< Takaisin välilehdelle

✔ Tehty: Käy oppitunti läpi loppuun asti

Tasks

A **relationship** in **Power BI** simply defines how different tables are related to one another. For example, your customer table may be related to your sales table on the customer key column. You could argue that the building of relationships is the most important aspect of **Power BI Desktop**. It is this process, the building of relationships, that makes everything else work like magic in **Power BI.** The automatic filtering of visuals and reports, the ease with which you can author measures with **Data Analysis Expressions (DAX)**, and the ability to quickly connect disparate data sources are all made possible through properly built relationships in the data model.

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

Task 1 - Examine key elements of relationships

A **PBIX file** is a document created by **Power BI Desktop**. It contains queries, data models, visualizations, settings, and reports added by the user. The file is used to store all of the data visualization information. (You can create the PBIX file by selecting File → Save or File →Save As.)

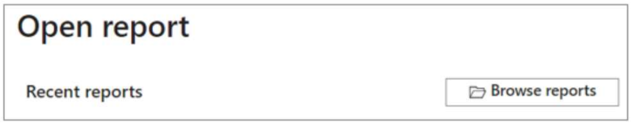
PBIX files are saved in the Office XML format, which is used to save .DOCX, .XLSX, and .PPTX files. The format stores a PBIX file as a collection of files and folders in a compressed zip package. If you need to manually view the contents you can rename the ".pbix" extension to ".zip" to create a .ZIP file and then extract the contents of the file.

Step 1: If not done earlier, click link to download **AdventureWorksDW.XLSX**, and save it to **C:\PBExams**.

Step 2: Click link to download **temp-data.XLSX**, and save it to **C:\PBExams**.

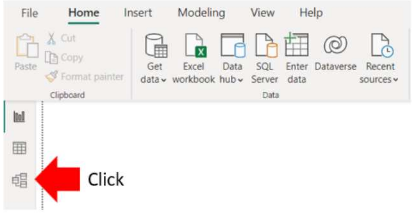
Step3: Click link to download **Lesson 4 - Building the Data Model.pbix**, and save it to **C:\PBExams**.

Step 4: Open **Power BI Desktop**. Click **File**, and select **Open Report**.

