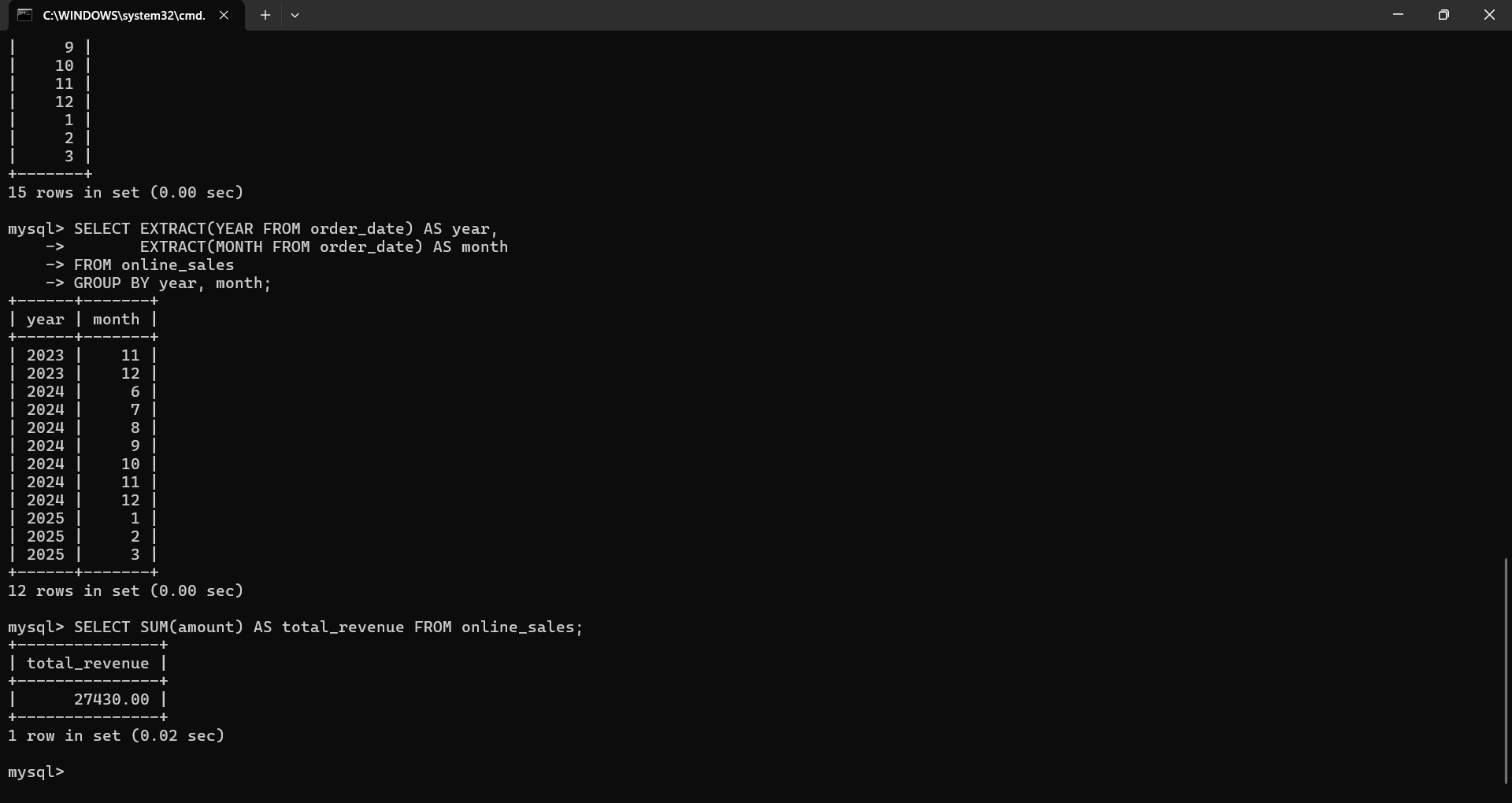
**Hint 2: GROUP BY year/month**

SELECT EXTRACT(YEAR FROM order\_date) AS year,  
 EXTRACT(MONTH FROM order\_date) AS month  
FROM online\_sales  
GROUP BY year, month;



