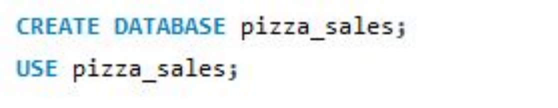
SQL PROJECT

Pizza Sales Analysis

Creating and Using Database

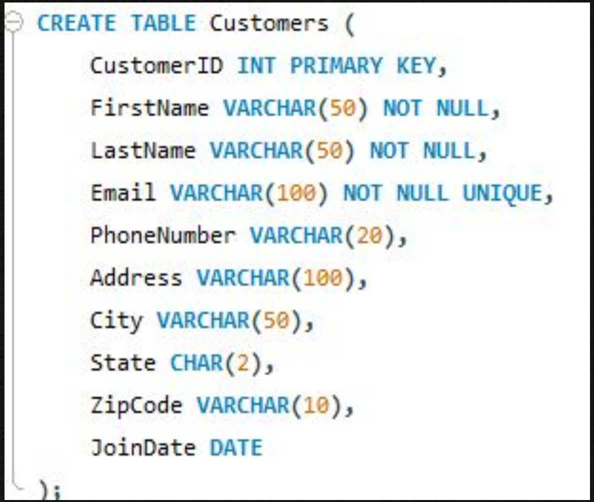
Query :



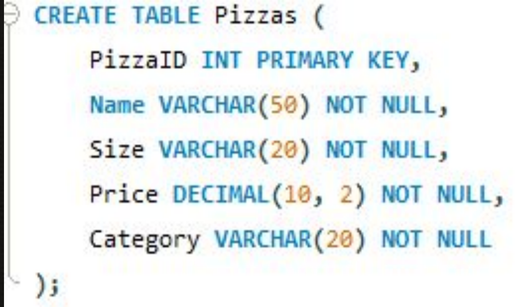
```sql
CREATE DATABASE pizza_sales;
USE pizza_sales;
```

Creating Tables (Customers, Pizzas, Orders, OrderItems)

Queries :

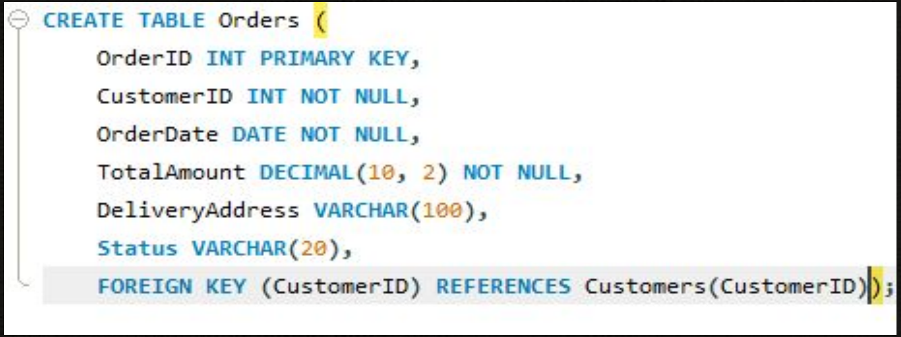


```sql
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    PhoneNumber VARCHAR(20),
    Address VARCHAR(100),
    City VARCHAR(50),
    State CHAR(2),
    ZipCode VARCHAR(10),
    JoinDate DATE
);
```

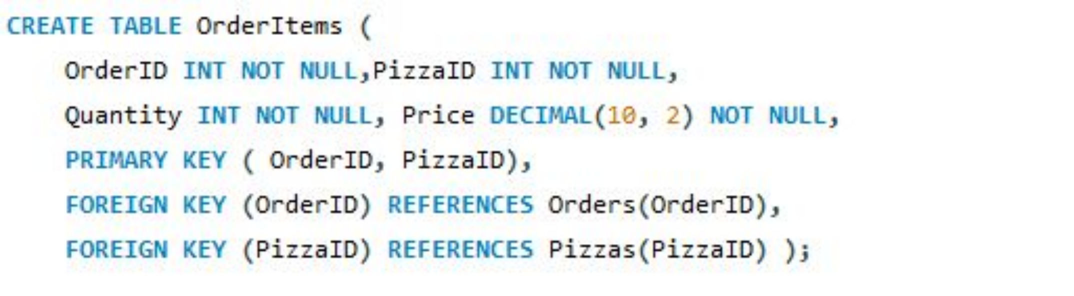


```sql
CREATE TABLE Pizzas (
    PizzaID INT PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Size VARCHAR(20) NOT NULL,
    Price DECIMAL(10, 2) NOT NULL,
    Category VARCHAR(20) NOT NULL
);
```

