SQL Practice Problems

My solutions

Vamshi Jandhyala

September 13, 2022

Contents

1 Basic Problems

Problem 1: Which shippers do we have?

We have a table called Shippers. Return all the fields from all the shippers.

Solution

```
select * from Shippers;
```

Problem 2: Certain fields from Categories

In the Categories table, selecting all the fields using this SQL: Select * from Categories …will return 4 columns. We only want to see two columns, CategoryName and Description.

Solution

```
select CategoryName, Description from Categories;
```

Problem 3: Sales Representatives

We'd like to see just the FirstName, LastName, and HireDate of all the employees with the Title of Sales Representative. Write a SQL statement that returns only those employees.

Solution

```
select FirstName, LastName, HireDate
from Employees
where Title = 'Sales Representative';
```

Problem 4: Sales Representatives in the United States

Now we'd like to see the same columns as above, but only for those employees that both have the title of Sales Representative, and also are in the United States.

Solution

```
select FirstName, LastName, HireDate
from Employees
where Title = 'Sales Representative' and Country = 'USA';
```

Problem 5: Orders placed by specific EmployeeID

Show all the orders placed by a specific employee. The EmployeeID for this Employee (Steven Buchanan) is 5.

Solution

```sql
select OrderID, OrderDate from Orders
where EmployeeID = 5;
```

Problem 6: Suppliers and ContactTitles

In the Suppliers table, show the SupplierID, ContactName, and ContactTitle for those Suppliers whose ContactTitle is not Marketing Manager.

Solution

```sql
select  SupplierID, ContactName, ContactTitle
from Suppliers
where ContactTitle != 'Marketing Manager';
```

Problem 7: Products with "queso" in ProductName

In the products table, we'd like to see the ProductID and ProductName for those products where the ProductName includes the string "queso".

Solution 1

```sql
select ProductID, ProductName
from Products
where CHARINDEX('queso', ProductName) != 0;
```

Solution 2

```sql
select ProductID, ProductName
from Products
where ProductName like '%queso%';
```

Problem 8: Orders shipping to France or Belgium

Looking at the Orders table, there's a field called ShipCountry. Write a query that shows the OrderID, CustomerID, and ShipCountry for the orders where the ShipCountry is either France or Belgium.

Solution

```sql
select OrderID, CustomerID, ShipCountry
from Orders
where ShipCountry = 'France' OR ShipCountry = 'Belgium';
```

Problem 9: Orders shipping to any country in Latin America

Now, instead of just wanting to return all the orders from France of Belgium, we want to show all the orders from any Latin American country. But we don't have a list of Latin American countries in a table in the Northwind database. So, we're going to just use this list of Latin American countries that happen to be in the Orders table:

1. Brazil 2. Mexico 3. Argentina 4. Venezuela

Solution

```
select OrderID, CustomerID, ShipCountry
from Orders
where ShipCountry in (
    'Brazil', 'Mexico', 'Argentina', 'Venezuela');
```

Problem 10: Employees, in order of age

For all the employees in the Employees table, show the FirstName, LastName, Title, and BirthDate. Order the results by BirthDate, so we have the oldest employees first.

Solution

```
select FirstName, LastName, Title, BirthDate
from Employees
order by BirthDate;
```

Problem 11: Showing only the Date with a DateTime field

In the output of the query above, showing the Employees in order of BirthDate, we see the time of the BirthDate field, which we don't want. Show only the date portion of the BirthDate field.

Solution

```
select FirstName, LastName, Title,
    CONVERT(DATE, BirthDate) AS DateOnlyBirthDate
from Employees
order by BirthDate;
```

Problem 12: Employees full name

Show the FirstName and LastName columns from the Employees table, and then create a new column called FullName, showing FirstName and LastName joined together in one column, with a space in between.

Solution

```
select FirstName, LastName, FirstName + ' ' + LastName  as FullName
from Employees;
```

Problem 13: OrderDetails amount per line item

In the OrderDetails table, we have the fields UnitPrice and Quantity. Create a new field, TotalPrice, that multiplies these two together. We'll ignore the Discount field for now. In addition, show the OrderID, ProductID, UnitPrice, and Quantity. Order by OrderID and ProductID.

Solution

```
select OrderID, ProductID, UnitPrice, Quantity,
    UnitPrice * Quantity As TotalPrice
from OrderDetails
order by OrderID, ProductID;
```

Problem 14: How many customers?

How many customers do we have in the Customers table? Show one value only, and don't rely on getting the record count at the end of a result set.

