## Introduction

As a data analyst, you'll frequently use schemas when designing your data models. So, it's important that you understand the different types of schemas available, along with the advantages and disadvantages of each. With this knowledge, you can choose the schema that best suits the needs of your database and business objectives.

This reading provides a comprehensive guide to the different types of schemas and their advantages and disadvantages.

## What is a schema?

In the context of Power BI, a schema is a logical blueprint that defines the structure, organization, and relationships of tables.

An analyst's role is to facilitate querying and reporting by establishing meaningful and well-defined relationships between tables. By understanding different schema types, data analysts can choose the most suitable schema design for their specific use cases.

An analyst might also need to ensure that their schema is normalized. Normalization refers to the process of organizing and structuring data in a way that eliminates redundant data and improves data integrity. It involves breaking down larger tables into smaller, more focused tables and establishing relationships between them.

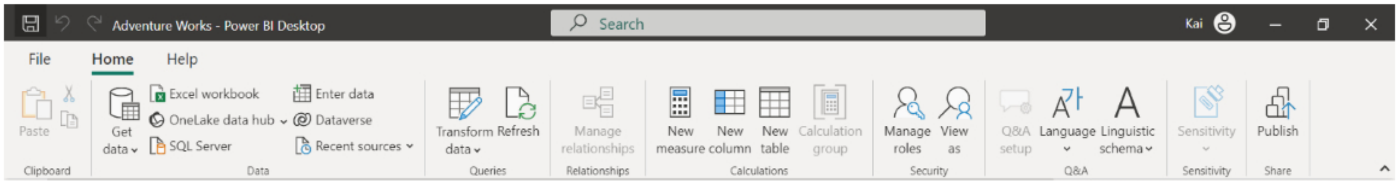
## Adventure Works schema

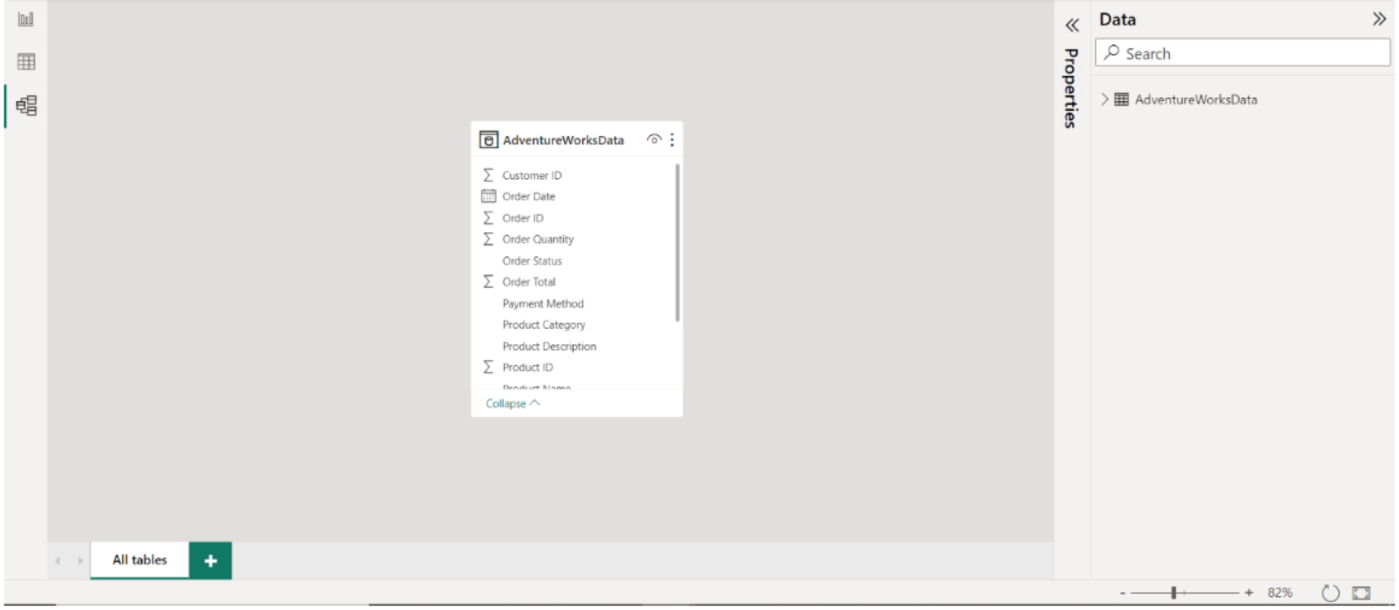
Adventure Works has decided to use Power BI for its data analytics. The company must design a schema with several interconnected tables to analyze and visualize its data effectively. Some of the tables in the company schema are as follows:

