

1. What is a primary key in a table?

In a table, primary key is a unique identifier for a record that is unique for each row in a table and does not repeat. Also, primary keys are essential to create relations between multiple tables.

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

One-to-One: relationship between two tables where there is only one record in one table corresponding to only one record in the other table.

One-to-Many: relationship between two tables where there is only one record in one table corresponding to multiple records in the other table

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

Simple. To create a relationship between two tables in Power BI, just drag and drop the column(pk) in one table to the column you want to reference that column to in the other.

4. What is a "star schema"?

"Star Schema" is a Datawarehouse schema where there is one fact table referenced by multiple dimensional tables.

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

In a sales dataset, the "Sales" table is often the fact table

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

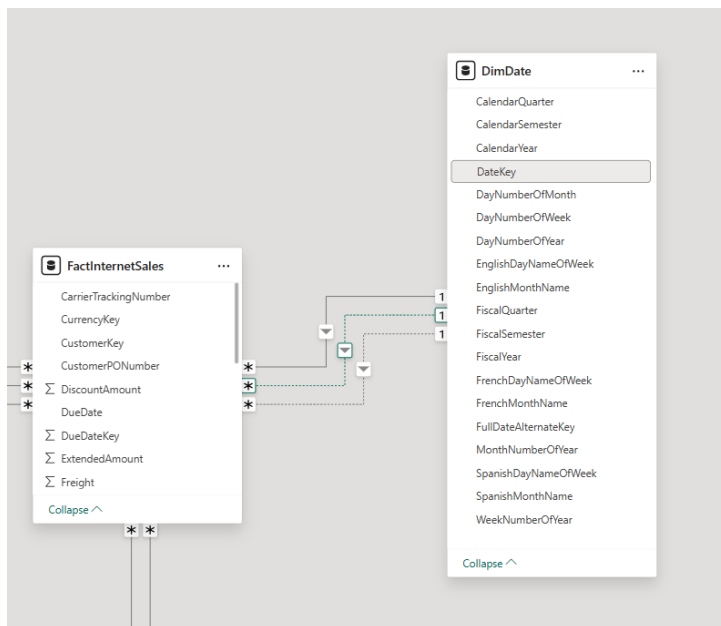
7. Why is ProductID in Sales.csv a foreign key?

ProductID is actually a primary key, but in Products table. And because ProductID in Sales table references another table, where is a primary key, it is considered a foreign key.

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

```
= Table.TransformColumnTypes("#Promoted Headers",{"ProductID", Int64.Type}, {"ProductName", type text}, {"Category", type text}, {"Price", Currency.Type}))
```

```
= Table.TransformColumnTypes("#Promoted Headers",{"OrderID", Int64.Type}, {"CustomerID", Int64.Type}, {"ProductID", Int64.Type}, {"Quantity", Int64.Type}, {"OrderDate", type date}))
```



9. Explain why a star schema improves performance.

Star schema improves performance by applying normalization to tables, thereby reducing the number of rows per table by creating other tables referencing them, which will speed up and take up less memory as redundancy created by denormalized tables disappears. For example, if in a single table where there are regions, customers, products, and other info, and we need to calculate the number of customers per region, then there's no need for the products info etc. This has a negative toll on the performance. Besides

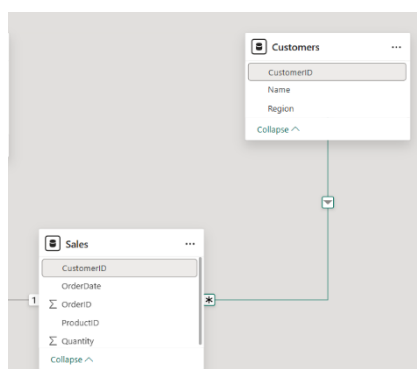
10. Add a new column TotalSales in Sales (Quantity * Price from Products).

= Table.AddColumn("#Expanded Products", "TotalSales", each [Quantity]*[Price])

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

I'd remove one of the connections to prevent circular loops from happening

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



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

To handle many-to-many relationship between Customer and Products, we need to create a new “Bridge table” that will connect both tables and have a primary key composed of primary key from customers and products tables and at the same time referencing them to the original tables. This way we can avoid any issues with many-to-many relationships.

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

Bidirectional filtering is appropriate when we need a calculation that needs two or more dimensional tables that are not referenced to each other. By using this filtering, each dimensional table can be referenced through a shared fact table as a bridge.

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

DAX **does not support enforcing referential integrity directly**, because DAX is a **query and calculation language**, not a data manipulation or constraint-enforcement language like SQL.