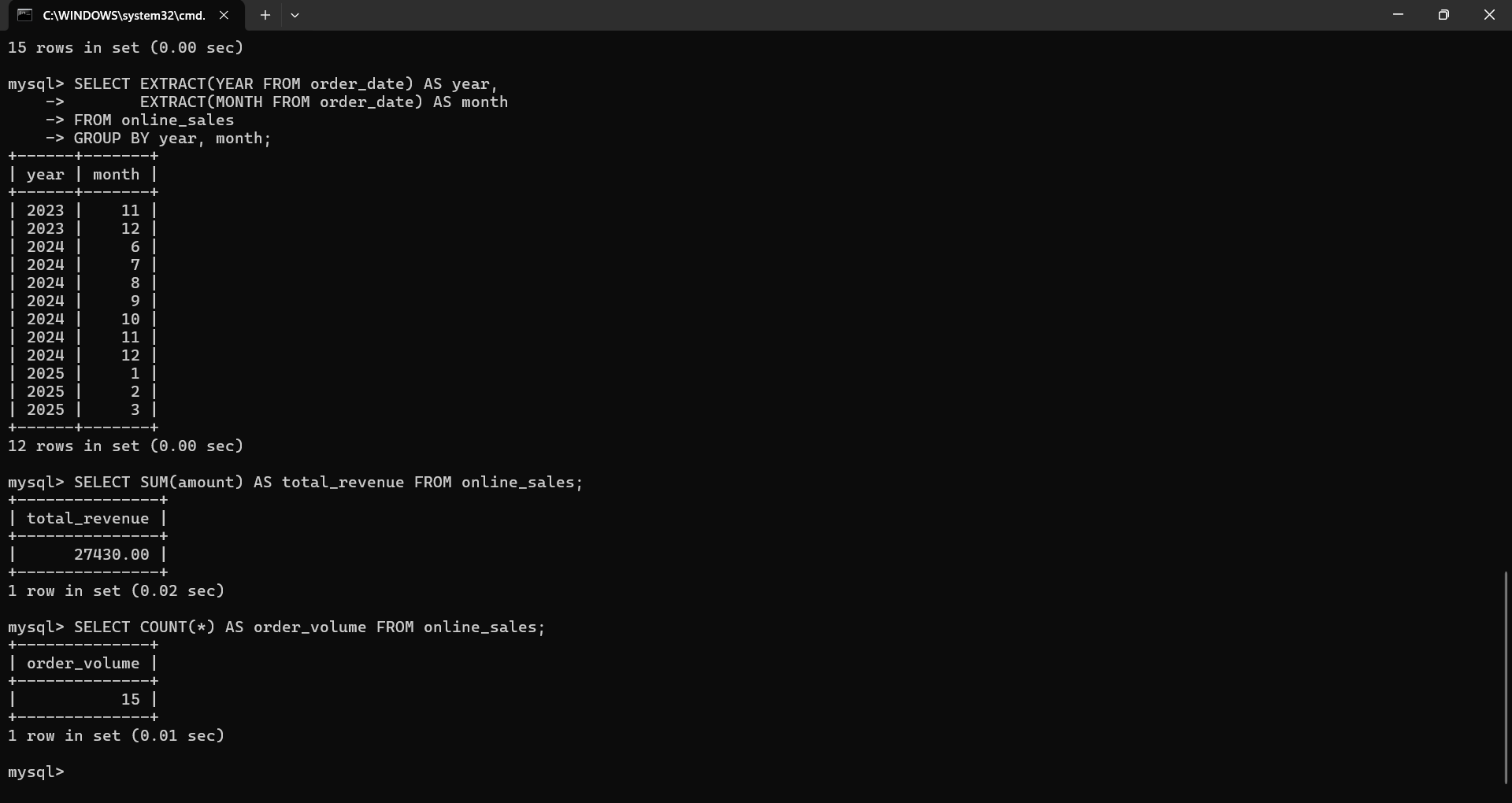
**Hint 3: Use SUM() for revenue**

SQL: SELECT SUM(amount) AS total\_revenue FROM online\_sales;