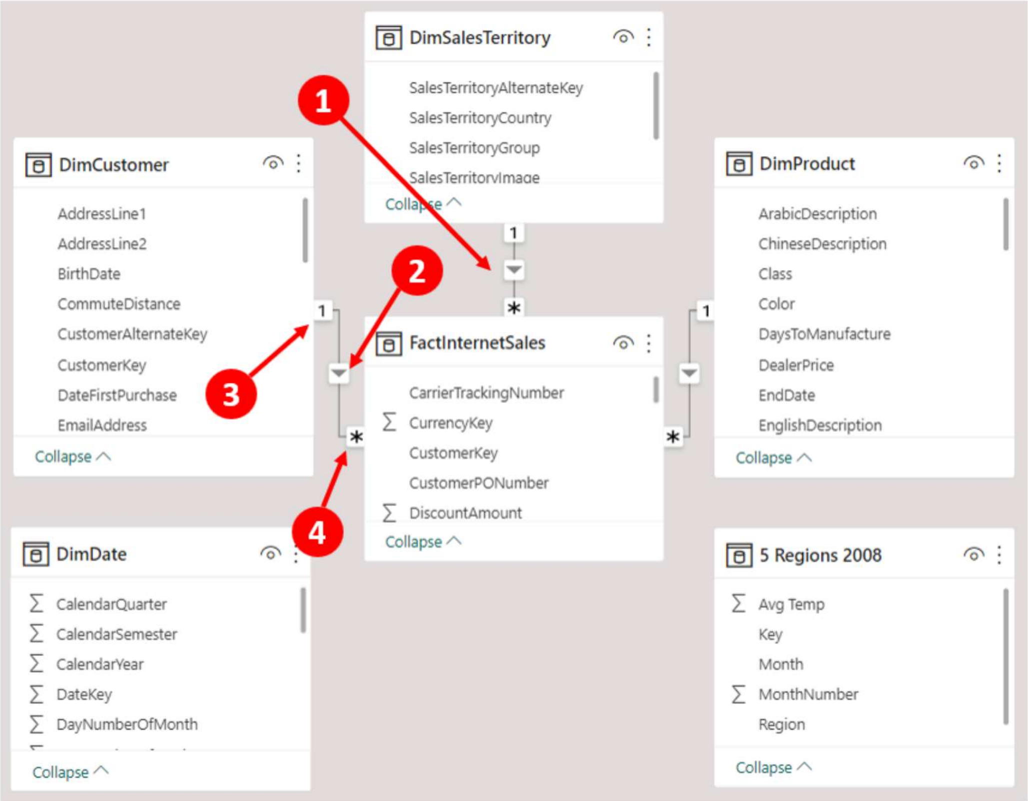


Step 5: Browse to ***C:\PBExams***, and select ***Lesson 4 - Building the Data Model.pbix.***

Step 6: Click the **Model** Icon:



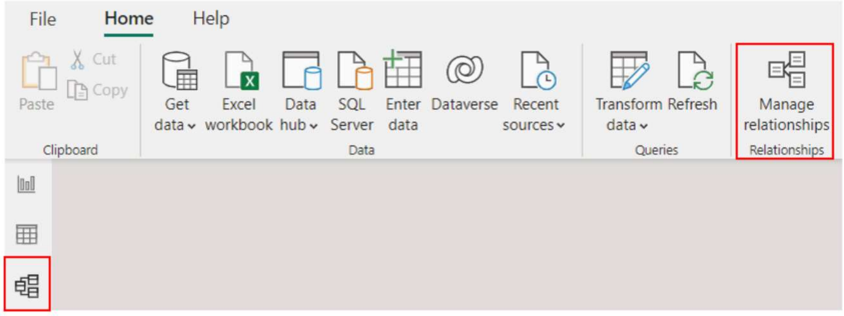
Step 9: The *Model View* opens. (You can scale it using slider at the bottom of the display.)



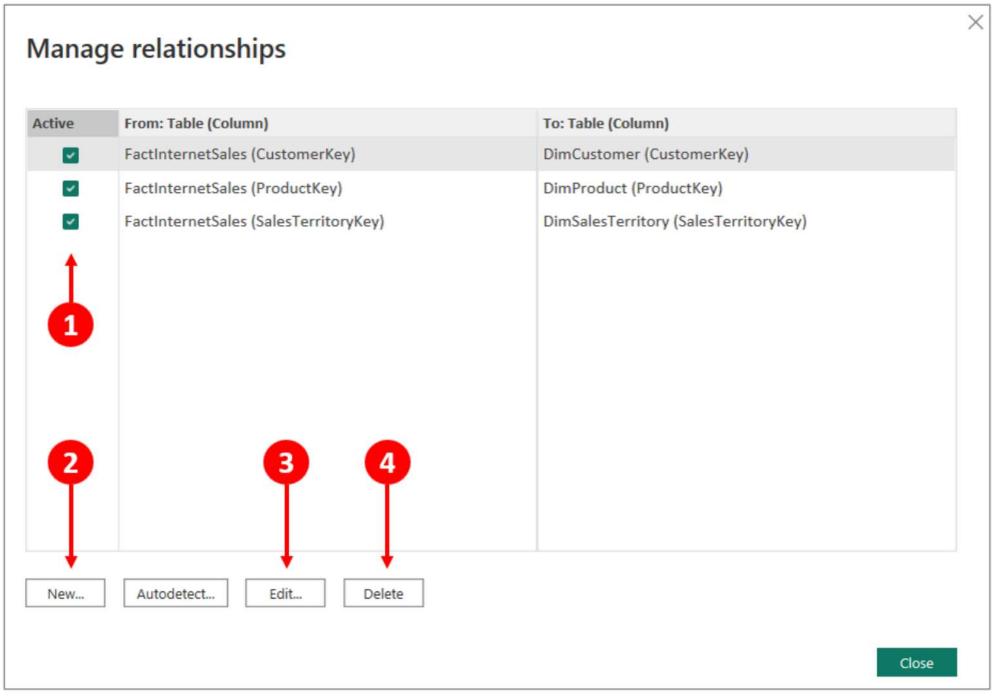
Following the numbering, let's take a closer look at each of the four items highlighted in the preceding Figure:

1. **Relationship**: The line between two tables represents that a relationship exists.
2. **Direction**: The arrow indicates in which direction filtering will occur.
3. **One side**: The **1** indicates the *DimCustomer* table as the *one side* of the relationship. This means the key from the *one side* of the relationship is always unique in that table.
4. **Many side**: The ***** indicates that the *FactInternetSales* table is the *many side* of the relationship. The key will appear in the sales table for each transaction; therefore, the key appears many times.