Queries :



```sql
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT NOT NULL,
    OrderDate DATE NOT NULL,
    TotalAmount DECIMAL(10, 2) NOT NULL,
    DeliveryAddress VARCHAR(100),
    Status VARCHAR(20),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));
```



```sql
CREATE TABLE OrderItems (
    OrderID INT NOT NULL,PizzaID INT NOT NULL,
    Quantity INT NOT NULL, Price DECIMAL(10, 2) NOT NULL,
    PRIMARY KEY ( OrderID, PizzaID),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (PizzaID) REFERENCES Pizzas(PizzaID) );
```

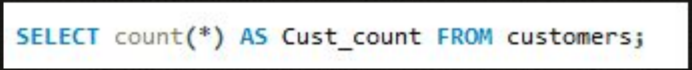
Imported the data into the tables by using table import wizard

Data Imported successfully…..

DATA EXPLORATION

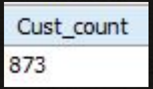
How many customers we have??

Query :



```
SELECT count(*) AS Cust_count FROM customers;
```

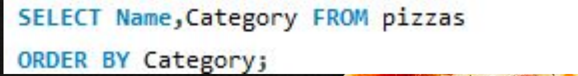
Output :



Cust_count
873

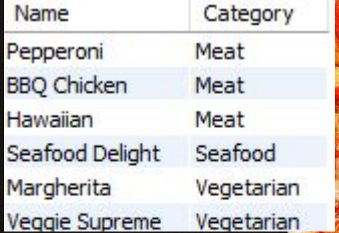
What types of pizzas & pizzas category we have ??

Query :



```
SELECT Name,Category FROM pizzas
ORDER BY Category;
```

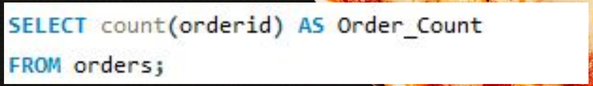
Output :



Name	Category
Pepperoni	Meat
BBQ Chicken	Meat
Hawaiian	Meat
Seafood Delight	Seafood
Margherita	Vegetarian
Veggie Supreme	Vegetarian

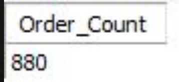
What are the total orders count??

Query :



```
SELECT count(orderid) AS Order_Count
FROM orders;
```

Output :

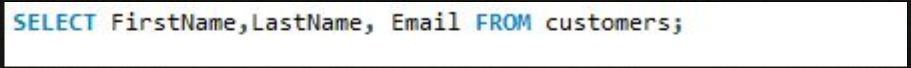


Order_Count
880

Analysis Questions

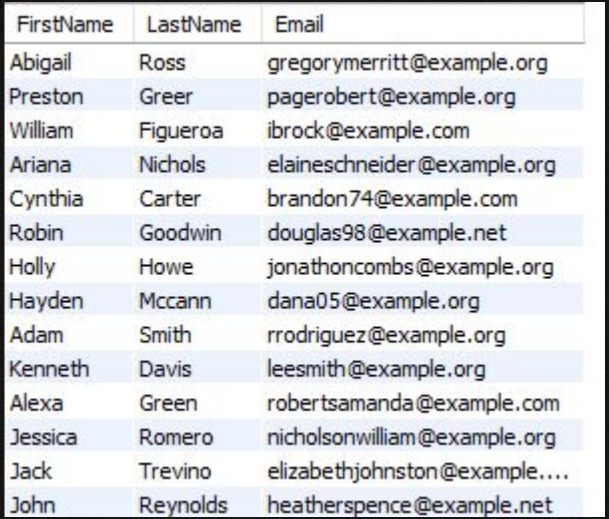
Q1: Retrieve the names and email addresses of all customers from the Customers table.

