Task 6-

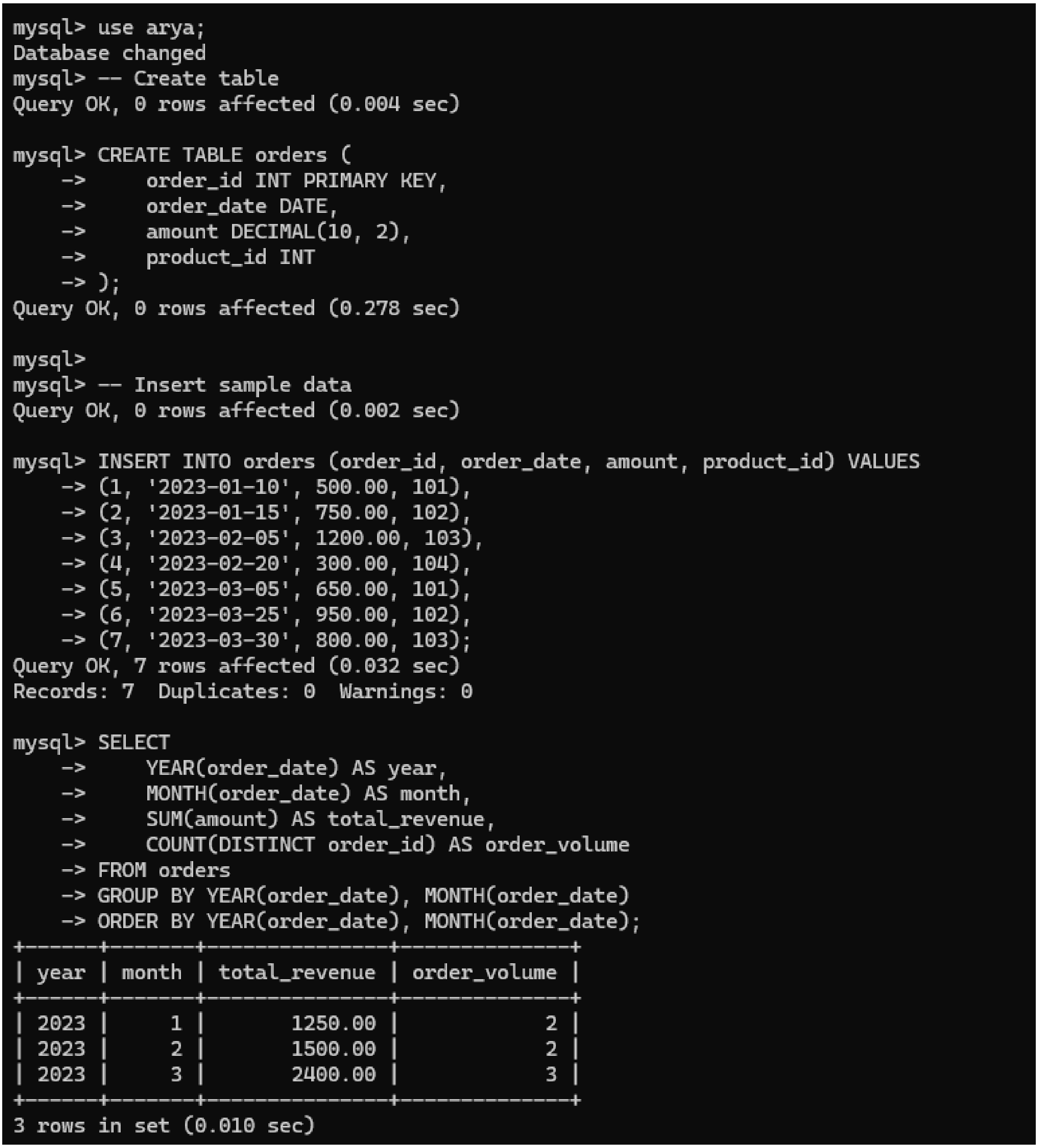
Perform **monthly sales trend analysis** using SQL aggregation functions on the orders table.

Aim :

- Extract **monthly revenue** using SUM(amount)
- Count **monthly order volume** using COUNT(DISTINCT order_id)
- Group and sort data by **year and month**

Outcome :

- SQL query to analyze trends from orders table
- Output showing **year, month, total revenue, and order volume**
- Ready to visualize or use for business insights



```
mysql> use arya;
Database changed
mysql> -- Create table
Query OK, 0 rows affected (0.004 sec)

mysql> CREATE TABLE orders (
    ->     order_id INT PRIMARY KEY,
    ->     order_date DATE,
    ->     amount DECIMAL(10, 2),
    ->     product_id INT
    -> );
Query OK, 0 rows affected (0.278 sec)

mysql>
mysql> -- Insert sample data
Query OK, 0 rows affected (0.002 sec)

mysql> INSERT INTO orders (order_id, order_date, amount, product_id) VALUES
    -> (1, '2023-01-10', 500.00, 101),
    -> (2, '2023-01-15', 750.00, 102),
    -> (3, '2023-02-05', 1200.00, 103),
    -> (4, '2023-02-20', 300.00, 104),
    -> (5, '2023-03-05', 650.00, 101),
    -> (6, '2023-03-25', 950.00, 102),
    -> (7, '2023-03-30', 800.00, 103);
Query OK, 7 rows affected (0.032 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> SELECT
    ->     YEAR(order_date) AS year,
    ->     MONTH(order_date) AS month,
    ->     SUM(amount) AS total_revenue,
    ->     COUNT(DISTINCT order_id) AS order_volume
    -> FROM orders
    -> GROUP BY YEAR(order_date), MONTH(order_date)
    -> ORDER BY YEAR(order_date), MONTH(order_date);
+------+-------+---------------+--------------+
| year | month | total_revenue | order_volume |
+------+-------+---------------+--------------+
| 2023 |     1 |       1250.00 |            2 |
| 2023 |     2 |       1500.00 |            2 |
| 2023 |     3 |       2400.00 |            3 |
+------+-------+---------------+--------------+
3 rows in set (0.010 sec)
```