Solution

```
select count(*) As TotalCustomers
from Customers;
```

Problem 15: When was the first order?

Show the date of the first order ever made in the Orders table.

Solution

```
select MIN(OrderDate) as FirstOrder
from Orders
```

Problem 16: Countries where there are customers

Show a list of countries where the Northwind company has customers.

Solution

```
select DISTINCT(Country) As Country
from Customers;
```

Problem 17: Contact titles for customers

Show a list of all the different values in the Customers table for ContactTitles. Also include a count for each ContactTitle. This is similar in concept to the previous question "Countries where there are customers", except we now want a count for each ContactTitle.

Solution

```
select ContactTitle, count(*) As  TotalContactTitle
from Customers
group by ContactTitle
order by TotalContactTitle DESC;
```

Problem 18: Products with associated supplier names

We'd like to show, for each product, the associated Supplier. Show the ProductID, ProductName, and the CompanyName of the Supplier. Sort by ProductID.

Solution

```
select P.ProductID, P.ProductName, S.CompanyName As Supplier
from Products AS P JOIN Suppliers AS S on P.SupplierID = S.SupplierID
order by P.ProductID;
```

Problem 19: Orders and the Shipper that was used

We'd like to show a list of the Orders that were made, including the Shipper that was used. Show the OrderID, OrderDate (date only), and CompanyName of the Shipper, and sort by OrderID. In order to not show all the orders (there's more than 800), show only those rows with an OrderID of less than 10300.

Solution

```
select O.OrderID, CONVERT(DATE, O.OrderDate) AS OrderDate,
    S.CompanyName
from Orders As O
    JOIN Shippers AS S
        ON O.ShipVia =  S.ShipperID
where O.OrderID < 10300;
```

2 Intermediate Problems

Problem 20: Categories, and the total products in each category

For this problem, we'd like to see the total number of products in each category. Sort the results by the total number of products, in descending order. We have a table called Shippers. Return all the fields from all the shippers.

Solution

```
select C.CategoryName, COUNT(P.ProductID) AS TotalProducts
from Products AS P JOIN Categories AS C ON P.CategoryID = C.CategoryID
group by C.CategoryName
ORDER BY COUNT(P.ProductID) DESC;
```

Problem 21: Total customers per country/city

In the Customers table, show the total number of customers per Country and City.

Solution

```
select Country, City, COUNT(CustomerID) AS TotalCustomers
from Customers
group by Country, City
ORDER BY  COUNT(CustomerID) DESC;
```

Problem 22: Products that need reordering

What products do we have in our inventory that should be reordered? For now, just use the fields UnitsInStock and ReorderLevel, where UnitsInStock is less than the ReorderLevel, ignoring the fields UnitsOnOrder and Discontinued. Order the results by ProductID.

Solution

```
select ProductID, ProductName, UnitsInStock, ReorderLevel
from Products
where  UnitsInStock < ReorderLevel
order by ProductID;
```

Problem 23: Products that need reordering, continued

Now we need to incorporate these fields— UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued—into our calculation. We'll define "products that need reordering" with the following:
1. UnitsInStock plus UnitsOnOrder are less than or equal to ReorderLevel 2. The Discontinued flag is false (0)

Solution

```
select ProductID, ProductName, UnitsInStock, UnitsOnOrder,
       ReorderLevel, Discontinued
from Products
where  UnitsInStock + UnitsOnOrder < ReorderLevel
and Discontinued = 0
order by ProductID;
```

Problem 24: Customer list by region

A salesperson for Northwind is going on a business trip to visit customers, and would like to see a list of all customers, sorted by region, alphabetically. However, he wants the customers with no region (null in the Region field) to be at the end, instead of at the top, where you'd normally find the null values. Within the same region, companies should be sorted by CustomerID.

Solution

```
Select T.CustomerID, T.CompanyName, T.Region
from
    (Select CustomerID, CompanyName, Region,
            Case when Region is null then 1 else 0 End As SortField
    from Customers) AS T
order by T.SortField, T.Region, T.CustomerID;
```

Problem 25: High freight charges

Some of the countries we ship to have very high freight charges. We'd like to investigate some more shipping options for our customers, to be able to offer them lower freight charges. Return the three ship countries with the highest average freight overall, in descending order by average freight.

Solution

```
select Top 3 ShipCountry, AVG(Freight) AS   AverageFreight
from Orders
group by ShipCountry
order by AVG(Freight) DESC;
```

Problem 26: High freight charges - 2015

We're continuing on the question above on high freight charges. Now, instead of using all the orders we have, we only want to see orders from the year 2015.