- **Reseller**: A table with reseller ID, contact information, and demographic data.
- **Regions**: Stores geographical location about the customers like region, country and city.
- **Sales**: Captures transaction data like date, sale amount, transaction ID, and quantities sold.
- **Products**: Stores data about various products, categories, and subcategories with Product ID.
- **Salesperson**: Records data about salespersons, employee ID, hiring data, and designation.

Let's review the components of various schema using Adventure Works' schema as our example.

## Flat schema





### Flat schema overview

A Flat schema is a simple database design where all data is stored in a single table. In this schema, each row represents a unique record, and each column represents the record's attributes.

For example, in the Adventure Works **Sales** table, each row represents a sales transaction, while the various columns specify who bought the product, the date on which it was sold, and so on. Since there is only one table in the schema, you don't need to manage relationships between different tables.

### Flat schema advantages

The main advantage of a Flat schema is its simplicity. Since all data is contained within one table, it's easy to understand, design, and maintain. Querying the database is straightforward, as no joins or complex queries are required.

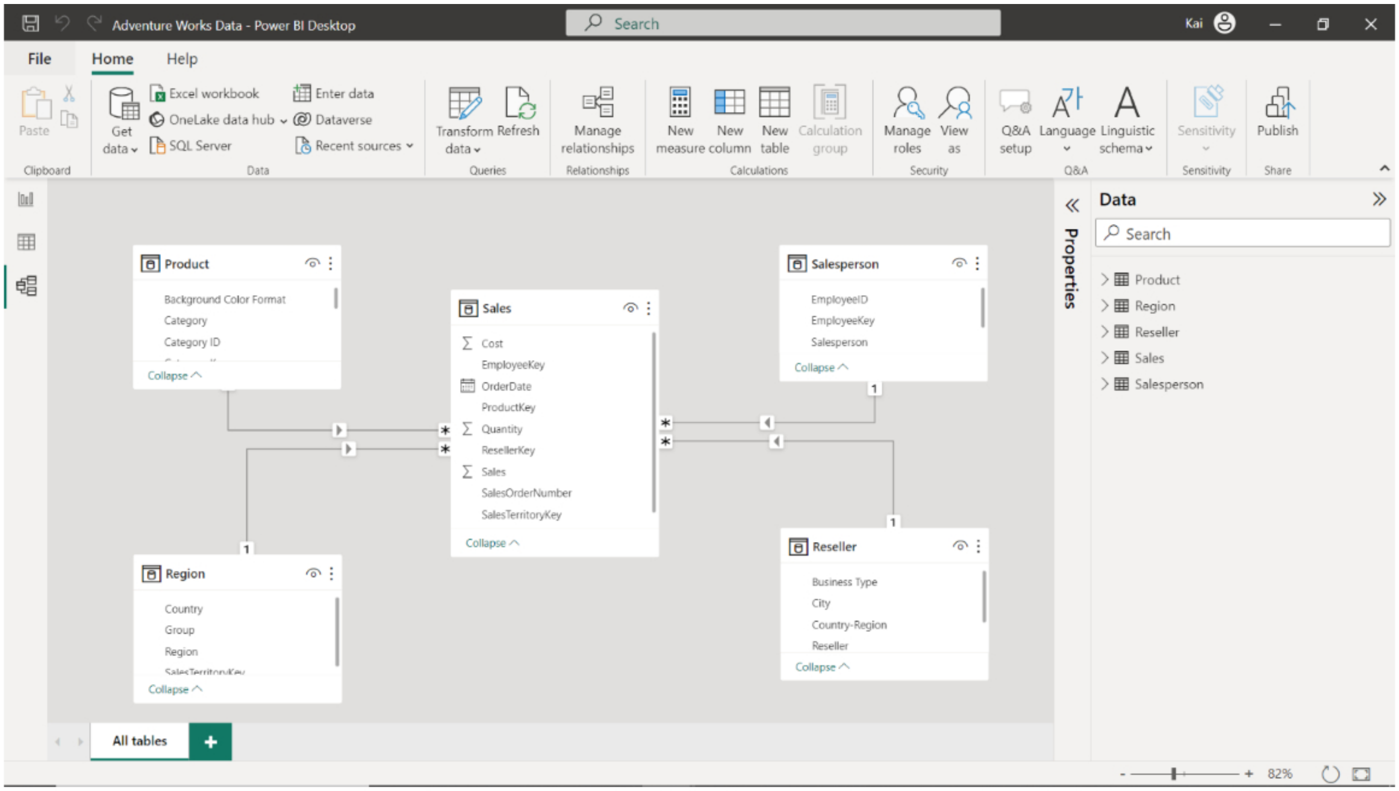
### Flat schema disadvantages

However, Flat schemas also come with limitations. They often result in data redundancy, as the same data may be repeated across multiple rows. This can lead to increased storage requirements and potential data inconsistencies.

Flat schemas are also unsuitable for handling and analyzing complex data and may perform poorly when dealing with large datasets. Some of the reasons for this are as follows:

- All attributes are combined into a single table. This means that there are multiple columns, which results in data clutter.
- The schema contains multiple records with repeated information giving rise to redundant data.
- It is difficult to establish meaningful relationships and hierarchies between data elements, so it's difficult to perform detailed analysis.

## Star schema



### Star schema overview

A Star schema is a type of schema used in data warehousing and dimensional modeling. In this schema, a central fact table is connected to one or more dimension tables based on the common field or column in both fact and dimension tables.

In the Adventure Works schema, the fact table contains quantitative data, such as sales amounts or product quantities. In contrast, the dimension tables store descriptive data, such as customer information, product details, or dates.

In the coming lessons, you'll learn more about fact and dimension tables and the relationships between them.

### Star schema advantages

The Star schema offers several advantages over the Flat schema.

One advantage is that it reduces data redundancy by separating facts and dimensions into distinct tables. This separation also allows for more efficient querying and aggregation of data, as dimension tables are typically smaller in size and can be indexed to improve query performance.

