EXPLORE AI ACADEMY

Data models in Power BI

Relationships and cardinality in Power BI

Data models in Power BI

Defining relationships in Power BI

As in relational databases, relationships are essential in Power BI for **connecting and associating data from different tables** in a data model.

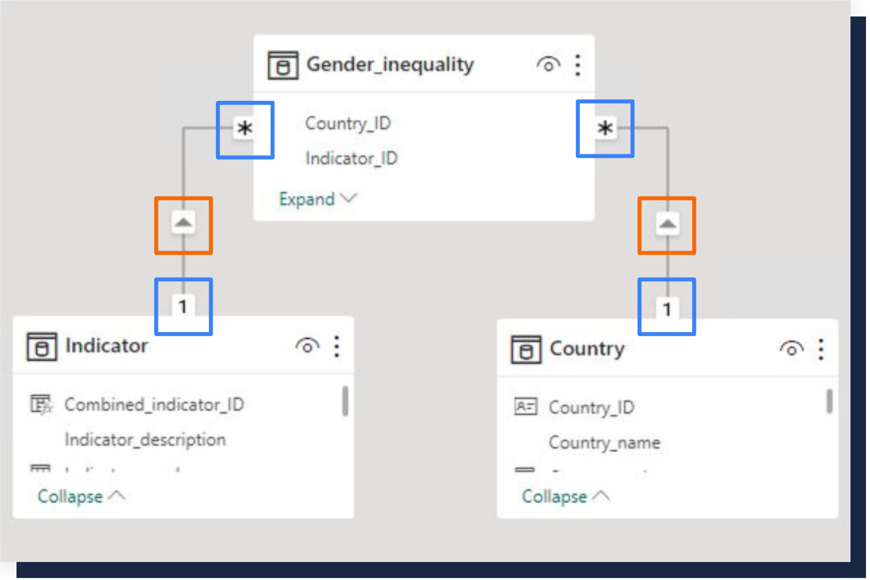
When we define relationships in Power BI, we need to consider the following:

Cardinality

The **number of related records** in one table that can be linked to a single record in another table.

Directionality

Defines **how filtering** in one table affects another table.



Cardinality

> We can have the same type of relationships in Power BI that we would normally have in a relational database, including **one-to-one**, **one-to-many**, **many-to-one**, and **many-to-many relationships**.

"Cardinality" is a mathematical term that refers to the **number of elements in a given set**. Similarly, in a database it represents the **number of related records in one table** that can be **linked to a single record** in another table.

The **four main types of cardinality** in Power BI are:

One-to-one (1:1)	One-to-many (1:*)	Many-to-one (*:1)	Many-to-many (*:*)
1 — 1	1 — *	* — 1	* — *

We can have all of these **cardinalities in a single data model**. However, **one-to-many cardinalities are more common** in structured data models such as the star schema.

Directionality

> Unlike SQL, Power BI introduces the concept of "cross-filter direction", also known as directionality, which **defines how filtering in one table affects another table**.

Single

Filtering in the **"One" table** (the table with a single record on one side of the relationship) **impacts the "Many" table**, allowing us to filter related records.

This is often called a **single relationship**.

Both

Filtering in **both tables** affects each other, enabling two-way filtering. Although useful, it introduces complexity.

This is often called **bi-directional relationships**. In some cases, we **may want both tables to impact each other's filtering**. **One-to-one** cardinalities can only ever be bi-directional relationships.

Many-to-many, **one-to-many**, and **many-to-one** cardinalities can **either be single or bi-directional**.

