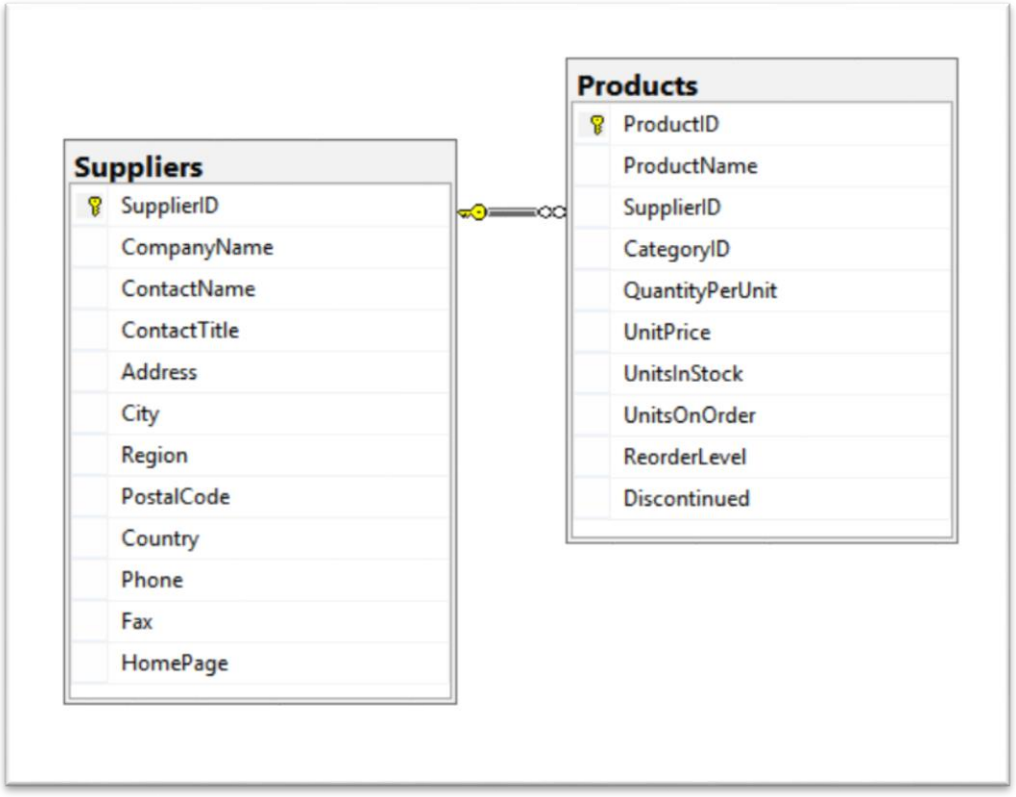
18. Products with their associated supplier names

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

This question will introduce what may be a new concept, the Join clause in SQL. The Join clause is used to join two or more relational database tables together in a logical way.

Here's a data model of the relationship between Products and Suppliers.



Expected Results

```
ProductID   ProductName                              Supplier
----------- ---------------------------------------- ------------------------------
1           Chai                                     Exotic Liquids
2           Chang                                    Exotic Liquids
3           Aniseed Syrup                            Exotic Liquids
```

```
4           Chef Anton's Cajun Seasoning            New Orleans Cajun Delights
5           Chef Anton's Gumbo Mix                  New Orleans Cajun Delights
6           Grandma's Boysenberry Spread            Grandma Kelly's Homestead
7           Uncle Bob's Organic Dried Pears         Grandma Kelly's Homestead
8           Northwoods Cranberry Sauce              Grandma Kelly's Homestead
9           Mishi Kobe Niku                         Tokyo Traders
10          Ikura                                   Tokyo Traders

... (skipping some rows)

66          Louisiana Hot Spiced Okra               New Orleans Cajun Delights
67          Laughing Lumberjack Lager               Bigfoot Breweries
68          Scottish Longbreads                     Specialty Biscuits, Ltd.
69          Gudbrandsdalsost                        Norske Meierier
70          Outback Lager                           Pavlova, Ltd.
71          Flotemysost                             Norske Meierier
72          Mozzarella di Giovanni                  Formaggi Fortini s.r.l.
73          Röd Kaviar                              Svensk Sjöföda AB
74          Longlife Tofu                           Tokyo Traders
75          Rhönbräu Klosterbier                    Plutzer Lebensmittelgroßmärkte AG
76          Lakkalikööri                            Karkki Oy
77          Original Frankfurter grüne Soße         Plutzer Lebensmittelgroßmärkte AG

(77 row(s) affected)
```

