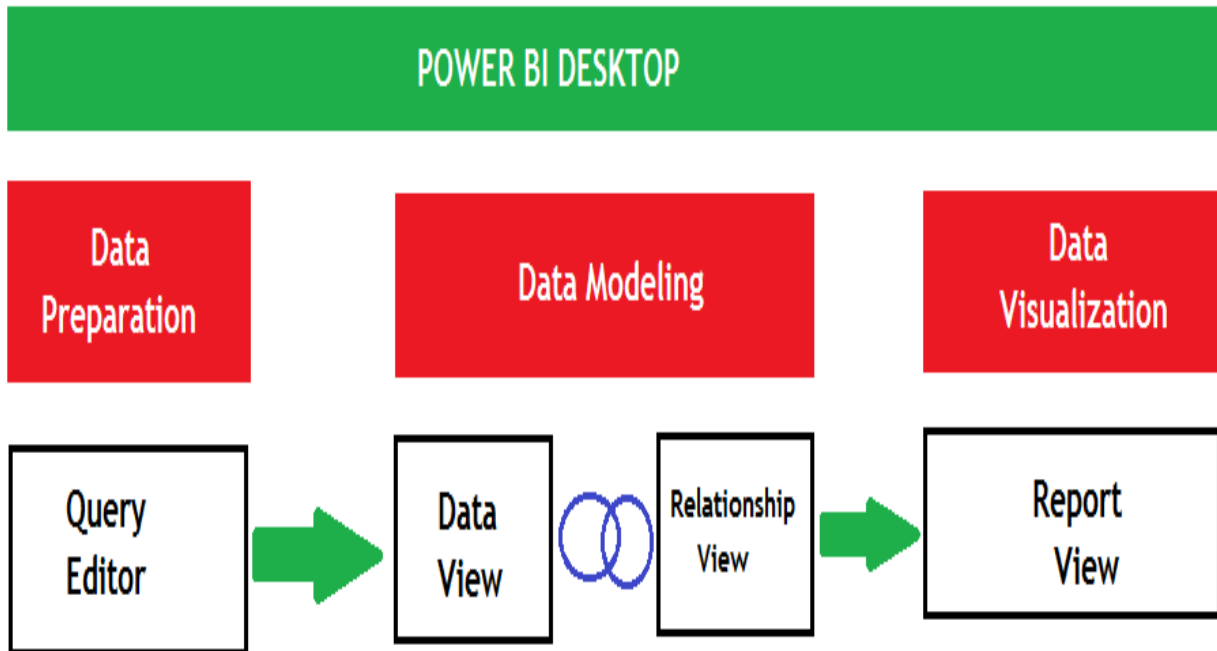


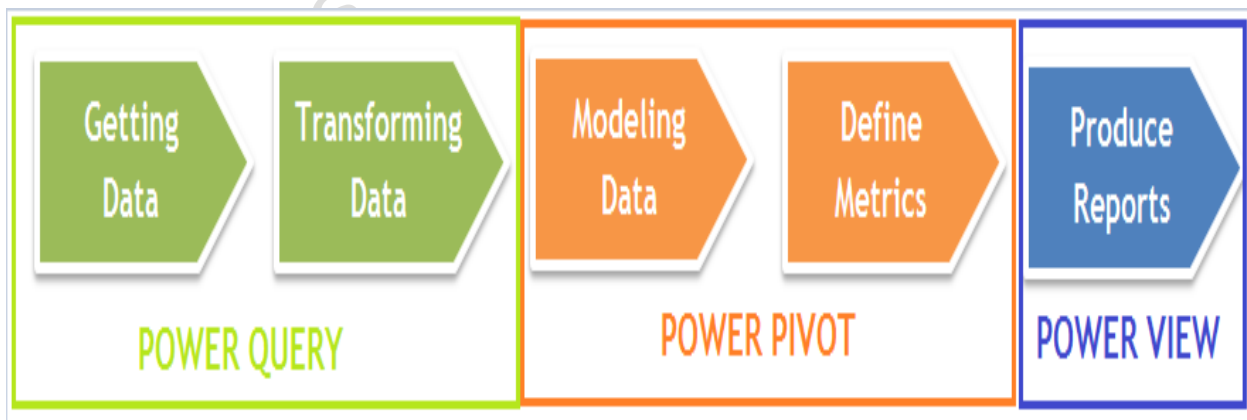
## Power BI Modeling

In this section, we will learn about Modeling Data, relationships, their definition, use, and how to create relationships, different types of relationships in Power BI. After this, we will learn how to manage relationships in Power BI. Managing relationships means adding, editing, and deleting relationships as well as making these active or inactive.

Below image explains different steps involved in Power BI Desktop.



## Power BI Desktop Steps



Power BI can connect to different data source, Transform or Prepare the Data using Power Query or Query Editor and then Load the transformed data into Power Pivot, to perform data modeling.

## Data Modeling

Modeling is to create a connection between different queries or tables, for that you create a relationship.

Often, you'll will use fields from more than one Query to create your Visualizations, and you'll need all of that data to work together. Modeling is how you get it there.

Usually in OLAP systems we stores discriptive information in Dimension Tables and Numeric information that is helpful for Analyzing the business will be stored in Fact Table. We always use the dimension table along with fact table to build reports or visualizations.