Step 7: Verify that all auto-detected relationships have been created correctly. From the Home ribbon, select **Manage relationships**. When in the *Report view*, *Manage relationships* will appear on the *Modeling* ribbon. When in the *Data view*, **Manage relationships** appears on the *T***able tools** ribbon, and when in the *Model view*, **Manage relationships** appears on the **Home** ribbon:



The **Manage relationships** editor opens.. The relationship editor is one of two places where you will go to create new relationships and edit or delete existing relationships. In this demo, the relationship editor will be used to verify the relationships that were automatically created by *Power BI Desktop*.



Manage relationships

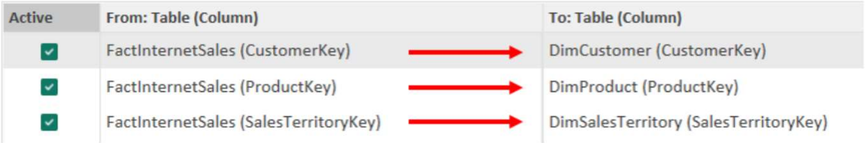
Active	From: Table (Column)	To: Table (Column)
☑	FactInternetSales (CustomerKey)	DimCustomer (CustomerKey)
☑	FactInternetSales (ProductKey)	DimProduct (ProductKey)
☑	FactInternetSales (SalesTerritoryKey)	DimSalesTerritory (SalesTerritoryKey)

Let's break down the editor using the numbered figure:

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

In figure above, 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.

There are currently three relationships, and all three relationships are currently active:



Active	From: Table (Column)	To: Table (Column)
☑	FactInternetSales (CustomerKey)	DimCustomer (CustomerKey)
☑	FactInternetSales (ProductKey)	DimProduct (ProductKey)
☑	FactInternetSales (SalesTerritoryKey)	DimSalesTerritory (SalesTerritoryKey)

The first row displays the relationship between the *FactInternetSales* table and the *DimCustomer* table. The relationship between these two tables was created automatically by *Power BI* on the *CustomerKey* column from each table. In this scenario, *Power BI* has correctly chosen the correct column names. However, if the relationship was created in error, then you would need to edit that relationship.

Task 2 - Editing relationships

In this task, you will edit the relationship between *FactInternetSales* and *DimCustomer*.

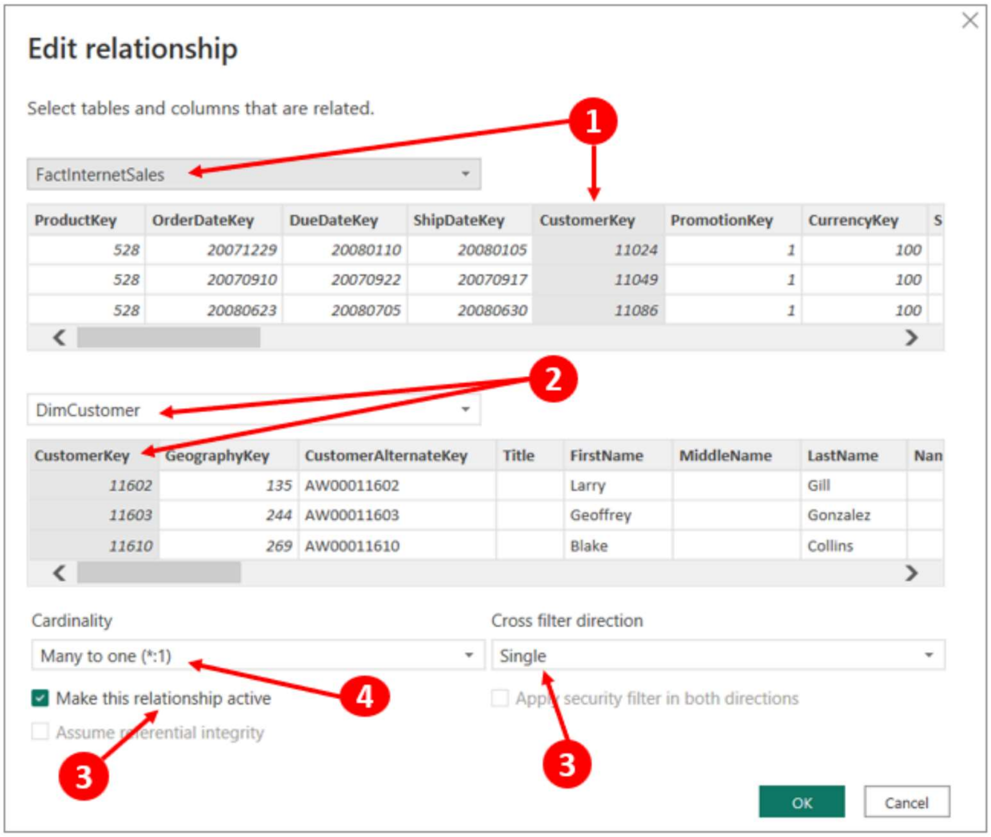
Step 1: In *Manage relationships* select *FactInternetSales* and then click on **Edit...** The **Edit relationship** editor opens.

