

# Power BI Data Modeling Exercise: Sales Analytics

## Exercise Overview

This exercise is broken down into several steps that coordinate with the lessons. Please only complete the step indicated by the instructor.

Beginning on page 13 you will find the same instructions with more guidance attached.

In this exercise, you will import five CSV files into Power BI and build a relational data model using best practices. You will create relationships between tables, build measures, and develop visualizations to analyze sales performance.

**Estimated Time:** 90-120 minutes

**Difficulty Level:** Intermediate

---

## Dataset Description

You have been provided with five CSV files representing sales data for a retail company:

### Dimension Tables

#### 1. Calendar.csv

- **Purpose:** Date dimension table for time-based analysis
- **Key Fields:**
  - Date (Primary Key)
  - Year, Quarter, Month, MonthName
  - Week, DayOfWeek, DayName
  - IsWeekend, FiscalYear, FiscalQuarter
- **Date Range:** January 1, 2022 to December 31, 2024

#### 2. Customers.csv

- **Purpose:** Customer dimension with demographic information
- **Key Fields:**
  - Customer\_Number (Primary Key)
  - Column1 (Customer Name), Email, Segment
  - GeographyID (Foreign Key)
  - Company, CustomerSince

- **Segments:** Consumer, Corporate, Home Office
- **Records:** 500 customers

### 3. Geography.csv

- **Purpose:** Geographic hierarchy for location-based analysis
- **Key Fields:**
  - Geo\_id (Primary Key)
  - Country, Region, State, StateCode
- **Coverage:** 12 US states across 4 regions
- **Regions:** East, West, Central, South

### 4. Products.csv

- **Purpose:** Product catalog with pricing information
- **Key Fields:**
  - ProductID (Primary Key)
  - ProductName, Category, SubCategory
  - UnitCost, UnitPrice
  - ProductCode, Manufacturer
- **Categories:** Technology, Furniture, Office Supplies
- **Records:** 150 products

## Fact Table

### 5. Sales\_Fact.csv

- **Purpose:** Transactional sales data (fact table)
  - **Key Fields:**
    - OrderID, LineItemID
    - OrderDate (Foreign Key to Calendar)
    - ShipDate, ShipMode
    - CustomerID (Foreign Key to Customers)
    - ProductID (Foreign Key to Products)
    - GeographyID (Foreign Key to Geography)
    - Quantity, UnitPrice, Discount
    - Sales, Profit, OrderStatus
  - **Ship Modes:** Standard, First Class, Express, Same Day
  - **Order Status:** Shipped, Returned, Cancelled
  - **Records:** ~10,000 line items
-

## Part 1: Data Import (15 minutes)

### Step 1: Import CSV Files

1. Open Power BI Desktop
2. Click **Get Data** → **Text/CSV**
3. Import each of the five CSV files:
  - Calendar.csv
  - Customers.csv
  - Geography.csv
  - Products.csv
  - Sales\_Fact.csv

### Step 2: Understand the Star Schema

This dataset follows a **star schema** design:

- **Fact Table (Center):** Sales\_Fact contains measurable transactions
- **Dimension Tables (Points):** Calendar, Customers, Geography, Products provide context

### Step 3: Create Relationships

Navigate to **Model View** in Power BI and create (or verify) the following relationships:

**Relationship 1: Sales\_Fact to Calendar**

**Relationship 2: Sales\_Fact to Customers**

**Relationship 3: Sales\_Fact to Products**

**Relationship 4: Sales\_Fact to Geography (Should you really do this?)**

**Relationship 5: Customers to Geography**

### Step 4: Validate the Model

Check that your model resembles a star schema with:

- Sales\_Fact in the center
- Dimension tables around the perimeter
- All relationships pointing from fact to dimensions (many-to-one)
- No circular dependencies

✓ **CHECKPOINT:** Your model should now have 5 tables and 5 relationships

## **Part 2: Configure the Data Model (15 minutes)**

### **Step 1: Data Type Validation**

For each table, verify and correct data types:

**Calendar Table:**

**Customers Table:**

**Geography Table:**

**Products Table:**

**Sales\_Fact Table:**

### **Step 2: Set Table Properties**

1. Mark the **Calendar** table as a Date Table:
2. Hide unnecessary columns from Report View:
3. Organize tables by type:

## Part 3: Create Calculated Columns & Measures (50 minutes)

### Step 1: Create Calculated Columns

Add the following calculated columns to enhance your data model. When needed, use Report View to create simple visualizations to test your results.