## Relationship

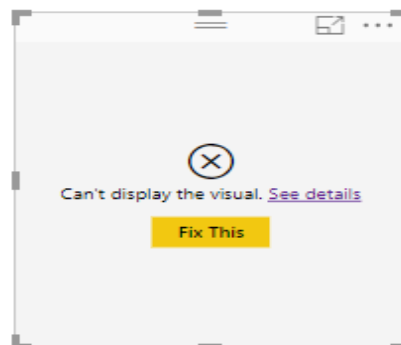
As the name suggests, Relationship in Power BI is used to define the connections or the relation between two or more tables. The relationship is used when we want to perform an analysis based on multiple tables. Relationship helps us to display the data and correct information between multiple tables. It is used to calculate the accurate results also.

A relationship between data sources enables Power BI to know how those tables relate to one another, allowing you to create interesting visuals and reports.

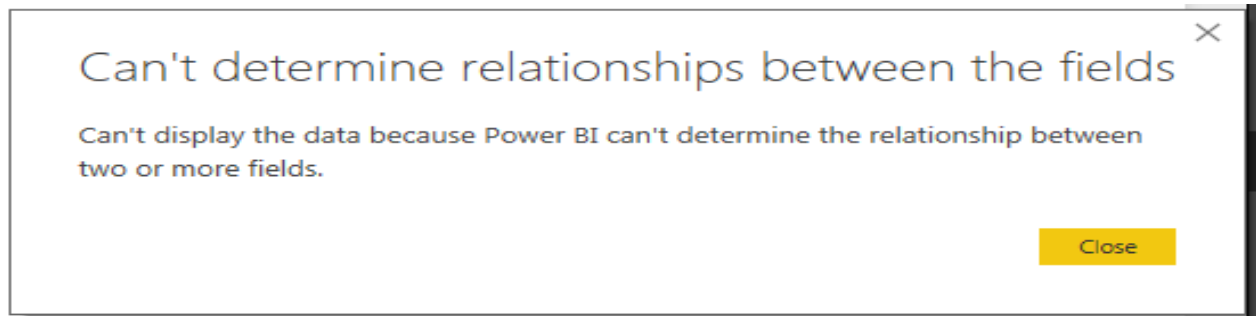
Power BI Desktop will create the relationships for you automatically. It is important to verify these auto-detected relationships to ensure accuracy.

## Need of Relationship

If we don't have relationship between two tables which are using in the same visualization for analysis, throws error as shown below or else it will show wrong results.



When you Click the **See details** hyperlink, you will be getting the message as shown in below image.