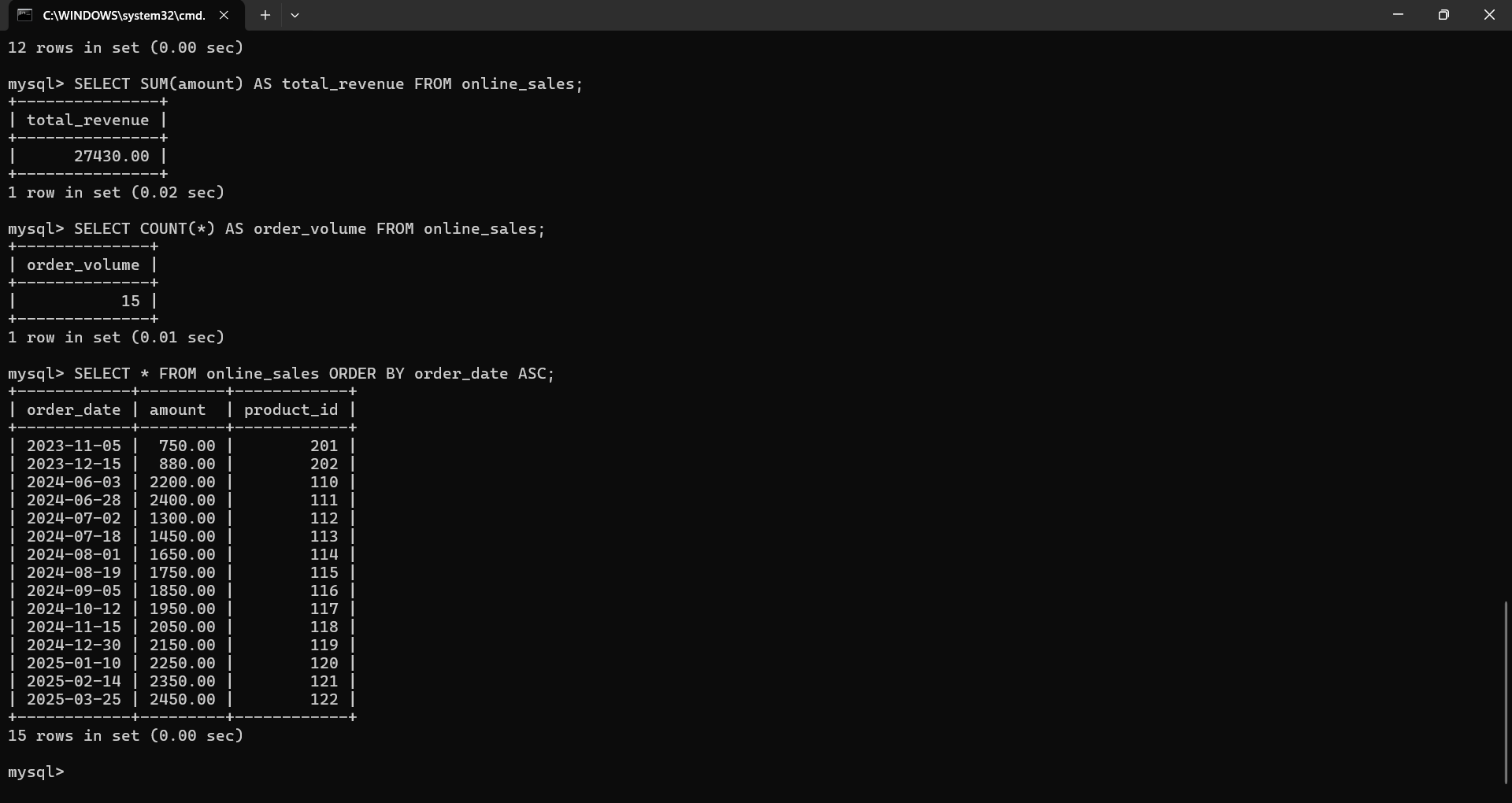
**Hint 4: COUNT(\*) for order volume (no order\_id in table)**

SQL: SELECT COUNT(\*) AS order\_volume FROM online\_sales;



