

WHERE CLAUSE QUESTIONS

--Question: How do you retrieve all columns from a table named "Customers"?

```
SELECT * FROM Customers;
```

--Question: Retrieve the names and cuisines of all restaurants from the "Restaurants" table.

```
SELECT Name, Cuisine FROM Restaurants;
```

--Question: Retrieve the unique locations (no duplicates) from the "Restaurants" table.

```
SELECT DISTINCT Location FROM Restaurants;
```

--Question: Retrieve customers from the "Customers" table who live in New York.

```
SELECT * FROM Customers WHERE City = 'New York';
```

--Question: Retrieve restaurants with a rating greater than 4.0 from the "Restaurants" table.

```
SELECT * FROM Restaurants WHERE Rating > 4.0;
```

--Question: Retrieve the total cost for two people from the "Restaurants" table for restaurants in the 'Italian' cuisine.

```
SELECT Name, CostForTwo FROM Restaurants WHERE Cuisine = 'Italian';
```

--Question: Retrieve the first 5 customers (sorted by CustomerID) from the "Customers" table.

```
SELECT TOP 5 * FROM Customers ORDER BY CustomerID;
```

--Question: Count the number of restaurants in each cuisine category from the "Restaurants" table.

```
SELECT Cuisine, COUNT(*) AS CountOfRestaurants FROM Restaurants GROUP BY Cuisine;
```

--Question: Retrieve the average rating of restaurants in each location from the "Restaurants" table.

```
SELECT Location, AVG(Rating) AS AvgRating FROM Restaurants GROUP BY Location;
```

--Question: Retrieve the total cost spent by customers in 'New York' from the "Orders" table.

```
SELECT SUM(TotalAmount) AS TotalSpent FROM Orders  
WHERE CustomerID IN (SELECT CustomerID FROM Customers WHERE City = 'New York');
```

--Intermediate Level

--Question: Retrieve the names of restaurants and the count of orders placed at each restaurant.

```
SELECT R.Name, COUNT(O.OrderID) AS OrderCount  
FROM Restaurants R  
LEFT JOIN Orders O ON R.RestaurantID = O.RestaurantID  
GROUP BY R.Name;
```

--Question: Retrieve the customers who have placed orders with a total amount greater than \$100.

```
SELECT C.FirstName, C.LastName  
FROM Customers C  
JOIN Orders O ON C.CustomerID = O.CustomerID  
WHERE O.TotalAmount > 100.00;
```

--Question: Retrieve the average delivery time for each cuisine category.
SELECT R.Cuisine, AVG(R.DeliveryTimeMinutes) AS AvgDeliveryTime
FROM Restaurants R
GROUP BY R.Cuisine;

--Question: Retrieve the top 5 customers who have spent the most money.
SELECT TOP 5 C.FirstName, C.LastName, SUM(O.TotalAmount) AS TotalSpent
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
GROUP BY C.FirstName, C.LastName
ORDER BY TotalSpent DESC;

--Question: Retrieve the restaurants with the highest ratings in each location.
SELECT R.Location, R.Name, R.Rating
FROM Restaurants R
WHERE (R.Location, R.Rating) IN
(SELECT Location, MAX(Rating) FROM Restaurants GROUP BY Location);

--Question: Retrieve the number of customers who have not placed any orders.
SELECT COUNT(*) AS CustomersWithoutOrders
FROM Customers C
WHERE C.CustomerID NOT IN (SELECT DISTINCT CustomerID FROM Orders);

--Question: Retrieve the customers who have placed orders on more than one occasion.
SELECT C.CustomerID, C.FirstName, C.LastName
FROM Customers C
JOIN (
SELECT CustomerID
FROM Orders
GROUP BY CustomerID
HAVING COUNT(OrderID) > 1
) AS RepeatCustomers ON C.CustomerID = RepeatCustomers.CustomerID;

--Question: Retrieve the orders placed in the year 2023, ordered by the order date in ascending order.
SELECT * FROM Orders
WHERE YEAR(OrderDate) = 2023
ORDER BY OrderDate;

--Question: Retrieve the names of customers who live in cities starting with the letter 'N'.
SELECT FirstName, LastName
FROM Customers
WHERE City LIKE 'N%';

