

**What is a primary key in a table?**

A primary key is a column or combination of columns in a table that uniquely identifies each record. No two rows can have the same primary key value, and it can't be blank. It's the unique identifier that makes linking tables possible.

**Name the two types of table relationships in Power BI.**

The two main types are **one-to-many** (most common, where one value in a table relates to many values in another) and **many-to-many** (where multiple values in both tables relate to each other). One-to-one exists but is less common.

**How do you create a relationship between two tables in Power BI?**

You can do it in Model view by dragging a field from one table to its matching field in another, or by going to "Manage Relationships," clicking "New," selecting the two tables and the matching columns, then setting the cardinality (for example, one-to-many) and cross-filter direction before confirming.

**What is a "star schema"?**

A star schema is a data model design where you have one central fact table containing measurable events like transactions, and several dimension tables around it containing descriptive attributes such as customer details, product information, or dates. The layout looks like a star, with the fact table in the center and dimension tables radiating outward.

**Which table is typically the fact table in a sales dataset?**

In a sales dataset, the **Sales** table is usually the fact table because it contains the individual sales transactions you want to measure and analyze.

**Why is ProductID in Sales.csv a foreign key?**

ProductID in Sales.csv is a foreign key because it points to the ProductID in the Products table. It isn't unique within Sales—many sales rows can reference the same product—but it links each sale back to product details.

**Explain why a star schema improves performance.**

A star schema improves performance because Power BI's engine is optimized for it. Fact tables stay narrow and efficient, dimension tables are easy to filter, and queries run faster with fewer complex joins. It also makes models easier to understand and maintain.

**Use bidirectional filtering sparingly—when is it appropriate?**

Bidirectional filtering is appropriate only when you need filters to flow in both directions for accurate results, such as when two dimension tables influence each other or when a specific report scenario requires cross-filtering both ways. Using it everywhere can slow performance and produce confusing totals.