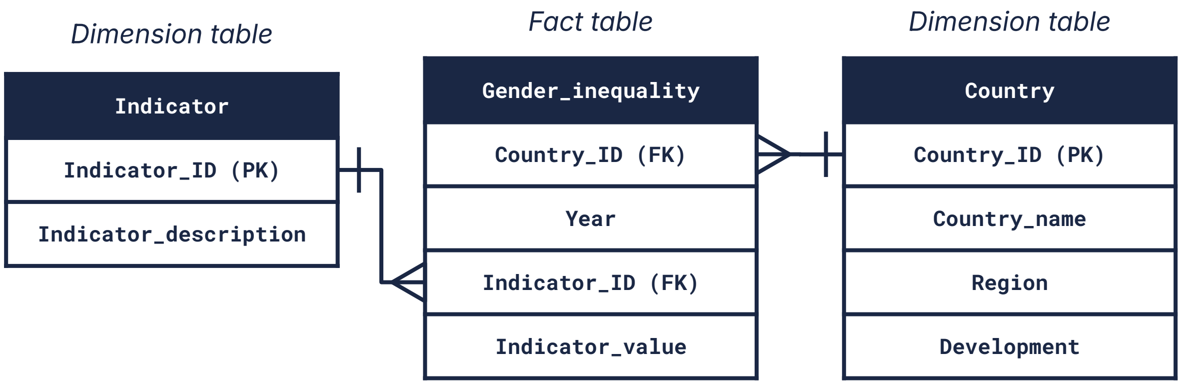
Data models in Power BI

Dataset example

UN SDG 5: Gender equality

This database hosts gender-disaggregated data and statistics covering **geographic location**, **education**, **health**, access to **economic opportunities**, public life and **decision-making**, and **agency**.

We want to look at **gender parity (or equality)** and inequality across different geographical areas. For a simple star schema data model, we simply break up the dataset into three tables:



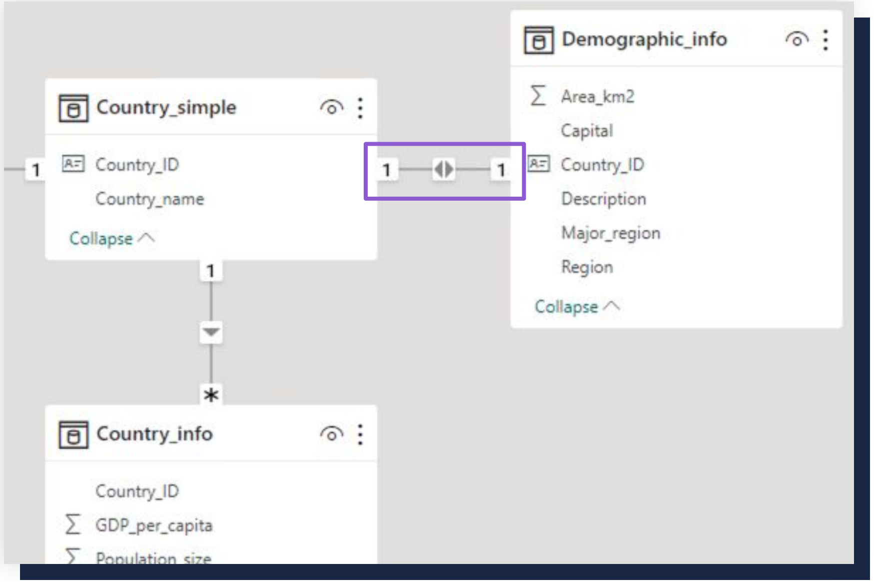
This dataset is at an **appropriate granularity** for the stories we want to tell and the **star schema** is **relatively simple**. However, we'll see how this **simple design may limit what we can visualise and how we can filter** across our tables.

One-to-one (1:1)

> In a one-to-one cardinality, **each record in the first table** is associated with **one and only one record in the second table**, and vice versa.

As an example, we could move all of our geographical information per country (that doesn't change over time) to a **dedicated table called `Demographic_info`** that we can **enrich** with additional information.

This new table would then have a **one-to-one cardinality** with the `Country_simple` table because **one country is related to single description**, **region**, **area**, etc.



