TASK 6

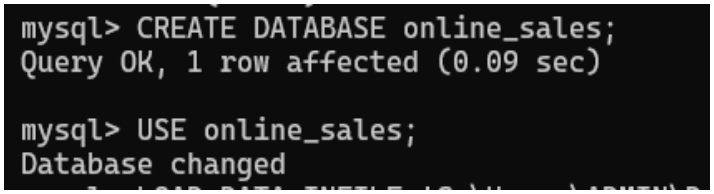
SQL QUERIES:

```
-- 1. Use database
CREATE DATABASE IF NOT EXISTS online_sales;
USE online_sales;

-- 2. Create table for imported data
CREATE TABLE orders (
    order_id    VARCHAR(20),
    order_date  DATE,
    quantity    INT,
    unit_price  DECIMAL(10,2)
);

-- 3. Load data from CSV (using variables to skip extra columns)
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ecommerce_dataset_10000.csv'
INTO TABLE orders
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(
  @customer_id,
  @first_name,
  @last_name,
  @gender,
  @age_group,
  @signup_date,
  @country,
  @product_id,
  @product_name,
  @category,
  @quantity,
  @unit_price,
  @order_id,
  @order_date,
```

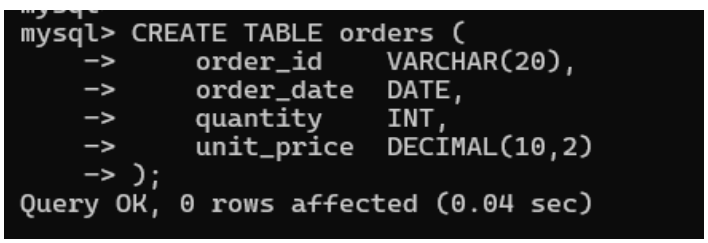
```
  @order_status,
  @payment_method,
  @rating,
  @review_text,
  @review_id,
  @review_date
)
SET
  order_id   = @order_id,
  order_date = STR_TO_DATE(@order_date, '%Y-%m-%d'),
  quantity   = @quantity,
  unit_price = @unit_price;

-- 4. Monthly Sales Trend Analysis
SELECT
    YEAR(order_date)  AS year,
    MONTH(order_date) AS month,
    SUM(quantity * unit_price) AS total_revenue,
    COUNT(DISTINCT order_id)   AS order_volume
FROM orders
GROUP BY year, month
ORDER BY year, month;
```

OUTPUTS- SCREENSHOTS

Creating the Database



Creating Table



Loading the dataset

```
mysql> LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ecommerce_dataset_10000.csv'
    -> INTO TABLE orders
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 ROWS
    -> (
    ->   @customer_id,
    ->   @first_name,
    ->   @last_name,
    ->   @gender,
    ->   @age_group,
    ->   @signup_date,
    ->   @country,
    ->   @product_id,
    ->   @product_name,
    ->   @category,
    ->   @quantity,
    ->   @unit_price,
    ->   @order_id,
    ->   @order_date,
    ->   @order_status,
    ->   @payment_method,
    ->   @rating,
    ->   @review_text,
    ->   @review_id,
    ->   @review_date
    -> )
    -> SET
    ->   order_id   = @order_id,
    ->   order_date = STR_TO_DATE(@order_date, '%Y-%m-%d'),
    ->   quantity   = @quantity,
    ->   unit_price = @unit_price;
Query OK, 10000 rows affected (0.36 sec)
Records: 10000  Deleted: 0  Skipped: 0  Warnings: 0
```

Displaying the content

```
mysql> SELECT COUNT(*) FROM orders;
+----------+
| COUNT(*) |
+----------+
|    10000 |
+----------+
1 row in set (0.01 sec)

mysql> SELECT * FROM orders LIMIT 5;
+----------+------------+----------+------------+
| order_id | order_date | quantity | unit_price |
+----------+------------+----------+------------+
| ORD10000 | 2023-07-13 |        3 |     229.00 |
| ORD10001 | 2024-08-12 |        4 |      59.00 |
| ORD10002 | 2024-08-04 |        2 |      59.00 |
| ORD10003 | 2025-05-23 |        4 |     399.00 |
| ORD10004 | 2023-07-02 |        1 |     110.00 |
+----------+------------+----------+------------+
5 rows in set (0.00 sec)
```

Aggregations

```
mysql> SELECT
    ->     YEAR(order_date)  AS year,
    ->     MONTH(order_date) AS month,
    ->     SUM(quantity * unit_price) AS total_revenue,
    ->     COUNT(DISTINCT order_id)   AS order_volume
    -> FROM orders
    -> GROUP BY year, month
    -> ORDER BY year, month;
+------+-------+---------------+--------------+
| year | month | total_revenue | order_volume |
+------+-------+---------------+--------------+
| 2022 |     8 |      59094.00 |           61 |
| 2022 |     9 |     161540.00 |          246 |
| 2022 |    10 |     226862.00 |          293 |
| 2022 |    11 |     217207.00 |          293 |
| 2022 |    12 |     182328.00 |          274 |
| 2023 |     1 |     194910.00 |          251 |
| 2023 |     2 |     160193.00 |          251 |
| 2023 |     3 |     226185.00 |          296 |
| 2023 |     4 |     229076.00 |          286 |
| 2023 |     5 |     236284.00 |          315 |
| 2023 |     6 |     197018.00 |          277 |
| 2023 |     7 |     207817.00 |          288 |
| 2023 |     8 |     190483.00 |          275 |
| 2023 |     9 |     231165.00 |          292 |
| 2023 |    10 |     197801.00 |          281 |
| 2023 |    11 |     230382.00 |          279 |
| 2023 |    12 |     188711.00 |          272 |
| 2024 |     1 |     193275.00 |          268 |
| 2024 |     2 |     180858.00 |          235 |
| 2024 |     3 |     201151.00 |          300 |
| 2024 |     4 |     212243.00 |          259 |
| 2024 |     5 |     198556.00 |          284 |
| 2024 |     6 |     240348.00 |          284 |
| 2024 |     7 |     207456.00 |          254 |
| 2024 |     8 |     247751.00 |          293 |
| 2024 |     9 |     219328.00 |          294 |
| 2024 |    10 |     208396.00 |          300 |
| 2024 |    11 |     211909.00 |          275 |
| 2024 |    12 |     184730.00 |          269 |
| 2025 |     1 |     207064.00 |          287 |
| 2025 |     2 |     217049.00 |          277 |
| 2025 |     3 |     196312.00 |          270 |
| 2025 |     4 |     217914.00 |          279 |
| 2025 |     5 |     197934.00 |          281 |
| 2025 |     6 |     208290.00 |          263 |
| 2025 |     7 |     215346.00 |          291 |
| 2025 |     8 |     147797.00 |          207 |
+------+-------+---------------+--------------+
37 rows in set (0.08 sec)
```