#### In the Calendar Table:

##### Year-Month

```
Format Calendar Date to display as YYYY-MM
```

##### Quarter-Year

```
Quarter-Year: QQ-YYYY
```

#### In the Products Table:

##### Profit Margin %

```
Profit Margin % = price - cost/price  
Round to 2 decimal positions
```

##### Price Tier

```
Price Tier =  
    UnitPrice < 100, "Low",  
    UnitPrice < 300, "Medium",  
    "High"
```

#### In the Sales\_Fact Table:

##### Days to Ship

```
Days to Ship = days between order date and ship date
```

##### Discount Tier

```
Discount Tier =  
    Discount = 0, "No Discount",  
    Discount <= 0.10, "Up to 10%",  
    Discount <= 0.20, "11-20%",  
    "Over 20%"  
)
```

---

## Step 2: Create Calculated Measures

Create a new table called **Measures** for organizing your DAX measures:

### Basic Measures

#### Total Sales

Total Sales

#### Total Quantity

Total Quantity

#### Total Orders

Total Orders

#### Total Customers

Total Customers

### Advanced Measures

#### Average Order Value

Average Order Value = sales/orders

#### Margin %

Profit Margin % = profit/sales

#### Return Rate

Return Rate orders returned /orders

#### Customer

Sales per Customer = sales/# of customers

### Intelligence Measures

#### Previous Year Sales

#### YoY Sales Growth %

#### YTD Sales

## Part 4: Build Visualizations (25 minutes)

Create a new report page and build the following visualizations. If necessary, adjust data types (e.g., Year in the calendar table is set to Sum. Will we ever aggregate Year?)

### Dashboard Layout

**1. KPI Cards (Top of page)** Create four card visuals showing:

- Total Sales
- Total Profit
- Profit Margin %
- Total Orders

**2. Sales by Category (Clustered Column Chart)**

- X-axis: Products[Category]
- Y-axis: [Total Sales]
- Add data labels

**3. Sales Trend Over Time (Line Chart)**

- X-axis: Calendar[Year]
- Y-axis: [Total Sales]
- Legend: Products[Category]

**4. Top 10 Products by Profit (Bar Chart)**

- Y-axis: Products[ProductName]
- X-axis: [Total Profit]
- Add filter: Top 10 by [Total Profit]

**5. Sales by Region (Map or Donut Chart)**

- Location: Geography[State]
- Values: [Total Sales]

**6. Customer Segment Performance (Clustered Column Chart)**

- X-axis: Customers[Segment]
- Y-axis: [Total Sales], [Total Profit]

**7. Monthly Profit Margin Trend (Line and Clustered Column Chart)**

- X-axis: Calendar[MonthName]
- Column Y-axis: [Total Sales]

- Line Y-axis: [Profit Margin %]

## Add Interactivity

1. Add a **Date Slicer** using Calendar[Year]
  2. Add a **Category Slicer** using Products[Category]
  3. Add a **Region Slicer** using Geography[Region]
  4. Test cross-filtering between visuals
-



## Part 5: Analysis Questions (Bonus)

Use your completed model and visualizations to answer these questions:

### Basic Analysis

1. What is the total sales for 2023?
2. Which product category generates the most profit?
3. Which customer segment has the highest average order value?
4. What is the overall profit margin percentage?
5. Which region has the highest sales?

### Intermediate Analysis

6. What is the return rate for each ship mode?
7. Which month of the year typically has the highest sales?
8. What is the relationship between discount and profit margin?
9. Which manufacturers have products with the highest profit margins?
10. How many days on average does it take to ship orders?

### Advanced Analysis

11. What is the year-over-year growth rate for 2024 compared to 2023?
12. Which products are underperforming (low sales, low profit margin)?
13. Are there any seasonal patterns in the data?
14. Which customer segments are most sensitive to discounts?
15. What is the customer retention rate? (Hint: Compare customers who ordered in multiple years)

---

## Validation Checklist

Before submitting your work, verify:

### Data Model

- ☐ All 5 tables are imported and loaded
- ☐ All data types are correct
- ☐ 5 relationships are created with correct cardinality
- ☐ Calendar table is marked as a date table
- ☐ Model resembles a star schema
- ☐ No circular dependencies exist

### Calculations

- ☐ All 6 calculated columns are created
- ☐ All 12 measures are created
- ☐ Measures are organized in a dedicated table
- ☐ Measures use proper DAX syntax and formatting

## Report

- ☐ At least 7 visualizations are created
- ☐ KPI cards show correct totals
- ☐ Charts have proper titles and labels
- ☐ Slicers are functioning correctly
- ☐ Cross-filtering is enabled between visuals