One-to-many (1:*)

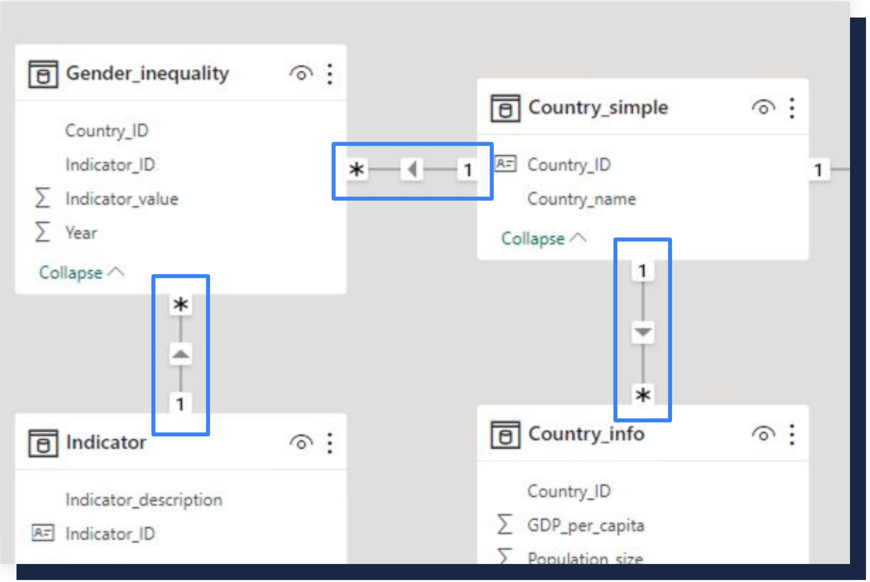
In a one-to-many cardinality, **each record in the first table can be related to multiple records in the second table**, but each record in the second table is associated with one record in the first table.

Here we have **three** **one-to-many** relationships:

01. A **single country** can have **multiple records for different years and inequality indicators**, but each record in `Gender_inequality` is related to one specific country.

02. A **single country** can have **multiple records for different years and GDP per capita**, **population size**, etc. combinations, but each record in `Country_info` is related to one specific country.

03. A **single indicator** can have **multiple records for different years and countries**, but each record in `Gender_inequality` is related to one specific indicator.

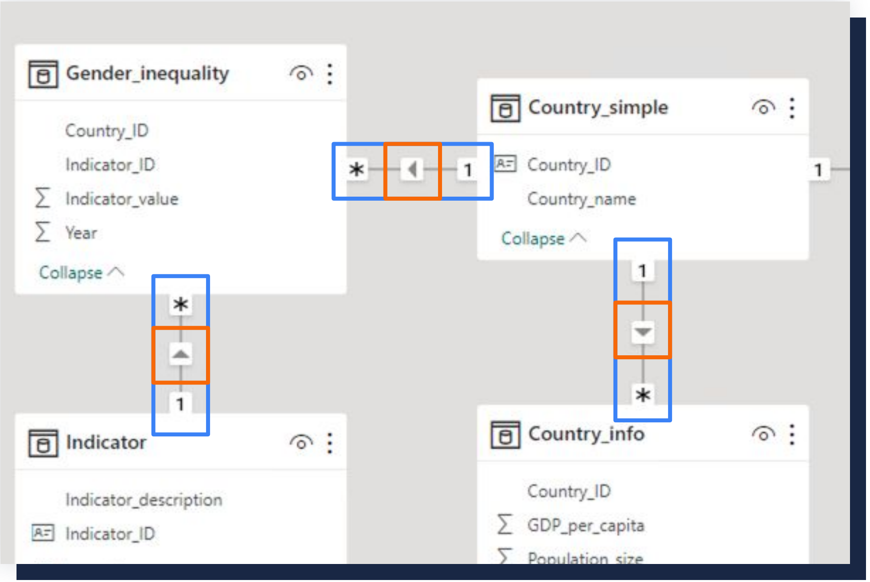


Many-to-one (∗:1)

> On the contrary, in a many-to-one cardinality, **each record in the first table is associated with a single record in the second table**, but each record in the second table can be linked to multiple records in the first table.

