# Table Relationships in Power BI

Our data so far has come from a single table source know as **Flat File** very much the way that a single spread sheet would present its data, however it may be that our data source may come from multiple tables, very much the way that a database stores table of related data known as **Relational.**

Databases are most *efficien*t stores of data i.e. avoid repetition and so they are relational. However, Power BI is a reporting tool and so more importance may be put on the easy of data presentation and so you may choose to either model you data as *flat file*, or *relational* or a *mixture* of both!

As part of the data cleansing and data modelling process you will be able to start making these choices. When we import multiple tables, chances are we'll do some analysis using data from all those tables.

**Fully Flat File**

If you choose to use flat file then you will most likely need to Merge or Append data. Power BI provides good tools for this.

**Mixed**

Power BI also accommodates a mixture of flat and relational and it may be that is how your data will end up.

**Relational**

Relationships between those tables are necessary to accurately calculate results and display the correct information in your reports.   
  
Power BI Desktop makes creating those relationships easy. In fact, in most cases you won’t have to do anything, the autodetect feature does it for you.

This is when terms like **Star Schema** and **Fact** and **Dim** tables comes into play

### Star Schema

A Star schema is named such as the usual relation set up of the relationships is that some maininformation is stored in what is termed a **Fact** table and that this fact data is supported by one or more dimension **Dim** tables

Diagram

Description automatically generatedHere we see that the main data is the InternetSales **(Fact)** table and that this is supported by details such as Products, Customers, Dates, SalesTerritory **(Dim)** tables

**Fact** is the **many side**

**Dim** the **one side**

A picture containing diagram

Description automatically generated

In most data environments there may be multiple **Fact** tables that each have multiple **Dim** tables that may even overlap

**Important Note:**

Power BI Data model only supports *one* active-relationship

In this case you will have to create relationships between the Fact and Dim tables, or need to make changes to a relationship. Either way, it’s important to understand relationships and the **joins** that we use in Power BI Desktop and how to create and edit them. There are three **Join Types** used.

In Power BI we use the **USERELATIONSHIP** formula and the **CROSSFILTER** formulas to help us solve the issue of only *one* active-relationship

Note: This should all be very familiar to those of you with SQL knowledge.

### Relationship Types

1-Many One Trainer delivers Many Workshops

Many-Many Many Employees attend Many Workshops

1-1 An Employee has one set of Private Data and One set of   
 Public Data

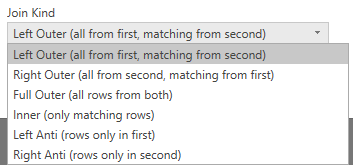
Join Types

There are 6 types of joins supported in Power BI as below, depends on the effect on the result set based on matching rows, each of these types works differently.  
  
These will be mostly familiar to T-SQL developers.

See here: <https://docs.microsoft.com/en-us/power-query/merge-queries-overview>

Graphical user interface, application, Teams

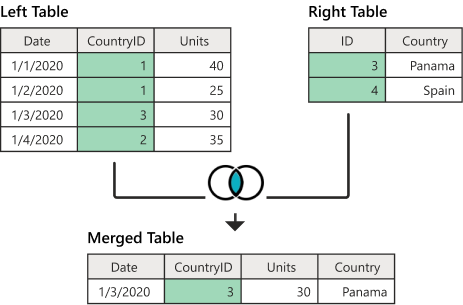
Description automatically generated

[](https://i0.wp.com/radacad.com/wp-content/uploads/2017/01/2017-01-05_21h18_39.png)

We won’t fully investigate all the variations, but Microsoft has some good details on each here:

**Inner:**

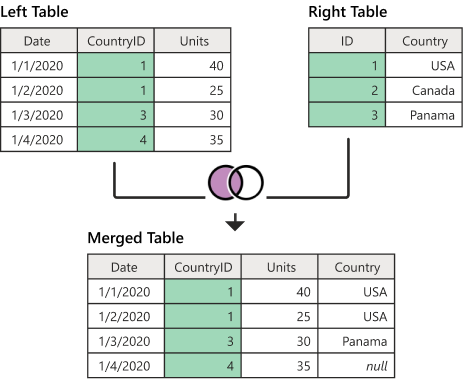
inner join, which brings in only matching rows from both the left and right tables



<https://docs.microsoft.com/en-us/power-query/merge-queries-inner>

**Left Outer:**

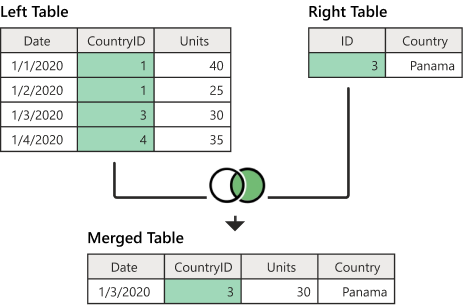
left outer join, keeps all the rows from the left table and brings in any matching rows from the right table.



<https://docs.microsoft.com/en-us/power-query/merge-queries-left-outer>

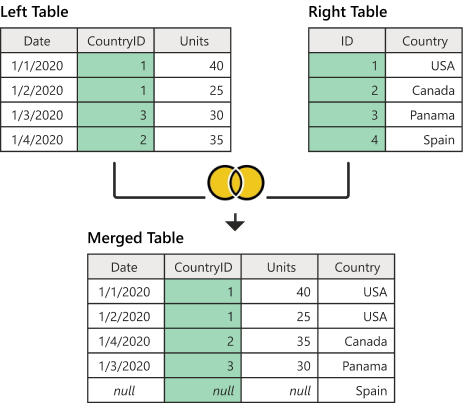
**Right Outer:**

right outer join, which keeps all the rows from the right table and brings in any matching rows from the left table.