---

## Challenge Extensions (Optional)

For students who complete the exercise early:

### Advanced DAX

1. Create a measure for Customer Lifetime Value
2. Build a cohort analysis showing customer acquisition by quarter
3. Create a dynamic "Top N" measure using parameters
4. Build a what-if parameter for scenario analysis

### Additional Visualizations

5. Create a waterfall chart showing profit breakdown by category
6. Build a decomposition tree for sales analysis
7. Create a key influencers visual to identify factors affecting profit
8. Design a second dashboard page for executive summary

### Data Model Enhancements

9. Create a role-playing dimension for ShipDate vs OrderDate
10. Build a bridge table for analyzing multiple subcategories
11. Create a parameter table for dynamic measure selection
12. Implement row-level security by region

---

## Common Troubleshooting

**Issue: Relationships won't create automatically**

**Solution:** Ensure data types match between related columns. CustomerID and ProductID must be the same data type in both tables.

**Issue: Circular dependency error**

**Solution:** Check that you haven't created bidirectional relationships that form a loop. Keep all cross-filter directions as "Single" initially.

**Issue: Time intelligence measures return blank**

**Solution:** Ensure the Calendar table is marked as a date table and has continuous dates without gaps.

**Issue: Totals don't match expectations**

**Solution:** Check for duplicate relationships, verify that measures use SUM or COUNT appropriately, and check filter context.

**Issue: Visualizations show wrong data**

**Solution:** Verify that relationships are active and have correct cardinality. Check that foreign keys match primary keys.

---

**The following pages provide more assistive instructions.**

# Part 1: Data Import (15 minutes)

## Step 1: Import CSV Files

4. Open Power BI Desktop
5. Click **Get Data** → **Text/CSV**
6. Import each of the five CSV files:
  - Calendar.csv
  - Customers.csv
  - Geography.csv
  - Products.csv
  - Sales\_Fact.csv

## Step 2: Understand the Star Schema

This dataset follows a **star schema** design:

- **Fact Table (Center):** Sales\_Fact contains measurable transactions
- **Dimension Tables (Points):** Calendar, Customers, Geography, Products provide context

## Step 3: Create Relationships

Navigate to **Model View** in Power BI and create (or verify) the following relationships:

### Relationship 1: Sales\_Fact to Calendar

- **From:** Sales\_Fact[OrderDate]
- **To:** Calendar[Date]
- **Cardinality:** Many to One (\*:1)
- **Cross Filter Direction:** Single
- **Active:** Yes

### Relationship 2: Sales\_Fact to Customers

- **From:** Sales\_Fact[CustomerID]
- **To:** Customers[CustomerID]
- **Cardinality:** Many to One (\*:1)
- **Cross Filter Direction:** Single
- **Active:** Yes

### Relationship 3: Sales\_Fact to Products

- **From:** Sales\_Fact[ProductID]
- **To:** Products[ProductID]
- **Cardinality:** Many to One (\*:1)

- **Cross Filter Direction:** Single
- **Active:** Yes

#### **Relationship 4: Sales\_Fact to Geography (Should you really do this?)**

- **From:** Sales\_Fact[GeographyID]
- **To:** Geography[GeographyID]
- **Cardinality:** Many to One (\*:1)
- **Cross Filter Direction:** Single
- **Active:** Yes

#### **Relationship 5: Customers to Geography**

- **From:** Customers[GeographyID]
- **To:** Geography[GeographyID]
- **Cardinality:** Many to One (\*:1)
- **Cross Filter Direction:** Single
- **Active:** Yes

### **Step 4: Validate the Model**

Check that your model resembles a star schema with:

- Sales\_Fact in the center
- Dimension tables around the perimeter
- All relationships pointing from fact to dimensions (many-to-one)
- No circular dependencies

✓ **CHECKPOINT:** Your model should now have 5 tables and 5 relationships

## Part 2: Configure the Data Model (15 minutes)

### Step 1: Data Type Validation

For each table, verify and correct data types:

#### Calendar Table:

- Date → Date
- Year → Whole Number
- Month → Whole Number
- Week → Whole Number
- DayOfWeek → Whole Number
- IsWeekend → Whole Number (0 or 1)
- All other text fields → Text

#### Customers Table:

- Rename Column1 to Customer Name
- CustomerID → Text
- GeographyID → Whole Number
- CustomerSince → Date
- All other fields → Text

#### Geography Table:

- GeographyID → Whole Number
- All other fields → Text

#### Products Table:

- ProductID → Whole Number
- UnitCost → Decimal Number
- UnitPrice → Decimal Number
- All other fields → Text

