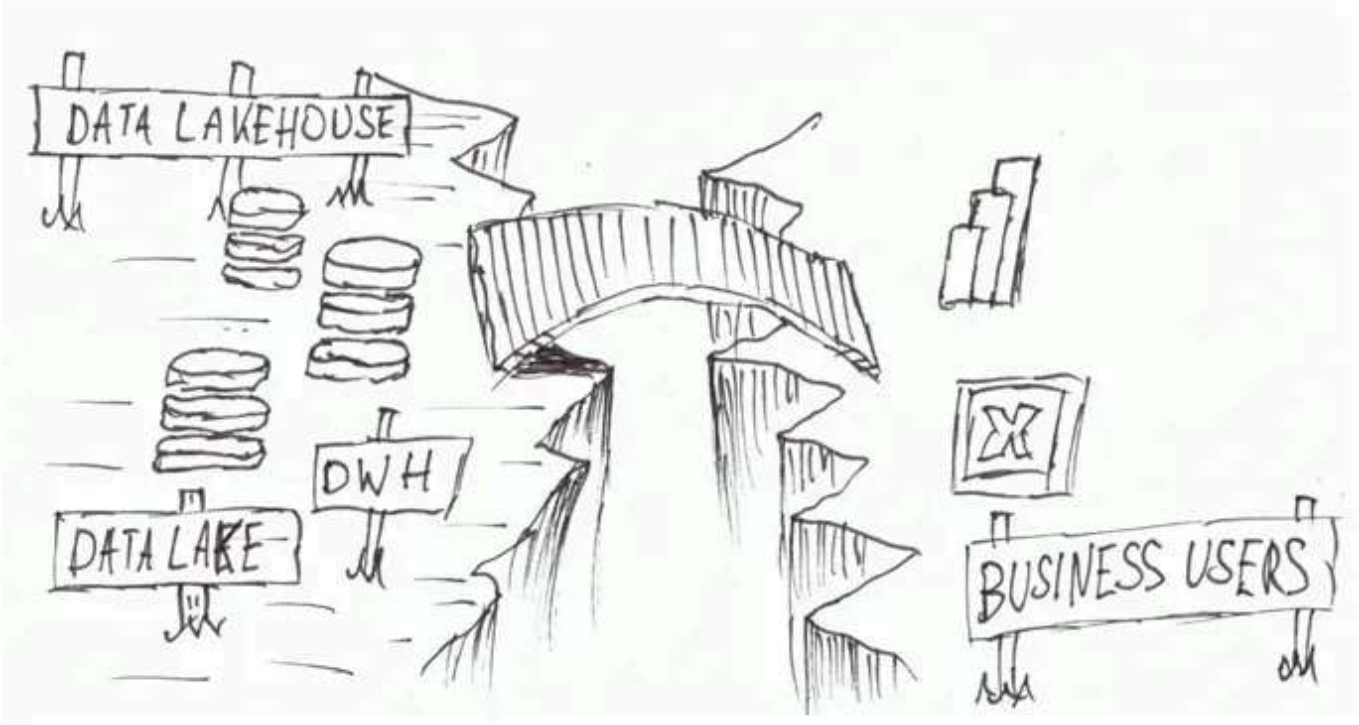
Improved Data Utilization with Semantic Models in Microsoft Fabric

Mariusz Kujawski · Follow

7 min read · Jun 27, 2024

Years ago, when I implemented my first OLAP cube, I quickly noticed a significant reduction in the number of requests for reports or data. This was because people using Excel were able to answer their questions on their own. To truly harness the power of data, business users need to have the ability to consume organizational data to meet their needs. This leads to a data-driven organization with a high level of data democratization. Achieving this requires a data platform integrated with data sources, centralized data accessibility, and the development of meaningful data models to answer analytical questions. While this concept is theoretically straightforward, practical implementation often reveals challenges such as difficulty accessing data, lack of tools, or insufficient user skills. Low-code tools like Power BI, Excel, or generative AI like Copilot can fill this gap, but they require a supporting layer — a Semantic Model that organizes data and metadata in the right schema, providing measures and hierarchies to facilitate easy report development via drag-and-drop or simple DAX functions.