Hint

Just as a reference, here's an example of what the syntax for the Join looks like, using different tables from the Northwind database. It will show all the products, with the associated CategoryName.

```
Select
    ProductID
    ,ProductName
    ,CategoryName
From Products
    Join Categories
        on Products.CategoryID = Categories.CategoryID
```

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.

Expected Results

```
CustomerID CompanyName                               Region
---------- ----------------------------------------- ---------------
OLDWO      Old World Delicatessen                    AK
BOTTM      Bottom-Dollar Markets                     BC
LAUGB      Laughing Bacchus Wine Cellars             BC
LETSS      Let's Stop N Shop                         CA
HUNGO      Hungry Owl All-Night Grocers              Co. Cork
GROSR      GROSELLA-Restaurante                      DF
SAVEA      Save-a-lot Markets                        ID
ISLAT      Island Trading                            Isle of Wight
LILAS      LILA-Supermercado                         Lara
THECR      The Cracker Box                           MT
RATTC      Rattlesnake Canyon Grocery                NM

... (skipping some rows)

SANTG      Santé Gourmet                             NULL
SEVES      Seven Seas Imports                        NULL
SIMOB      Simons bistro                             NULL
SPECD      Spécialités du monde                      NULL
SUPRD      Suprêmes délices                          NULL
TOMSP      Toms Spezialitäten                        NULL
TORTU      Tortuga Restaurante                       NULL
VAFFE      Vaffeljernet                              NULL
VICTE      Victuailles en stock                      NULL
VINET      Vins et alcools Chevalier                 NULL
WANDK      Die Wandernde Kuh                         NULL
WARTH      Wartian Herkku                            NULL
WILMK      Wilman Kala                               NULL
WOLZA      Wolski  Zajazd                            NULL

(91 row(s) affected)
```

Hint

You won't be able to sort directly on the Region field here. You'll need to sort on the Region field, and also on a computed field that you create, which will give you a secondary sort for when Region is null

First, without ordering, create a computed field that has a value which will sort the way you want. In this case, you can create a field with the Case statement, which allows you do to if/then logic. You want a field that is 1 when Region is null.

Take a look at the Examples section in the SQL Server documentation for Case.

Note that when filtering for null values, you can't use “FieldName = Null”. You must use “FieldName is null”.

Hint

You should have something like this:

```
Select
    CustomerID
    ,CompanyName
    ,Region
    ,Case
        when Region is null then 1
        else 0
    End
From Customers
```

When the Region contains a null, you will have a 1 in the final column. Now, just add the fields for the Order By clause, in the right order.

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.

Expected Result

```
CustomerID CompanyName                               OrderID     TotalOrderAmount
---------- ----------------------------------------- ----------- --------------------
QUICK      QUICK-Stop                                10865       17250.00
SAVEA      Save-a-lot Markets                        11030       16321.90
HANAR      Hanari Carnes                             10981       15810.00
KOENE      Königlich Essen                           10817       11490.70
RATTC      Rattlesnake Canyon Grocery                10889       11380.00
HUNGO      Hungry Owl All-Night Grocers              10897       10835.24

(6 row(s) affected)
```

Hint

First, let's get the necessary fields for all orders made in the year 2016. Don't bother grouping yet, just work on the Where clause. You'll need the CustomerID, CompanyName from Customers; OrderID from Orders; and Quantity and unit price from OrderDetails. Order by the total amount of the order, in descending order.