## Relationship Types / Cardinality in General

There are three types of relationships or Cardinality between tables

- One-to-One
- One-to-Many (or Many-to-One)
- Many-to-Many

### One-to-One

A row in table **A** can have only one matching row in table **B**, and vice versa.

Ex: Person and Passport Table

### One-to-Many (or Many-to-One)

This is the most common relationship type. In this type of relationship, a row in table **A** can have many matching rows in table **B**, but a row in table **B** can have only one matching row in table **A**.

One-to-Many relationships can also be viewed as Many-to-One relationships, depending on which way you look at it.

Ex: Customer and Sales Table

Sales table is the “many” and the Customer table is the “one”. Customer Table will have customer record only once but a Sales table will have multiple sales values for the same customer. When you see from Sales table to Customer Table it is Many-to-One relationship similarly when you see from Customer Table to Sales Table it is One-to-Many relationship.

### Many-to-Many

In a many-to-many relationship, a row in table **A** can have many matching rows in table **B**, and vice versa. Customer can buy many products and a Product can be purchased by many Customers.

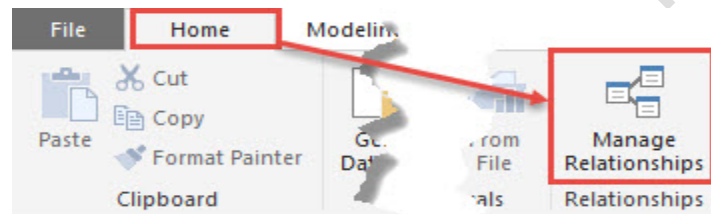
Ex: Customers and Products

Relationships cannot be built directly between tables that have a many-to-many relationship in Power BI.

A many-to-many relationship could be thought of as two one-to-many relationships, linked by an intermediary table.

The intermediary table is typically referred to as a “junction table” (also as a “cross-reference table”). This table is used to link the other two tables together. It does this by having two fields that reference the primary key of each of the other two tables.

The first thing you should do after importing data is to verify that all auto-detected relationships have been created correctly. From the modeling ribbon, select Manage Relationships.