Step 2: Change an existing relationship, change a relationship to active or inactive, and the cardinality of the current relationship:

1. This identifies the *FactInternetSales* table and the column that the relationship was built on, *CustomerKey*.
2. This identifies the *DimCustomer* table and the column that the relationship was built on, also *CustomerKey*.
3. This checkbox identifies whether the relationship is *active* or *inactive*.
4. This is the current cardinality 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. The cross-filter option allows you to change this behavior. Cross-filtering will be

discussed later in this chapter.

6. Click **OK**, and **Close** to close the editor.



Edit relationship

Select tables and columns that are related.

FactInternetSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	S
528	20071229	20080110	20080105	11024	1	100	
528	20070910	20070922	20070917	11049	1	100	
528	20080623	20080705	20080630	11086	1	100	

DimCustomer

CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	Nan
11602	135	AW00011602		Larry		Gill	
11603	244	AW00011603		Geoffrey		Gonzalez	
11610	269	AW00011610		Blake		Collins	

Cardinality: Many to one (*:1)

Cross filter direction: Single

☑ Make this relationship active

□ Assume referential integrity

□ Apply security filter in both directions

Note! If you need to change an existing relationship, then you would do that in the relationship editor. To change the column that a relationship has been created on, simply select a different column. It is important to point out that a relationship between two tables may only be created on a single column.

If you have a relationship that needs to be defined on multiple columns, also known as a composite key, then you would need to first combine those keys into a single column before creating your relationship. You saw how to combine columns in *Lesson 3*, *Data Transformation Strategies*.

Task 3 - Create a new relationship

In the previous task, you saw how to verify existing relationships, and even how to edit them. In this section, you are going to learn how to create a new relationship. There are six tables in the data model thus far, and **Power BI** automatically created a relationship for all the tables, except for two.

Your task is to create a relationship between the *FactInternetSales* and *DimDate* tables.