Hint

You should have something like this:

```
Select
    Customers.CustomerID
    ,Customers.CompanyName
    ,Orders.OrderID
    ,Amount = Quantity * UnitPrice
From Customers
    join Orders
        on Orders.CustomerID = Customers.CustomerID
    join OrderDetails
        on Orders.OrderID = OrderDetails.OrderID
Where
    OrderDate >= '20160101'
    and OrderDate  < '20170101'
```

This gives you the total amount for each Order Detail item in 2016 orders, at the Order Detail level. Now, which fields do you need to group on, and which need to be summed?

Hint

```
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
    OrderDate >= '20160101'
    and OrderDate  < '20170101'
Group By
    Customers.CustomerID
    ,Customers.CompanyName
    ,Orders.OrderID
```

The fields at the Customer and Order level need to be grouped by, and the TotalOrderAmount needs to be summed.

How would you filter on the sum, in order to get orders of $10,000 or more? Can you put it straight into the where clause?

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.

Expected Result

```
CustomerID CompanyName                               TotalOrderAmount      CustomerGroup
---------- ----------------------------------------- --------------------- -------------
ALFKI      Alfreds Futterkiste                       2302.20               Medium
ANATR      Ana Trujillo Emparedados y helados        514.40                Low
ANTON      Antonio Moreno Taquería                   660.00                Low
AROUT      Around the Horn                           5838.50               High
BERGS      Berglunds snabbköp                        8110.55               High
BLAUS      Blauer See Delikatessen                   2160.00               Medium
BLONP      Blondesddsl père et fils                  730.00                Low
BOLID      Bólido Comidas preparadas                 280.00                Low
BONAP      Bon app'                                  7185.90               High
BOTTM      Bottom-Dollar Markets                     12227.40              Very High
BSBEV      B's Beverages                             2431.00               Medium
CACTU      Cactus Comidas para llevar                1576.80               Medium
CHOPS      Chop-suey Chinese                         4429.40               Medium

... (skipping some rows)

SPLIR      Split Rail Beer & Ale                     1117.00               Medium
SUPRD      Suprêmes délices                          11862.50              Very High
THEBI      The Big Cheese                            69.60                 Low
THECR      The Cracker Box                           326.00                Low
```

```
TOMSP      Toms Spezialitäten                    910.40              Low
TORTU      Tortuga Restaurante                   1874.50             Medium
TRADH      Tradição Hipermercados                4401.62             Medium
TRAIH      Trail's Head Gourmet Provisioners     237.90              Low
VAFFE      Vaffeljernet                          4333.50             Medium
VICTE      Victuailles en stock                  3022.00             Medium
WANDK      Die Wandernde Kuh                     1564.00             Medium
WARTH      Wartian Herkku                        300.00              Low
WELLI      Wellington Importadora                1245.00             Medium
WHITC      White Clover Markets                  15278.90            Very High
WILMK      Wilman Kala                           1987.00             Medium
WOLZA      Wolski  Zajazd                        1865.10             Medium

(81 row(s) affected)
```

Hint

This is the SQL from the problem “High-value customers - total orders”, but without the filter for order totals over 10,000.

```sql
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
    OrderDate >= '20160101'
    and OrderDate  < '20170101'
Group By
    Customers.CustomerID
    ,Customers.CompanyName
Order By TotalOrderAmount Desc;
```

Hint

You can use the above SQL in a CTE (common table expression), and then build on it, using a Case statement on the TotalOrderAmount.

ANSWERS

18. Products with their associated supplier names

Answer

```
Select
    ProductID
    ,ProductName
    ,Supplier = CompanyName
From Products
    Join Suppliers
        on Products.SupplierID = Suppliers.SupplierID
```

Discussion

Joins can range from the very simple, which we have here, to the very complex. You need to understand them thoroughly, as they're critical in writing anything but the simplest SQL.

One thing you'll see when reading SQL code is, instead of something like the answer above, something like this:

```
Select
    ProductID
    ,ProductName
    ,Supplier = CompanyName
From Products P         -- Aliased table
    Join Suppliers S    -- Aliased table
        on P.SupplierID = S.SupplierID
```

Notice that the Products table and Suppliers table is aliased, or renamed, with one letter aliases—P and S. If this is done, the P and S need to be used in the On clause as well.

