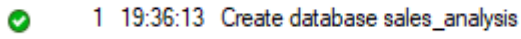
SQL QUERIES AND OUTPUT

1) write a query to create database named ‘sales_analysis’

Query :

```
#Creating database
CREATE DATABASE sales_analysis;
```

Output :

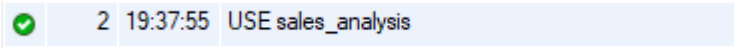


2) Write a query to use that database

Query :

```
#Use Database
USE sales_analysis;
```

Output :

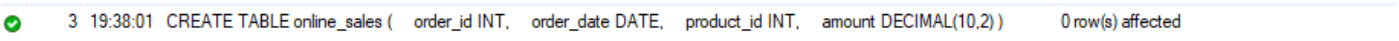


3) Write query to create table named ‘online_sales’

Query :

```
#Creating Table
CREATE TABLE online_sales (
    order_id INT,
    order_date DATE,
    product_id INT,
    amount DECIMAL(10,2)
);
```

Output :



Name : Akshay Patil

4) Write a query to insert atleast 50 records in a table

Query :

```
#inserting records
INSERT INTO online_sales (order_id, order_date, product_id, amount) VALUES
(1, '2024-01-05', 101, 500.00),
(2, '2024-01-06', 102, 300.00),
(3, '2024-01-07', 103, 450.00),
(4, '2024-01-08', 104, 700.00),
(5, '2024-01-09', 105, 650.00),
(6, '2024-01-10', 106, 200.00),
(7, '2024-01-11', 107, 900.00),
(8, '2024-01-12', 101, 120.00),
(9, '2024-01-13', 102, 350.00),
(10, '2024-01-14', 103, 550.00),
(11, '2024-01-15', 104, 250.00),
(12, '2024-01-16', 105, 450.00),
(13, '2024-01-17', 106, 800.00),
(14, '2024-01-18', 107, 600.00),
(15, '2024-01-19', 101, 1000.00),
(16, '2024-01-20', 102, 300.00),
(17, '2024-01-21', 103, 450.00),
(18, '2024-01-22', 104, 700.00),
(19, '2024-01-23', 105, 600.00),
(20, '2024-01-24', 106, 150.00),
(21, '2024-01-25', 107, 950.00),
(22, '2024-01-26', 101, 500.00),
(23, '2024-01-27', 102, 400.00),
(24, '2024-01-28', 103, 800.00),
(25, '2024-01-29', 104, 700.00),
(26, '2024-01-30', 105, 550.00),
(27, '2024-01-31', 106, 300.00),
```

(28, '2024-02-01', 107, 450.00),

(29, '2024-02-02', 101, 600.00),

(30, '2024-02-03', 102, 500.00),

(31, '2024-02-04', 103, 700.00),

(32, '2024-02-05', 104, 800.00),

(33, '2024-02-06', 105, 600.00),

(34, '2024-02-07', 106, 350.00),

(35, '2024-02-08', 107, 120.00),

(36, '2024-02-09', 101, 900.00),

(37, '2024-02-10', 102, 250.00),

(38, '2024-02-11', 103, 650.00),

(39, '2024-02-12', 104, 550.00),

(40, '2024-02-13', 105, 750.00),

(41, '2024-02-14', 106, 300.00),

(42, '2024-02-15', 107, 800.00),

(43, '2024-02-16', 101, 400.00),

(44, '2024-02-17', 102, 150.00),

(45, '2024-02-18', 103, 500.00),

(46, '2024-02-19', 104, 600.00),

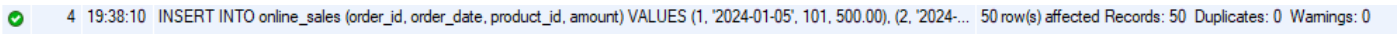
(47, '2024-02-20', 105, 300.00),

(48, '2024-02-21', 106, 450.00),

(49, '2024-02-22', 107, 550.00),

(50, '2024-02-23', 101, 700.00);

Output :