Illustrated by author

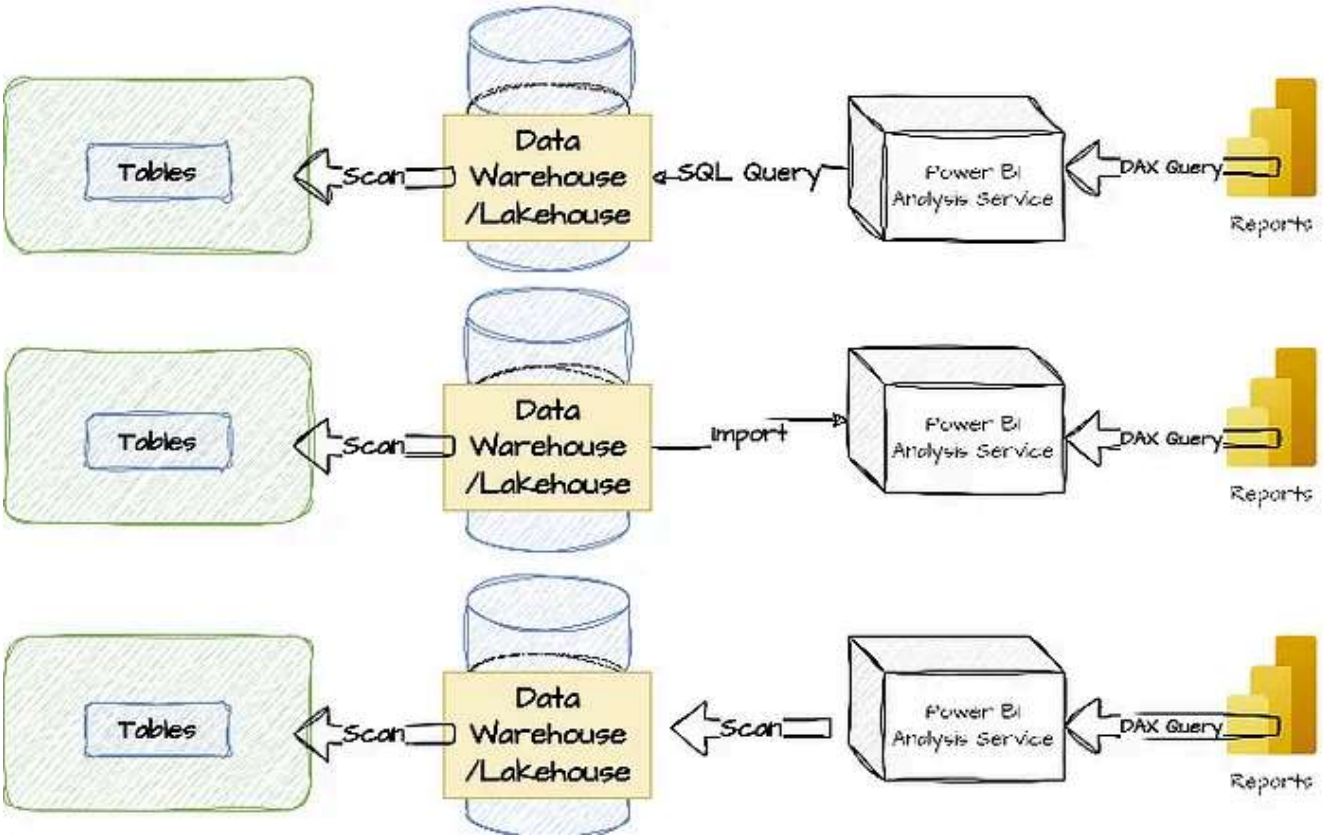
Power BI Semantic Model in Fabric

A Power BI semantic model is a logical data description that includes metrics, predefined calculations, hierarchies, and business-friendly terminology. Typically structured as a star schema with fact and dimension tables, it allows efficient data filtering, slicing, and dicing. Microsoft Fabric can generate a default semantic model for Lakehouse or Warehouse environments, which can be utilized by Power BI or Excel for visualization and reporting. Key features of the Power BI semantic model include:

- Establishing relationships between tables, descriptions, and hierarchies.
- Creating measures and standardized metrics for repeatable analysis.
- Defining data types and categories for columns.
- Enhancing security with row-level security.
- Supporting data consumption from Power BI, Excel, and Tableau.