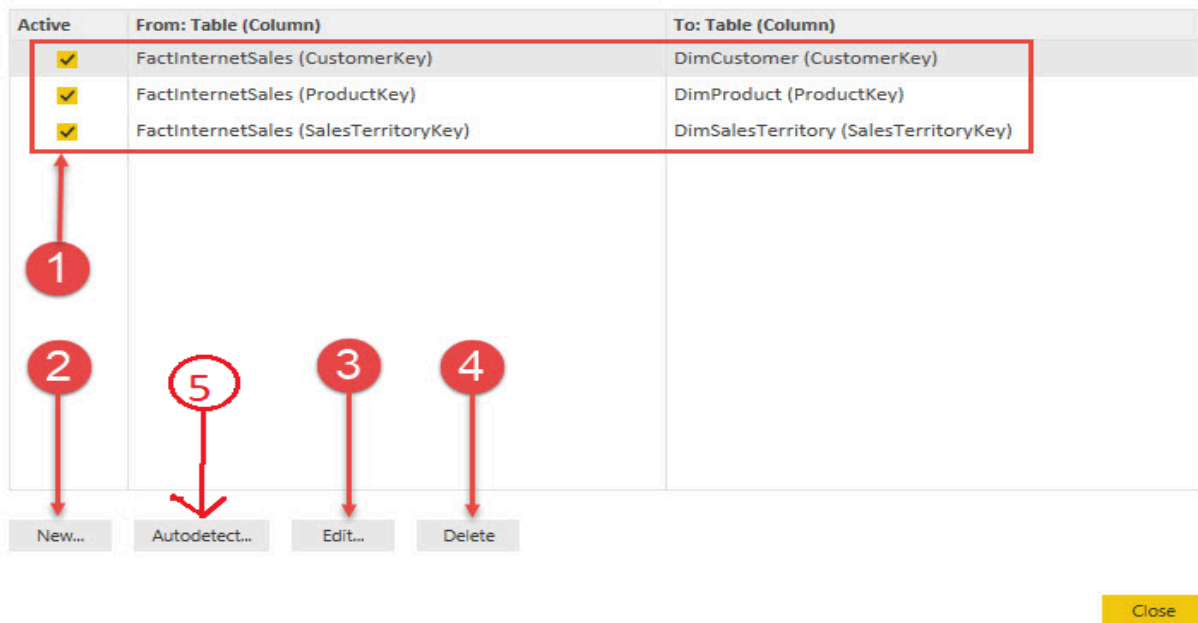


This will open up the Manage Relationships editor. The relationship editor is where you will go to create new relationships, Autodetect Relationship and edit or delete existing relationships. The relationship editor will be used to verify the relationships that were automatically created by Power BI Desktop.

Let's take a look at the Manage Relationships editor, in which you can manage or perform the following.

1. Current relationships in the data model
2. Create a new relationship
3. Edit existing relationships
4. Delete a relationship
5. Autodetect the relationship

## Manage relationships



First, you need to verify auto-detected relationships. The top half of the relationship editor gives you a quick and easy way to see what tables have relationships between them, what columns the relationships have been created on, and if the relationship is an active relationship.

### To create the new relationship

You can select Autodetect option in Manage relationship window to identify the relationships exists between the tables automatically by Power BI or else you can create manually using below steps.

- Click Manage Relationships. In Manage Relationships Window, click New. This opens the Create Relationship dialog / Window, where we can select the tables, columns, and any additional settings we want for our relationship.
- For the first table, select SALES\_DATA, then select the CustomerKey column in SALES\_DATA table. This is the many side of our relationship.
- For the second table, select CUSTOMERS\_DATA, then select the CustomerKey column. This is the one side of our relationship.
- Go ahead and click OK in the Create Relationship dialog and the Manage Relationships dialog. Click on close.

×

## Create relationship

Select tables and columns that are related.

SALES\_DATA

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	S
537	769	781	776	11793	1	100	
537	773	785	780	11233	1	100	
537	773	785	780	11635	1	100	

CUSTOMERS\_DATA

CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	Nam
11003	11	AW00011003	null	Christy	null	Zhu	
11005	22	AW00011005	null	Julio	null	Ruiz	
11015	301	AW00011015	null	Chloe	null	Young	

Cardinality

Many to one (\*:1)