Query :



```
SELECT FirstName,LastName, Email FROM customers;
```

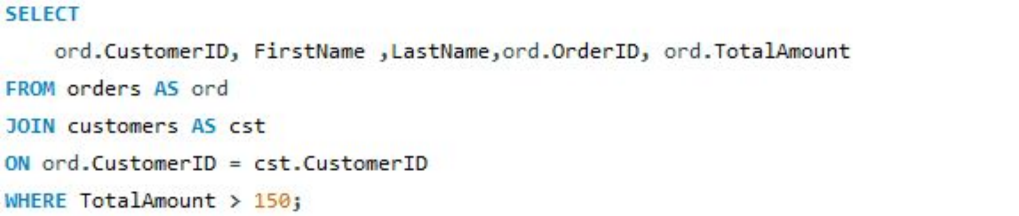
Output :



FirstName	LastName	Email
Abigail	Ross	gregorymerritt@example.org
Preston	Greer	pagerobert@example.org
William	Figueroa	ibrock@example.com
Ariana	Nichols	elaineschneider@example.org
Cynthia	Carter	brandon74@example.com
Robin	Goodwin	douglas98@example.net
Holly	Howe	jonathoncombs@example.org
Hayden	Mccann	dana05@example.org
Adam	Smith	rrodriguez@example.org
Kenneth	Davis	leesmith@example.org
Alexa	Green	robertsamanda@example.com
Jessica	Romero	nicholsonwilliam@example.org
Jack	Trevino	elizabethjohnston@example....
John	Reynolds	heatherspence@example.net

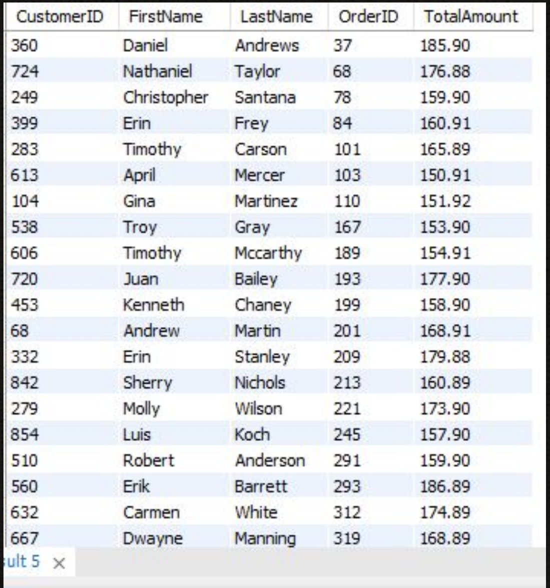
Q2: Find all orders with a total amount greater than $150. Include First Name, Last Name, OrderID, CustomerID and Total Amount in the results.

Query :



```
SELECT
    ord.CustomerID, FirstName ,LastName,ord.OrderID, ord.TotalAmount
FROM orders AS ord
JOIN customers AS cst
ON ord.CustomerID = cst.CustomerID
WHERE TotalAmount > 150;
```

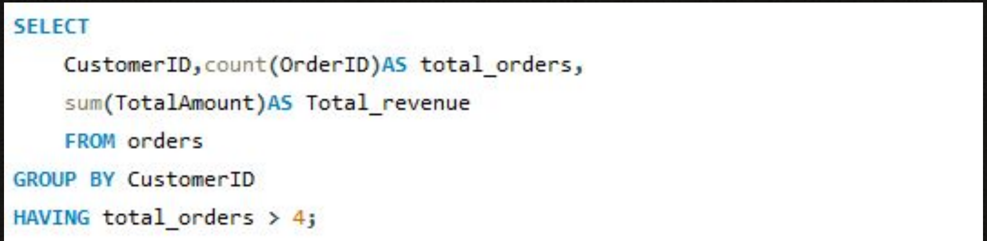
Output :



CustomerID	FirstName	LastName	OrderID	TotalAmount
360	Daniel	Andrews	37	185.90
724	Nathaniel	Taylor	68	176.88
249	Christopher	Santana	78	159.90
399	Erin	Frey	84	160.91
283	Timothy	Carson	101	165.89
613	April	Mercer	103	150.91
104	Gina	Martinez	110	151.92
538	Troy	Gray	167	153.90
606	Timothy	Mccarthy	189	154.91
720	Juan	Bailey	193	177.90
453	Kenneth	Chaney	199	158.90
68	Andrew	Martin	201	168.91
332	Erin	Stanley	209	179.88
842	Sherry	Nichols	213	160.89
279	Molly	Wilson	221	173.90
854	Luis	Koch	245	157.90
510	Robert	Anderson	291	159.90
560	Erik	Barrett	293	186.89
632	Carmen	White	312	174.89
667	Dwayne	Manning	319	168.89