Name : Akshay Patil

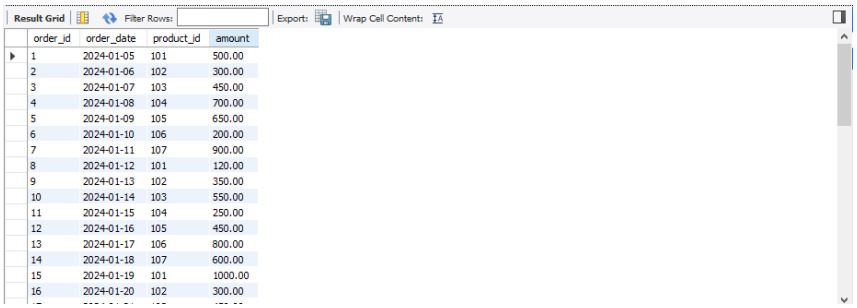
5) Write a query to view all records from the table

Query :

#View all records

SELECT * FROM online_sales;

Output :



order_id	order_date	product_id	amount
1	2024-01-05	101	500.00
2	2024-01-06	102	300.00
3	2024-01-07	103	450.00
4	2024-01-08	104	700.00
5	2024-01-09	105	650.00
6	2024-01-10	106	200.00
7	2024-01-11	107	900.00
8	2024-01-12	101	120.00
9	2024-01-13	102	350.00
10	2024-01-14	103	550.00
11	2024-01-15	104	250.00
12	2024-01-16	105	450.00
13	2024-01-17	106	800.00
14	2024-01-18	107	600.00
15	2024-01-19	101	1000.00
16	2024-01-20	102	300.00

6) Show total revenue for each month.

Query :

SELECT MONTH(order_date) AS month, SUM(amount) AS total_revenue

FROM online_sales

GROUP BY MONTH(order_date);

Output :

month	total_revenue
1	14220.00
2	11970.00

7) Count the number of distinct orders each month.

Query :

SELECT MONTH(order_date) AS month, COUNT(DISTINCT order_id) AS total_orders

FROM online_sales

GROUP BY MONTH(order_date);

Output :

month	total_orders
1	27
2	23

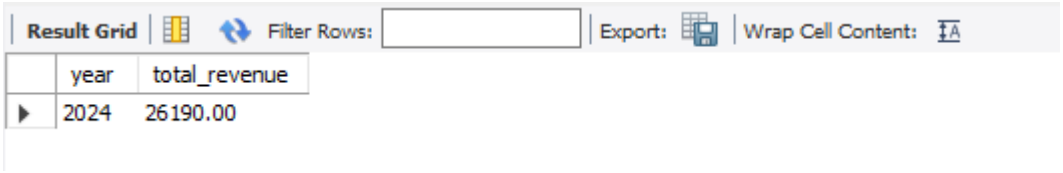
Name : Akshay Patil

8) Calculate total revenue for each year.

Query :

```
SELECT YEAR(order_date) AS year, SUM(amount) AS total_revenue
FROM online_sales
GROUP BY YEAR(order_date);
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

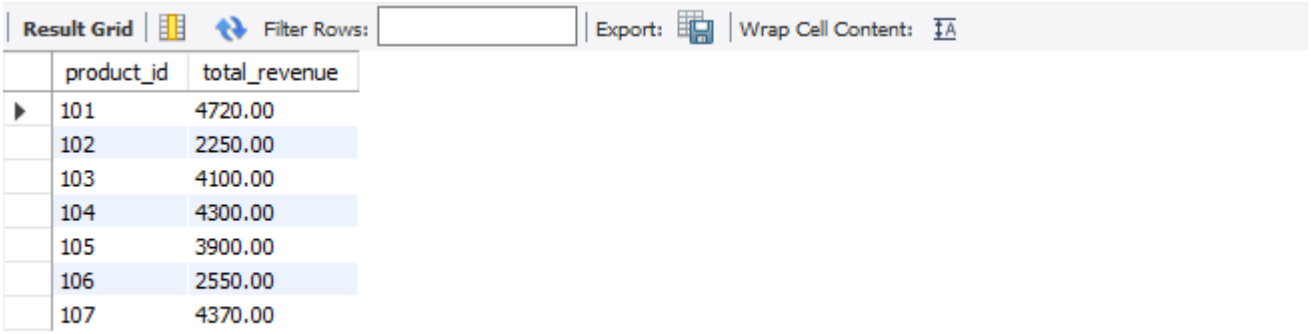
year	total_revenue
2024	26190.00

9) Show total revenue for each product.

Query :

