8. Demonstrate different types of relationship present between tables with an example.

In Power BI, there are three main types of relationships between tables: One-to-Many, Many-to-One, and Many-to-Many. Let's demonstrate each type of relationship with an example using three fictional tables: "Sales," "Products," and "Customers."

1. One-to-Many Relationship (1:N): A one-to-many relationship is the most common type of relationship in Power BI. It means that for each unique record in the "one" table, there can be multiple related records in the "many" table.

Example:

- **Tables:**
    - "Sales" (One side): Contains sales transactions with details like date, product ID, and quantity.
    - "Products" (Many side): Contains product information, including product names and categories.
- **Relationship:** Each sale is associated with a single product, but each product can be sold in multiple transactions. Therefore, there's a one-to-many relationship between "Products" and "Sales" based on the "Product ID" column.

2. Many-to-One Relationship (N:1): This type of relationship is essentially the reverse of a one-to-many relationship. It means that multiple records in the "many" table can be associated with a single record in the "one" table.

Example:

- **Tables:**
    - "Sales" (Many side): Contains sales transactions with details like date, product ID, and quantity.
    - "Customers" (One side): Contains customer information, including customer names and addresses.
- **Relationship:** Each sale is made by a single customer, but each customer can have multiple sales transactions. Therefore, there's a many-to-one relationship between "Customers" and "Sales" based on the "Customer ID" column.

3. Many-to-Many Relationship (N:N): A many-to-many relationship is less common in Power BI but is still possible. It indicates that multiple records in both the "many" and "one" tables can be related to each other.

Example:

- **Tables:**
    - "Sales" (Many side): Contains sales transactions with details like date and customer ID.
    - "Products" (Many side): Contains product information, including product names and categories.
    - "Customers" (Many side): Contains customer information, including customer names and addresses.
- **Relationship:** In this example, a sale can involve multiple products, and each product can be sold to multiple customers. Therefore, there's a many-to-many relationship between "Sales," "Products," and "Customers." To create such a relationship, an intermediate table called "SalesProducts" could be used to map the products sold in each transaction.

Here's a simplified Power BI model to represent these relationships:

- "Sales" has a many-to-one relationship with "Customers" (Customer ID in "Sales" related to Customer ID in "Customers").
- "Sales" has a one-to-many relationship with "Products" (Product ID in "Sales" related to Product ID in "Products").
- If needed, you can establish a many-to-many relationship between "Sales," "Products," and "Customers" using an intermediate table.

Refernce :

- [Create and manage relationships in Power BI Desktop](#)

9. What is DAX?

DAX (Data Analysis Expressions) is a formula language and expression language used in Microsoft Power BI, Power Pivot, and Analysis Services. It is specifically designed for creating custom calculations, aggregations, and other data manipulations in these tools. DAX is similar to Excel functions but tailored for more complex and larger data models. DAX formulas can be used to create custom calculated columns, measures, and table-level expressions.

Features of Power BI

1. Data Connectivity

Power BI allows you to connect to a wide range of data sources, including databases, cloud services, spreadsheets, and web services. You can import, transform, and load data from various sources to create meaningful visualizations.

2. Data Transformation

With Power BI's Power Query, you can clean, shape, and transform your data to suit your analytical needs. This ETL (Extract, Transform, Load) functionality enables you to handle data preparation efficiently.

3. Data Modeling

Power BI provides a robust data modeling feature that allows you to create relationships between tables, define hierarchies, and optimize your data structure for analysis.

4. Visualization

Create interactive and insightful visualizations with a variety of chart types, maps, and tables. Power BI offers extensive customization options to make your reports visually appealing and informative.

5. Natural Language Query

You can use natural language queries to ask questions about your data. Power BI's Q&A feature can understand your questions and provide answers in the form of visuals.

6. DAX Formulas

DAX (Data Analysis Expressions) is a powerful formula language for creating custom calculations, measures, and calculated columns. It enables you to perform advanced calculations on your data, such as aggregations, time intelligence, and more.

7. Collaboration and Sharing

Power BI allows you to collaborate with others by sharing reports and dashboards. You can publish your reports to the Power BI service or embed them in other applications.

8. Mobile Accessibility

Access your reports and dashboards on the go using Power BI's mobile app. It provides a seamless experience for users on smartphones and tablets.