Considering the definitions of the many-to-one and one-to-many cardinalities, we see some similarities. The **difference is the direction of the relationship**.

So, while in our example all of our relationships are currently one-to-many (with a single direction), we **could change the directions to be bi-directional**. This would allow **filtering in both directions** (even in the direction a many-to-one cardinality would filter) but it won't change the cardinality.



Many-to-many (*:*)

In a many-to-many cardinality, **multiple records in the first table can be associated with multiple records in the second** table.

Since the `Country_simple` table **only includes two features**, namely `Country_ID` and `Country_name`, we could easily get rid of the table and simply move `Country_name` to the `Country_info` table.

Now we'll need a **many-to-many** cardinality since both `Gender_inequality` and `Country_info` have **multiple instances of the same country record**.

How to choose the cardinality and directionality

Understand the data	Start by **understanding the nature** of the dataset. **Identify the relationships** between tables and how they should logically relate.
Consider the story we want to tell	What is the **purpose** of the visualisations or report we're building? How do we want **visualisations to respond** to each other and what should the **interactions** look like?
Avoid unnecessary bi-directionality	While bi-directional relationships can be useful, they can make our **data model more complex and harder to manage**.
Test and iterate	It's common to **refine** relationships as we are building and testing our visualisations and reports. We always **test** our data model to **ensure filtering and calculations work** as expected.

In summary, the cardinality and directionality we choose should align with the **data structure**, **visualisation and reporting needs**, and the **logical flow of data** in our model.

Challenges in relationships

Creating and managing relationships in Power BI is a critical part of data modelling and reporting. Unsurprisingly, it comes with some **common challenges that need to be addressed** to **ensure data accuracy**, **reporting performance**, and **usability**.

Ambiguous paths

When there are **multiple pathways between tables**, it can create ambiguity in filter propagation.

Can lead to:

- **Unexpected filtering results** that can result in data inconsistencies or inaccuracies.
- **"Filter leakage"** where filters propagate to unintended tables or visuals, which can cause confusion and misinterpretation of data.
- **Inefficient queries** when trying to resolve ambiguous paths, which can result in longer query execution times and slower report responsiveness.

Circular dependencies

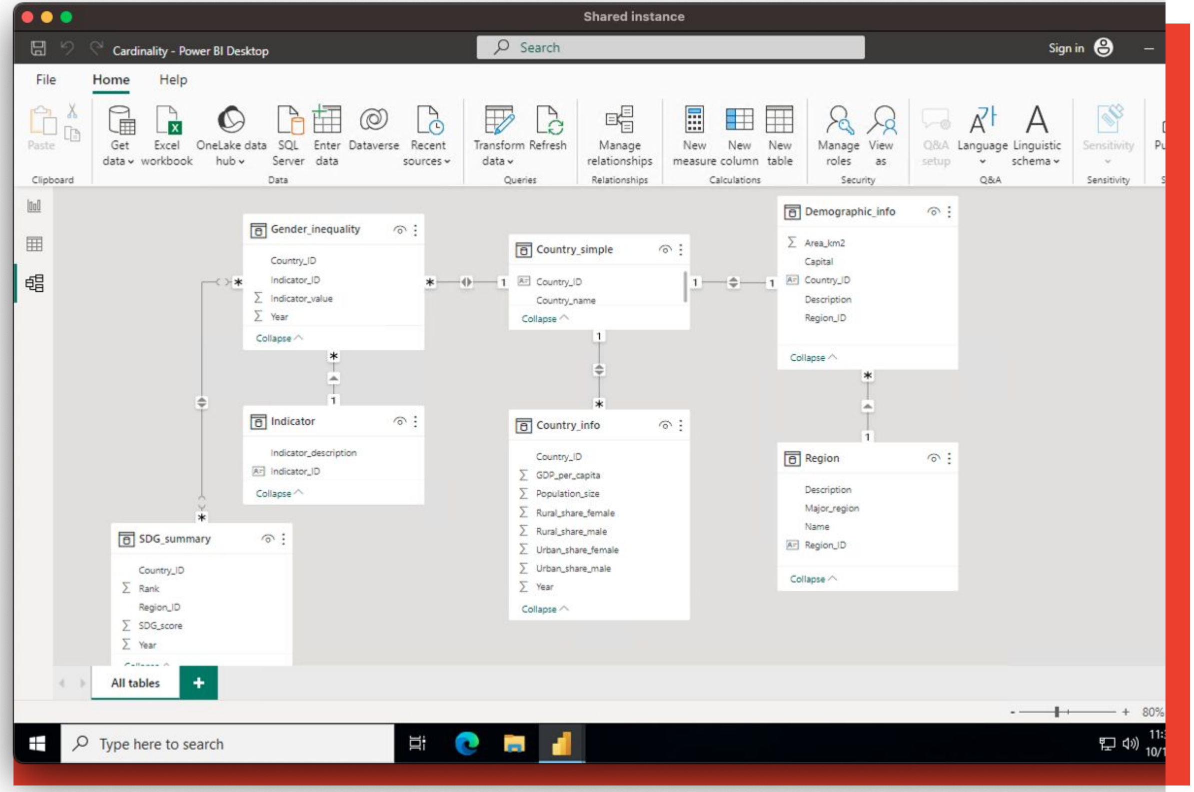
These occur **when the relationships form a loop**, and it's unclear how data should flow between tables.

