

## Schema Type Explanation – Star & Snowflake Schema

### Overview

In this project, a hybrid data model has been implemented using both Star Schema and Snowflake Schema concepts. The primary structure follows a Star Schema, while an additional fact table introduces a Snowflake extension.

### Star Schema Design

A Star Schema consists of a central fact table connected directly to multiple dimension tables. This design improves query performance and simplifies data analysis.

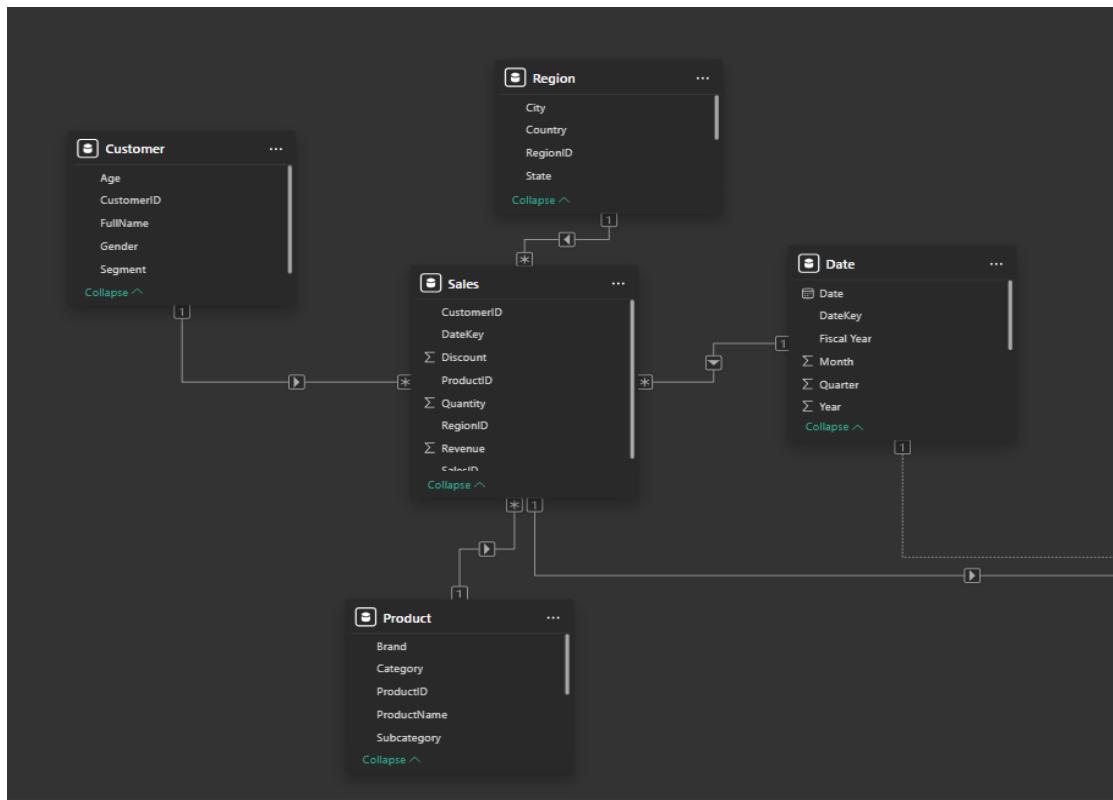
#### Fact Table

- Sales

#### Dimension Tables

- **Customer** – Customer details
- **Product** – Product category and product details
- **Region** – Geographic information
- **Date** – Date, month, quarter, and year information