#### Sales\_Fact Table:

- OrderID → Text
- LineItemID → Whole Number
- OrderDate → Date
- ShipDate → Date
- CustomerID → Text

- ProductID → Whole Number
- GeographyID → Whole Number
- Quantity → Whole Number
- UnitPrice → Decimal Number
- Discount → Decimal Number
- Sales → Decimal Number
- Profit → Decimal Number
- ShipMode → Text
- OrderStatus → Text

## Step 2: Set Table Properties

4. Mark the **Calendar** table as a Date Table:
  - Right-click Calendar table → **Mark as date table**
  - Select Date column as the date column
5. Hide unnecessary columns from Report View:
  - Hide GeographyID in Customers table
  - Hide GeographyID in Sales\_Fact table
  - Hide foreign key columns that users don't need to see
6. Organize tables by type:
  - Create a **Measures** table for all calculated measures
  - Keep dimension tables on one side of the model
  - Keep fact table in the center



## Part 3: Create Calculated Columns & Measures (50 minutes)

### Step 1: Create Calculated Columns

Add the following calculated columns to enhance your data model. When needed, use Report View to create simple visualizations to test your results.

#### In the Calendar Table:

##### 1. Year-Month

```
Year-Month =  
FORMAT(Calendar[Date], "YYYY-MM")
```

##### 2. Quarter-Year

```
Quarter-Year =  
Calendar[Quarter] & " " & FORMAT(Calendar[Year], "0000")
```

#### In the Products Table:

##### 3. Profit Margin %

```
Profit Margin % =  
Round(DIVIDE(  
    Products[UnitPrice] - Products[UnitCost],  
    Products[UnitPrice],  
    0  
) * 100, 2)
```

##### 4. Price Tier

```
Price Tier =  
SWITCH(  
    TRUE(),  
    Products[UnitPrice] < 100, "Low",  
    Products[UnitPrice] < 300, "Medium",  
    "High"  
)
```

#### In the Sales\_Fact Table:

##### 5. Days to Ship

```
Days to Ship =  
DATEDIFF(Sales_Fact[OrderDate], Sales_Fact[ShipDate], DAY)
```

## 6. Discount Tier

```
Discount Tier =  
SWITCH(  
    TRUE(),  
    Sales_Fact[Discount] = 0, "No Discount",  
    Sales_Fact[Discount] <= 0.10, "Up to 10%",  
    Sales_Fact[Discount] <= 0.20, "11-20%",  
    "Over 20%"  
)
```

---

## Step 2: Create Calculated Measures

Create a new table called **Measures** for organizing your DAX measures:

### Basic Measures

#### 1. Total Sales

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

#### 2. Total Quantity

```
Total Quantity = SUM(Sales_Fact[Quantity])
```

#### 4. Total Orders

```
Total Orders = DISTINCTCOUNT(Sales_Fact[OrderID])
```

#### 5. Total Customers

```
Total Customers = DISTINCTCOUNT(Sales_Fact[CustomerID])
```

### Advanced Measures

#### 6. Average Order Value

```
Average Order Value =  
DIVIDE(  
    [Total Sales],  
    [Total Orders],  
    0  
)
```

#### 7. Profit Margin %

```
Profit Margin % =  
DIVIDE(  
    [Total Profit],
```

```
    [Total Sales],  
    0  
  ) * 100
```

## 8. Return Rate

```
Return Rate =  
DIVIDE(  
    CALCULATE([Total Orders], Sales_Fact[OrderStatus] = "Returned"),  
    [Total Orders],  
    0  
  ) * 100
```

## 9. Sales per Customer

```
Sales per Customer =  
DIVIDE(  
    [Total Sales],  
    [Total Customers],  
    0  
  )
```

## Time Intelligence Measures

### 10. Previous Year Sales

```
Previous Year Sales =  
CALCULATE(  
    [Total Sales],  
    SAMEPERIODLASTYEAR(Calendar[Date])  
  )
```

### 11. YoY Sales Growth %

```
YoY Sales Growth % =  
DIVIDE(  
    [Total Sales] - [Previous Year Sales],  
    [Previous Year Sales],  
    0  
  ) * 100
```

### 12. YTD Sales

```
YTD Sales =  
TOTALYTD([Total Sales], Calendar[Date])
```

---

## Part 4: Build Visualizations (25 minutes)

Create a new report page and build the following visualizations. If necessary, adjust data types (e.g., Year in the calendar table is set to Sum. Will we ever aggregate Year?)

