EXPLORE AI ACADEMY

Data models in Power BI

An introduction to data models in Power BI

What is a data model?

> A data model is a **conceptual representation of data and its structure** within a database.

Data models in SQL and in Power BI serve **different purposes but are closely related** when it comes to business intelligence and data analysis.

SQL

Data models, such as entity-relationship data models, are primarily used for **database management** and **data storage**.

They define **how data are structured**, **stored**, and **organised** within relational database systems.

Power BI

Data models help us **build analytical models** that enhance our analytical capabilities by providing relationships between data tables.

This enables us to **aggregate** and **filter** across and analyse from multiple sources.

SQL and Power BI data models

> Since SQL and Power BI data models are **related**, we can apply our knowledge of data models in SQL to Power BI, and also use them together.

Data integration

- Power BI can retrieve data from SQL databases and other sources, such as spreadsheets and cloud services.
- These various data sources can be integrated into a single coherent data model.

Data extraction and transformation

- SQL data models play a critical role in providing structured data in Power BI.
- We can use Power BI's Power Query Editor to transform data from other sources, as we would've done in SQL.

Enhanced analysis and reporting

- The relationships defined in the data model allow for powerful cross-filtering and data exploration.
- We can also create calculated columns and measures in Power BI, which performs calculations similar to SQL queries.

Characteristics of a (good) data model

A good data model in Power BI is essential for **ensuring that our reports and dashboards provide valuable insights** and are **efficient to work with**. Some of the key characteristics of a good data model include:

Efficient data exploration

Users should be able to **quickly navigate through data, filter, and drill down** to gain insights without experiencing significant delays or performance issues. This requires **optimising the model for responsiveness**.

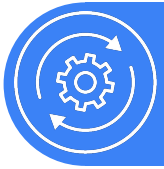
Simplified aggregations

Aggregations are used to summarise and pre-calculate data for improved performance, especially when dealing with large datasets. A good data model should make it **simple to create and manage these aggregations**.

Data accuracy

Accuracy is paramount in any data model. It should **reflect the true state of the underlying data**, and **transformations and calculations should be carried out accurately**. Inaccuracies can lead to incorrect insights and decisions.

Characteristics of a (good) data model



Maintainability, reusability

A good data model should be modular and designed for reuse. **Well-named tables, columns, and measures**, along with **clear relationships and calculations**, contribute to maintainability and reusability.

Performance optimisation

Performance is crucial for a responsive and user-friendly experience. A **well-designed data model should optimise queries and calculations** to minimise load times and maximise interactivity.

Flexibility

Business requirements change over time. A good data model should be flexible enough to **adapt to evolving needs**. This might involve modifying relationships or adding new data sources, while maintaining data integrity.

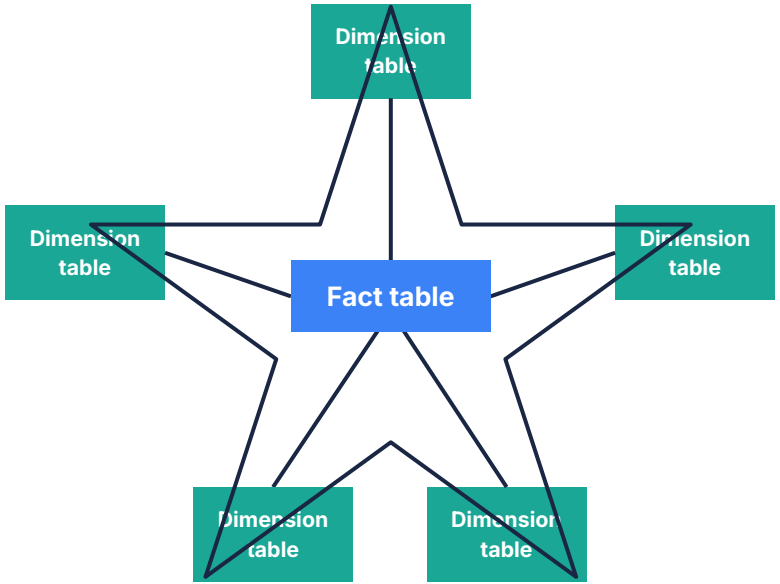
It’s important to note that **not all “good” data models will always fit our requirements**, i.e. it isn’t always one-size-fits-all. Therefore, it’s important to know the characteristics of a good data model because, chances are, we’ll have to experiment until we find the right balance.