Solution

```
select Top 3 ShipCountry, AVG(Freight) AS   AverageFreight
from Orders
where YEAR(OrderDate) = 2015
group by ShipCountry
order by AVG(Freight) DESC;
```

Problem 27: High freight charges with between

Another (incorrect) answer to the problem above is this:

```
Select Top 3  ShipCountry, AverageFreight = avg(freight)
From Orders Where OrderDate between '1/1/2015' and '12/31/2015'
Group By ShipCountry
Order By AverageFreight desc;
```

Notice when you run this, it gives Sweden as the ShipCountry with the third highest freight charges. However, this is wrong - it should be France. What is the OrderID of the order that the (incorrect) answer above is missing?

Solution

```
select OrderID, Freight, ShipCountry
from Orders
Where YEAR(OrderDate) = 2015
and
OrderID not in
(select OrderID
from Orders
where OrderDate between '1/1/2015' and '12/31/2015');
```

Problem 28: High freight charges - last year

We're continuing to work on high freight charges. We now want to get the three ship countries with the highest average freight charges. But instead of filtering for a particular year, we want to use the last 12 months of order data, using as the end date the last OrderDate in Orders.

Solution

```
select Top 3 ShipCountry, AVG(Freight) AS AverageFreight
from Orders
where OrderDate between
    Dateadd(yy, -1, (Select Max(OrderDate) from Orders))
    and
    (Select Max(OrderDate) from Orders)
group by ShipCountry
order by AVG(Freight) DESC;
```

Problem 29: Inventory list

We're doing inventory, and need to show information like the below, for all orders. Sort by OrderID and Product ID.

Solution

```
select  O.EmployeeID, E.LastName, O.OrderID, P.ProductName, OD.Quantity
from Employees AS E, Orders AS O, Products AS P, OrderDetails as OD
where
O.EmployeeID = E.EmployeeID
and
OD.OrderID = O.OrderID
```

```
and
OD.ProductID = P.ProductID;
```

Problem 30: Customers with no orders

There are some customers who have never actually placed an order. Show these customers.

Solution

```
select CustomerID
from Customers
where
CustomerID not in (select DISTINCT(CustomerID) from Orders);
```

Problem 31: Customers with no orders

Customers with no orders for EmployeeID 4 One employee (Margaret Peacock, EmployeeID 4) has placed the most orders. However, there are some customers who've never placed an order with her. Show only those customers who have never placed an order with her.

Solution

```
select CustomerID
from Customers
where
CustomerID not in
    (select DISTINCT(CustomerID)
    from Orders where EmployeeID = 4);
```

3 Advanced Problems

Problem 32: High-value customers

We want to send all of our high-value customers a special VIP gift. We're defining high-value customers as those who've made at least 1 order with a total value (not including the discount) equal to $10,000 or more. We only want to consider orders made in the year 2016.

Solution

```
Select Customers.Customerid,
       Customers.Companyname,
       Orders.Orderid,
       Totalorderamount = Sum(Quantity * Unitprice)
From Customers
    Join Orders
        On Orders.Customerid = Customers.Customerid
    Join Orderdetails
        On Orders.Orderid = Orderdetails.Orderid
Where
    Year(Orders.Orderdate) = 2016
Group By Customers.Customerid, Customers.Companyname, Orders.Orderid
Having Sum(Quantity * Unitprice) > 10000
Order By Totalorderamount Desc;
```

Problem 33: High-value customers - total orders

The manager has changed his mind. Instead of requiring that customers have at least one individual orders totaling $10,000 or more, he wants to define high-value customers as those who have orders totaling $15,000 or more in 2016. How would you change the answer to the problem above?

Solution

```
Select Customers.CustomerID, Customers.CompanyName,
       TotalOrderAmount = SUM(Quantity * UnitPrice)
From Customers
    join Orders
        on Orders.CustomerID = Customers.CustomerID
    join OrderDetails
        on Orders.OrderID = OrderDetails.OrderID
Where
    YEAR(Orders.OrderDate) = 2016
group by Customers.CustomerID, Customers.CompanyName
having SUM(Quantity * UnitPrice) > 15000
order by TotalOrderAmount desc;
```

Problem 34: High-value customers - with discount

Change the above query to use the discount when calculating high-value customers. Order by the total amount which includes the discount.