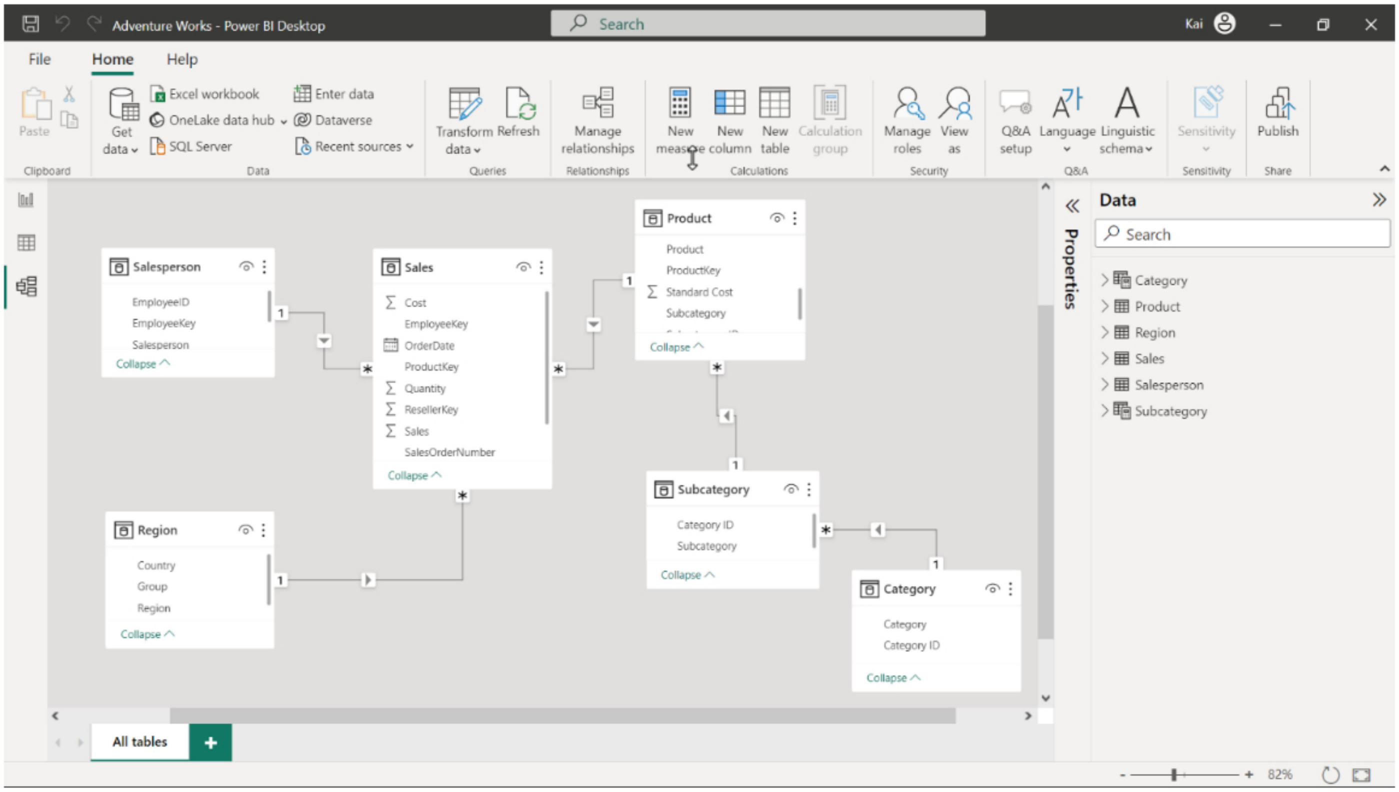
Furthermore, the Star schema is highly intuitive and easy to understand. The fact table serves as the focal point of the schema, and all other tables are linked to it through clear relationships.

### Star schema disadvantages

On the downside, the Star schema can be less flexible than other schema types. For example, actions like adding new dimensions or modifying existing ones may require extensive changes to the schema.

Also, the Star schema may not be suitable for handling complex relationships between dimensions. The Snowflake schema better accommodates greater levels of complexity.

## Snowflake schema



### Snowflake schema overview

A Snowflake schema is an extension of the Star schema. In this schema, dimension tables are split into multiple related tables to reduce data redundancy and improve data integrity. This process is referred to as normalization.

In the Adventure Works schema, each dimension table is connected to one or more related tables, forming a hierarchical structure that resembles a Snowflake.

### Snowflake schema advantages

The Snowflake schema offers several benefits, including reduced data redundancy and improved data integrity due to the normalization of dimension tables.

This schema also provides greater flexibility in handling complex relationships between dimensions, as new tables can be easily added to accommodate new relationships.

### Snowflake schema disadvantages

However, the Snowflake schema is more complex than the Star schema. This makes it harder to understand and maintain.

Additionally, querying a Snowflake schema can be more challenging and time-consuming. This is because multiple join operations may be required to retrieve data from the related tables.

## The importance of choosing the right schema

Understanding schemas and the available types is crucial for effective database design and management.

The Flat schema is simple and easy to work with but may not be suitable for complex data relationships or large datasets.

The Star schema is a popular choice for dimensional modeling, offering reduced data redundancy and intuitive design. Although it may be less flexible when handling complex relationships between dimensions.

Lastly, the Snowflake schema provides greater flexibility and improved data integrity through normalizing dimension tables. However, its complexity can make it more challenging to understand and query.