The star schema in Power BI

> As in SQL, a **variety of data models** can be implemented in Power BI. However, the most commonly used data modelling approach in Power BI is the **star schema**.

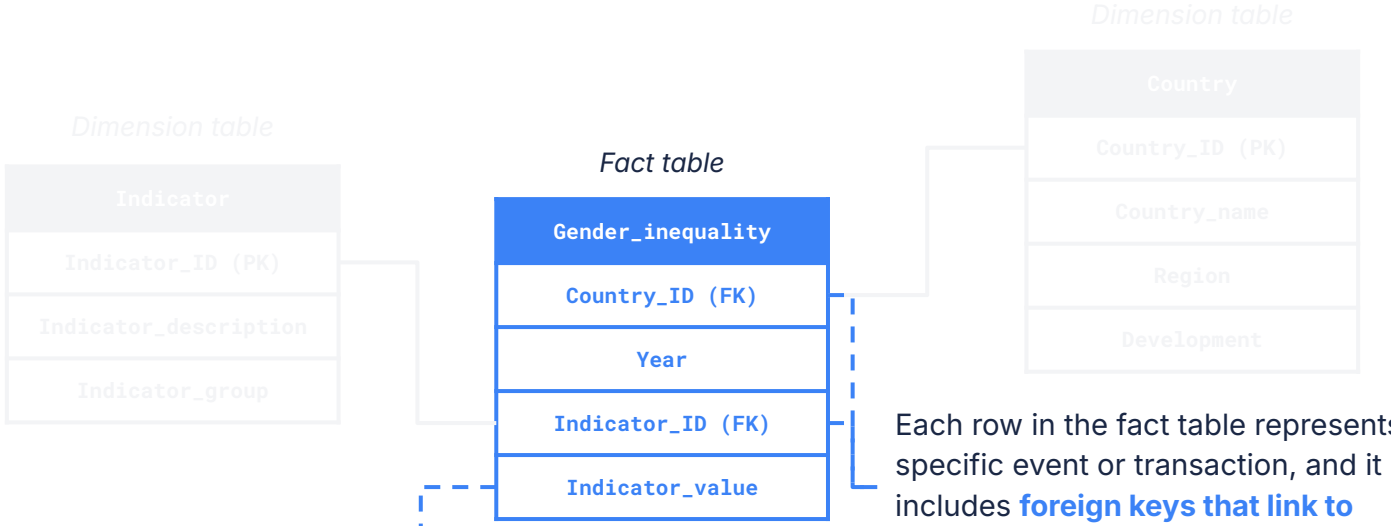
A star schema is **a specific type of data modelling approach** commonly used in Power BI and other data warehouse and business intelligence systems. It is characterised by a **central fact table** connected to multiple **dimension tables**, forming a star-like structure when visualised graphically.

The star schema is popular because it's **simple**, **optimised for query performance**, **scalable**, **easy to maintain**, and **consistent**. In other words, it ticks all of the boxes of being a "good" data model.



The star schema in Power BI

Let’s consider an example dataset on gender inequality:

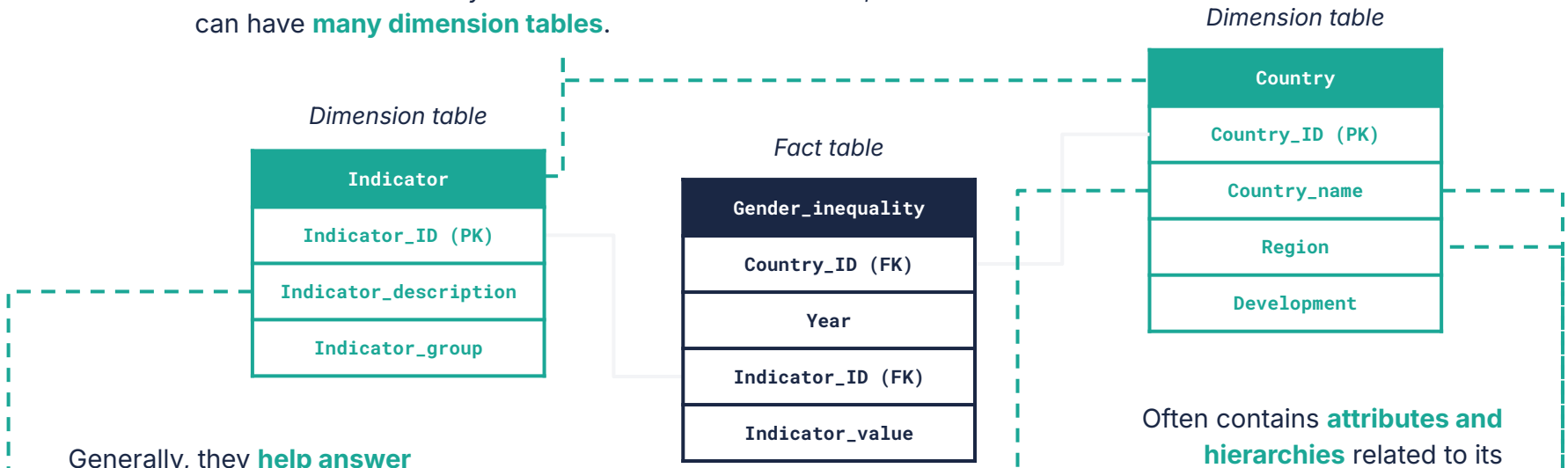


A single table at the centre of the schema that contains **quantitative measures or facts**, such as `Indicator_value`.

Each row in the fact table represents a specific event or transaction, and it includes **foreign keys that link to dimension tables**.

The star schema in Power BI

While we can have only one fact table in a star schema, we can have **many dimension tables**.



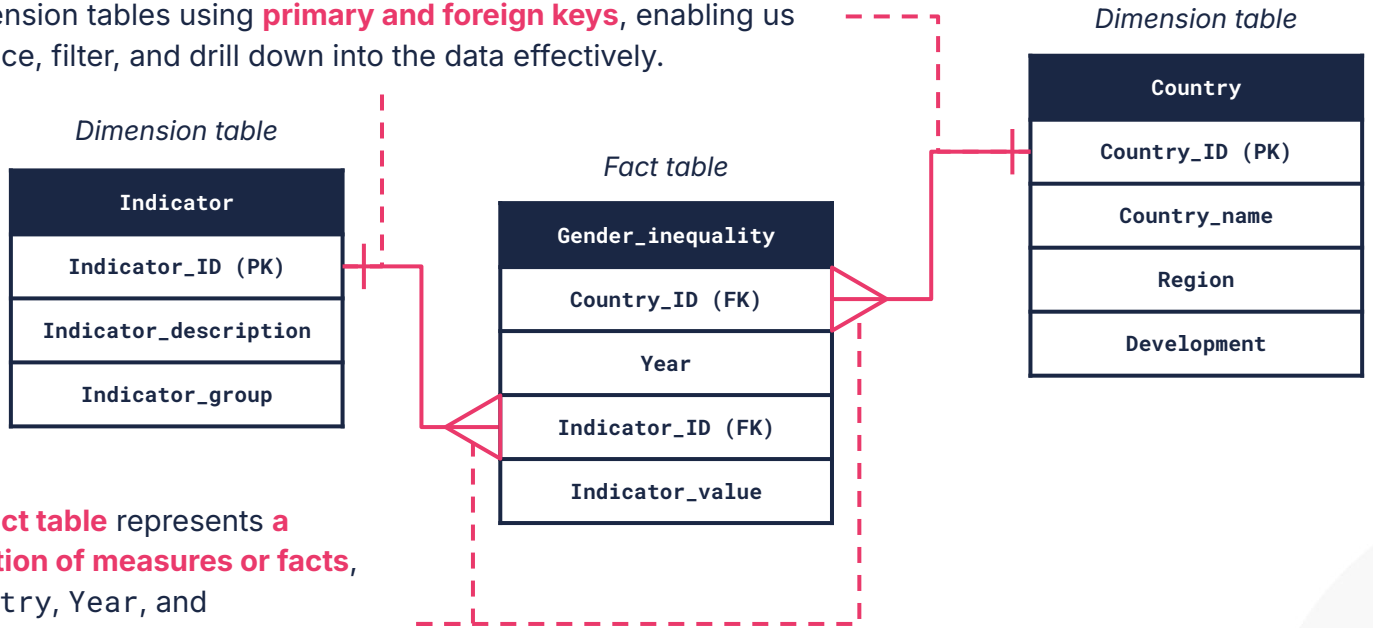
Generally, they **help answer questions like "who", "what", "where", "when", and "how"** regarding the measures in the fact table.