Solution

```
Select Customers.Customerid, Customers.Companyname,
       Totalwithoutdiscount = Sum(Quantity * Unitprice),
       Totalwithdiscount = Sum(Quantity * Unitprice * (1-Discount))
From Customers
    Join Orders
        On Orders.Customerid = Customers.Customerid
    Join Orderdetails
        On Orders.Orderid = Orderdetails.Orderid
Where
    Year(Orders.Orderdate) = 2016
Group By Customers.Customerid, Customers.Companyname
Having Sum(Quantity * Unitprice * (1-Discount)) > 10000
Order By Totalwithdiscount Desc;
```

Solution

(using window functions)

```
Select Customerid, Companyname, Totalswithoutdiscount, Totalswithdiscount
From
    (Select Distinct Customers.Customerid, Companyname,
        Sum(Quantity * Unitprice)
            Over(Partition By Customers.Customerid) As Totalswithoutdiscount,
        Sum(Quantity * Unitprice * (1-Discount))
            Over(Partition By Customers.Customerid) As Totalswithdiscount
    From Customers
        Join Orders
            On Orders.Customerid = Customers.Customerid
        Join Orderdetails
            On Orders.Orderid = Orderdetails.Orderid
    Where
        Year(Orders.Orderdate) = 2016) As T
Where Totalswithdiscount > 10000
```

Problem 35: Month-end orders

At the end of the month, salespeople are likely to try much harder to get orders, to meet their month-end quotas. Show all orders made on the last day of the month. Order by EmployeeID and OrderID.

Solution

```
Select Employeeid, Orderid, Orderdate
From Orders
Where Eomonth(Orderdate) = Convert(Date, Orderdate)
Order By Employeeid, Orderid
```

Problem 36: Orders with many line items

The Northwind mobile app developers are testing an app that customers will use to show orders. In order to make sure that even the largest orders will show up correctly on the app, they'd like some samples of orders that have lots of individual line items. Show the 10 orders with the most line items, in order of total line items.

Solution

(with window functions)

```
Select Top 10 Orderid, Totalorderdetails
From
    (Select Distinct Orderid,
            Count(Orderid) Over(Partition By Orderid) As Totalorderdetails
    From Orderdetails) As T
Order By Totalorderdetails Desc
```

Solution

```
Select Top 10 Orderid, Count(Orderid) As Totalorderdetails
From Orderdetails
Group By Orderid
Order By Totalorderdetails Desc
```

Problem 37: Orders - random assortment

The Northwind mobile app developers would now like to just get a random assortment of orders for beta testing on their app. Show a random set of 2

Solution

```
Select  Top 2 Percent Orderid
From Orders
Order By Newid()
```

Problem 38: Orders - accidental double-entry

Janet Leverling, one of the salespeople, has come to you with a request. She thinks that she accidentally double-entered a line item on an order, with a different ProductID, but the same quantity. She remembers that the quantity was 60 or more. Show all the OrderIDs with line items that match this, in order of OrderID.

Solution

(with window functions)

```
select distinct OrderID
from
    (select OrderID,
        count(ProductID) over(partition by orderid, quantity) as nl
    from OrderDetails
    where Quantity >= 60) As T
where nl > 1
order by OrderID
```

Problem 39: Orders - accidental double-entry details

Based on the previous question, we now want to show details of the order, for orders that match the above criteria.

Solution

(with window functions)

```
With DE_Orders(OrderID) As
    (select distinct OrderID
    from
        (select OrderID,
            count(ProductID) over(partition by orderid, quantity) as nl
        from OrderDetails
        where Quantity >= 60) As T
    where nl > 1)
select OrderID, ProductID, UnitPrice, QUantity, Discount
from OrderDetails
where OrderID in (select OrderID from DE_Orders)
```

Problem 41: Late orders

Some customers are complaining about their orders arriving late. Which orders are late?

Solution

```
select OrderID, OrderDate, RequiredDate, ShippedDate
from Orders
where ShippedDate >= RequiredDate
```

Problem 42: Late orders - which employees?

Some salespeople have more orders arriving late than others. Maybe they’re not following up on the order process, and need more training. Which salespeople have the most orders arriving late?

Solution

```
select Orders.EmployeeID, Employees.LastName,
       count(OrderID) As TotalLateOrders
from Orders
join Employees on Orders.EmployeeID = Employees.EmployeeID
where ShippedDate >= RequiredDate
group by Orders.EmployeeID,  Employees.LastName
order by TotalLateOrders desc
```

Problem 43: Late orders vs. total orders

Andrew, the VP of sales, has been doing some more thinking some more about the problem of late orders. He realizes that just looking at the number of orders arriving late for each salesperson isn't a good idea. It needs to be compared against the total number of orders per salesperson.