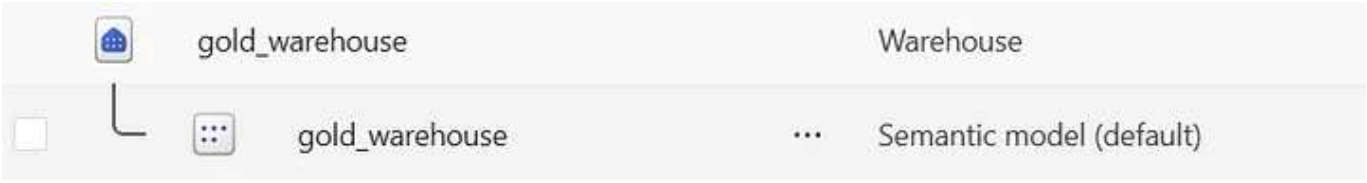
Direct Lake Mode

Direct Lake mode is a new engine that loads Parquet files from a data lake into a Power BI semantic model without querying data in the Warehouse, SQL endpoint, or importing it into the semantic model. This method is faster than traditional DirectQuery or Import modes, as it bypasses the need for query execution or memory-based data refreshes.



Creating a Power BI Semantic Model

The default Power BI semantic model can be modified or entirely new models can be created, defining relations, measures, hierarchies, and descriptions. The default model serves as a starter template, and tables can be removed if necessary. Semantic models are located in workspaces assigned to a warehouse or lakehouse.

