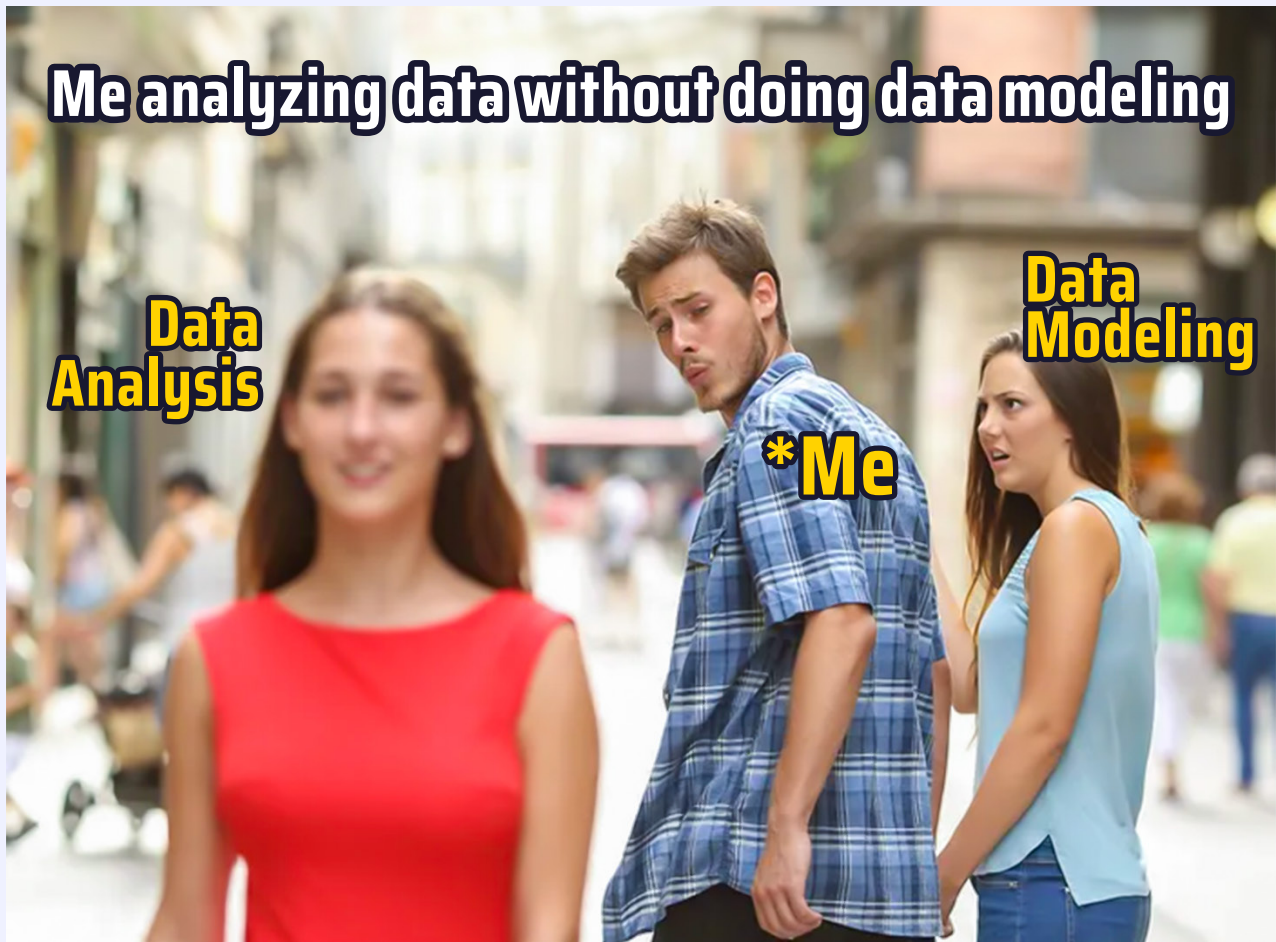


# Data Modeling

**The Foundation of Data**

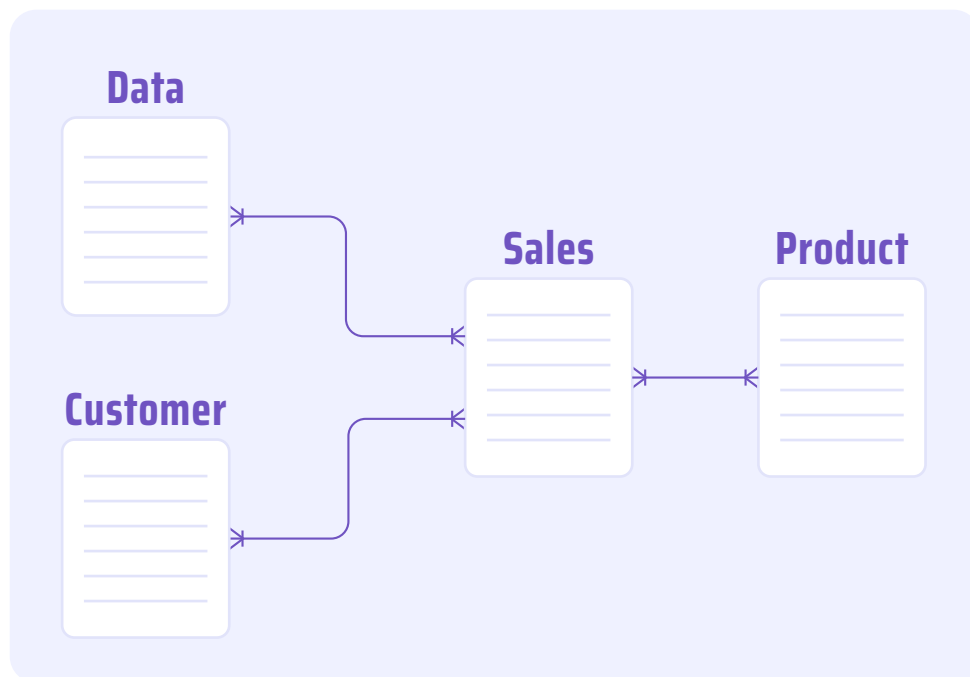
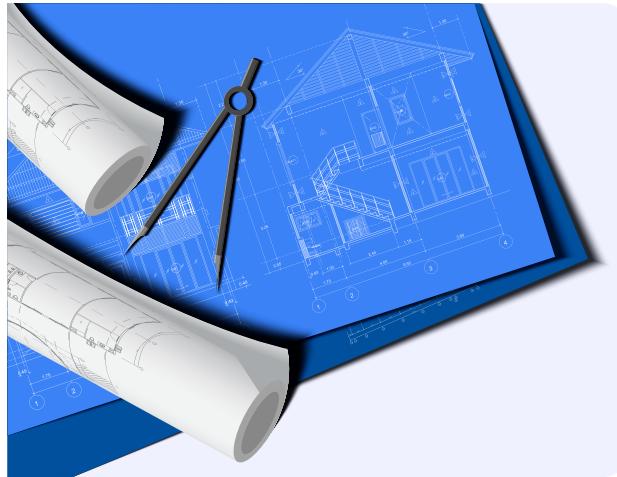




Looking at the meme above, you might understand the importance of data modeling.