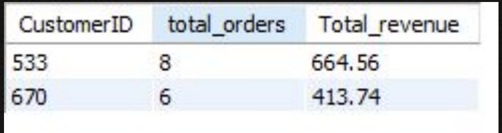
Q3: Calculate the total revenue generated for each customer where the total order is greater than 4. Include CustomerID, Total Orders, and Total Revenue in the results.

Query :



```
SELECT
    CustomerID,count(OrderID)AS total_orders,
    sum(TotalAmount)AS Total_revenue
    FROM orders
GROUP BY CustomerID
HAVING total_orders > 4;
```

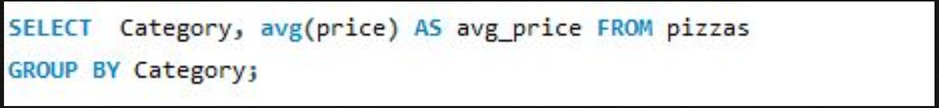
Output :



CustomerID	total_orders	Total_revenue
533	8	664.56
670	6	413.74

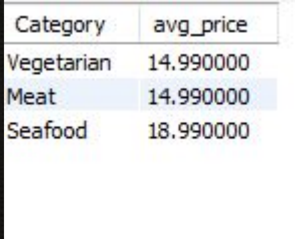
Q4: Find the average price of pizzas in each category. Include Category and Average Price in the results.

Query :



```
SELECT  Category, avg(price) AS avg_price FROM pizzas
GROUP BY Category;
```

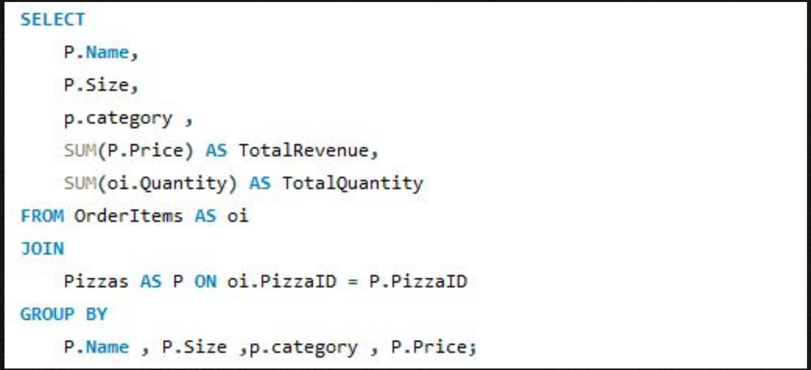
Output :



Category	avg_price
Vegetarian	14.990000
Meat	14.990000
Seafood	18.990000

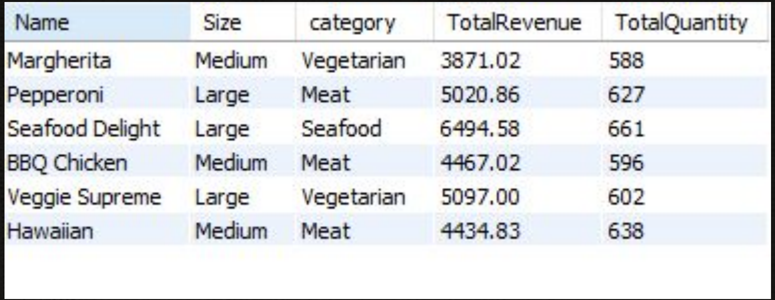
Q5: Calculate the total revenue and total quantity for each type of pizza. Include the pizza name, size, category, Total revenue generated and Total quantity sold by each pizza type.