Cross filter direction

Single

☒ Make this relationship active
 ☐ Apply security filter in both directions

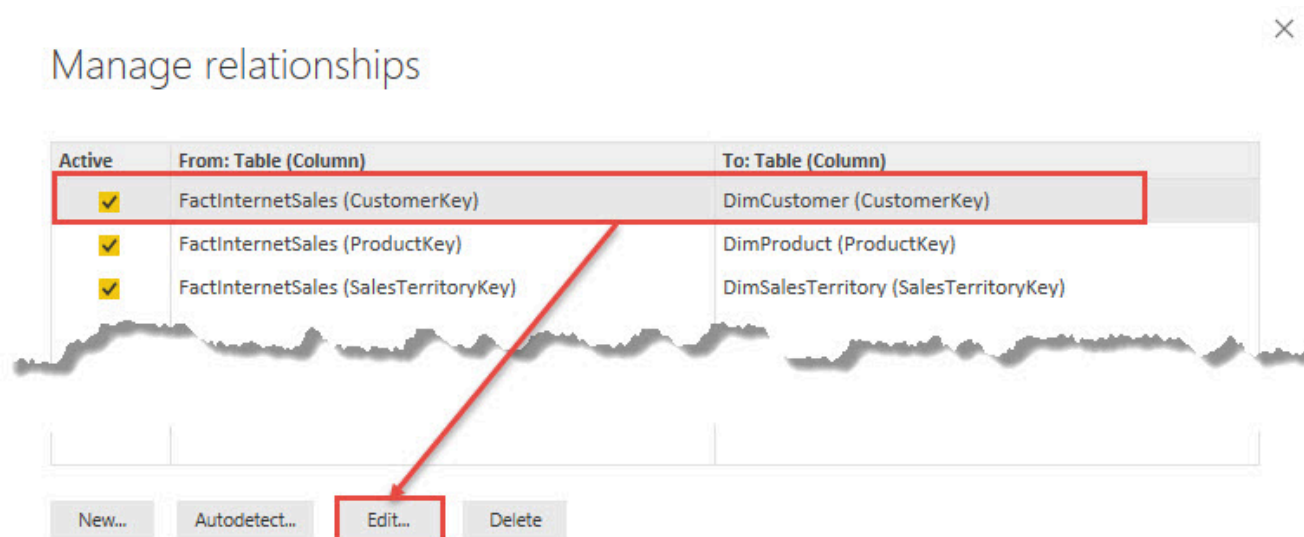
☐ Assume referential integrity

OK

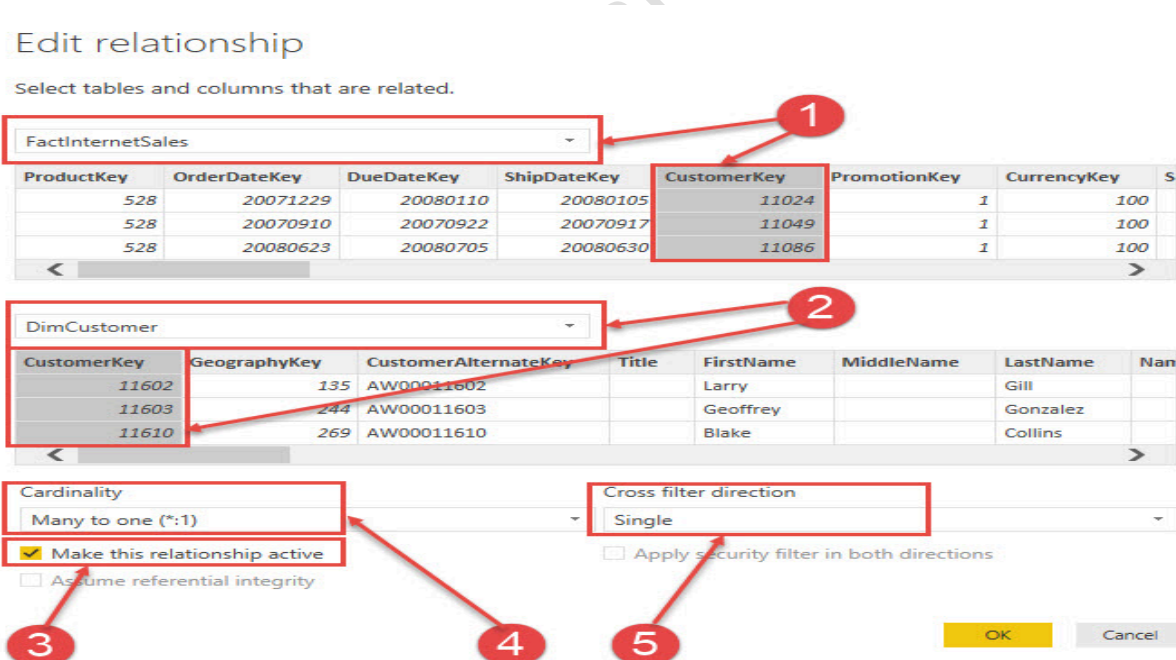
Cancel

### Editing relationships

Now, let's take a look at how to edit an existing relationship. In this example, you will edit the relationship between FactInternetSales and DimCustomer. To edit an existing relationship, select that relationship and then click on Edit.



Once you select Edit... you will receive a new dialog box, this is the Edit Relationship editor. In this view, you will see how to change an existing relationship, how to change a relationship to active or inactive, and the cardinality of the current relationship, this is also where you can change the cross filter direction if required.