Can lead to:

- **Query errors** that might prevent data from loading or displaying as expected in visualisations.
- **Performance bottlenecks** that can result in slow report rendering and user frustration.
- **Inconsistent results** where the same query or visualisation may yield different outcomes depending on the order of execution or how the relationships are resolved.

Ambiguous paths

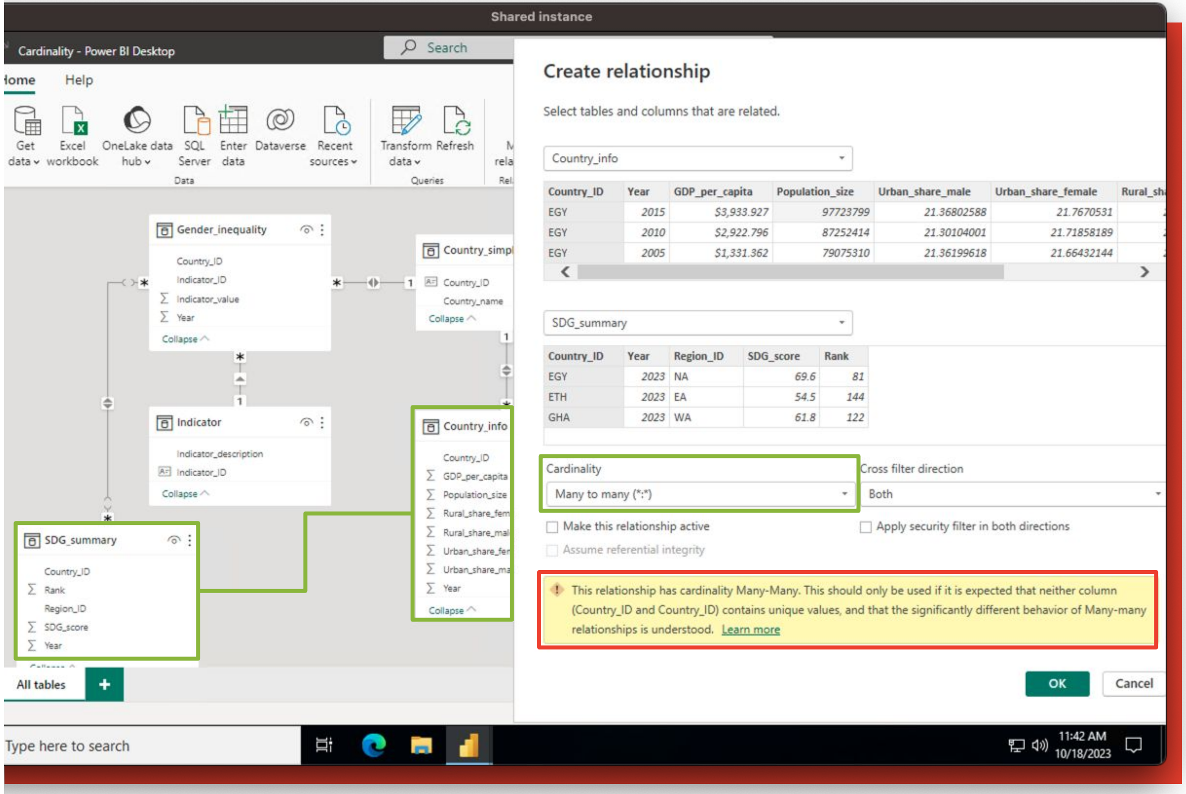
Ambiguous paths often occur when our **data model becomes more complex**.