I'm not a fan of this type of aliasing, although it's common. The only benefit is avoiding some typing, which is trivial. But the downside is severe—it leads to code that is much harder to read.

It's not so much a problem in small chunks of SQL like this one. However, in long, convoluted SQL, you'll find yourself wondering what the one-letter aliases mean, always needing to refer back to the From clause, and translate in your head.

The only time I use tables aliases is if the table name is extremely long. And then, I use table alias names that are understandable, just shortened.

24. Customer list by region

Answer

```
Select
    CustomerID
    ,CompanyName
    ,Region
From Customers
Order By
    Case
        when Region is null then 1
        else 0
    End
    ,Region
    ,CustomerID
```

Discussion

Once we have the Case expression set up correctly, you just need to create an Order By clause for it, and add the additional fields for sorting (Region and CustomerID).

If we had wanted to include the sorting field in the output , you could write this:

```
Select
    CustomerID
    ,CompanyName
    ,Region
    ,RegionOrder=
        Case
        when Region is null then 1
        else 0
    End
From Customers
```

```
Order By
    RegionOrder
    ,Region
    ,CustomerID
```

You would not need to repeat the case statement in the Order By, you can just refer to the alias - RegionOrder.

32. High-value customers

Answer

```
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
    OrderDate >= '20160101'
    and OrderDate  < '20170101'
Group by
    Customers.CustomerID
    ,Customers.CompanyName
    ,Orders.Orderid
Having Sum(Quantity * UnitPrice) > 10000
Order by TotalOrderAmount DESC
```

Discussion

If you tried putting this filter

```
and sum(Quantity * UnitPrice)  >= 10000
```

… in the where clause, you got an error. Aggregate functions can only be used to filter (with some exceptions) in the Having clause, not the Where clause.

48. Customer grouping

Answer

```
;with Orders2016 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
        OrderDate >= '20160101'
        and OrderDate  < '20170101'
    Group by
        Customers.CustomerID
        ,Customers.CompanyName
)
Select
    CustomerID
    ,CompanyName
    ,TotalOrderAmount
    ,CustomerGroup =
        Case
            when TotalOrderAmount between 0 and 1000 then 'Low'
            when TotalOrderAmount between 1001 and 5000 then 'Medium'
            when TotalOrderAmount between 5001 and 10000 then 'High'
            when TotalOrderAmount > 10000 then 'Very High'
        End
from Orders2016
Order by CustomerID
```

Discussion

(Note - there's a small bug in the above SQL, which we'll review in the next problem.)

The CTE works well for this problem, but it's not strictly necessary. You could also use SQL like this:

```
Select
    Customers.CustomerID
    ,Customers.CompanyName
    ,TotalOrderAmount = SUM(Quantity * UnitPrice)
    ,CustomerGroup =
        Case
            when SUM(Quantity * UnitPrice) between 0 and 1000 then 'Low'
            when SUM(Quantity * UnitPrice) between 1001 and 5000 then 'Medium'
            when SUM(Quantity * UnitPrice) between 5001 and 10000 then 'High'
            when SUM(Quantity * UnitPrice) > 10000 then 'Very High'
        End
```

```
From Customers
    Join Orders
        on Orders.CustomerID = Customers.CustomerID
    Join OrderDetails
        on Orders.OrderID = OrderDetails.OrderID
Where
    OrderDate >= '20160101'
    and OrderDate  < '20170101'
Group By
    Customers.CustomerID
    ,Customers.CompanyName
```

This gives the same result, but notice that the calculation for getting the TotalOrderAmount was repeated 5 times, including the 4 times in the Case statement.

It's best to avoid repeating calculations like this. The calculations will usually be quite complex and difficult to read, and you want to have them only in one place. In something simple, like Quantity * UnitPrice, it's not necessarily a problem. But most of the time, you should avoid repeating any calculations and code. An easy way to remember this is with the acronym DRY - “Don’t Repeat Yourself”. Here’s an article on the topic.