There are five things we want to look at in the edit relationship window

1. This identifies the FactInternetSales table and the column that the relationship was built on.
2. This identifies the DimCustomer table and the column that the relationship was built on.

3. This checkbox identifies whether the relationship is active or inactive.
4. This is the current cardinality or relationship type between the two tables. Here we see that there is a many-to-one relationship between FactInternetSales and DimCustomer. Power BI does an excellent job of identifying the correct cardinality, but it is important to always verify that the cardinality is correct.
5. The cross filter direction can be single or both. *The one side of a relationship always filters the many side of the relationship, and this is the default behavior in Power BI.* If you filter values in dimension table corresponding values in fact table will be shown. The cross filter option allows you to change this behavior. Cross filtering will be discussed later in this chapter.

### Assume referential integrity settings in Power BI Desktop

When connecting to a data source using **DirectQuery**, you can use the Assume Referential Integrity selection to enable running more efficient queries against your data source. When we are using **DirectQuery**, every time when we drag the fields to visualizations, Power BI will run Queries on top of Source Databases. This feature has a few requirements of the underlying data, and it is only available when using **DirectQuery**.

Setting Assume referential integrity enables queries on the data source to use **INNER JOIN** statements rather than **OUTER JOIN**, which improves query efficiency.

**Edit Relationship**

Select tables and columns that relate to one another.

Orders

ID	ProductID	DepotID	CustomerID	Qty
1	ProductA	null	CustA	10
2	ProductB	DepotA	CustX	20
3	ProductC	DepotB	CustB	10

Products

ProductID	Price	Color
ProductA	9.99	null
ProductB	10	Blue
ProductC	20	Red

Cardinality: Many to One (\*:1) Cross filter direction: Single

☒ Make this relationship active

☐ Assume Referential Integrity [Learn More](#)

OK Cancel

### Requirements for using Assume referential integrity to Give Correct Values

This is an advanced setting, and is only enabled when connecting to data using **DirectQuery**. The following requirements are necessary for Assume referential integrity to work properly.



- Data in the **From** column in the relationship is never Null or blank.
- For each value in the **From** column, there is a corresponding value in the **To** column.
- In this context, the **From** column is the Many in a One-to-Many relationship, or it is the column in the first table in a One-to-One relationship.

### Example of using Assume referential integrity

The following example demonstrates how Assume referential integrity behaves when used in data connections. The example connects to a data source that includes an Orders table, a Products table, and a Depots table.

1. In the following image that shows the Orders table and the Products table, note that referential integrity exists between Orders[ProductID] and Products[ProductID]. The [ProductID] column in the Orders table is never Null, and every value also appears in the Products table. As such, Assume Referential Integrity should be set to get more efficient queries (using this setting does not change the values shown in visuals).

ID	ProductID	DepotID	CustomerID	Qty
1	ProductA	null	CustA	10
2	ProductB	DepotA	CustX	20
3	ProductC	DepotB	CustB	10

ProductID	Price	Color
ProductA	9.99	null
ProductB	10	Blue
ProductC	20	Red

2. In the next image, notice that no referential integrity exists between Orders[DepotID] and Depots[DepotID], because the DepotID is Null for some Orders. As such, Assume Referential Integrity should not be set.

<b>Orders table</b>	ID	ProductID	DepotID	CustomerID	Qty
	1	ProductA	null	CustA	10
	2	ProductB	DepotA	CustX	20
	3	ProductC	DepotB	CustB	10

<b>Depots table</b>	DepotID	City
	DepotA	Seattle
	DepotB	New York

3. Finally, no referential integrity exists between Orders[CustomerID] and Customers[CustID] in the following tables, the CustomerID contains some values (in this case, CustX) that do not exist in the Customers table. As such, Assume Referential Integrity should not be set.