Solution

(with window functions)

```
select  EmployeeID, LastName, TotalOrders, TotalLateOrders
from
    (select distinct O.EmployeeID, E.LastName,
    sum(case when ShippedDate >= RequiredDate then 1 else 0 end)
        over (partition by O.EmployeeID) As TotalLateOrders,
    count(O.OrderID) over (partition by O.EmployeeID) As TotalOrders
    from Orders AS O
        join Employees AS E
            on O.EmployeeID = E.EmployeeID) AS T
where TotalLateOrders > 0
```

Problem 46: Late orders vs. total orders - percentage

Now we want to get the percentage of late orders over total orders.

Solution

(with window functions)

```
select  EmployeeID, LastName, TotalOrders, TotalLateOrders,
    (TotalLateOrders*100.0/TotalOrders) AS PercentLateOrders
from
    (select distinct O.EmployeeID, E.LastName,
    sum(case when ShippedDate >= RequiredDate then 1 else 0 end)
        over (partition by O.EmployeeID) As TotalLateOrders,
    count(O.OrderID) over (partition by O.EmployeeID) As TotalOrders
    from Orders AS O
        join Employees AS E
            on O.EmployeeID = E.EmployeeID) AS T
where TotalLateOrders > 0
```

Problem 47: Late orders vs. total orders - fix decimal

Solution

(with window functions)

```
select  EmployeeID, LastName, TotalOrders, TotalLateOrders,
        CONVERT(decimal(10,2),(TotalLateOrders*100.0/TotalOrders))
            AS PercentLateOrders
from
    (select distinct O.EmployeeID, E.LastName,
    sum(case when ShippedDate >= RequiredDate then 1 else 0 end)
        over (partition by O.EmployeeID) As TotalLateOrders,
    count(O.OrderID) over (partition by O.EmployeeID) As TotalOrders
    from Orders AS O
        join Employees AS E
            on O.EmployeeID = E.EmployeeID) AS T
where TotalLateOrders > 0
```

Problem 48: Customer grouping

Andrew Fuller, the VP of sales at Northwind, would like to do a sales campaign for existing customers. He'd like to categorize customers into groups, based on how much they ordered in 2016. Then, depending on which group the customer is in, he will target the customer with different sales materials. The customer grouping categories are 0 to 1,000, 1,000 to 5,000, 5,000 to 10,000, and over 10,000. A good starting point for this query is the answer from the problem "High-value customers - total orders. We don't want to show customers who don't have any orders in 2016. Order the results by CustomerID.

Solution

(with CTE)

```
With CustomerTotalOrder_2016(CustomerID, CompanyName, TotalOrderAmount) AS
    (Select Customers.CustomerID, Customers.CompanyName,
        TotalOrderAmount = SUM(Quantity * UnitPrice)
    From Customers
        join Orders
            on Orders.CustomerID = Customers.CustomerID
        join OrderDetails
            on Orders.OrderID = OrderDetails.OrderID
    Where
        YEAR(Orders.OrderDate) = 2016
    group by Customers.CustomerID, Customers.CompanyName)
select CustomerID, CompanyName, TotalOrderAmount,
case
    when 0 <= TotalOrderAmount AND TotalOrderAmount <= 1000 Then 'Low'
    when 1000 < TotalOrderAmount AND TotalOrderAmount <= 5000 Then 'Medium'
    when 5000 < TotalOrderAmount AND TotalOrderAmount <= 10000 Then 'High'
    when TotalOrderAmount > 10000  Then 'Very High'
end as CustomerGroup
from CustomerTotalOrder_2016
order by CustomerID
```

Problem 50: Customer grouping with percentage

Based on the above query, show all the defined CustomerGroups, and the percentage in each. Sort by the total in each group, in descending order.