Writing DAX Formulas

DAX formulas in Power BI are written using a combination of functions and operators. Here's a basic syntax for writing DAX formulas:

```
Result = FunctionName([Column1], [Column2], ...)
```

- **Result**: This is the name you give to the calculated column or measure you're creating.

- **FunctionName**: The name of the DAX function you want to use for your calculation (e.g., SUM, AVERAGE, COUNT, etc.).

- **[Column1], [Column2], ...**: These are the columns or values you want to use as inputs for your function. You can reference columns in your data model or provide explicit values.

Here's an example of a simple DAX formula to calculate the total sales:

```
Total Sales = SUM(Sales[SalesAmount])
```

In this formula:

- "Total Sales" is the name of the calculated measure.
- "SUM" is the DAX function used to sum the values.
- "Sales[SalesAmount]" references the "SalesAmount" column in the "Sales" table.

Refernce :

- [Power BI documentation.](#)

10. Demonstrate the following DAX formulas with a context:

let's demonstrate these DAX formulas in the context of analyzing sales data for a fictional retail company.

Assume we have a Power BI data model with the following tables:

1. Sales: Contains information about each sale, including the sales date, product ID, quantity, and total sales amount.

2. Products: Contains details about the products, including product name and category.

Now, let's use the mentioned DAX functions in various scenarios:

1. **COUNT**: Calculate the total number of sales transactions.

```
Total Sales Count = COUNTROWS(Sales)
```

2. **DISTINCTCOUNT**: Calculate the total number of distinct products sold.

```
Unique Products Sold = DISTINCTCOUNT(Sales[Product ID])
```

3. **SUM**: Calculate the total sales amount for all transactions.

```
Total Sales Amount = SUM(Sales[Sales Amount])
```

4. **AVERAGE**: Calculate the average sales amount per transaction.

```
Average Sales Amount = AVERAGE(Sales[Sales Amount])
```

5. **MIN and MAX**: Find the minimum and maximum sales amounts.

```
Minimum Sales Amount = MIN(Sales[Sales Amount])
Maximum Sales Amount = MAX(Sales[Sales Amount])
```

6. **SUMMARIZE**: Create a summary table to aggregate sales data by product category.

```
SummaryTable = SUMMARIZE(
    Sales,
    Products[Category],
    "Total Sales", SUM(Sales[Sales Amount]),
    "Total Quantity", SUM(Sales[Quantity])
)
```

7. **CALCULATE**: Calculate the total sales amount for a specific product category (e.g., "Electronics").

```
Electronics Sales = CALCULATE(SUM(Sales[Sales Amount]), Products[Category] =
"Electronics")
```

8. **IF**: Create a calculated column that categorizes sales as "High" or "Low" based on a threshold (e.g., $500).

```
Sales Category = IF(Sales[Sales Amount] > 500, "High", "Low")
```

9. **IFERROR**: Handle errors when calculating the profit margin for each sale.

```
Profit Margin = IFERROR((Sales[Sales Amount] - Sales[Cost]) / Sales[Sales Amount],
0)
```

10. **ISBLANK**: Calculate the count of sales transactions with missing product IDs.

```
Missing Product ID Count = COUNTROWS(FILTER(Sales, ISBLANK(Sales[Product ID])))
```

11. **EOMONTH**: Calculate the last day of the current month for a given sales date.

```
Last Day of Current Month = EOMONTH(MAX(Sales[Sales Date]), 0)
```

12. **DATEDIFF**: Calculate the number of days between the order date and the ship date for each sale.

```
Days to Ship = DATEDIFF(Sales[Order Date], Sales[Ship Date], DAY)
```

Refernce :

- DAX functions - Microsoft Docs

11. Describe the terminology used in Power BI with a example/use case:

1. **Table**:
   - **Definition**: A table in Power BI is a collection of data. It can represent various data entities, such as sales data, product information, or customer details.
   - **Example**: The "Sales" table contains information about sales transactions, including columns for date, product, quantity, and sales amount.

2. **Fact**:

- **Definition**: A fact table in Power BI contains numerical data (facts) that are typically used for measurements or calculations. It often includes foreign keys to link to dimension tables.
- **Example**: The "Sales" table is a fact table that contains sales amounts and quantities.

3. **Dimension**:

- **Definition**: A dimension table in Power BI stores descriptive data that can be used to categorize, filter, or group facts in a fact table.
- **Example**: The "Products" table is a dimension table that provides information about products, such as product names, categories, and suppliers.