<https://docs.microsoft.com/en-us/power-query/merge-queries-right-outer>

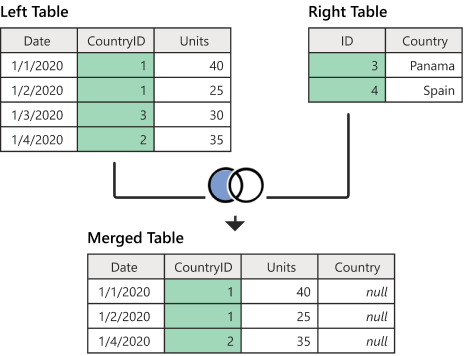
**Full Outer:**  *full outer* join, which brings in all the rows from both the left and right tables



<https://docs.microsoft.com/en-us/power-query/merge-queries-full-outer>

**Left anti:**

 left anti join, which brings in only rows from the left table that don't have any matching rows from the right table

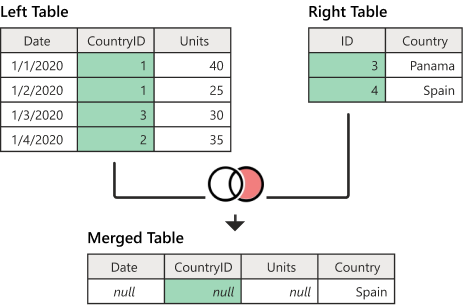


<https://docs.microsoft.com/en-us/power-query/merge-queries-left-anti>

**Right anti:**

right anti join, which brings in only rows from the right table that don't have

any matching rows from the left table.

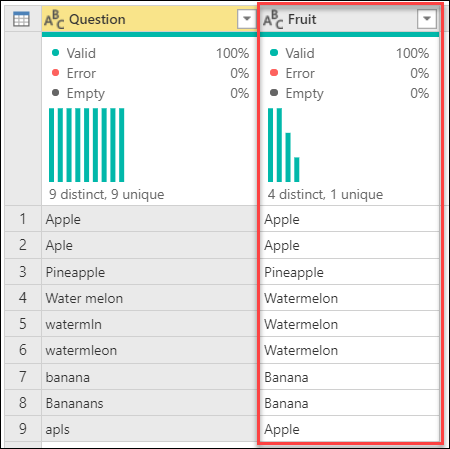


<https://docs.microsoft.com/en-us/power-query/merge-queries-right-anti>

**Fuzzy Merge:**

Fuzzy merge is a smart data preparation feature you can use to apply fuzzy matching algorithms when comparing columns, to try to find matches across the tables that are being merged.

You can enable fuzzy matching at the bottom of the **Merge** dialog box by selecting the **Use fuzzy matching to perform the merge** option button.

**For example**

The nine records reflect the survey submissions. The problem with the survey submissions is that some have typos, some are plural, some are singular, some are uppercase, and some are lowercase.

To help standardize these values, in this example you have a **Fruits** reference/look up table and match the submissions to the reference/look up values.

<https://docs.microsoft.com/en-us/power-query/merge-queries-fuzzy-match>

**Note:**

In T-SQL we use Join types like **Inner Join** , **Right Join**, **Left Join** and **Full Join,** however in Power BI we use the **CALCULATE** formula and the **RELATED** formula

Let’s look at this example :

Diagram

Description automatically generated

### ACTIVE RELATIONSHIP

Power Bi only supports one-active relationship at any one time

Graphical user interface, application

Description automatically generated

So we see here that the **OrderDate** is the Active relationship and **DueDate** and **ShipDate** are not currently active

Graphical user interface, application, Word

Description automatically generated

So we can easily get, say, the Sum of Orders based on OrderDate, (as this is the active relationship)

But what if we need the Sum of Orders based on ShipDate or DueDate ? (inactive relationship)

Here, we need to create a measure that uses DAX **USERELATIONSHIP formula:**

**Total Sales(Ship date) =**

**CALCULATE( SUM(‘Reseller Sales’[SalesAmount]), USERELATIONSHIP(‘Reseller Sales’[Ship Date],’Date’[DateKey])**

**)**

Logo, company name

Description automatically generated

Graphical user interface, text, application

Description automatically generated

### CROSS FILTER DIRECTION

Graphical user interface

Description automatically generated

Power BI offers three types of CrossFilter direction : **Single, Both, None**

Graphical user interface, table

Description automatically generated

**CROSSFILTER** Specifies cross filtering direction to be used in the evaluation of a DAX expression.

A picture containing text

Description automatically generated

CrossFilter is similar to USERELATIONSHIP , but it also takes a third argument for CrossFilterType : **Both/ None/OneWay**

In this direction set up it means that the Product (Dim table) can filter the

Reseller Sales (Fact table)

Graphical user interface

Description automatically generated

But what if we want the Reseller Sales (Fact table) to filter the

Product (Dim table)?

For example we want the number of products based on the Product table?

Table

Description automatically generated

*If we try this we will get the wrong data result (i.e. a cartesian error) i.e. 606 for all sales*

So, to solve this, we have two options:

Either we can set the direction to BOTH :

Graphical user interface, text, application

Description automatically generated

This will work, but it does come at a cost as it is resource intensive/performance issues (so ok for smaller models but not good for larger models).

OR we can use **CROSSFILTER** formula to create a measure

**Num of products (cross both ) =**

**CALCULATE( DISTINCTCOUNT(‘Product’[ProductKey]), CROSSFILTER(‘ResellerSales’[ProductKey],’Product’[ProductKey]), Both)**

**)**

Table

Description automatically generated

This is not as resource intensive or create performance issues!