Query :



```
SELECT
    P.Name,
    P.Size,
    p.category ,
    SUM(P.Price) AS TotalRevenue,
    SUM(oi.Quantity) AS TotalQuantity
FROM OrderItems AS oi
JOIN
    Pizzas AS P ON oi.PizzaID = P.PizzaID
GROUP BY
    P.Name , P.Size ,p.category , P.Price;
```

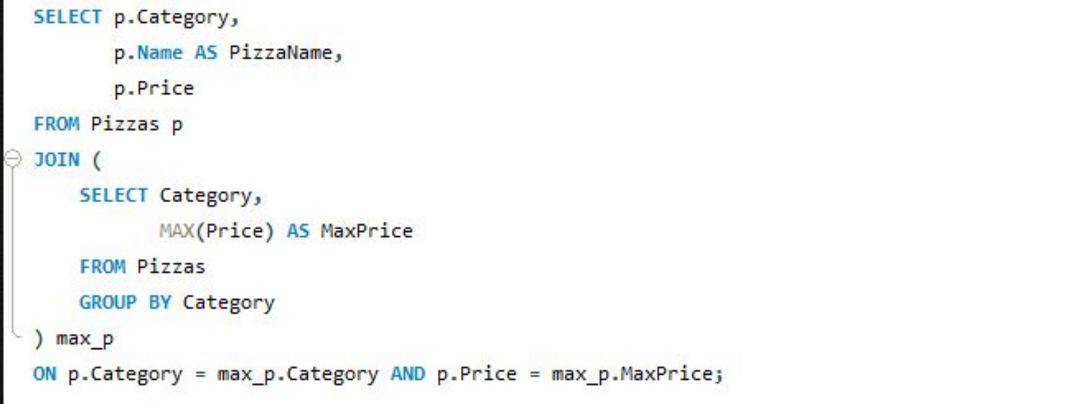
Output :



Name	Size	category	TotalRevenue	TotalQuantity
Margherita	Medium	Vegetarian	3871.02	588
Pepperoni	Large	Meat	5020.86	627
Seafood Delight	Large	Seafood	6494.58	661
BBQ Chicken	Medium	Meat	4467.02	596
Veggie Supreme	Large	Vegetarian	5097.00	602
Hawaiian	Medium	Meat	4434.83	638

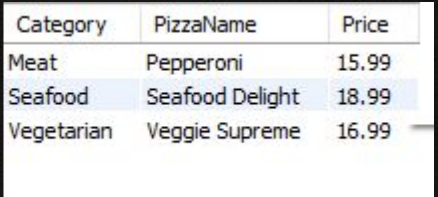
Q6: Retrieve the highest priced pizza in each category. Include Category, PizzaName, and Price in the results.

Query :



```
SELECT p.Category,
       p.Name AS PizzaName,
       p.Price
FROM Pizzas p
JOIN (
    SELECT Category,
           MAX(Price) AS MaxPrice
    FROM Pizzas
    GROUP BY Category
) max_p
ON p.Category = max_p.Category AND p.Price = max_p.MaxPrice;
```

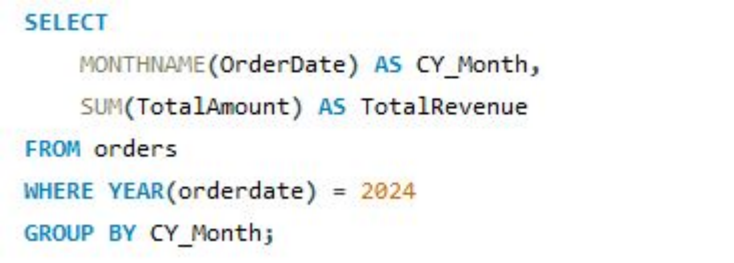
Output :



Category	PizzaName	Price
Meat	Pepperoni	15.99
Seafood	Seafood Delight	18.99
Vegetarian	Veggie Supreme	16.99

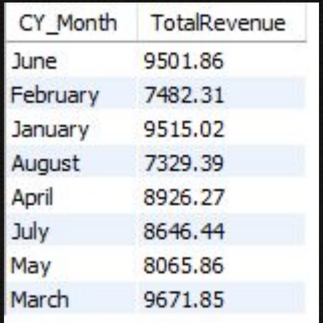
Q7: Find the total revenue generated by month for the current year. Include Month and Total Revenue in the results.