```
SELECT product_id, SUM(amount) AS total_revenue
FROM online_sales
GROUP BY product_id;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

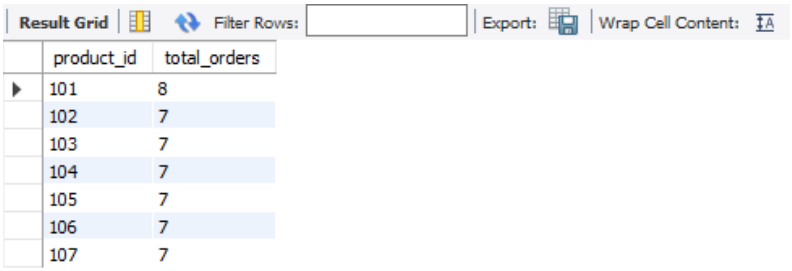
product_id	total_revenue
101	4720.00
102	2250.00
103	4100.00
104	4300.00
105	3900.00
106	2550.00
107	4370.00

10) Count total orders per product.

Query :

```
SELECT product_id, COUNT(order_id) AS total_orders
FROM online_sales
GROUP BY product_id;
```

Output :



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	total_orders
101	8
102	7
103	7
104	7
105	7
106	7
107	7

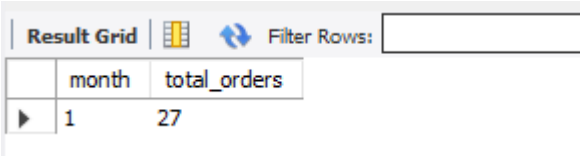
Name : Akshay Patil

11) Find the month with the highest number of orders.

Query :

```
SELECT MONTH(order_date) AS month, COUNT(DISTINCT order_id) AS total_orders
FROM online_sales
GROUP BY MONTH(order_date)
ORDER BY total_orders DESC
LIMIT 1;
```

Output :



Result Grid | Filter Rows:

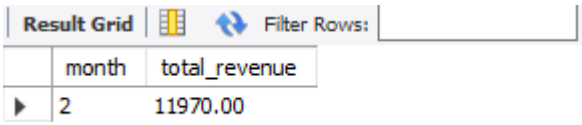
month	total_orders
1	27

12) Find the month with the lowest revenue.

Query :

```
SELECT MONTH(order_date) AS month, SUM(amount) AS total_revenue
FROM online_sales
GROUP BY MONTH(order_date)
ORDER BY total_revenue ASC
LIMIT 1;
```

Output :



Result Grid | Filter Rows:

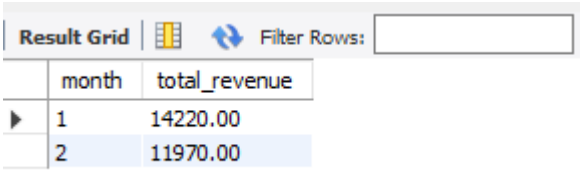
month	total_revenue
2	11970.00

13) Display monthly revenue sorted from highest to lowest.

Query :

```
SELECT MONTH(order_date) AS month, SUM(amount) AS total_revenue
FROM online_sales
GROUP BY MONTH(order_date)
ORDER BY total_revenue DESC;
```

Output :



Result Grid | Filter Rows:

month	total_revenue
1	14220.00
2	11970.00

14) Display monthly order volume (count of distinct orders).

Query :

```
SELECT MONTH(order_date) AS month, COUNT(DISTINCT order_id) AS order_volume
FROM online_sales
GROUP BY MONTH(order_date);
```

Output :

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month	order_volume
1	27
2	23

15) Get total revenue per product for January.

Query :

```
SELECT product_id, SUM(amount) AS revenue
FROM online_sales
WHERE MONTH(order_date) = 1
GROUP BY product_id;
```

Output :

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_id	revenue
101	2120.00
102	1350.00
103	2250.00
104	2350.00
105	2250.00
106	1450.00
107	2450.00

Name : Akshay Patil