<b>Orders table</b>	ID	ProductID	DepotID	CustomerID	Qty
	1	ProductA	null	CustA	10
	2	ProductB	DepotA	CustX	20
	3	ProductC	DepotB	CustB	10

<b>Customers table</b>	CustID	Name
	CustA	John Doe
	CustB	Jack Smith

### Setting Assume referential integrity

To enable this feature, select the checkbox next to Assume Referential Integrity as shown in the following image.

**Edit Relationship**

Select tables and columns that relate to one another.

Orders

ID	ProductID	DepotID	CustomerID	Qty
1	ProductA	null	CustA	10
2	ProductB	DepotA	CustX	20
3	ProductC	DepotB	CustB	10

Products

ProductID	Price	Color
ProductA	9.99	null
ProductB	10	Blue
ProductC	20	Red

Cardinality: Many to One (\*:1) Cross filter direction: Single

☒ Make this relationship active

☐ Assume Referential Integrity [Learn More](#)

OK Cancel

When selected, the setting is validated against the data to ensure there are *no Null or mismatched rows*. However, for cases with a very large number of values, the validation is not a guarantee that there are no referential integrity issues.

In addition, the validation occurs at the time of editing the relationship, and does not reflect any subsequent changes to the data.

### What happens if you incorrectly set Assume referential integrity?

If you set Assume Referential Integrity when there are referential integrity issues in the data, this will not result in errors. However, it will result in apparent inconsistencies in the data. For example, in the case of the relationship to the Depots table described above, it would result in the following.




A visual showing the total Order Qty would show a value of 40

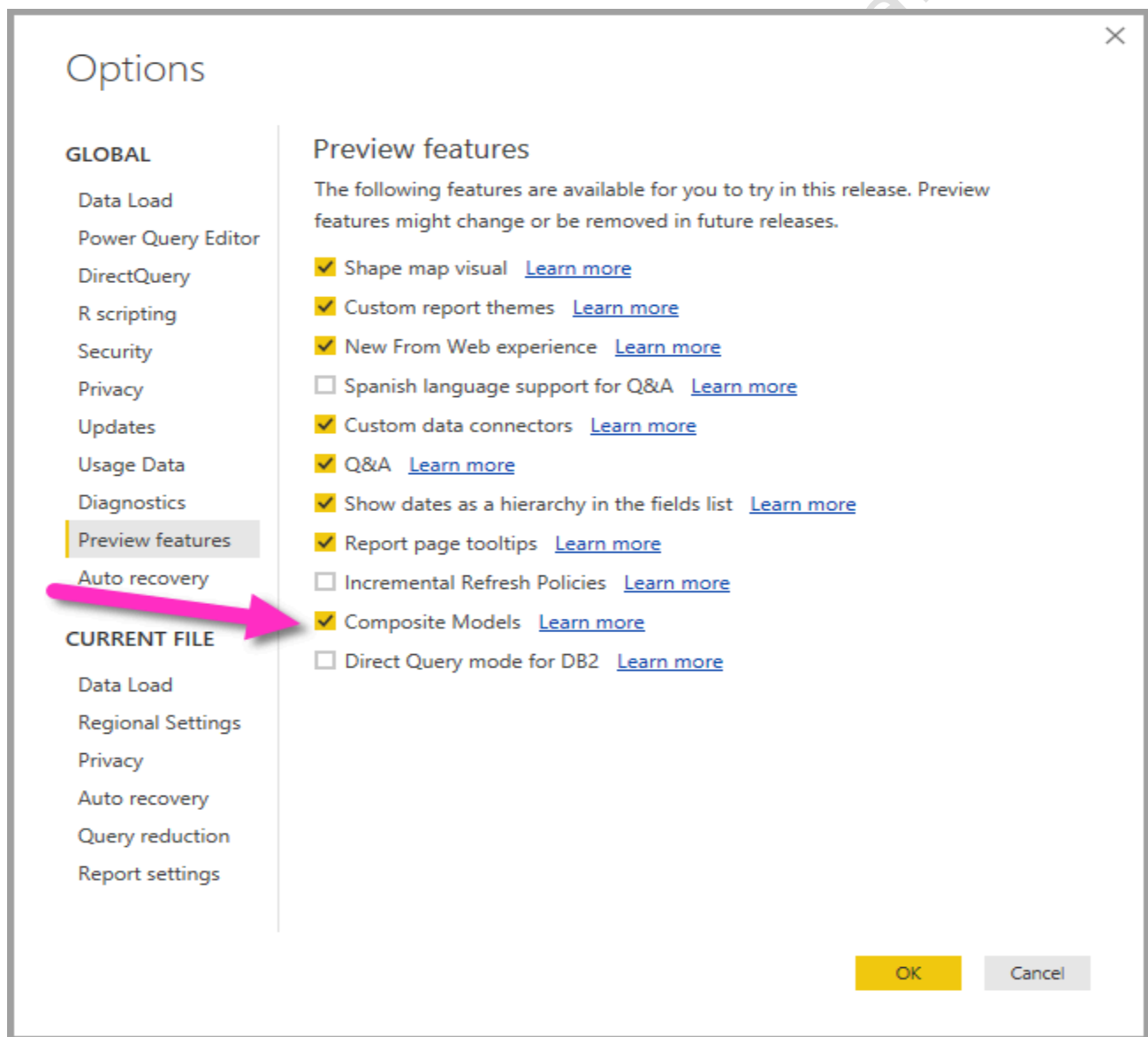
A visual showing the total Order Qty by Depot would show a total value of only 30, because it would not include Order ID 1, where DepotID is Null.

