Power BI Data Modeling & Relationships Cheat Sheet

1. What is a primary key in a table?

A primary key is a column (or set of columns) that uniquely identifies each row in a table.
Example: CustomerID in Customers table.

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

- One-to-Many (1:*) – Most common, e.g., one customer → many sales.
- Many-to-Many (*:*) – More complex, e.g., customers linked to multiple loyalty programs.

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

Go to Model view → Drag a field (e.g., CustomerID) from one table to the matching field in another table.
Or use Manage Relationships → New → Select tables/columns → Define cardinality (1:* or *:*)

4. What is a 'star schema'?

A star schema is a data model with:
- A central fact table (e.g., Sales).
- Surrounded by dimension tables (e.g., Products, Customers, Dates).

Called 'star' because of its structure.

5. Which table is typically the fact table in a sales dataset?

The Sales table (contains transactions, numeric measures, and foreign keys).

6. Link Sales.csv to Customers.csv using CustomerID (one-to-many).

- Customers[CustomerID] → One side (unique)
- Sales[CustomerID] → Many side

Relationship: 1 → * (one-to-many).

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

Because ProductID in Sales references the primary key ProductID in the Products table. It links transactions to product details.

8. Fix a relationship error where ProductID has mismatched data types.

Ensure both ProductID in Sales and Products are the same type (Whole Number or Text) in Power Query, then recreate the relationship.

9. Explain why a star schema improves performance.

- Reduces table size (no duplication of descriptive data).
- Simplifies DAX queries.

- VertiPaq engine compresses data efficiently.
- Better query folding and aggregation.

10. Add a new column TotalSales in Sales (Quantity * Price from Products).

DAX: TotalSales = Sales[Quantity] * RELATED(Products[Price])

11. Optimize a model with circular relationships—how would you resolve it?

- Remove redundant relationships.
- Use bridge tables if needed.
- Replace bidirectional filters with single-directional.
- Flatten into a star schema.

12. Create a role-playing dimension for OrderDate and ShipDate.

- Load one Date table.
- Create multiple relationships (OrderDate, ShipDate).
- Keep one active at a time.
- Use USERELATIONSHIP in DAX.

Example: ShipSales = CALCULATE([TotalSales], USERELATIONSHIP(Sales[ShipDate], Date[Date]))

13. Handle a many-to-many relationship between Customers and Products.

Introduce a bridge table (CustomerProductMapping) with unique CustomerID–ProductID pairs. Connect both Customers and Products to the bridge with one-to-many relationships.

14. Use bidirectional filtering sparingly—when is it appropriate?

Only when cross-filtering is required both ways (e.g., security roles, many-to-many bridge tables). Avoid on large fact tables to prevent performance issues.

15. Write DAX to enforce referential integrity if a CustomerID is deleted.

DAX Example:
OrphanedSales = CALCULATE (COUNTROWS(Sales),
ISBLANK(RELATED(Customers[CustomerID])))
→ Returns the number of sales records with missing customers.