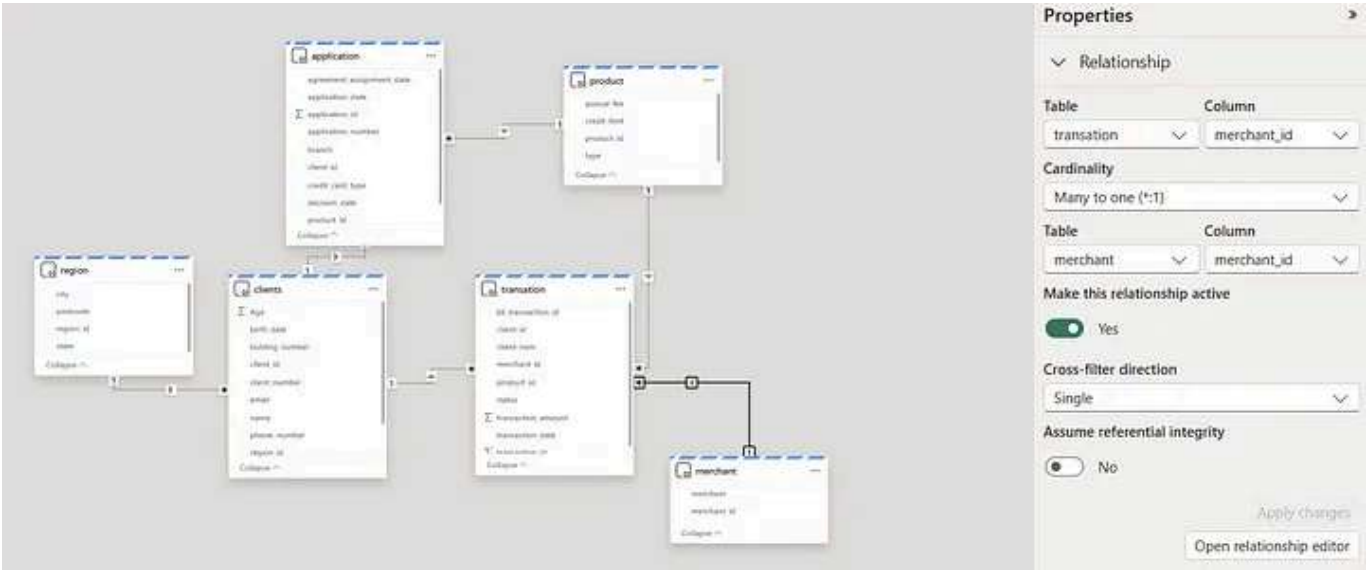


Warehouse semantic model

Lakehouse semantic model

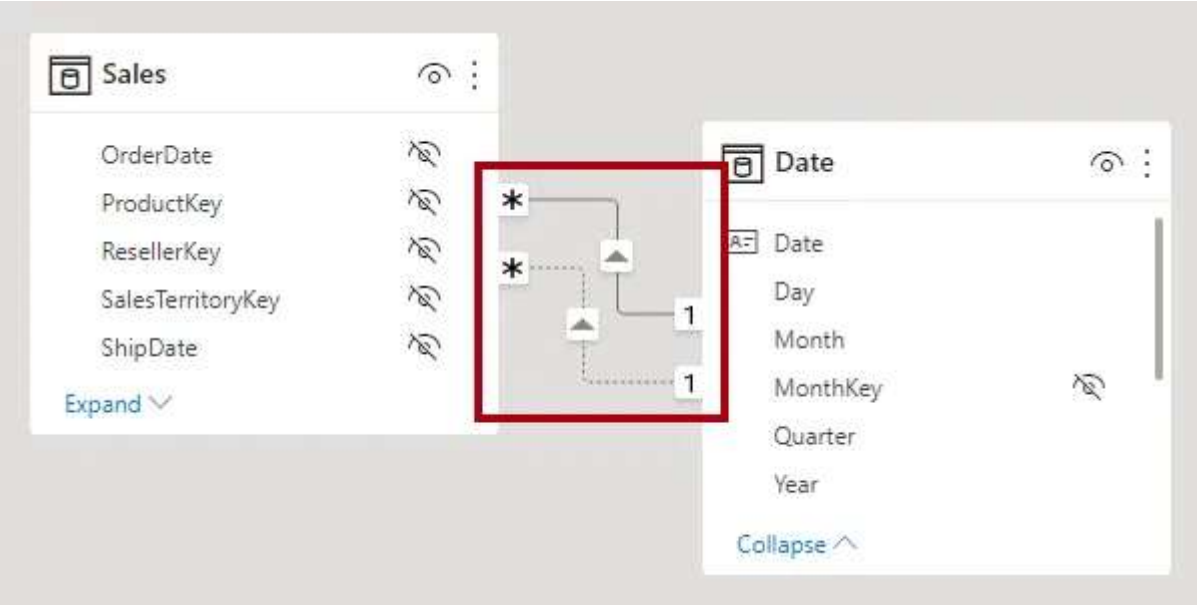
Relationships

Relationships allow filtering or grouping data in one table based on columns in another by linking equivalent key columns. These relationships, defined in the model, use in-memory data structures for efficient navigation across related tables. Relationships can be created via drag-and-drop in the model tab.



Active relationship

Most physical relationships should be active unless using role-playing dimensions. Active relationships support filtering related values. For example, filtering transactions by a year column linked to a calendar table filters transactions from the selected year. Role-playing dimensions, such as order date, delivery date, and payment date in a fact table, require using the DAX function `USERELATIONSHIP` to set inactive relationships.



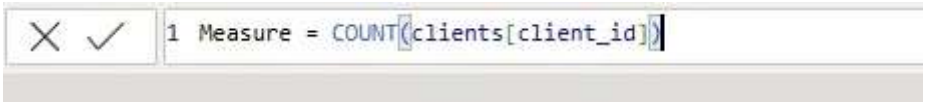
Role-play-dimension

Assume referential integrity

Enabling this option uses INNER JOIN instead of OUTER JOIN. This should not be enabled if there are NULL values in foreign keys.

Creating Measures

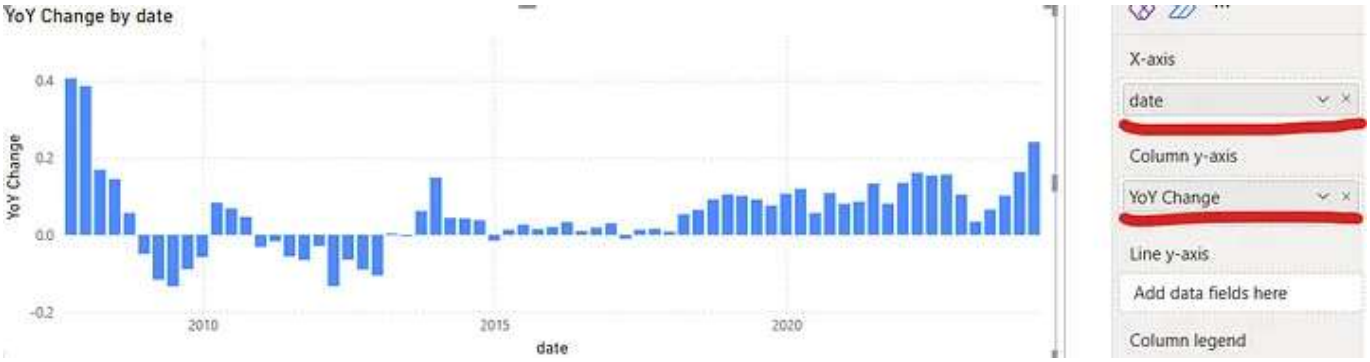
Measures are standardized metrics created using the DAX language (Data Analysis Expressions), which includes over 200 functions, operators, and constructs for complex calculations. Measures can be simple (sum, average) or complex (Year-over-Year change). Categories include Date and Time, Time Intelligence, Information, Logical, Mathematical, Statistical, Text, Parent/Child, etc.



