

Category Germany India USA **Total**

Clothing	10369	14869	16095	41333
Electronics	9880	11354	21394	42628
Furniture	10545	16889	26981	54415
Total	30794	4311	6447	138376

Year

<input checked="" type="checkbox"/> 2022
<input type="checkbox"/> 2023

Fiscal Year Damaged Not Needed Wrong Item **Total**

FY2022	22	21	17	60
Total	22	21	17	60

Category

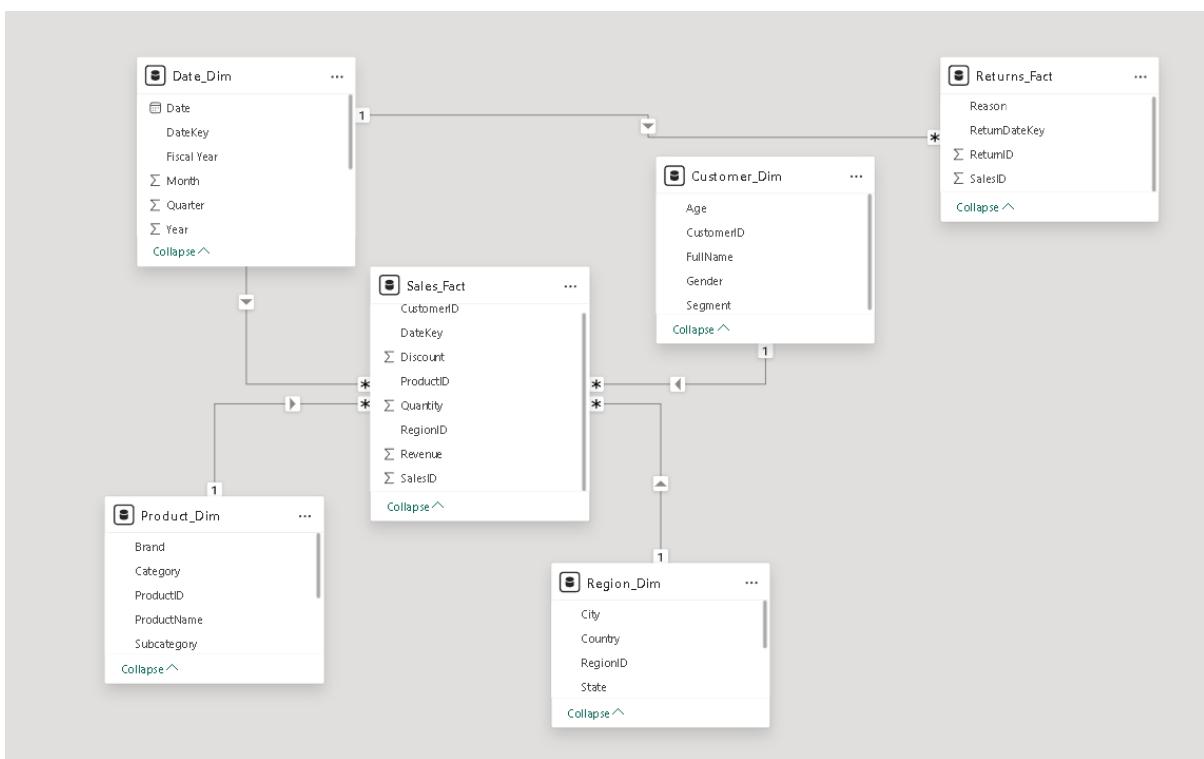
<input type="checkbox"/> Clothing
<input type="checkbox"/> Electronics
<input type="checkbox"/> Furniture

Segment Sum of Revenue

Gold	46155
Platinum	41021
Silver	51200
Total	138376

Country

<input type="checkbox"/> Germany
<input type="checkbox"/> India
<input type="checkbox"/> USA



Data Modeling Explanation Summary

1 Why a Star Schema Was Used

A **Star Schema** was chosen because it is the most efficient and widely accepted data modeling approach for analytical and reporting systems such as Power BI.

In this model:

- **Sales_Fact** acts as the central fact table containing transactional measures such as *Revenue, Quantity, and Discount*.
- All descriptive tables (**Customer_Dim, Product_Dim, Region_Dim, and Date_Dim**) connect directly to the fact table through **one-to-many relationships**.

This design provides:

- Simple and predictable filter flow
- Better query performance
- Easier drill-down analysis using hierarchies
- Clear separation between facts (numerical data) and dimensions (descriptive attributes)

The star schema ensures that business users can analyze sales data across multiple dimensions without creating complex or ambiguous relationship paths.

2 Why Returns_Fact Is Modeled as a Separate Table

Returns data represents a **different business process** than sales transactions.

While sales capture completed purchases, returns track:

- Returned items
- Return reasons
- Return dates
- References to original sales transactions

Keeping **Returns_Fact** as a separate fact table:

- Maintains data normalization
- Prevents unnecessary duplication of sales records
- Allows independent analysis of return behavior
- Supports scenarios where multiple returns can occur for a single sale

Returns_Fact is linked to Sales_Fact using **SalesID**, ensuring referential integrity while preserving analytical flexibility.

3 Why the ReturnDate Relationship Is Inactive

Both **Sales_Fact** and **Returns_Fact** connect to the same **Date_Dim**, but they represent **different date contexts**:

- Sales_Fact uses *Sale Date*
- Returns_Fact uses *Return Date*

If both relationships were active simultaneously, Power BI would face **ambiguous filter paths** when filtering by date.

To prevent this:

- The Sales_Fact → Date_Dim relationship remains **active**
- The Returns_Fact → Date_Dim relationship (ReturnDateKey) is set to **inactive**

This design:

- Avoids incorrect aggregations
- Preserves correct time-based analysis
- Demonstrates real-world modeling best practices for handling multiple date roles

The inactive relationship clearly communicates intent and ensures model stability.

4 Ambiguity Issues & How They Were Avoided

Potential ambiguity arises when:

- Multiple active relationships exist between the same tables
- Bidirectional filters are applied unnecessarily
- Multiple fact tables share common dimensions

These issues were avoided by:

- Using **single-direction filtering** across all relationships
- Maintaining a clear **star schema structure**
- Setting only one active date relationship per fact table
- Keeping Returns_Fact isolated as a secondary fact table
- Avoiding many-to-many relationships

This approach ensures:

- Predictable filter propagation
- Accurate aggregations
- No circular dependencies
- Clean and scalable model architecture