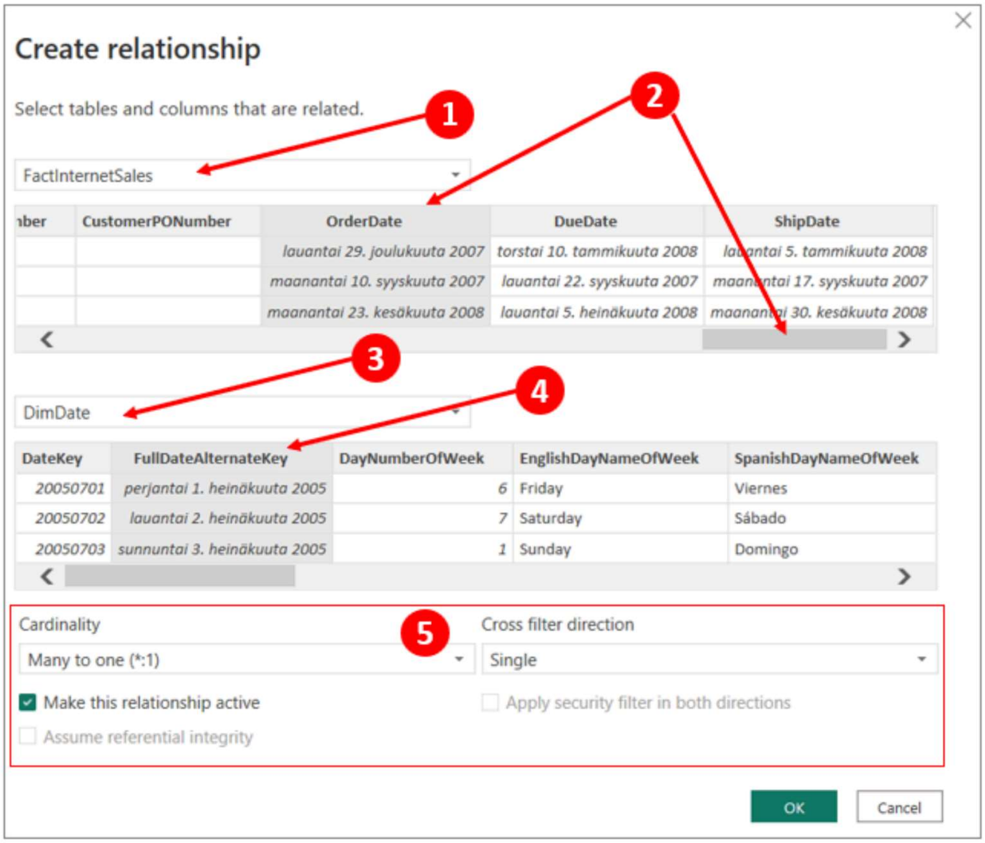
- The *FactInternetSales* table stores three different dates: *OrderDate*, *ShipDate*, and *DueDate*. There can be only one active relationship between tables in Power BI, and all filtering occurs through the active relationship. In other words, oe must choose which date to see your *total sales*, *profit*, and *profit margin calculations* on?
- Choose *OrderDate.* The relationship will be on the *OrderDate* column from the *FactInternetSales* table to the *FullDateAlternateKey* column in the *DimDate* table.

Step 1: Open **Manage relationships** from the *Home* ribbon. Click on **New...** to open the **Create relationship** editor.

Step 2: Create a relationship from the *OrderDate* column in *FactInternetSales* to the *FullDateAlternateKey* column in *DimDate*:

1. Select *FactInternetSales* from the list of tables in the dropdown.
2. Select *OrderDate* from the list of columns; use the scroll bar to scroll all the way to the right.

3. Select *DimDate* from the next drop-down list.
4. Select *FullDateAlternateKey* from the list of columns.
5. *Cardinality*, Cross filter direction, and whether the relationship is active or inactive are updated automatically by Power BI; remember to always verify these items.
6. Click **OK**, and **Close** to close the editor.



Create relationship

Select tables and columns that are related.

FactInternetSales

ıber	CustomerPONumber	OrderDate	DueDate	ShipDate
		lauantai 29. joulukuuta 2007	torstai 10. tammikuuta 2008	lauantai 5. tammikuuta 2008
		maanantai 10. syyskuuta 2007	lauantai 22. syyskuuta 2007	maanantai 17. syyskuuta 2007
		maanantai 23. kesäkuuta 2008	lauantai 5. heinäkuuta 2008	maanantai 30. kesäkuuta 2008

DimDate

DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek
20050701	perjantai 1. heinäkuuta 2005	6	Friday	Viernes
20050702	lauantai 2. heinäkuuta 2005	7	Saturday	Sábado
20050703	sunnuntai 3. heinäkuuta 2005	1	Sunday	Domingo

Cardinality: Many to one (*:1)