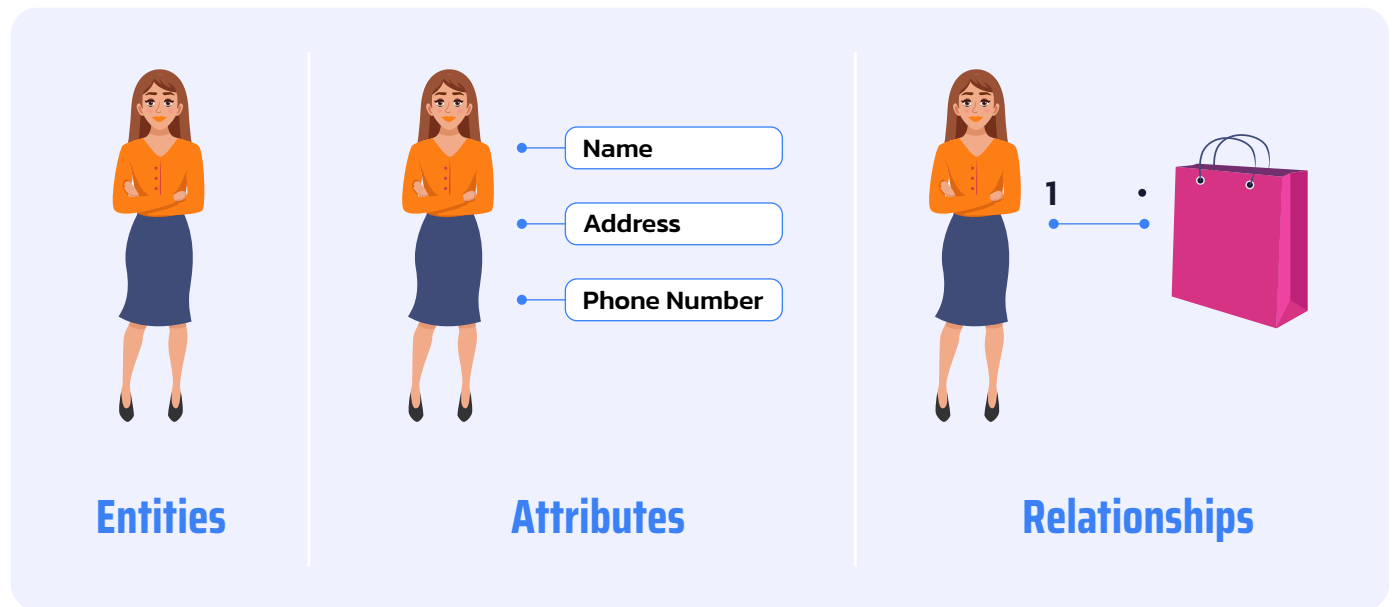
Let's dive into the topic 🤪

# DATA MODELING:



- **Data modeling is an important step in designing and building a database.**
- Just like a blueprint visualizes the plan and details for a house, data modeling creates a visual representation of data entities and the relationships between data elements.

# COMPONENTS OF DATA MODELING:



## A data model consists of 3 components



### Entities:

These are the main things we want to store information about. For instance, in a business, an entity could be "Customer" or "Product."



### Attributes:

These are the specific pieces of information about an entity. For a "Customer," attributes could include name, address, and phone number.



### Relationships:

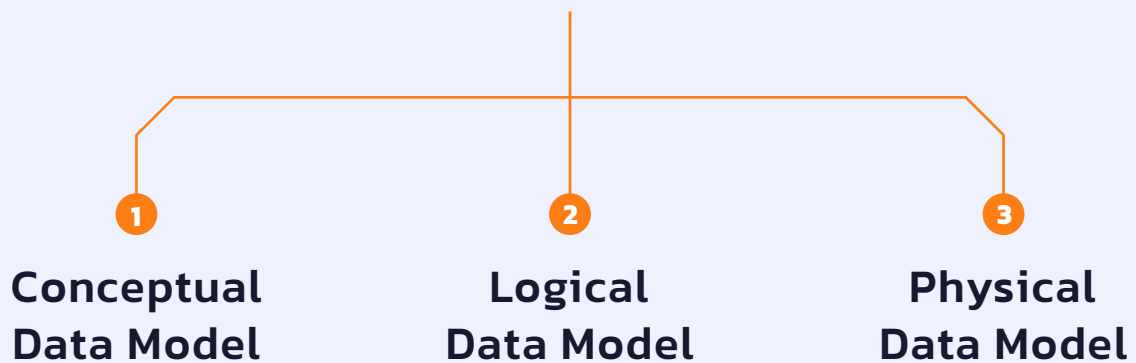
These define how entities are connected or related to each other. For example, a "Customer" can have a relationship with an "Order."

## TYPES OF DATA MODELS:

There are 3 types of data models:

**Conceptual Data Model, Logical Data Model, Physical Data Model**

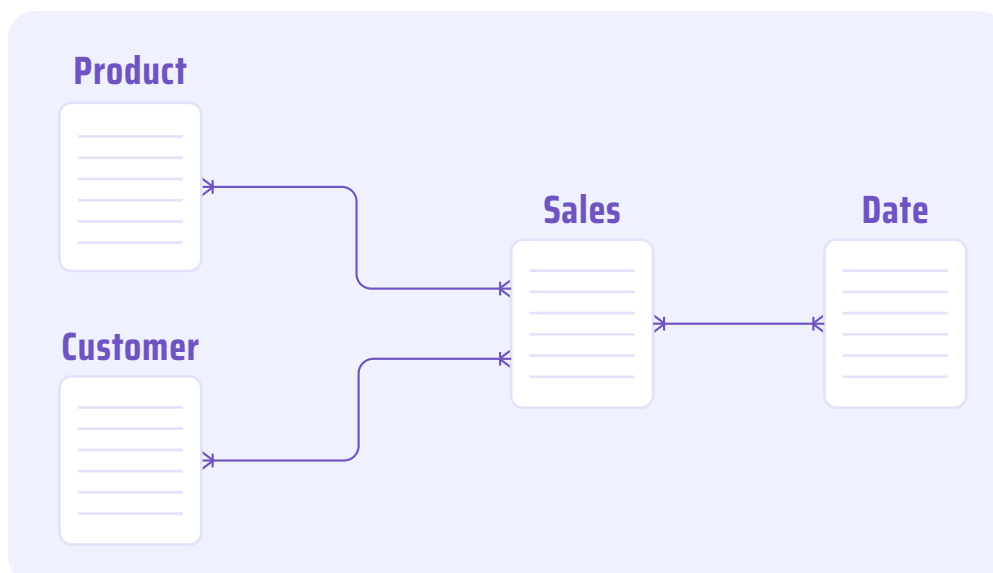
### TYPES OF DATA MODELS:



1

### CONCEPTUAL DATA MODEL

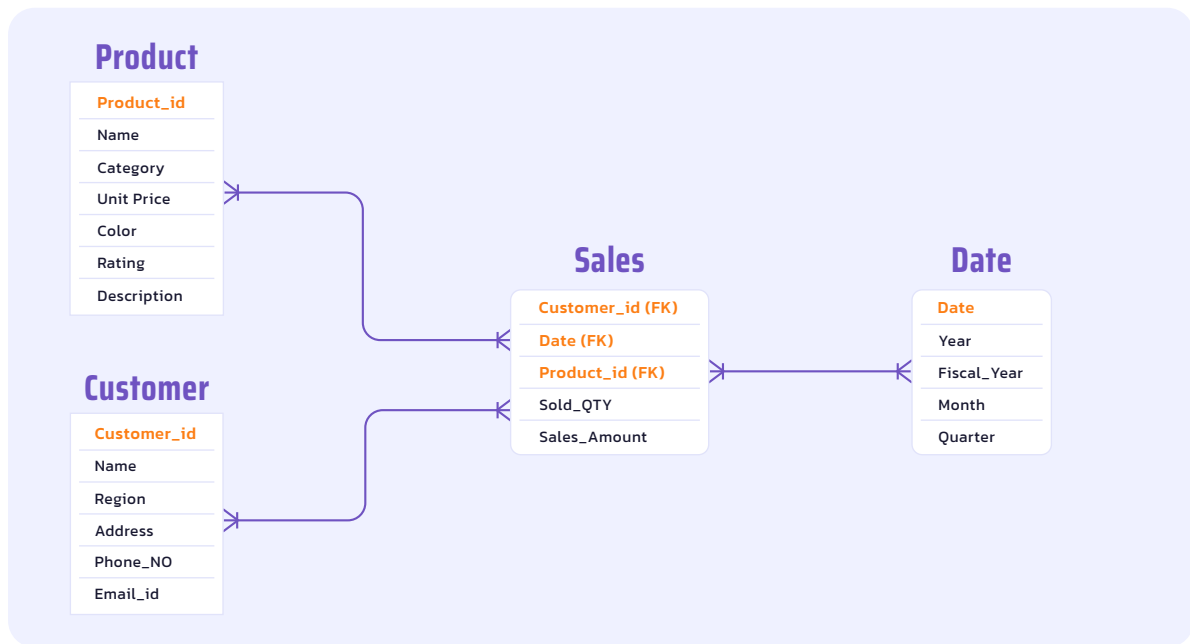
A high-level view of what needs to be stored and how different entities relate to each other. It's like a bird's eye view.



## 2

## LOGICAL DATA MODEL

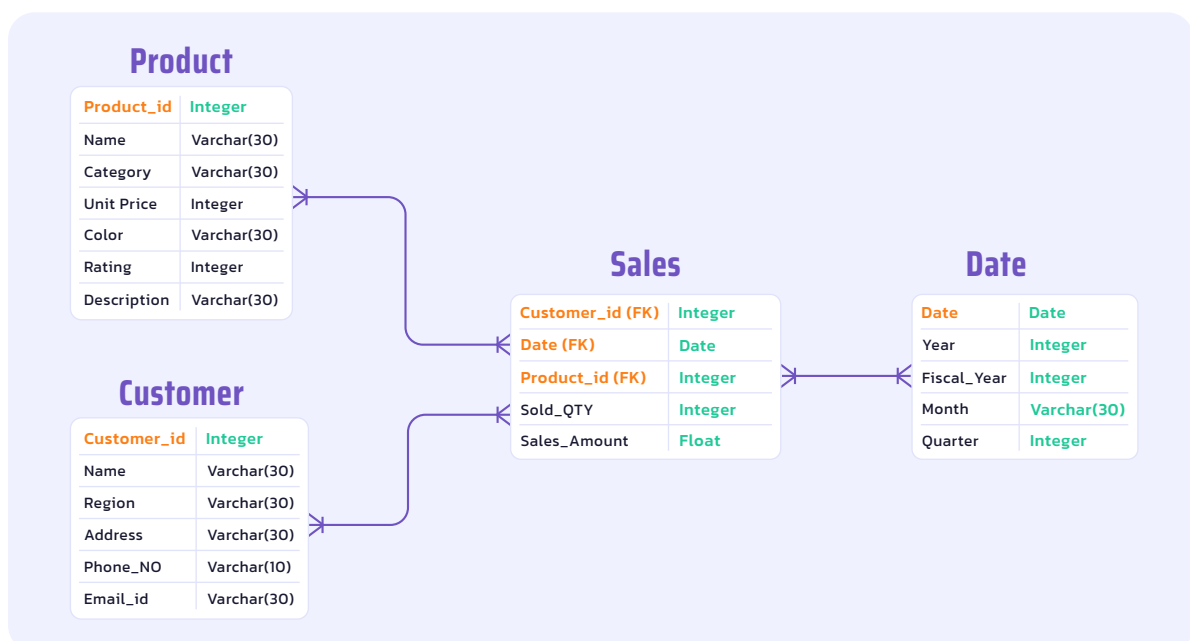
More detailed than the conceptual model, specifying attributes and relationships. It's like a floor plan of a house.