Year-over-Year Change DAX Example:

```
YoY Change =
VAR CurrentDate = MAX('flat_report'[Date])
VAR PreviousYearDate = SAMEPERIODLASTYEAR('flat_report'[date])
VAR CurrentYearValue =
    CALCULATE(
        SUM('flat_report'[flat_price]),
        'Calendar'[Date] = CurrentDate
    )
VAR PreviousYearValue =
    CALCULATE(
        SUM('flat_report'[flat_price]),
        PreviousYearDate
    )
RETURN
IF(
    NOT ISBLANK(CurrentYearValue) && NOT ISBLANK(PreviousYearValue),
    (CurrentYearValue - PreviousYearValue) / PreviousYearValue,
    BLANK()
)
```

Predefined measures can be easily used in Power BI for meaningful visualizations.

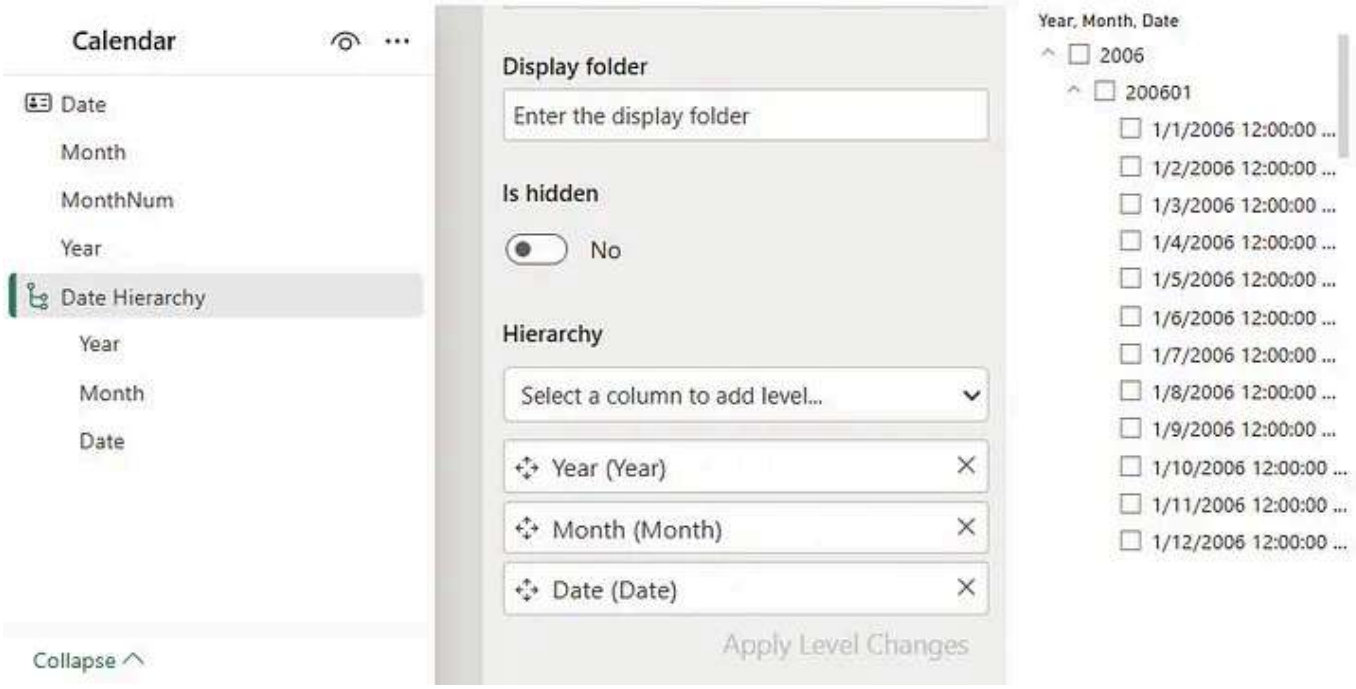


Relationship Types

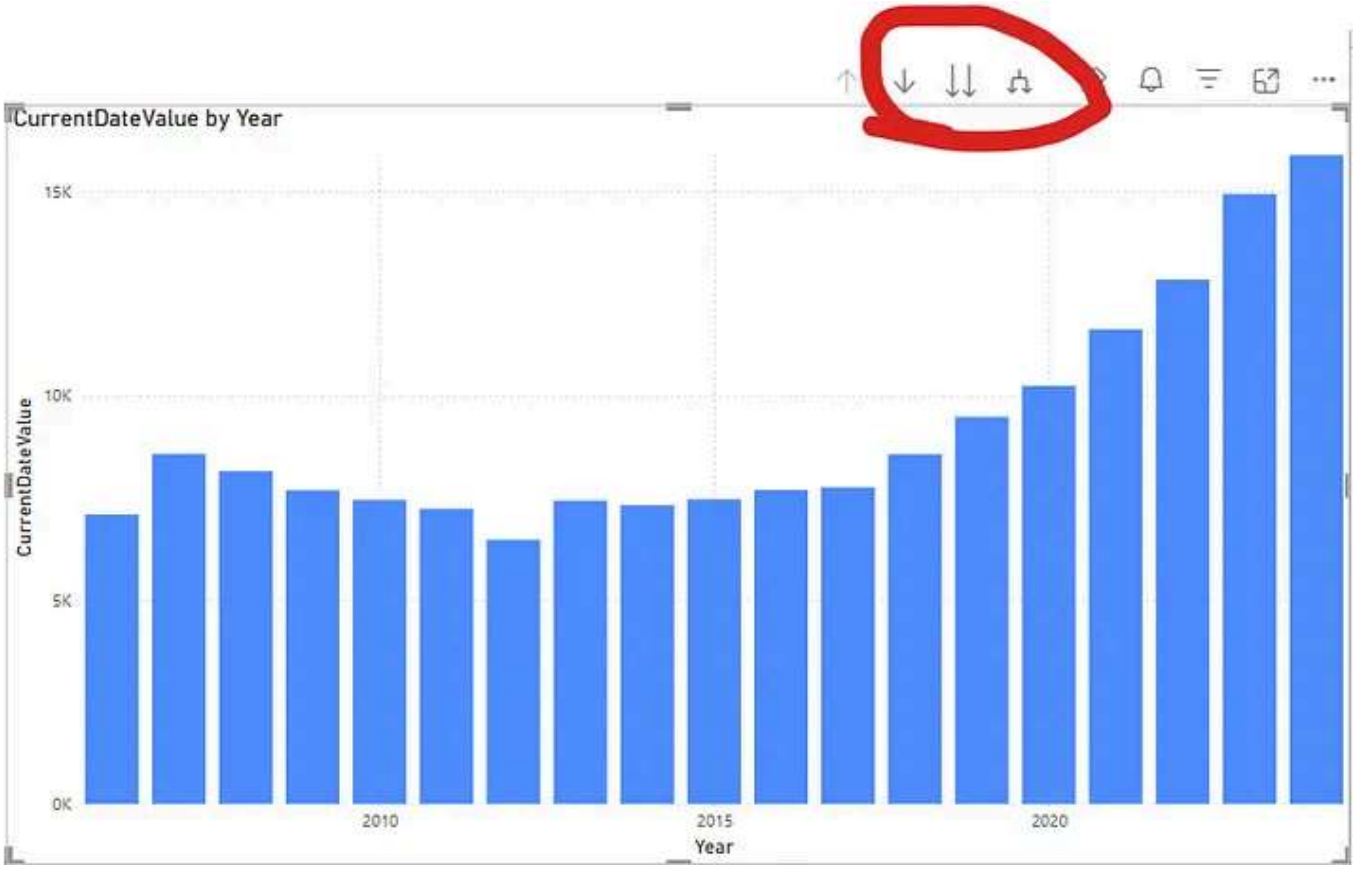
We have three types of cardinality options, representing the type of relationship. ‘One’ indicates unique values, which can be found in a key column in a dimension table, while ‘many’ means a column can contain duplicated values, such as in a foreign key in a fact table. At the relationship level, we can configure the cross-filter direction, which informs the engine how filtering should be propagated. We have two options: ‘Single’ and ‘Both.’ The arrow on the relationship line indicates the direction. This behavior can be changed using CROSSFILTER DAX functions.