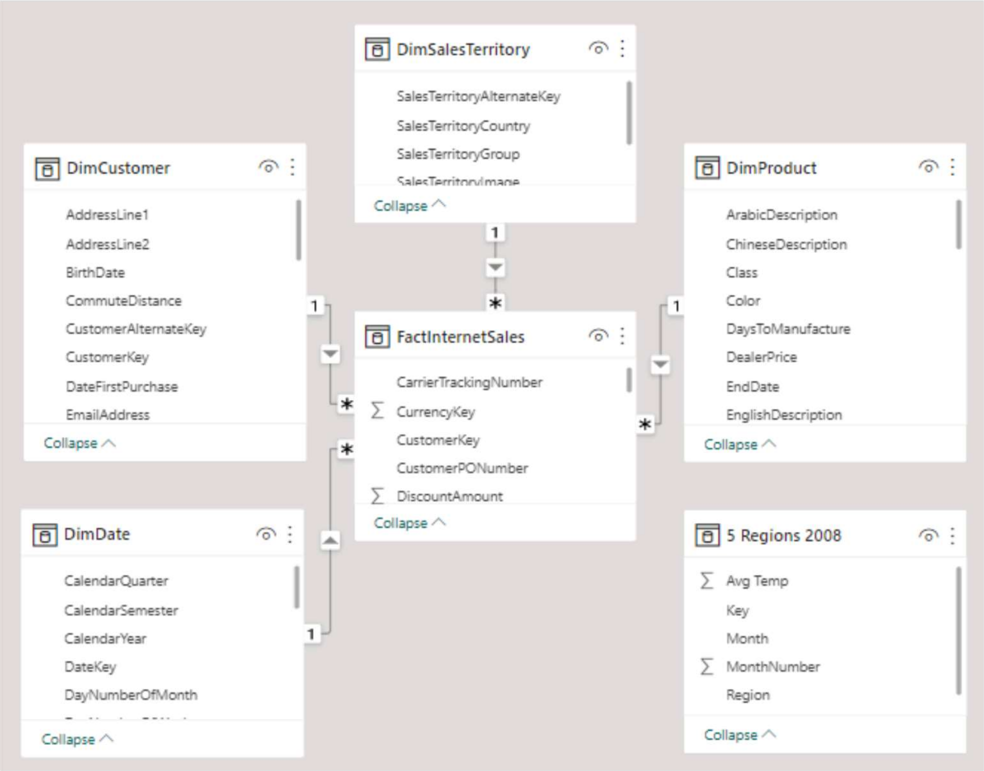
Cross filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity

The Model is now as shown below.

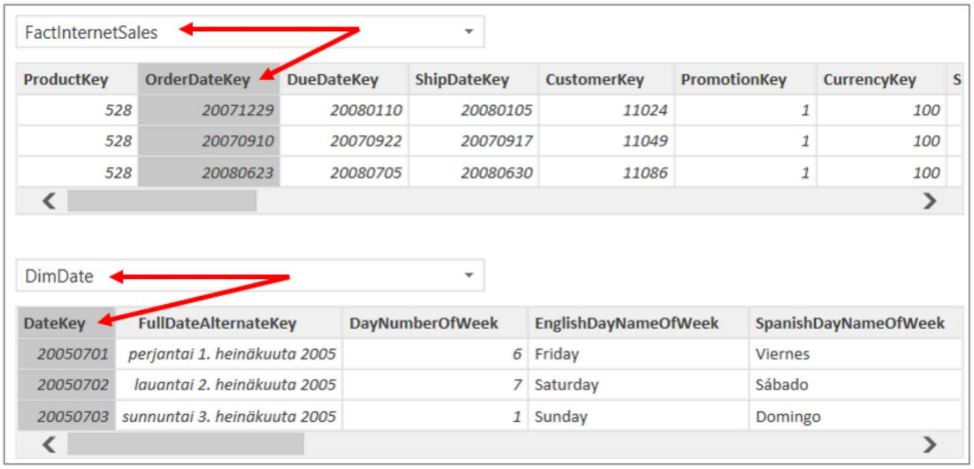


Task 4 - Create a relationship on the date key

You may have noticed that the previous step used the **actual date columns** from each table instead of the **date keys**. This is because most *Power BI* models will not contain a date key. However, if you are retrieving your data from a relational database or data warehouse, then a date key will most likely exist on both tables, and the relationship can be created on the date key.

In data modeling, we generally store dates as integer values. More specifically, you can store the date as a **smart key**, for example, 20200706. This type of integer value is called a smart key because the date is stored as an integer, but you can still derive the date from the integer value. For example, the first four characters are the year, the next two represent the month, and the final two represent the day number of the month. Storing your dates as integer values is a best practice and will help save space in the model.