## 3

## PHYSICAL DATA MODEL

It specifies how the data will be stored, considering database technologies and constraints. It's like the actual construction of the house.



## TYPES OF TABLES:

Data modeling establishes a connection and flow of data between tables, typically consisting of fact tables surrounded by dimension tables, along with the relationships between these tables.

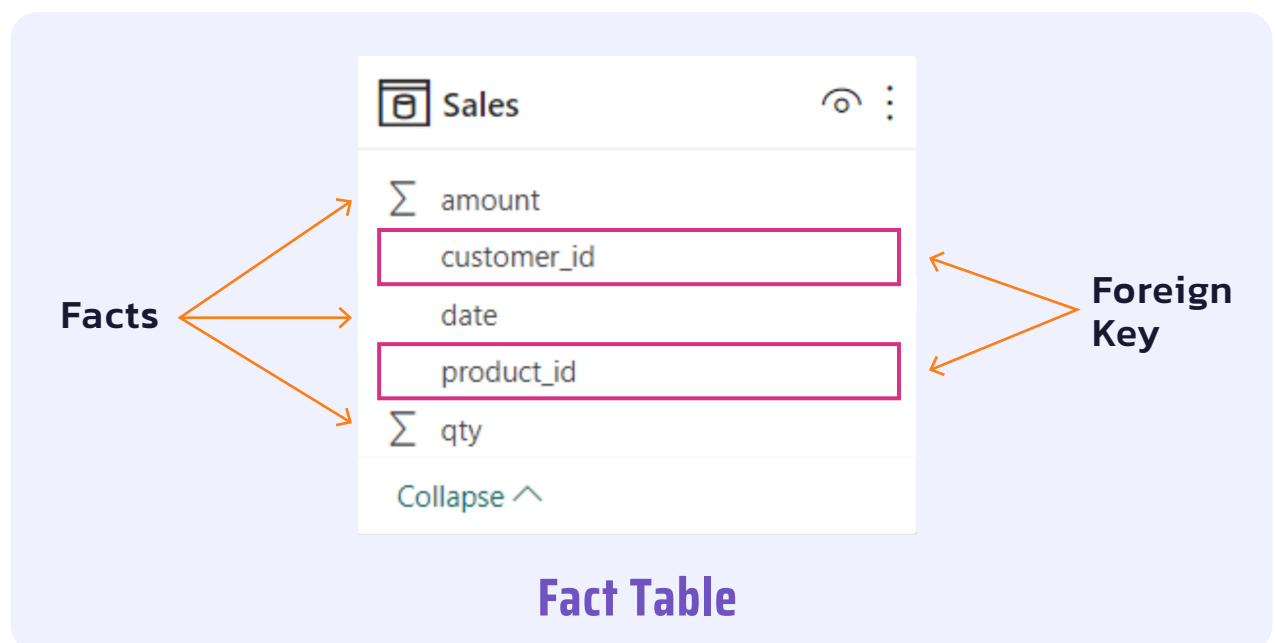
1

### FACT TABLE:

A fact table contains measurements, metrics, or facts about a business process. It generally compresses transactional data.

**It has two types of columns:** one representing facts of the business and another containing foreign keys to dimension tables.

**Example:** A Sales fact table contains data on store sales, detailing the quantity of each product sold and the revenue generated from each sale.



**Facts:** amount, date, quantity. These columns represent the business facts.

**Foreign keys:** customer\_id, product\_id. These columns contain foreign keys that link to dimension tables.

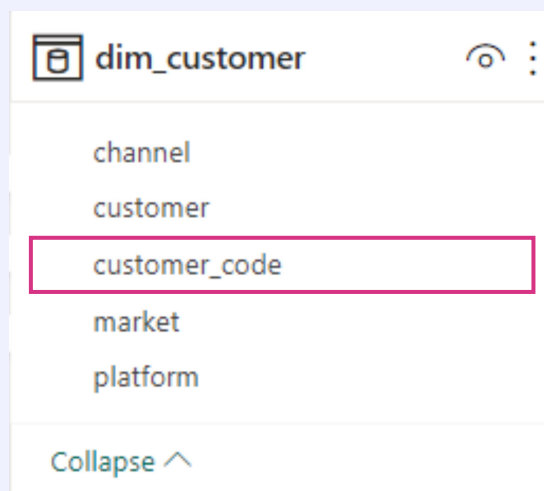
## 2 DIMENSION TABLE:

Fact tables are connected to dimension tables using foreign keys.

