CREATING A DATA MODEL

Steps to create a data model

Creating a strong data model is core to data analytics. Taking care to select the right tables and columns, and choosing the correct primary keys, cardinalities, relationships and data types are all essential to ensuring a data model produces the correct outputs and is not slowed down by redundant data; this is especially important for larger datasets.

1. The first checks should be made once the tables are loaded into Power Query. Removing any redundant columns is a quick step that helps keep the data model lean, saving disk space and processing power, and reducing the number of potential issues, such as data errors. In this example, there are additional fields: Month, Year, and Store|Product, which are not needed. Month and year are contained in DimDate. On the other hand, Store|Product is not useful here, and is not a primary key shared by any other table, so can be safely removed.
2. Checking that data is in the correct format and that Power Query has applied the correct automatic transformation steps is generally considered good practice, and helps to reduce future problems, such as incorrect DAX calculations due to a field in the wrong format. In this example, all data was recognised correctly by Power BI, so no transformations are needed.
3. Once the data checks have been completed and the data loaded into Power BI, it is wise to check the relationships in the model that Power BI has automatically generated, deleting and/or redoing those that are incorrect. As you go, think carefully about any issues that may be generated by the relationship, such as those that are circular. In this example, Power BI automatically detected some of the relationships correctly, leaving only a few more to establish. Note that when establishing the relationship with DimClimate in the model, it is important to think about the bigger picture. A relationship could be established with both fact tables; however this would create unnecessary complexity (an additional relationship) and would mean DimClimate could not have an active relationship with DimDate due to a circular relationship. DimClimate could be considered complementary to DimDate, and so the relationship is established between the two. In this example, Power BI automatically detected some of the relationships correctly, leaving only a few more to establish.
4. Note that when establishing the relationship with DimClimate in the model, it is important to think about the bigger picture. A relationship could be established with both fact tables; however this would create unnecessary complexity (an additional relationship) and would mean DimClimate could not have an active relationship with DimDate due to a circular relationship. DimClimate could be considered complementary to DimDate, and so the relationship is established between the two. There is a similar issue with FactNPS: to analyse any connections between sales and NPS, NPS should be brought in as a dimension to the FactSales table. Establishing the relationship directly with FactSales ensures the relationships between sales and NPS ratings can be measured.

Documentation of data model – list related tables, primary keys and cardinality.

Table	Table	Primary Key	Cardinality
FactSales	DimDate	MonthEnd	Many-to-one
FactSales	DimStore	StoreID	Many-to-one
FactSales	DimProduct	SKU	Many-to-one
FactSales	FactNPS	StoreMonth	Many-to-one
DimStore	DimRating	Postcode	One-to-one
DimDate	DimClimate	MonthEnd	One-to-one
Distance Between Sensors and Com	Pedestrian_Traffic_Data	Sensor_Name	Many-to-Many
Distance Between Sensors and Com	Competition_Data	Sensor_Name	Many-to-Many
FactOnlineSales	DimStore	StoreID	Many-to-one

A fact table is a table that contains the measures of interest.
Dimension tables are used to describe the dimensions of the fact table.

For example, sales take place on specific dates, they can be associated to specific stores and products. DimDate, DimStore and DimProduct are therefore joined to FactSales to add the date, stores, and product dimensionality (respectively) to FactSales.

In this example, the FactSales is joined to FactNPS. Usually, fact tables would not be joined. However in this case, the relevant relationship is between sales and net promoter score (NPS) ratings, not between NPS and other factors, such as temperature.

For this reason, FactNPS is treated as supplementary to (a dimension of) FactSales. If the main concern were, for example, to visualise temperature and NPS side by side, NPS would be treated purely as a fact table with similar relationships to those that FactSales has with the dimension tables.

Alternatively, FactNPS could be merged onto FactSales in Power Query, which would remedy most issues, though this is beyond the scope of this task.

As DimClimate has exactly the same number of rows as DimDate has (and identical records), a bi-directional, one-to-one relationship can be established, causing DimClimate to act as an extension of DimDate. A similar situation arises between DimStore and DimRating.