In Figure below, you see an example of creating a relationship on the date keys in each table:



FactInternetSales

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	S
528	20071229	20080110	20080105	11024	1	100	
528	20070910	20070922	20070917	11049	1	100	
528	20080623	20080705	20080630	11086	1	100	

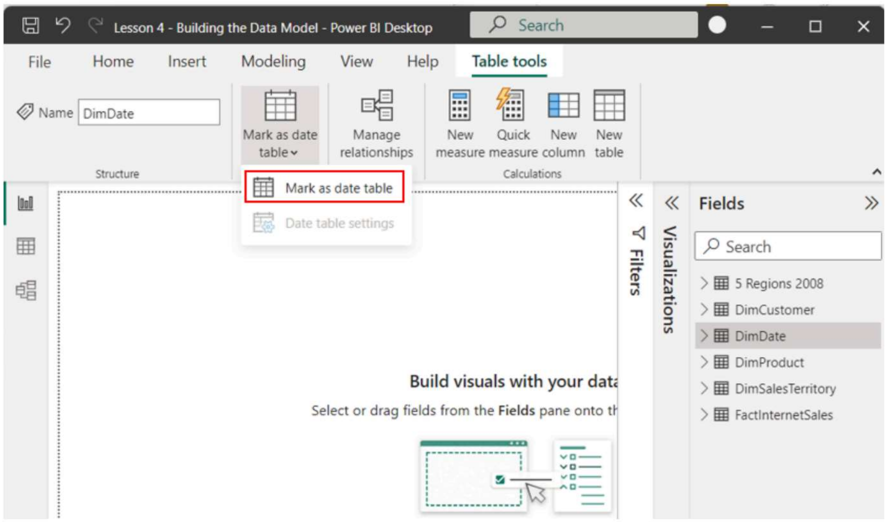
DimDate

DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek
20050701	perjantai 1. heinäkuuta 2005	6	Friday	Viernes
20050702	lauantai 2. heinäkuuta 2005	7	Saturday	Sábado
20050703	sunnuntai 3. heinäkuuta 2005	1	Sunday	Domingo

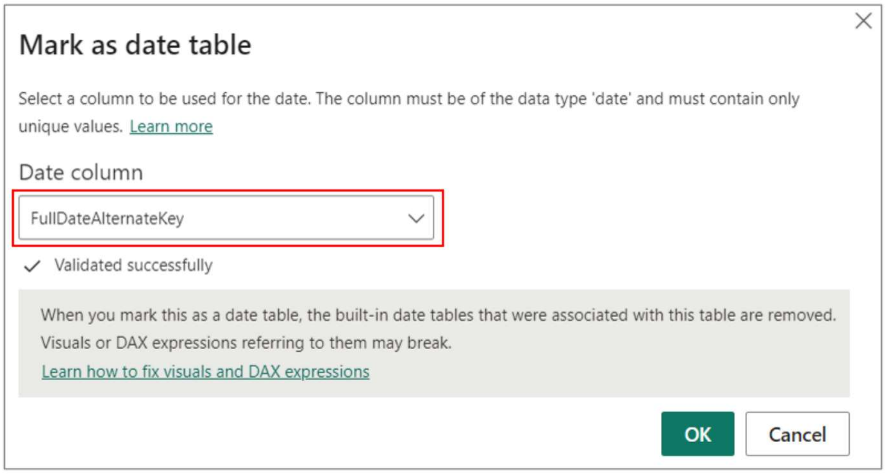
The **date table** in the Power BI Data model is important for developing time intelligence calculations. When you define your relationship on the date key, rather than the date column, it is important to also mark your date table as a date table! If this step is not performed, some built-in time intelligence functions may not work as expected.

Time intelligence is discussed in further detail in *Lesson 5*, *Leveraging DAX*.

Step 1: To mark a table as a **date table,** select the **Date** table, go to the **Table tools** ribbon and select **Mark as date table**.



Step 2: Once *Mark as date table* is selected, you will then be prompted to select the *Date column* from your date table:



Mark as date table

Select a column to be used for the date. The column must be of the data type 'date' and must contain only unique values. Learn more

Date column

FullDateAlternateKey

✓ Validated successfully

When you mark this as a date table, the built-in date tables that were associated with this table are removed. Visuals or DAX expressions referring to them may break.
Learn how to fix visuals and DAX expressions

Step 3: Click **OK**.