Dimension tables consist of attributes that describe the objects of a fact table.

Each dimension table includes a primary key that uniquely identifies each record and using this key dimension table associates with fact tables.

**Example:** A dim\_Customer table stores information about the customers who made purchases.



| dim_customer  |  |
|---------------|--|
| channel       |  |
| customer      |  |
| customer_code |  |
| market        |  |
| platform      |  |
| Collapse ^    |  |

← **Primary Key**

### Dimension Table

**Primary Key:** Customer\_code.

This column contains unique, non-null values associated with records in fact tables.

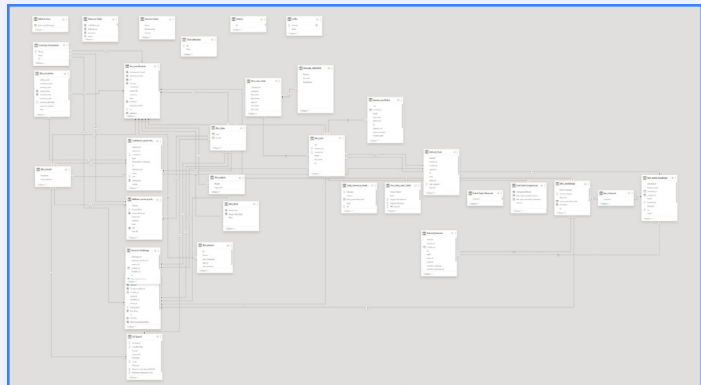


## TYPES OF RELATIONSHIPS:

### Only Data Analysts Can Relate



**The Relationship  
I Want**



**The Relationship  
I Get**

| dim_customer  |
|---------------|
| channel       |
| customer      |
| customer_code |
| market        |
| platform      |
| Collapse ^    |

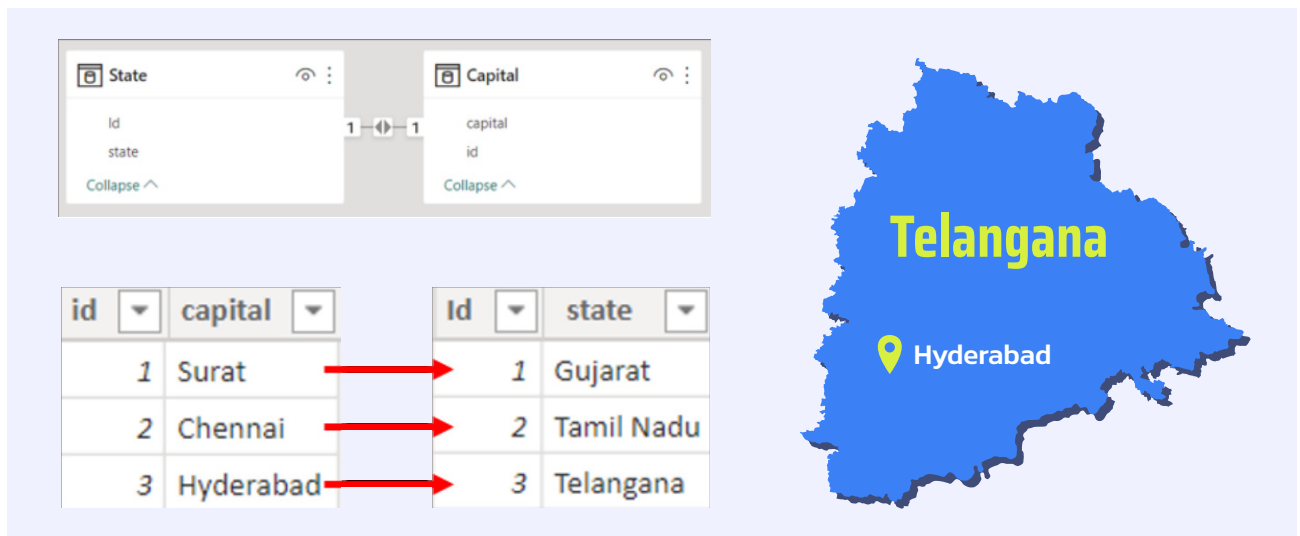
**Dimension Table**

| Sales       |
|-------------|
| Σ amount    |
| customer_id |
| date        |
| product_id  |
| Σ qty       |
| Collapse ^  |