Provides the **context and descriptive attributes** for the data in the fact table.

Often contains **attributes and hierarchies** related to its dimension, for example, `Country_name` and `Region`.

The star schema in Power BI

Relationships are established between the fact table and dimension tables using **primary and foreign keys**, enabling us to slice, filter, and drill down into the data effectively.



Each row in the fact table represents **a specific combination of measures or facts**, for example, `Country`, `Year`, and `Indicator_ID`, which implies that multiple indicator values can be associated with a single country.

How should we structure tables?

> A good and simple table structure allows us to organise the tables in the data model in a way that makes it **easy to navigate, work with, and create meaningful reports and visualisations**.

Specific and accessible column and table properties

Each table should have a **clear and specific purpose**, representing a **distinct entity or dimensions**. For example, having distinct dimension tables such as `Indicator` and `Country`.

Tables should have **well-defined and appropriately named columns**. Column names should be descriptive and unambiguous, such as `Indicator_description` and `Country_name`.

Meaningful **descriptions and annotations on tables and columns** can also help with accessibility. This metadata helps us understand the purpose and content of each table and column, especially when collaborating on a report or dashboard.

How should we structure tables?

Merge and/or append tables to simplify

In some cases, data from multiple sources will have to be **merged (combined) or appended (stacked)** to tables in the data model in order to **reduce complexity and redundancy**.

We can **merge** tables when we want to **combine related data from different tables into a single table**. For example, having `Year` as a column in the `Gender_inequality` fact table rather than in a separate dimension table.

We can **append** tables when we want to **stack similar data structures** on top of each other. For example, having the `Indicator_value` for all `Year` instances in the fact table, rather than having distinct tables for each year and the related indicator values.

Good-quality relationships between tables

Establishing appropriate relationships between tables is crucial for data analysis. They define **how tables are connected** and **how data can be combined and analysed** across different dimensions.

Maintain referential integrity by avoiding orphaned records and ensuring that related data remain consistent.

Appropriate relationships include using **unique values for primary and foreign keys** and considering the **cardinality** (such as one-to-one and one-to-many) of relationships and the **directions** (both or single direction) of these relationships.

What is data granularity?

> Data granularity refers to the **level of detail or specificity** at which data are recorded and stored in a dataset or database. It defines how fine-grained or coarse-grained data observations are.

Fine-grained data (high granularity)		Coarse-grained data (low granularity)
Allows for more **precise and detailed analysis**.		Provides a more **summarised view** of the data.
The **data is more complex** which may require more effort to design and maintain the data model.		A more **user-friendly data model** as the number of tables and relationships to manage is reduced.
More resource-intensive to store and process data which may result in slower query performance.		Requires **less storage** and may result in **faster query performance**.

So, what's the right granularity?

The choice of data granularity really **depends on the problem we need to solve** and the **data we have available**.

There will always be a **trade-off between the complexity of analysis and performance**. Some analyses require highly granular data to detect subtle trends and patterns, while others can be adequately served with summarisation.

Fine-grained data can be summarised where necessary before being used in dashboarding or reporting to optimise performance. However, **coarse-grained data cannot be reverted** into data that can provide the same level of analysis.