Query :



```
SELECT
    MONTHNAME(OrderDate) AS CY_Month,
    SUM(TotalAmount) AS TotalRevenue
FROM orders
WHERE YEAR(orderdate) = 2024
GROUP BY CY_Month;
```

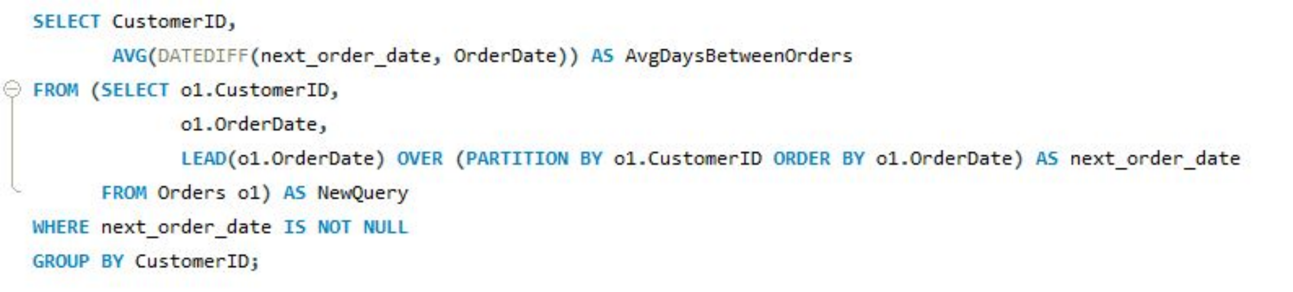
Output :



CY_Month	TotalRevenue
June	9501.86
February	7482.31
January	9515.02
August	7329.39
April	8926.27
July	8646.44
May	8065.86
March	9671.85

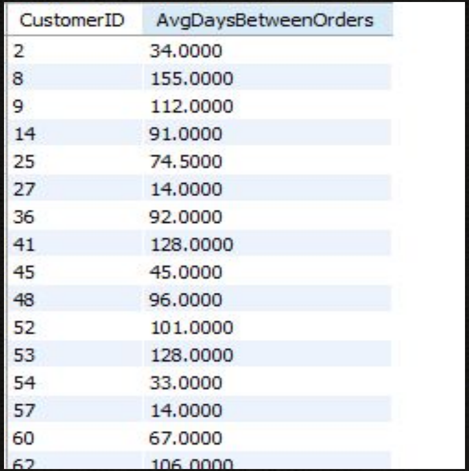
Q8: Determine the average time between orders for each customer. Assume each order is placed on the same day and calculate the average days between consecutive orders.

Query :



```
SELECT CustomerID,
       AVG(DATEDIFF(next_order_date, OrderDate)) AS AvgDaysBetweenOrders
FROM (SELECT o1.CustomerID,
             o1.OrderDate,
             LEAD(o1.OrderDate) OVER (PARTITION BY o1.CustomerID ORDER BY o1.OrderDate) AS next_order_date
    FROM Orders o1) AS NewQuery
WHERE next_order_date IS NOT NULL
GROUP BY CustomerID;
```

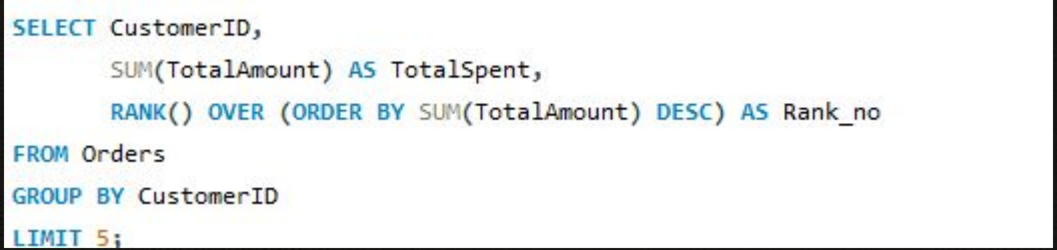
Output :



CustomerID	AvgDaysBetweenOrders
2	34.0000
8	155.0000
9	112.0000
14	91.0000
25	74.5000
27	14.0000
36	92.0000
41	128.0000
45	45.0000
48	96.0000
52	101.0000
53	128.0000
54	33.0000
57	14.0000
60	67.0000
62	106.0000

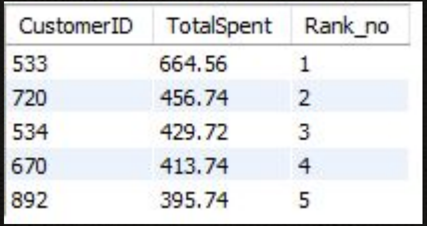
Q9: List the top 5 customers who have spent the most on pizza, including CustomerID, Total Spent, and Rank. Use window functions to rank the customers.