**Fact Table**

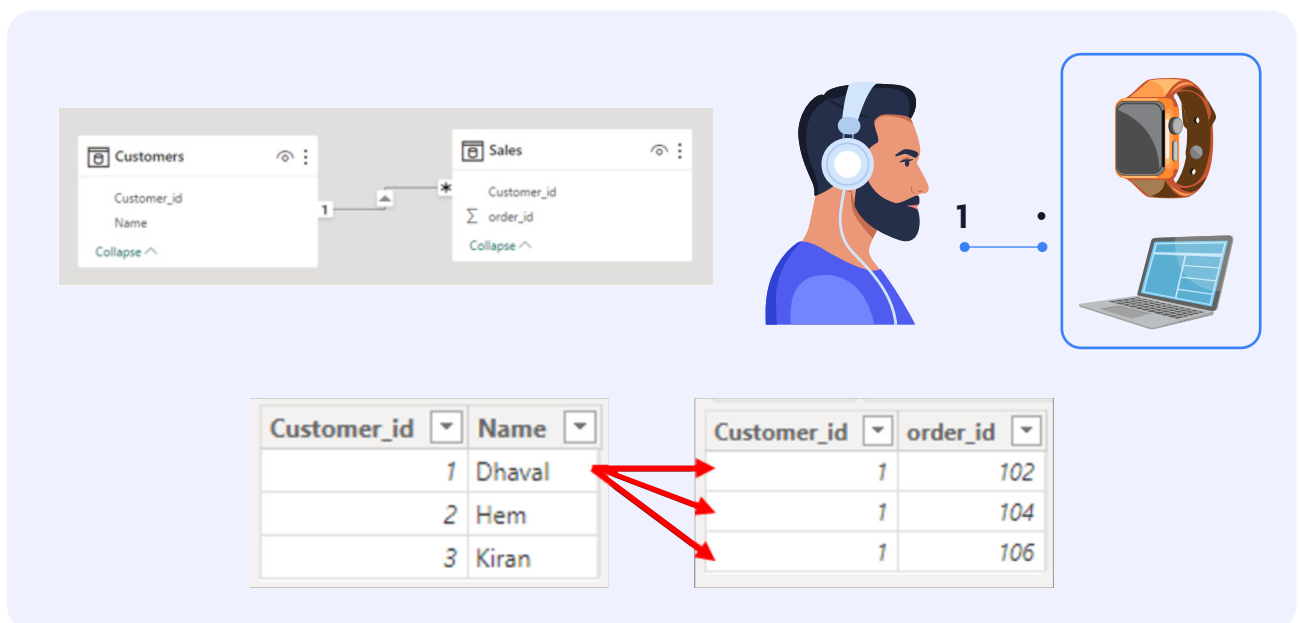
- ▶ Here dim\_customers is a dimension table, and Sales is a fact table
- ▶ The tables should consist of a common column attribute to make a relationship between tables.
- ▶ There are 4 types of relationships:
  1. One-to-One relationship
  2. One-to-Many relationship
  3. Many-to-One relationship
  4. Many-to-Many relationship

## 1 One-to-One Relationship:



Each row in the first table is mapped to only one row in the second table.  
For example, each state has only one capital.

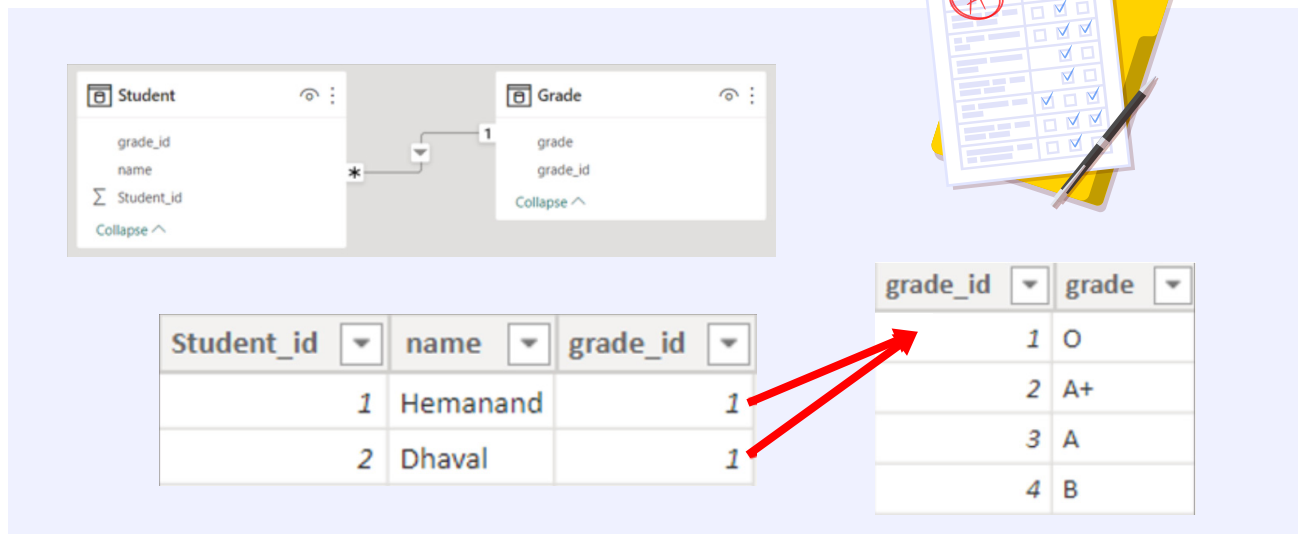
## 2 One-to-Many Relationship:



