

Homework 5.

Lesson 5: Data Modeling Basics in Power BI

1. What is a Primary Key in a Table?

A **primary key** is a unique identifier for each record in a table. It ensures no duplicates exist and helps establish relationships between tables.

- Example: `CustomerID` in the **Customer** table.
-

2. Two Types of Table Relationships in Power BI

1. **One-to-Many (1:N)** – One record in Table A relates to multiple records in Table B (e.g., one customer can have multiple sales).
 2. **Many-to-One (N:1)** – The inverse of one-to-many (e.g., multiple sales belong to one product).
-

3. Creating a Relationship Between Two Tables in Power BI

1. Go to **Model View**.
 2. Drag `CustomerID` from **Sales.csv** to `CustomerID` in **Customers.csv**.
 3. Power BI auto-detects a **one-to-many** relationship.
-

4. What is a "Star Schema"?

A star schema is a data model where:

- **One central fact table** (e.g., **Sales**) connects to **multiple dimension tables** (e.g., **Customer**, **Product**).
 - Improves query performance and simplifies analysis.
-

5. Which Table is Typically the Fact Table in a Sales Dataset?

The **Sales.csv** table is the fact table because it contains transactional data (e.g., OrderID, Quantity, OrderDate).

6. Link Sales.csv to Customers.csv Using CustomerID (One-to-Many)

- **Sales** (many) → **Customer** (one) via **CustomerID**.
 - Power BI automatically sets this as a **one-to-many** relationship.
-

7. Why is ProductID in Sales.csv a Foreign Key?

A **foreign key** references a primary key in another table.

- **ProductID** in **Sales.csv** links to **ProductID** (primary key) in **Products.csv**.
-

8. Fix a Relationship Error Where ProductID Has Mismatched Data Types

- Ensure **ProductID** in both **Sales.csv** and **Products.csv** has the **same data type** (e.g., Whole Number).
 - Steps:
 1. Right-click **ProductID** → **Change Data Type**.
 2. Set both columns to **Whole Number**.
 3. Recreate the relationship.
-

9. Why Does a Star Schema Improve Performance?

- Reduces redundant data.

- Simplifies queries (Power BI scans fewer tables).
 - Optimizes filter propagation.
-

11. Optimize a Model with Circular Relationships

Circular relationships (e.g., $A \rightarrow B \rightarrow C \rightarrow A$) cause ambiguity.

Solution:

- Remove redundant relationships.
 - Use **bridge tables** for many-to-many relationships.
-

12. Create a Role-Playing Dimension for OrderDate and ShipDate

If **Sales.csv** has both **OrderDate** and **ShipDate**:

1. Duplicate the **Date** table (or create a separate date table).
 2. Create two relationships:
 - **Sales[OrderDate] → Date[Date]**
 - **Sales[ShipDate] → Date[Date]**
-

13. Handle a Many-to-Many Relationship Between Customers and Products

If customers buy multiple products (and vice versa):

1. Create a **bridge table** (e.g., **CustomerProduct**).
 2. Link **CustomerID** and **ProductID** to the bridge table.
-

14. Use Bidirectional Filtering Sparingly—When is it Appropriate?

- **Default:** One-directional (filters flow from "one" to "many").
 - **Bidirectional** is useful in:
 - Complex star schemas.
 - When dimensions filter each other (e.g., **Product Category** affecting **Customer Region**).
 - **Risk:** May cause performance issues if overused.
-

15. Write DAX to Enforce Referential Integrity if a CustomerID is Deleted

Use **CROSSFILTER()** to manage filtering:

DAX

Copy

Download

SalesFiltered =

```
CALCULATETABLE(  
    Sales,  
    CROSSFILTER(Sales[CustomerID], Customer[CustomerID], ONEDIRECTIONAL)  
)
```

- Prevents ambiguity when deleting a customer.
-

Practical Example: Model View Setup

Tables & Relationships

Table	Key Column	Relationship
Sales	OrderID (PK)	Fact Table
Customer	CustomerID (PK)	1 → N with Sales
Products	ProductID (PK)	1 → N with Sales