Solution

```
With CustomerTotalOrder_2016(CustomerID, CompanyName, TotalOrderAmount) AS
    (Select Customers.CustomerID, Customers.CompanyName,
    TotalOrderAmount = SUM(Quantity * UnitPrice)
    From Customers
        join Orders
            on Orders.CustomerID = Customers.CustomerID
    join OrderDetails
      on Orders.OrderID = OrderDetails.OrderID
    Where
      YEAR(Orders.OrderDate) = 2016
    group by Customers.CustomerID, Customers.CompanyName)
select distinct
 CustomerGroup,
 count(CustomerGroup) over(partition by CustomerGroup) as TotalInGroup,
 (count(CustomerGroup) over(partition by CustomerGroup))*100.0/(select count(*)
from CustomerTotalOrder_2016) as PercentageInGroup
from
  (select
  case
   when 0 <= TotalOrderAmount AND TotalOrderAmount <= 1000 Then 'Low'
   when 1000 < TotalOrderAmount AND TotalOrderAmount <= 5000 Then 'Medium'
   when 5000 < TotalOrderAmount AND TotalOrderAmount <= 10000 Then 'High'
   when TotalOrderAmount > 10000  Then 'Very High'
  end as CustomerGroup
  from CustomerTotalOrder_2016) AS T
order by TotalInGroup desc
```

Problem 51: Customer grouping - flexible

Andrew, the VP of Sales is still thinking about how best to group customers, and define low, medium, high, and very high value customers. He now wants complete flexibility in grouping the customers, based on the dollar amount they've ordered. He doesn't want to have to edit SQL in order to change the boundaries of the customer groups. How would you write the SQL? There's a table called CustomerGroupThreshold that you will need to use. Use only orders from 2016.

Solution

```
With CustomerTotalOrder_2016(CustomerID, CompanyName, TotalOrderAmount) AS
  (Select Customers.CustomerID, Customers.CompanyName,
    TotalOrderAmount = SUM(Quantity * UnitPrice)
  From Customers
    join Orders
      on Orders.CustomerID = Customers.CustomerID
    join OrderDetails
      on Orders.OrderID = OrderDetails.OrderID
    Where
      YEAR(Orders.OrderDate) = 2016
    group by Customers.CustomerID, Customers.CompanyName)
select CustomerID, CompanyName, TotalOrderAmount, CustomerGroupName
from CustomerTotalOrder_2016
  join CustomerGroupThresholds
    on (RangeBottom < TotalOrderAmount)
    and (RangeTop > TotalOrderAmount)
order by CustomerID
```

Problem 52: Countries with suppliers or customers

Some Northwind employees are planning a business trip, and would like to visit as many suppliers and customers as possible. For their planning, they'd like to see a list of all countries where suppliers and/or customers are based.

Solution

```
select Country from Suppliers
Union
select Country from Customers
```

Problem 53: Countries with suppliers or customers version 2

The employees going on the business trip don't want just a raw list of countries, they want more details. We'd like to see output like the below, in the Expected Results.

Solution

```
select S.Country as SupplierCountry, C.Country as CustomerCountry
from Suppliers as S
    left outer join Customers AS C
        on S.Country = C.Country
Union
select S.Country as SupplierCountry, C.Country as CustomerCountry
from Customers as C
    left outer join Suppliers AS S
        on S.Country = C.Country
```

Problem 54: Countries with suppliers or customers version 3

The output of the above is improved, but it's still not ideal What we'd really like to see is the country name, the total suppliers, and the total customers.

Solution

```
With All_Countries(Country) AS
    (select Country from Suppliers
    Union
    select Country from Customers)
select CC.Country, count(S.SupplierID) AS TotalSuppliers, TotalCustomers
from
    (select AC.Country, count(C.CustomerID) AS TotalCustomers
    from All_Countries as AC
        left outer join Customers as C on AC.Country = C.Country
    group by AC.Country) AS CC
    left outer join Suppliers as S
        on CC.Country = S.Country
group by CC.Country, TotalCustomers
```

Problem 55: First order in each country

Looking at the Orders table—we'd like to show details for each order that was the first in that particular country, ordered by OrderID. So, we need one row per ShipCountry, and CustomerID, OrderID, and OrderDate should be of the first order from that country.

Solution

```
select O.ShipCountry, O.OrderID, O.CustomerID, FirstOrderDate
from Orders as O
    join
        (select ShipCountry, Min(OrderDate) as FirstOrderDate
        from Orders
        group by ShipCountry) AS SCFOD
            on O.ShipCountry = SCFOD.ShipCountry
            and O.OrderDate = SCFOD.FirstOrderDate
```

Solution

(with window functions)

```
select  ShipCountry,CustomerID, OrderID, OrderDate
from
(select CustomerID, ShipCountry, OrderID, OrderDate,
    (rank() over(partition by ShipCountry order by OrderDate)) as rnk
from Orders) AS T
where rnk = 1
group by  CustomerID, ShipCountry, OrderID, OrderDate
```