## Many-to-many relationships in Power BI Desktop - Preview July2018

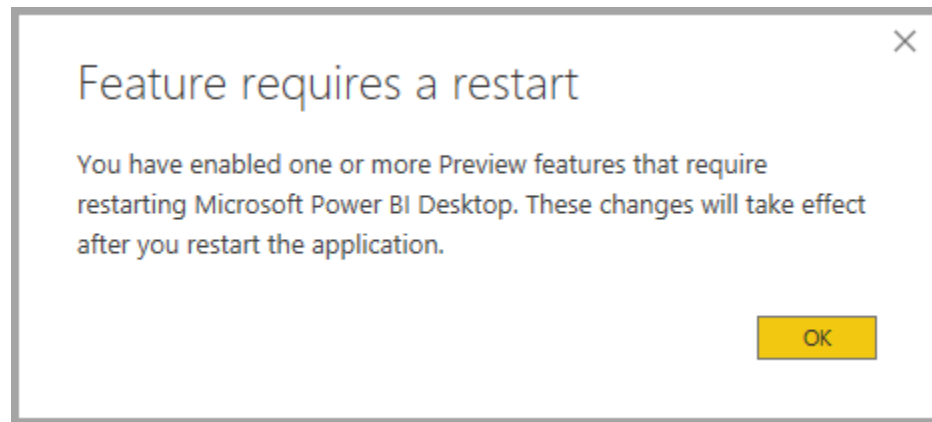
With the Many-to-many relationship feature in Power BI Desktop you can join tables using a cardinality of Many to Many, and create data models that contain multiple data sources easier and more intuitively.

### Enabling the many-to-many relationships preview feature

The many-to-many relationships feature is part of the composite models capabilities and is in Preview, and must be enabled in Power BI Desktop. To enable composite models, select File  Options and Settings  Options  Preview Features, then select the composite models checkbox.



You'll need to restart Power BI Desktop for the feature to be enabled.



### What many-to-many relationships solves

Prior to availability of many-to-many relationships, when defining a relationship between two tables in Power BI, at least one of the columns involved in the relationship had to contain unique values. In many circumstances though, no column in the table contained unique values.

For example, two tables may have a column containing the Country, but the values of Country were not unique in either table. To join between such tables, it was necessary to create a workaround such as introducing additional tables into the model that contained the necessary unique values. The feature many-to-many relationships provides an alternative approach, allowing such tables to be joined directly using a relationship with a cardinality of Many-to-many.