Ambiguous paths

Ambiguous paths often occur when our **data model becomes more complex**.

It also very often occurs when we try to activate unnecessary **many-to-many** relationships. Power BI will **always warn us on many-to-many relationships**.

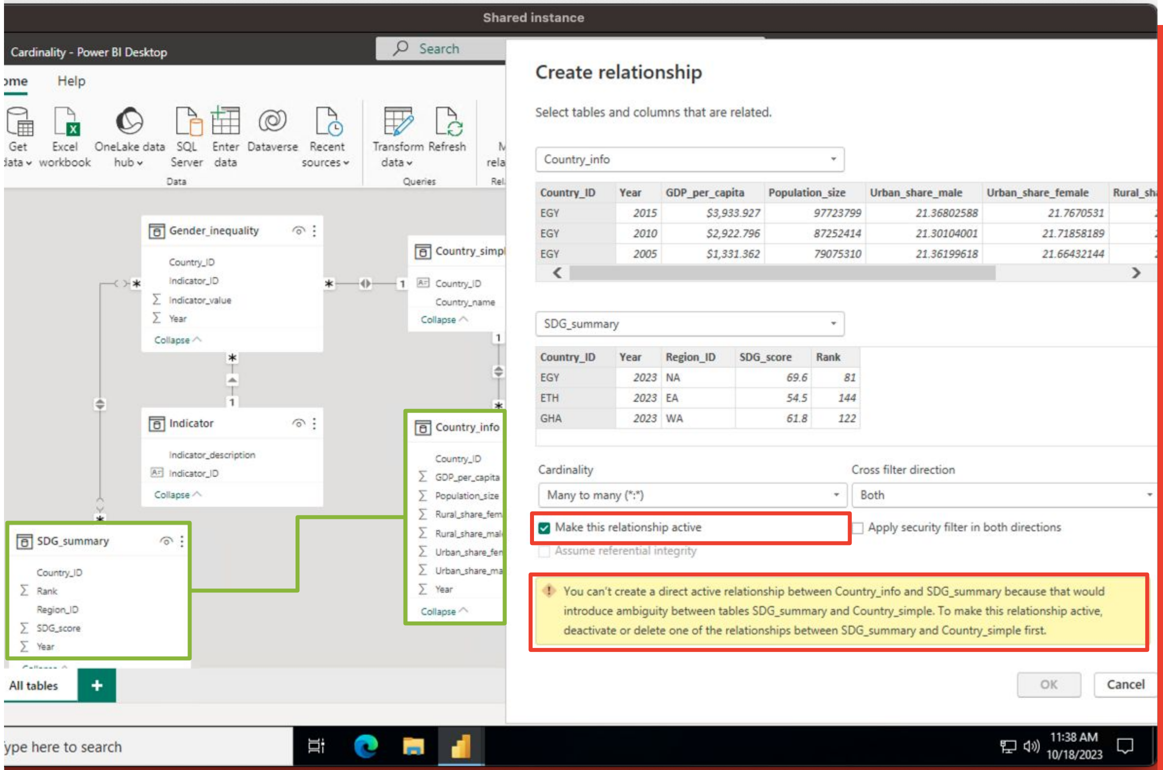


Ambiguous paths

Ambiguous paths often occur when our **data model becomes more complex**.