### Dashboard Layout

**1. KPI Cards (Top of page)** Create four card visuals showing:

- Total Sales
- Total Profit
- Profit Margin %
- Total Orders

**2. Sales by Category (Clustered Column Chart)**

- X-axis: Products[Category]
- Y-axis: [Total Sales]
- Add data labels

**3. Sales Trend Over Time (Line Chart)**

- X-axis: Calendar[Year]
- Y-axis: [Total Sales]
- Legend: Products[Category]

**4. Top 10 Products by Profit (Bar Chart)**

- Y-axis: Products[ProductName]
- X-axis: [Total Profit]
- Add filter: Top 10 by [Total Profit]

**5. Sales by Region (Map or Donut Chart)**

- Location: Geography[State]
- Values: [Total Sales]

**6. Customer Segment Performance (Clustered Column Chart)**

- X-axis: Customers[Segment]
- Y-axis: [Total Sales], [Total Profit]

**7. Monthly Profit Margin Trend (Line and Clustered Column Chart)**

- X-axis: Calendar[MonthName]
- Column Y-axis: [Total Sales]

- Line Y-axis: [Profit Margin %]

## **Add Interactivity**

5. Add a **Date Slicer** using Calendar[Year]
  6. Add a **Category Slicer** using Products[Category]
  7. Add a **Region Slicer** using Geography[Region]
  8. Test cross-filtering between visuals
-

## Part 5: Analysis Questions (Bonus)

Use your completed model and visualizations to answer these questions:

### Basic Analysis

6. What is the total sales for 2023?
7. Which product category generates the most profit?
8. Which customer segment has the highest average order value?
9. What is the overall profit margin percentage?
10. Which region has the highest sales?

### Intermediate Analysis

11. What is the return rate for each ship mode?
12. Which month of the year typically has the highest sales?
13. What is the relationship between discount and profit margin?
14. Which manufacturers have products with the highest profit margins?
15. How many days on average does it take to ship orders?

### Advanced Analysis

16. What is the year-over-year growth rate for 2024 compared to 2023?
17. Which products are underperforming (low sales, low profit margin)?
18. Are there any seasonal patterns in the data?
19. Which customer segments are most sensitive to discounts?
20. What is the customer retention rate? (Hint: Compare customers who ordered in multiple years)

---

## Validation Checklist

Before submitting your work, verify:

### Data Model

- ☐ All 5 tables are imported and loaded
- ☐ All data types are correct
- ☐ 5 relationships are created with correct cardinality
- ☐ Calendar table is marked as a date table
- ☐ Model resembles a star schema
- ☐ No circular dependencies exist

### Calculations

- ☐ All 6 calculated columns are created
- ☐ All 12 measures are created
- ☐ Measures are organized in a dedicated table
- ☐ Measures use proper DAX syntax and formatting

## Report

- ☐ At least 7 visualizations are created
- ☐ KPI cards show correct totals
- ☐ Charts have proper titles and labels
- ☐ Slicers are functioning correctly
- ☐ Cross-filtering is enabled between visuals

---

## Challenge Extensions (Optional)

For students who complete the exercise early:

### Advanced DAX

5. Create a measure for Customer Lifetime Value
6. Build a cohort analysis showing customer acquisition by quarter
7. Create a dynamic "Top N" measure using parameters
8. Build a what-if parameter for scenario analysis

### Additional Visualizations

9. Create a waterfall chart showing profit breakdown by category
10. Build a decomposition tree for sales analysis
11. Create a key influencers visual to identify factors affecting profit
12. Design a second dashboard page for executive summary

### Data Model Enhancements

13. Create a role-playing dimension for ShipDate vs OrderDate
  14. Build a bridge table for analyzing multiple subcategories
  15. Create a parameter table for dynamic measure selection
  16. Implement row-level security by region
-

# Common Troubleshooting

## **Issue: Relationships won't create automatically**

**Solution:** Ensure data types match between related columns. CustomerID and ProductID must be the same data type in both tables.

## **Issue: Circular dependency error**

**Solution:** Check that you haven't created bidirectional relationships that form a loop. Keep all cross-filter directions as "Single" initially.

## **Issue: Time intelligence measures return blank**

**Solution:** Ensure the Calendar table is marked as a date table and has continuous dates without gaps.

## **Issue: Totals don't match expectations**

**Solution:** Check for duplicate relationships, verify that measures use SUM or COUNT appropriately, and check filter context.

## **Issue: Visualizations show wrong data**

**Solution:** Verify that relationships are active and have correct cardinality. Check that foreign keys match primary keys.

---