ELEVATE LABS TASK 3

Task 3: SQL for Data Analysis

Objective: Use SQL queries to extract and analyze data from a database

Pizza Sales Analysis Using MYSQL

1.Creating a table.

```
create table orders(
order_id int not null ,
order_date date not null ,
order_time time not null ,
primary key (order_id));
```

2.Retrieve the total number of orders placed.

```
1    # Retriving the total number of orders placed
2
3 •  select count(order_id) as Total_orders from pizzastore.orders
```

Output:

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

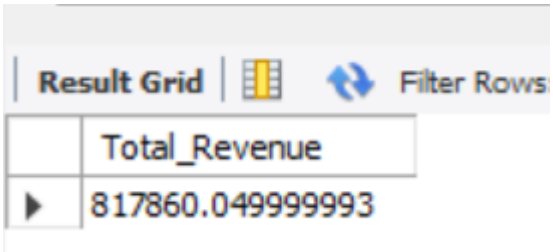
Total_orders
21350

\

3.Calculate the total revenue generated from pizza sales.

```
# Calculate total revenue generated from sales
select sum(pizzas.price * order_details. quantity ) as Total_Revenue
from
pizzas
join
order_details
on pizzas.pizza_id=order_details.pizza_id;
```

Output:

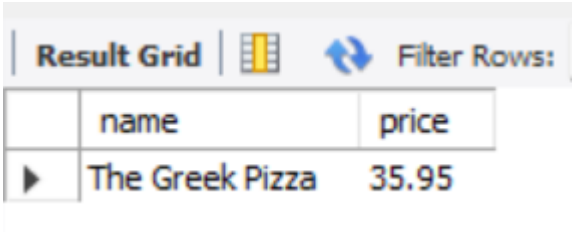


	Total_Revenue
▶	817860.049999993

4.Identify the highest-priced pizza.

```
#Identify the highest-priced pizza
SELECT
    name, price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

Output:

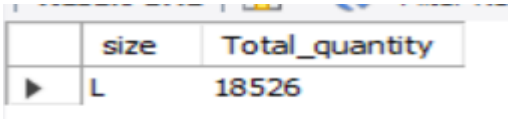


	name	price
▶	The Greek Pizza	35.95

5.Identify the most common pizza size ordered.

```
1  #Identify the most common pizza size ordered.
2
3  SELECT
4      size, COUNT(quantity) AS Total_quantity
5  FROM
6      pizzas
7          JOIN
8      order_details ON pizzas.pizza_id = order_details.pizza_id
9  GROUP BY size
10 ORDER BY Total_quantity DESC
11 LIMIT 1;
```

Output:

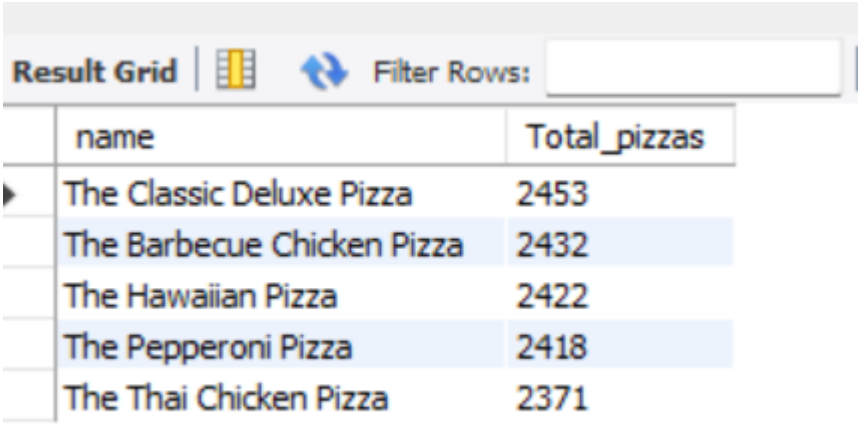


	size	Total_quantity
▶	L	18526

6.List the top 5 most ordered pizza types along with their quantities.

