Lesson 5 – Data Modeling Basics

1. What is a primary key in a table?
A column that uniquely identifies each row in a table.

2. Name the two types of table relationships in Power BI.
One-to-many and many-to-many.

3. How do you create a relationship between two tables in Power BI?
Drag and drop a column from one table to the matching column in another table in the Model view.

4. What is a "star schema"?
A data model with a central fact table connected to multiple dimension tables.

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

6. Link Sales.csv to Customers.csv using CustomerID (one-to-many).
CustomerID in Customers is the primary key; CustomerID in Sales is the foreign key. This forms a one-to-many relationship.

7. Why is ProductID in Sales.csv a foreign key?
Because it refers to ProductID in the Products table, allowing connection between the sales and product data.

8. Fix a relationship error where ProductID has mismatched data types.
Ensure both ProductID columns have the same data type (e.g., whole number) using the "Data Type" option in Power BI.

9. Explain why a star schema improves performance.
It reduces data redundancy, simplifies queries, and improves performance for aggregations and filtering.

10. Add a new column TotalSales in Sales (Quantity * Price from Products).
Use DAX:
TotalSales = Sales[Quantity] * RELATED(Products[Price])

11. Optimize a model with circular relationships—how would you resolve it?
Avoid creating circular relationships. Use DAX functions like USERELATIONSHIP() or re-evaluate the model to remove redundant paths.

12. Create a role-playing dimension for OrderDate and ShipDate.
Duplicate the Date table and create two relationships: one with OrderDate, another with ShipDate.

13. Handle a many-to-many relationship between Customers and Products.
Use a bridge (intermediate) table or composite models to eliminate ambiguity.

14. Use bidirectional filtering sparingly—when is it appropriate?
When both tables need to filter each other, such as in slicers or complex reporting needs. Use it with caution to avoid performance issues.

15. Write DAX to enforce referential integrity if a CustomerID is deleted.
Use DAX with TREATAS to ensure only existing CustomerIDs are evaluated:

DAX

CopyEdit

CALCULATE([Measure], TREATAS(VALUES(Customer[CustomerID]), Sales[CustomerID]))