In a one-to-many relationship, each row in the first table can be associated with multiple rows in the second table.

For example, a customer can place several orders over time, but each order is tied to a specific customer.

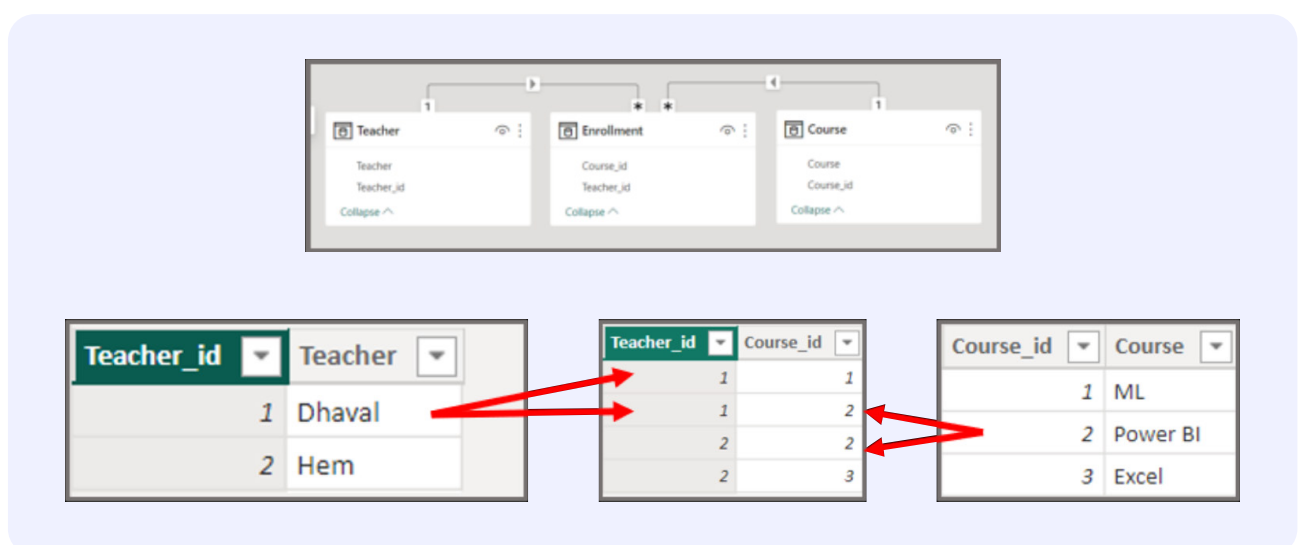
### 3 Many-to-One Relationship:



Multiple rows in the first table are mapped to a single related row in the second table.

For instance, different students can receive the same grade.

### 4 Many-to-Many Relationship:



Multiple records from one table relate to multiple records from another table.

## Many-to-Many Relationship:

Dhaval teaches  
ML, Power BI



Power BI taught by  
Dhaval and Hem



For example, more than one teacher can teach multiple courses. Dhaval teaches ML and Power BI; Power BI is taught by both Dhaval and Hem.

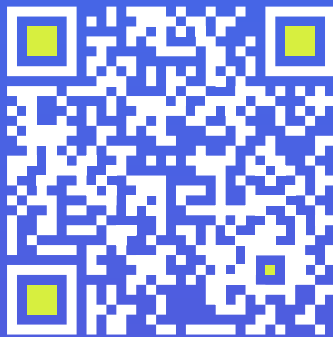


Enabling Careers

# Found this interesting?

To gain access to more valuable content,  
**join our WhatsApp Channel**

SCAN TO JOIN



codebasics.io