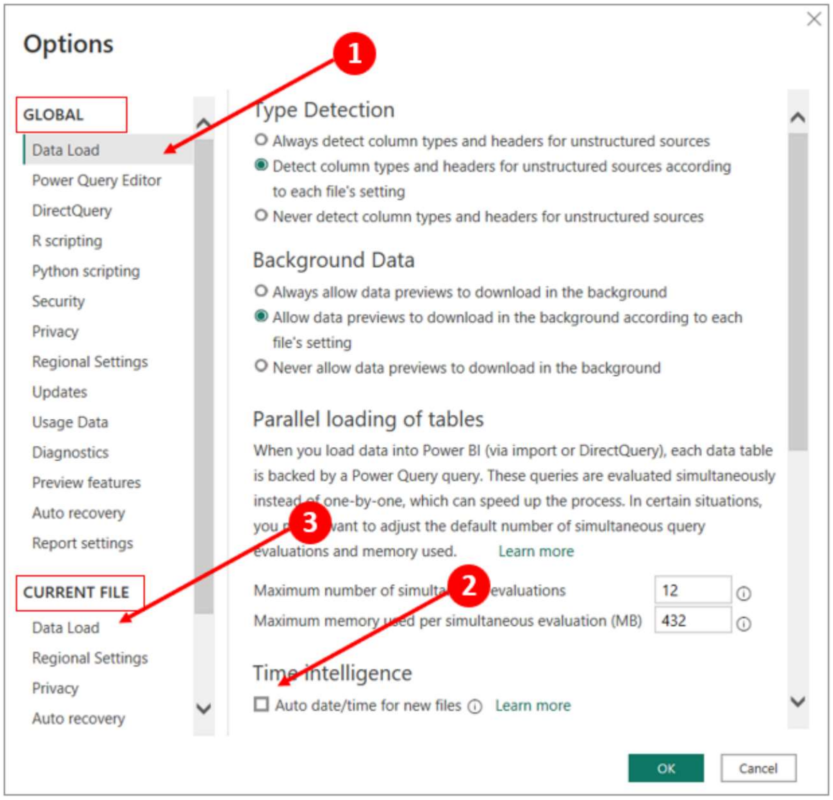
Power BI will automatically limit the list of available options to columns that are set to a date type and have only unique values.

Task 5 - Disable automatically created date tables

Power BI automatically creates hidden date tables for each date or date/time field in your data model. Depending on the number of date fields in your data model, this can create a lot of extra objects and will consume more memory in your data model! The date tables created by Power BI can be disabled from the *Options* menu. This functionality is automatically enabled on your current Power BI file and will be enabled on future files. Therefore, you will need to disable the auto date/time capability in two separate locations from the Options menu.

Step 1: Open the **Options** menu, in Power BI Desktop: **File | Options and Settings | Options**. Once the **Options** window has been opened, complete the following steps:

1. Under GLOBAL, choose **Data Load**.
2. Uncheck *Auto date/time for new files.*
3. Under CURRENT FILE, choose **Data Load** and uncheck *Auto date/time*.
4. Click **OK**.



Options

Type Detection

○ Always detect column types and headers for unstructured sources
● Detect column types and headers for unstructured sources according to each file's setting
○ Never detect column types and headers for unstructured sources

Background Data

○ Always allow data previews to download in the background
● Allow data previews to download in the background according to each file's setting
○ Never allow data previews to download in the background

Parallel loading of tables

When you load data into Power BI (via import or DirectQuery), each data table is backed by a Power Query query. These queries are evaluated simultaneously instead of one-by-one, which can speed up the process. In certain situations, you may want to adjust the default number of simultaneous query evaluations and memory used. Learn more

Maximum number of simultaneous evaluations 12

Maximum memory used per simultaneous evaluation (MB) 432

Time intelligence

□ Auto date/time for new files ⓘ Learn more

End-of-Exercise

Olet suorittanut 100 % oppitunnista

100%

◄ Building the Data Model

Siirry...

Exercise 9 - Working with complex relationships ►