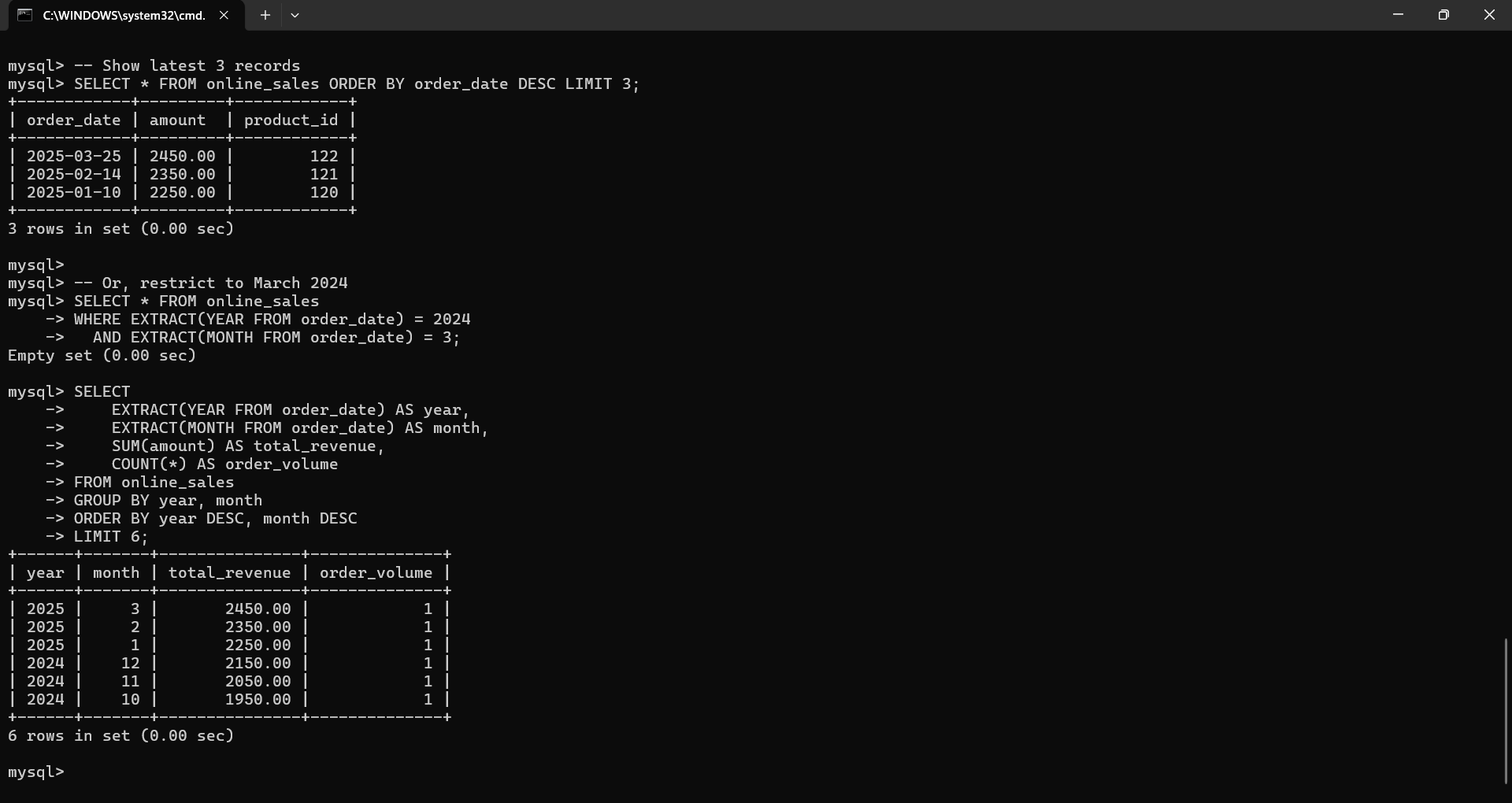
**Hint 5: Use ORDER BY for sorting**

SQL: SELECT \* FROM online\_sales ORDER BY order\_date ASC;



**Hint 6: Limit results for specific time periods**

SQL:   
SELECT EXTRACT(YEAR FROM order\_date) AS year,  
 EXTRACT(MONTH FROM order\_date) AS month,  
 SUM(amount) AS total\_revenue,  
 COUNT(\*) AS order\_volume  
FROM online\_sales  
GROUP BY year, month  
ORDER BY year DESC, month DESC  
LIMIT 6;



## 5. Conclusion

This task involved implementing SQL-based sales analysis using functions like EXTRACT, SUM, COUNT, ORDER BY, and LIMIT. The data was grouped and filtered to show monthly trends in revenue and volume. The exercise was completed entirely using the MySQL command-line interface, reinforcing skills in aggregation and time-based query logic.