Hierarchies

Hierarchies allow structured data drill-downs, enabling analysis at different granularity levels. For example, creating a year-month-date hierarchy facilitates data traversal and aggregation. Hierarchies can be created for dates, products, regions, etc., and are easily navigable using drill-down buttons.



Using Drill-down buttons, it is possible to easily change the level of data aggregation.



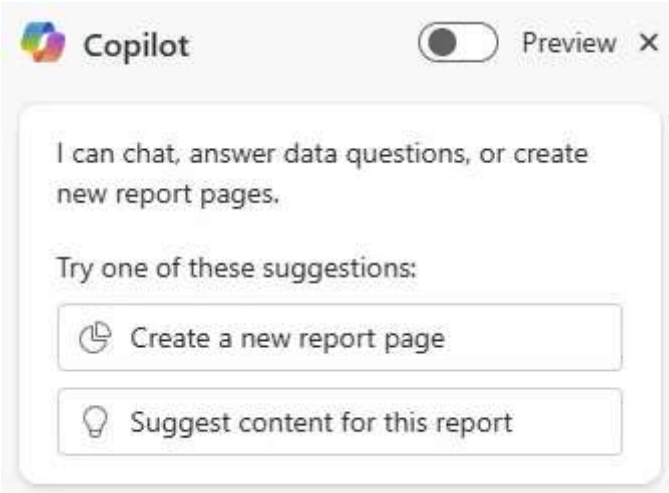
Utilizing the Semantic Model in Power BI and Excel

When we deploy a Power BI Semantic Model, we can consume it using Power BI to build reports and dashboards with impressive visualizations. Power BI is a low-code tool that allows report creation through drag-and-drop techniques. Alternatively, Excel and pivot tables can be used to analyze data. With these tools, business users can create reports, explore, and analyze data without needing to know SQL, Python, or other technical skills. Their simplicity helps make data more accessible.

Copilot for Power BI

Copilot for Microsoft Fabric is now generally available in Power BI. Copilot leverages generative AI to assist with data-related tasks, such as DAX, report generation, and report content. It can also answer questions about data (Preview). To enable this feature, we need to prepare a data model in a way that helps Copilot “understand” the data.

Preparing a schematic model for Copilot requires proper naming conventions; for example, use “Average_Customer_Rating” instead of “AvgRating.” The model should contain predefined measures that users are most likely to request. Additionally, it is important to maintain correct data types, hierarchies, and relationships. With a well-defined schematic model, Copilot can effectively provide useful information from data to non-technical business users.



Semantic Link

If you prefer Python, you can consume data from the Semantic model using the Semantic link. This feature allows you to establish a connection between semantic models and Synapse Data Science in Microsoft Fabric. Note that the use of semantic links is supported only in Microsoft Fabric. With this feature, you can read measures defined in a model and query data with Python or execute DAX expressions to retrieve data.

Methods to Integrate with the Power BI Semantic Model:

- **SemPy Python Library**: SemPy offers features similar to Pandas but with additional user-friendly enhancements for Data Scientists. It extends Pandas'

standard capabilities by incorporating extra metadata from data sources and Power BI Data Categories, such as geographic information, barcodes, and URLs. It also supports relationships, hierarchies, and more.

- **Spark Native Connector**: This connector provides native Spark API functionalities, enabling you to leverage Spark SQL, PySpark, R, and Scala according to your specific needs.

Semantic Link Usage Example:

```
import sempy.fabric as sf

df = (
    sf
    .evaluate_measure(
        workspace="My Workspace",
        dataset="Sales",
        groupby_columns=["'Date'[Yr-Mth]"],
        measure='Sales'

    )
)
```

Summary

The Power BI composite model is a powerful feature that supports data access using low-code tools such as Power BI and Excel. This capability helps democratize data access for business users who do not have technical skills like SQL or Python. It reduces the involvement of data engineering or BI teams and shortens the time needed for new report implementations, as more users within an organization can access data and build their reports. The composite model serves as an excellent bridge between data lakes, data warehouses, and data consumption by business users.