It also very often occurs when we try to activate unnecessary **many-to-many** relationships. Power BI will **always warn us on many-to-many relationships**.

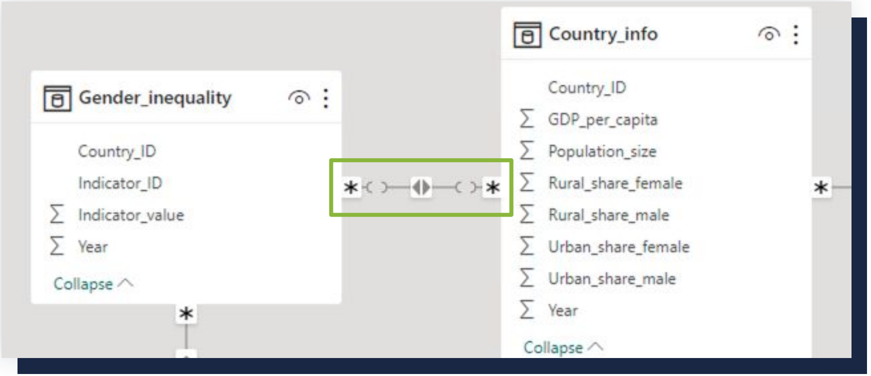
Luckily, Power BI will most often **not allow us to make a relationship active** that could potentially cause ambiguous paths.



How to avoid challenging relationships

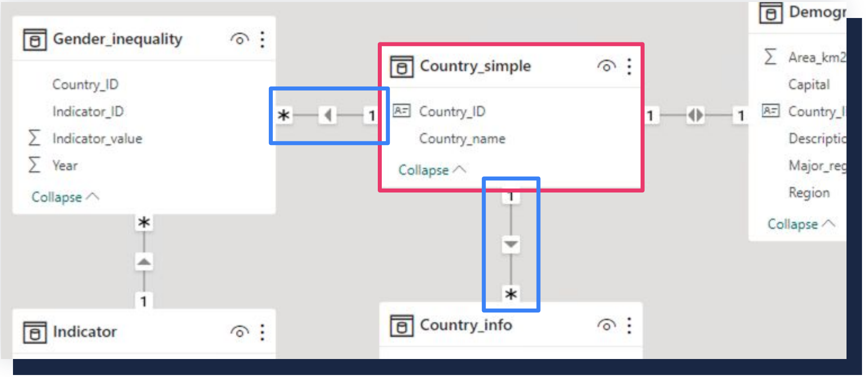
We've seen that many-to-many relationships often contribute to the common challenges we see in Power BI.

A simple **alternative to many-to-many relationships** is to use a **bridge table**, also known as a **junction table** or **association table**.



Think back to the **many-to-many** example we had and compare that to the **one-to-many** example.

So, we've technically already seen an example of a **bridge table** in action. We've **replaced the many-to-many relationship with two one-to-many** relationships by introducing the `Country_simple` table.



Tips for relationships in Power BI

Keep relationships simple

Try to keep to **one-to-many** relationships with a single directionality. They are **easy to interpret** and avoid over-complicating the data model.

While one-to-one and many-to-many relationships are useful in very specific situations, they may cause ambiguous paths and circular dependencies. Try to rather **use bridge tables** in many-to-many scenarios.

Iterate on the data structure

Restructure the data by moving features to a different table or **pivoting the data** if the interactivity between visualisations is not as expected.

Change the granularity of the dataset. **Simplify** the dataset if the data model and visualisations become too complex, or **enrich** the dataset if more detail is required to the data story.