4. **Calendar**:

- **Definition**: A calendar table is a special type of dimension table used to track and analyze date-related data, such as days, months, and years.
- **Example**: The "Calendar" table contains a date column, allowing users to analyze sales data over time.

5. **Relationship**:

- **Definition**: A relationship in Power BI defines how tables are connected or related to each other, typically using keys, to enable cross-table analysis and filtering.
- **Example**: The "Sales" table has a relationship with the "Products" table through the "Product ID" key, enabling users to analyze sales by product.

6. **One to Many, One to One, Many to Many**:

- **Definition**: These terms describe the types of relationships between tables. One to Many means one row in one table can relate to multiple rows in another table, One to One means one row in one table relates to only one row in another table, and Many to Many implies that many rows in one table relate to many rows in another.
- **Example**: A one-to-many relationship exists between "Customers" and "Sales" because one customer can have multiple sales transactions.

7. **Keys - Primary & Foreign Keys**:

- **Definition**: Primary keys are unique identifiers in a table, while foreign keys in another table reference those primary keys to establish relationships.
- **Example**: The "Product ID" in the "Products" table is a primary key, and the "Product ID" in the "Sales" table is a foreign key.

8. **Star Schema**:

- **Definition**: A star schema is a data modeling technique where one central fact table is connected to multiple dimension tables. It simplifies querying and improves performance.
- **Example**: In a star schema, the "Sales" fact table is connected to dimension tables like "Products," "Customers," and "Calendar."

9. **Snowflake Schema**:

- **Definition**: A snowflake schema is an extension of the star schema, where dimension tables are normalized into sub-dimensions. It's used to reduce redundancy in dimension data.
- **Example**: In a snowflake schema, the "Products" dimension table might be normalized into "Categories" and "Suppliers" sub-dimension tables.

10. **Measures**:

- **Definition**: Measures are calculations applied to the data, such as sums, averages, or ratios. They provide insights into the facts.
- **Example**: A "Total Sales" measure calculates the sum of all sales amounts in the "Sales" table.

11. **Values, Aggregation**:

- **Definition**: Values are individual data points, while aggregation involves summarizing values using functions like SUM, AVERAGE, COUNT, etc.
- **Example**: In a sales table, individual sales amounts are values, and the "Total Sales" measure is an aggregation.

12. **Data Modeling**:

- **Definition**: Data modeling is the process of designing how data tables and relationships should be structured to facilitate analysis.
- **Example**: Creating relationships between the "Sales" and "Products" tables is part of data modeling in Power BI.

13. **Slicer**:

- **Definition**: A slicer is a visual element in a report that allows users to filter data interactively.
- **Example**: A slicer for product categories lets users filter sales data by selecting specific categories.

14. **Filter**:

- **Definition**: A filter restricts the data displayed in a report to specific criteria, helping focus on relevant information.
- **Example**: Applying a filter to show only sales from a particular region.

15. **Query**:

- **Definition**: A query is a request for data from a data source or the Power BI model, often involving data transformation.
- **Example**: Writing a query to extract sales data from a database and load it into Power BI.

16. **ETL (Extract, Transform, Load)**:

- **Definition**: ETL is the process of extracting data from various sources, transforming it to meet analysis requirements, and loading it into a data model.
- **Example**: Extracting sales data from a CSV file, transforming it to calculate total sales, and loading it into Power BI.

17. **Batch**:

- **Definition**: A batch refers to a group of data or changes processed together in data loading or refresh operations.
- **Example**: Running a nightly batch job to update the data in the Power BI model with the latest sales data.

18. **Data Pipeline**:

- **Definition**: A data pipeline is a series of data processing steps, often used in ETL processes, to move and transform data from source to destination.
- **Example**: Designing a data pipeline to ingest data from multiple sources, clean and transform it, and load it into a data warehouse or Power BI.

19. **Source**:

- **Definition**: A source is the origin of data, such as a database, file, or web service, from which data is extracted for analysis.
- **Example**: The source of sales data can be a SQL database, an Excel spreadsheet, or a web API.

20. **Refresh**:

- **Definition**: Refreshing data involves updating the dataset with the latest data from the source. This is essential to keep reports up to date.
- **Example**: Scheduling regular data refresh to ensure that the Power BI reports always show the most recent sales data.

Refernce :

- [Power BI Glossary - Microsoft Docs](#)