#### Star Schema Structure



## Snowflake Schema Extension

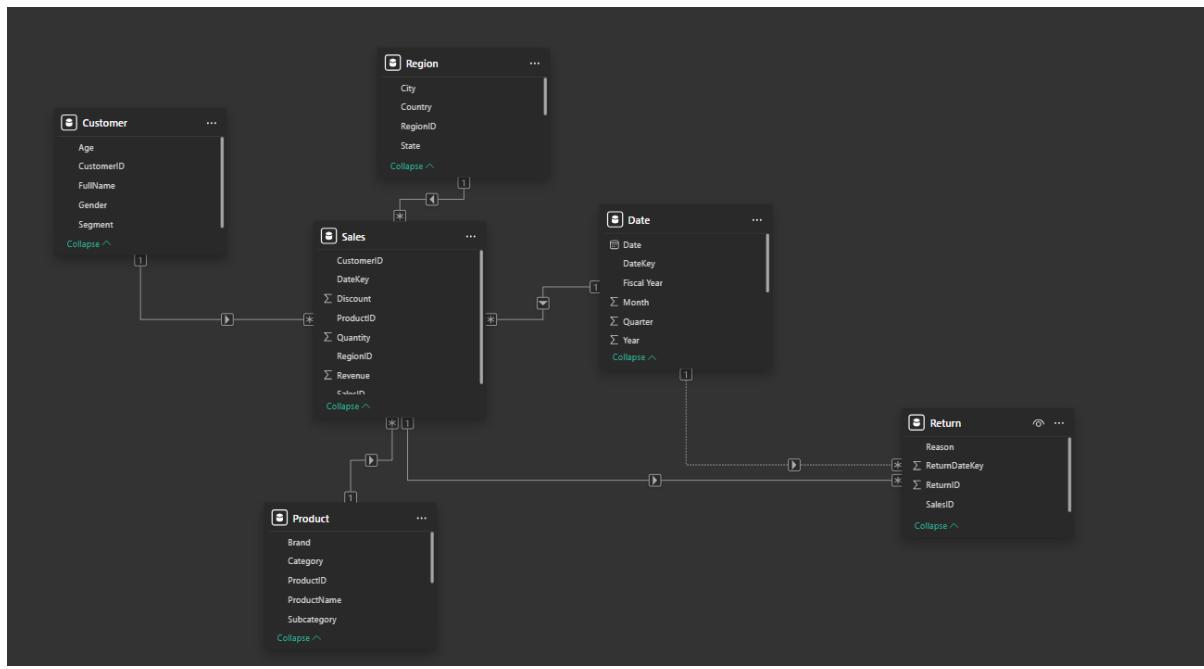
In addition to the star schema, the model includes a **secondary fact table**:

- **Returns**

The **Returns** table is connected to:

- **Sales** (via SalesID)
- **Date** (via ReturnDateKey)

## **Snowflake Structure**



## Relationships and Filter Flow

### **Relationships**

In this project, relationships are created between fact and dimension tables using primary and foreign keys. The data model follows a star schema design, where the central fact table is connected to multiple dimension tables.

### **List of Relationships**

From Table	To Table	Relationship Type	Cardinality	Status
Sales	Customer	CustomerID	Many-to-One	Active
Sales	Product	ProductID	Many-to-One	Active

<b>Sales</b>	Region	RegionID	Many-to-One	Active
<b>Sales</b>	Date	DateKey	Many-to-One	Active
<b>Returns</b>	Sales	SalesID	Many-to-One	Active
<b>Returns</b>	Date	ReturnDateKey	Many-to-One	Inactive

## Filter Flow Direction

### Filter Direction Used: Single Direction

- Filters flow from Dimension tables to Fact tables.
- This ensures:
  - Better performance
  - No ambiguity in relationships
  - Controlled and predictable filtering

### Example:

- Filtering by *Region* filters data in **Sales**
- Filtering by *Date* filters both **Sales** and **Returns**

### Inactive Relationship Explanation

- The relationship between **Returns** and **Date** is set as **Inactive**.
- This is done to avoid ambiguity caused by multiple date relationships.
- Inactive relationships are used only when needed (e.g., return date analysis).

## Issues Encountered and How They Were Resolved

During the development of the data model, several common challenges were encountered. Each issue was analysed and resolved using best practices in Power BI data modelling.