Query :



```
SELECT CustomerID,
       SUM(TotalAmount) AS TotalSpent,
       RANK() OVER (ORDER BY SUM(TotalAmount) DESC) AS Rank_no
FROM Orders
GROUP BY CustomerID
LIMIT 5;
```

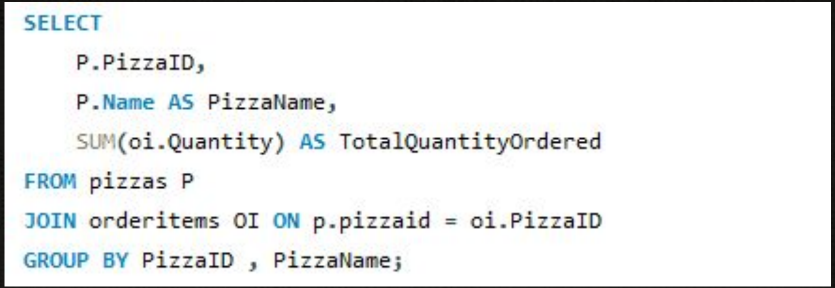
Output :



CustomerID	TotalSpent	Rank_no
533	664.56	1
720	456.74	2
534	429.72	3
670	413.74	4
892	395.74	5

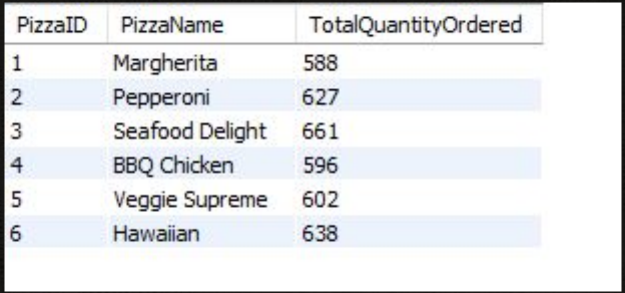
Q10: Identify the pizzas that are ordered the most frequently, including PizzaID, PizzaName, and Total Quantity Ordered.

Query :



```
SELECT
    P.PizzaID,
    P.Name AS PizzaName,
    SUM(oi.Quantity) AS TotalQuantityOrdered
FROM pizzas P
JOIN orderitems OI ON p.pizzaid = oi.PizzaID
GROUP BY PizzaID , PizzaName;
```

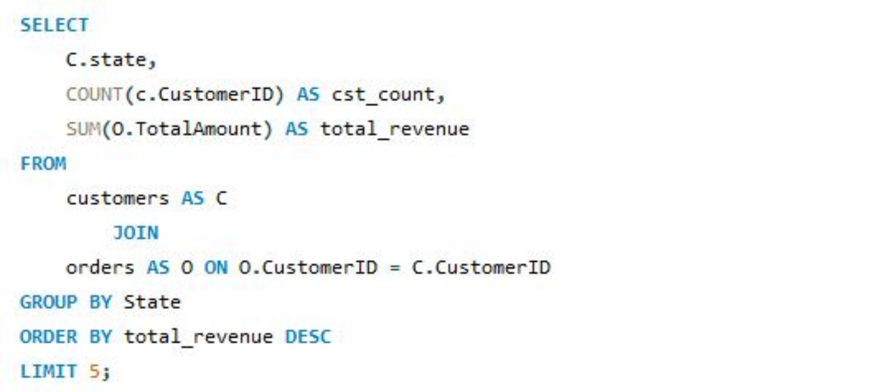
Output :



PizzaID	PizzaName	TotalQuantityOrdered
1	Margherita	588
2	Pepperoni	627
3	Seafood Delight	661
4	BBQ Chicken	596
5	Veggie Supreme	602
6	Hawaiian	638

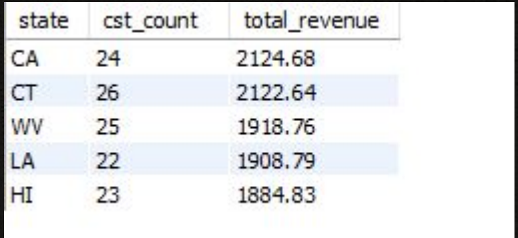
Q11. Identify the top 5 states with the highest total revenue generated from customer orders. For each state, include the total number of customers and the total revenue.

Query :



```
SELECT
    C.state,
    COUNT(c.CustomerID) AS cst_count,
    SUM(O.TotalAmount) AS total_revenue
FROM
    customers AS C
        JOIN
    orders AS O ON O.CustomerID = C.CustomerID
GROUP BY State
ORDER BY total_revenue DESC
LIMIT 5;
```

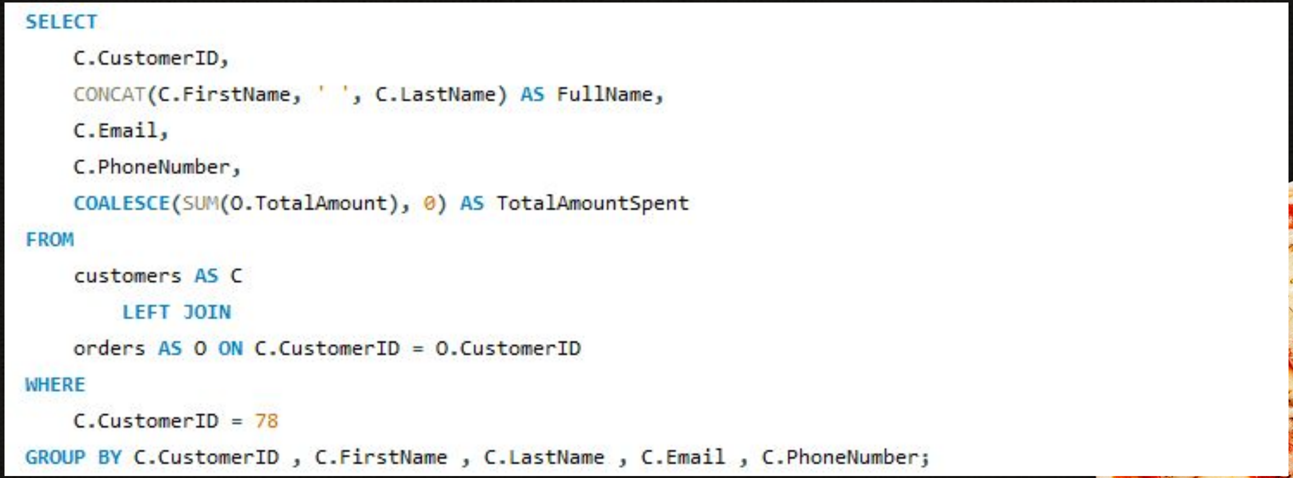
Output :



state	cst_count	total_revenue
CA	24	2124.68
CT	26	2122.64
WV	25	1918.76
LA	22	1908.79
HI	23	1884.83

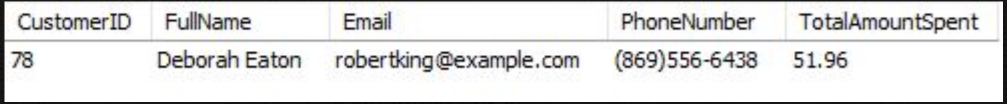
Q12. Find all the details for a specific customer, including their full name, email address, phone number, and the total amount they have spent on orders. (CustomerID = 78)

Query :



```
SELECT
    C.CustomerID,
    CONCAT(C.FirstName, ' ', C.LastName) AS FullName,
    C.Email,
    C.PhoneNumber,
    COALESCE(SUM(O.TotalAmount), 0) AS TotalAmountSpent
FROM
    customers AS C
        LEFT JOIN
    orders AS O ON C.CustomerID = O.CustomerID
WHERE
    C.CustomerID = 78
GROUP BY C.CustomerID , C.FirstName , C.LastName , C.Email , C.PhoneNumber;
```

Output :



```
CustomerID   FullName        Email                     PhoneNumber     TotalAmountSpent
78           Deborah Eaton   robertking@example.com    (869)556-6438   51.96
```

END OF THE PROJECT

By Jatin Thakur

LinkedIN

Github

E-mail