--Question: Retrieve the restaurants with the highest and lowest ratings.
SELECT R.Name, R.Rating
FROM Restaurants R
WHERE R.Rating = (SELECT MAX(Rating) FROM Restaurants)
UNION ALL
SELECT R.Name, R.Rating
FROM Restaurants R
WHERE R.Rating = (SELECT MIN(Rating) FROM Restaurants);

--Advanced Level

--Question: Retrieve the customers who have placed orders with a total amount greater than the average total amount of all orders.

```
SELECT C.FirstName, C.LastName
FROM Customers C
WHERE EXISTS (
    SELECT 1
    FROM Orders O
    WHERE O.CustomerID = C.CustomerID
    GROUP BY O.CustomerID
    HAVING AVG(O.TotalAmount) < (SELECT AVG(TotalAmount) FROM Orders)
);
```

--Question: Retrieve the customers who have placed orders at both 'Italian' and 'Indian' restaurants.

```
SELECT C.FirstName, C.LastName
FROM Customers C
WHERE EXISTS (
    SELECT 1
    FROM Orders O
    JOIN Restaurants R ON O.RestaurantID = R.RestaurantID
    WHERE O.CustomerID = C.CustomerID
    AND R.Cuisine = 'Italian'
)
AND EXISTS (
    SELECT 1
    FROM Orders O
    JOIN Restaurants R ON O.RestaurantID = R.RestaurantID
    WHERE O.CustomerID = C.CustomerID
    AND R.Cuisine = 'Indian'
);
```

--Question: Retrieve the customers who have placed the most orders.

```
WITH CustomerOrderCounts AS (
    SELECT CustomerID, COUNT(*) AS OrderCount
    FROM Orders
    GROUP BY CustomerID
)
SELECT C.FirstName, C.LastName
FROM Customers C
JOIN CustomerOrderCounts COC ON C.CustomerID = COC.CustomerID
WHERE COC.OrderCount = (SELECT MAX(OrderCount) FROM CustomerOrderCounts);
```

--Question: Retrieve the restaurants where the average delivery time is less than 40 minutes and the cost for two is less than \$30.00.

```
SELECT Name, DeliveryTimeMinutes, CostForTwo
FROM Restaurants
WHERE DeliveryTimeMinutes < 40 AND CostForTwo < 30.00;
```

--Question: Retrieve the customers who have placed orders with different payment methods.

```
SELECT C.FirstName, C.LastName
FROM Customers C
WHERE (
    SELECT COUNT(DISTINCT PaymentMethod)
    FROM Orders O
    WHERE O.CustomerID = C.CustomerID
) > 1;
```

--Question: Retrieve the customers who have placed orders with a total amount greater than their average total amount spent.

```
SELECT C.FirstName, C.LastName
FROM Customers C
WHERE (
    SELECT AVG(TotalAmount)
    FROM Orders O
    WHERE O.CustomerID = C.CustomerID
) < ALL (
    SELECT TotalAmount
    FROM Orders O
    WHERE O.CustomerID = C.CustomerID
);
```

--Question: Retrieve the customers who have placed orders at least once in every city.

```
SELECT C.FirstName, C.LastName
FROM Customers C
WHERE (
    SELECT COUNT(DISTINCT City)
    FROM Orders O
    JOIN Customers CC ON O.CustomerID = CC.CustomerID
    WHERE CC.CustomerID = C.CustomerID
) = (SELECT COUNT(DISTINCT City) FROM Customers);
```

--Question: Retrieve the restaurants where the sum of total amounts for all their orders is greater than \$1,000.00.

```
SELECT R.Name, SUM(O.TotalAmount) AS TotalOrderAmount
FROM Restaurants R
JOIN Orders O ON R.RestaurantID = O.RestaurantID
GROUP BY R.Name
HAVING SUM(O.TotalAmount) > 1000.00;
```

--Question: Retrieve the customers who have placed orders on consecutive days.

```
SELECT DISTINCT C.FirstName, C.LastName
FROM Customers C
JOIN Orders O1 ON C.CustomerID = O1.CustomerID
JOIN Orders O2 ON C.CustomerID = O2.CustomerID
WHERE DATEDIFF(day, O1.OrderDate, O2.OrderDate) = 1;
```

--Question: Retrieve the names of customers who have placed orders at restaurants located in the same city as their address city.

```
SELECT C.FirstName, C.LastName
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
JOIN Restaurants R ON O.RestaurantID = R.RestaurantID
WHERE C.City = R.Location;
```