### 1. Relationship Ambiguity Issue

#### Problem:

Multiple relationships existed between the same tables (especially between Returns and Date), which caused ambiguity and incorrect filtering.

#### Solution:

- One relationship was set as **Active**, and the other was kept **Inactive**.

- The inactive relationship was used only when required for return date analysis.

This ensured correct filter behaviour without conflicts.

## 2. Incorrect Aggregation in Visuals

### Problem:

Some visuals were displaying incorrect or blank values when using text fields like *Reason* in the Values section.

### Solution:

- Replaced text fields with numeric fields such as ReturnID or SalesID.
- Used **Count** aggregation instead of text values.

This corrected the aggregation logic and displayed accurate results.

## 3. Incorrect Filter Flow Direction

### Problem:

Filters were not propagating correctly across tables due to incorrect filter direction.

### Solution:

- Set all relationships to **Single Direction**.
- Ensured filters flow from **Dimension tables to Fact tables** only.

This improved model performance and avoided ambiguity.

## 4. Hierarchy Not Working Properly

### Problem:

Drill-down functionality was not working correctly for date and product hierarchies.

### Solution:

- Created proper hierarchies (Year → Quarter → Month → Date).
- Ensured correct data types and sorting (e.g., Month sorted by Month Number).

Drill-down functionality worked correctly after correction.

## 5. Data Type Mismatch Issues

### Problem:

Some key columns had mismatched data types (Text vs Number), causing relationship failures.

### Solution:

- Standardized data types across all tables.
- Ensured primary and foreign keys had identical data types.

Relationships were established successfully.

## 6. Blank or Missing Values in Visuals

### Problem:

Some visuals showed blank results due to missing or unmatched keys.

### Solution:

- Verified matching keys between fact and dimension tables.
- Removed or corrected null values during data cleaning.

Visuals displayed complete and accurate data.

## Matrix

The screenshot shows a Power BI interface with a matrix visual on the left and the Data pane on the right.

**Matrix Visual Data:**

Year	Sum of Revenue
2023	148550
2022	138376
Total	286926

Country	Sum of Revenue
Germany	77000
India	85764
USA	124162
NY	31256
CA	29801
DL	18834
MH	16567
NY	19421
TX	8223
Total	286926

Category	Sum of Revenue
Clothing	92261
Jeans	23766
Laptop	9589
Mobile	24863
Shirt	19866
Sofa	3385
Table	10792
Electronics	102893
Jeans	17116
Laptop	8504
Mobile	27650
Shirt	12212
Total	286926

**Data Pane:**

- Visualizations:** Build visual, Filters, R, Py, ...
- Data:**
  - Customer:** CustomerID, DateKey, Discount, Quantity, RegionID, Revenue, SalesID
  - Date:** Brand, Category, ProductID, ProductName, Subcategory
  - Product:** ProductID, ProductName, Subcategory
  - Region:** RegionID
  - Return:** Sales
  - Sales:** CustomerID, DateKey, Discount, ProductID, RegionID, Revenue, SalesID

The screenshot shows four additional matrix visual examples below the main one.

**Matrix Visual 1:**

Year	Sum of Revenue
2023	60265
2022	42628
Total	102893

**Matrix Visual 2:**

Country	Sum of Revenue
USA	44712
India	31692
Germany	26489
Total	102893

**Matrix Visual 3:**

Category	Germany	India	USA	Total
Clothing	26521	27725	38015	92261
Electronics	26489	31692	44712	102893
Furniture	23990	26347	41435	91772
Total	77000	85764	124162	286926

**Matrix Visual 4:**

Segment	Sum of Revenue
Gold	31555
Platinum	31614
Silver	39724
Total	102893

**Matrix Visual 5:**

Fiscal Year	Count of ReturnID
FY2022	10
Damaged	1
Not Needed	4
Wrong Item	5
FY2023	19
Damaged	10
Not Needed	5
Wrong Item	4
Total	29