```sql
# List the top 5 most ordered pizza types along with their quantities.
SELECT
    name, sum(quantity) as Total_pizzas
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY name
order by Total_pizzas desc
limit 5;
```

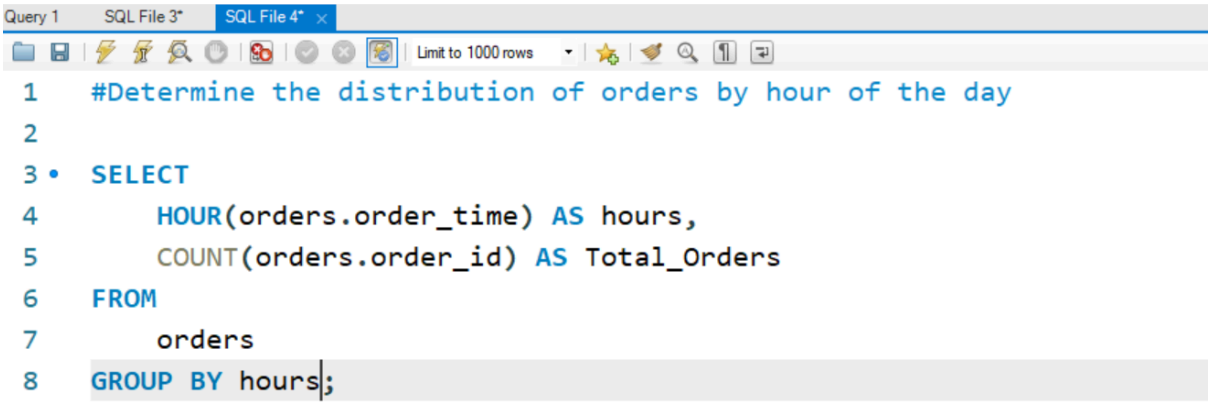
Output:



Result Grid | Filter Rows:

name	Total_pizzas
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

7.Determine the distribution of orders by hour of the day.

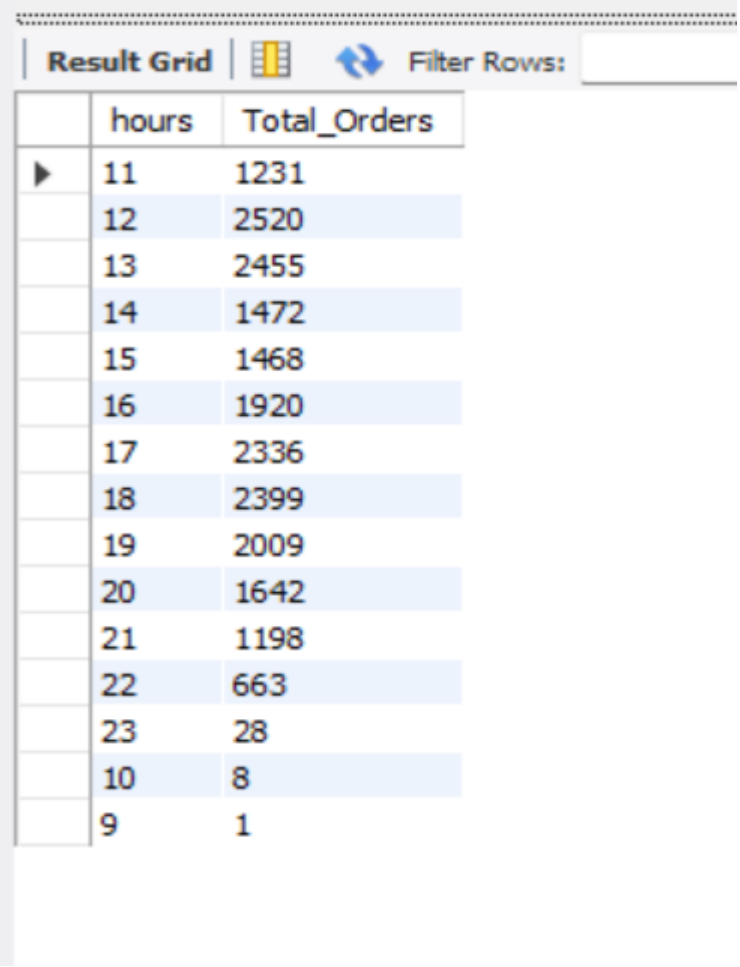


Query 1 | SQL File 3* | SQL File 4* | Limit to 1000 rows

```sql
#Determine the distribution of orders by hour of the day

SELECT
    HOUR(orders.order_time) AS hours,
    COUNT(orders.order_id) AS Total_Orders
FROM
    orders
GROUP BY hours;
```

Output:

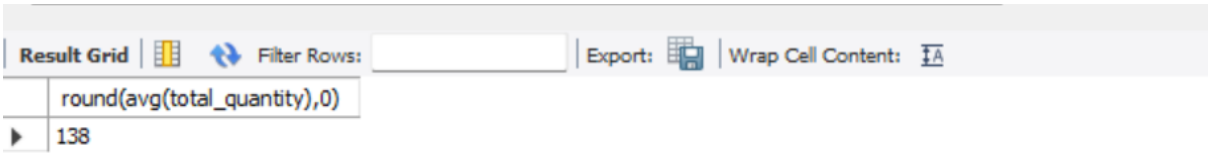


hours	Total_Orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

8.Group the orders by date and calculate the average number of pizzas ordered per day.

```
# Group the orders by date and calculate the average number of pizzas ordered per day
select round(avg(total_quantity),0) from (SELECT
    orders.order_date AS order_date, SUM(order_details.quantity) as Total_quantity
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
GROUP BY order_date)as quantity;
```

Output:



round(avg(total_quantity),0)
138

9.Determine the top 3 most ordered pizza types based on revenue

```sql
#Determine the top 3 most ordered pizza types based on revenue
SELECT
    pizza_types.name,
    ROUND(SUM(pizzas.price * order_details.quantity),
            2) AS Total_Revenue
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY name
ORDER BY Total_Revenue DESC
LIMIT 3;
```

Output: