

What is ER Modeling? Learn with Example

What is ER Modeling?

Entity Relationship Model (ER Modeling) is a graphical approach to database design. It is a high-level data model that defines data elements and their relationship for a specified software system. An ER model is used to represent real-world objects.

An **Entity** is a thing or object in the real world that is distinguishable from the surrounding environment. For example, each employee of an organization is a separate entity. Following are some of the major characteristics of entities.

- An entity has a set of properties.
- Entity properties can have values.

In this tutorial, you will learn-

- Enhanced Entity Relationship (EER) Model
- [Why use the ER Model?](#)
- [Entities in the "MyFlix" library](#)
- [Defining the Relationships Among Entities](#)

Let's consider our first example again. An employee of an organization is an entity. If "Peter" is a programmer (an **employee**) at Microsoft, he can have **attributes** (properties) like name, age, weight, height, etc. It is obvious that those do hold values relevant to him.

Each attribute can have **Values**. In most cases a single attribute has one value. But it is possible for attributes to have **multiple values** also. For example Peter's age has a single value. But his "phone numbers" property can have multiple values.

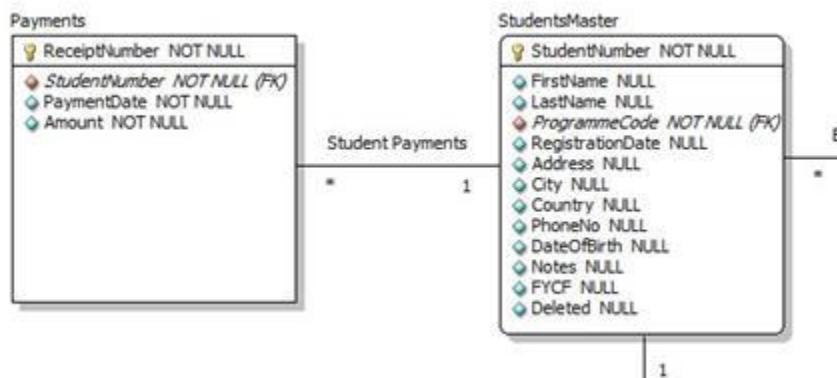
Entities can have **relationships** with each other. Let's consider the simplest example. Assume that each Microsoft Programmer is given a Computer. It is clear that **Peter's Computer** is also an entity. Peter is using that computer, and the same computer is used by Peter. In other words, there is a mutual relationship between Peter and his computer.

In **Entity Relationship Modeling**, we model entities, their attributes and relationships among entities.

Enhanced Entity Relationship (EER) Model

Enhanced Entity Relationship (EER) Model is a high-level data model which provides extensions to the original **Entity Relationship** (ER) model. EER Models supports more detailed design. EER Modeling emerged as a solution for modeling highly complex databases.

EER uses UML notation. UML is the acronym for Unified Modeling Language; it is a general-purpose modeling language used when designing object-oriented systems. Entities are represented as class diagrams. Relationships are represented as associations between entities. The diagram shown below illustrates an ER diagram using the UML notation.



Why use the ER Model?

Now you may think why use ER modeling when we can simply create the database and all of its objects without ER modeling? One of the challenges

faced when designing a database is the fact that designers, developers, and end-users tend to view data and its usage differently. If this situation is left unchecked, we can end up producing a database system that does not meet the requirements of the users.

Communication tools understood by all stakeholders(technical as well as non-technical users) are critical in producing database systems that meet the requirements of the users. ER models are examples of such tools.

ER diagrams also increase user productivity as they can be easily translated into relational tables.

Case Study: ER diagram for "MyFlix" Video Library

Let's now work with the MyFlix Video Library database system to help understand the concept of ER diagrams. We will use this database for all hand-on in the remainder of this tutorial

MyFlix is a business entity that rents out movies to its members. MyFlix has been storing its records manually. The management now wants to move to a DBMS

Let's look at the steps to develop EER diagram for this database-

1. Identify the entities and determine the relationships that exist among them.
2. Each entity, attribute, and relationship, should have appropriate names that can be easily understood by the non-technical people as well.
3. Relationships should not be connected directly to each other.
Relationships should connect entities.
4. Each attribute in a given entity should have a unique name.

Entities in the "MyFlix" library

The entities to be included in our ER diagram are;

- **Members** - this entity will hold member information.
- **Movies** - this entity will hold information regarding movies
- **Categories** - this entity will hold information that places movies into different categories such as "Drama", "Action", and "Epic" etc.
- **Movie Rentals** - this entity will hold information that about movies rented out to members.
- **Payments** - this entity will hold information about the payments made by members.

Defining the Relationships Among Entities

Members and movies

The following holds true regarding the interactions between the two entities.

- A member can rent more than one movie in a given period.
- A movie can be rented by more than one member in a given period.

From the above scenario, we can see that the nature of the relationship is many-to-many. **Relational databases do not support many-to-many relationships. We need to introduce a junction entity.** This is the role that the MovieRentals entity plays. It has a one-to-many relationship with the members table and another one-to-many relationship with the movies table.

Movies and categories entities

The following holds true about movies and categories.

- A movie can only belong to one category but a category can have more than one movie.

We can deduce from this that the nature of the relation between categories and movies table is one-to-many.

Members and payments entities

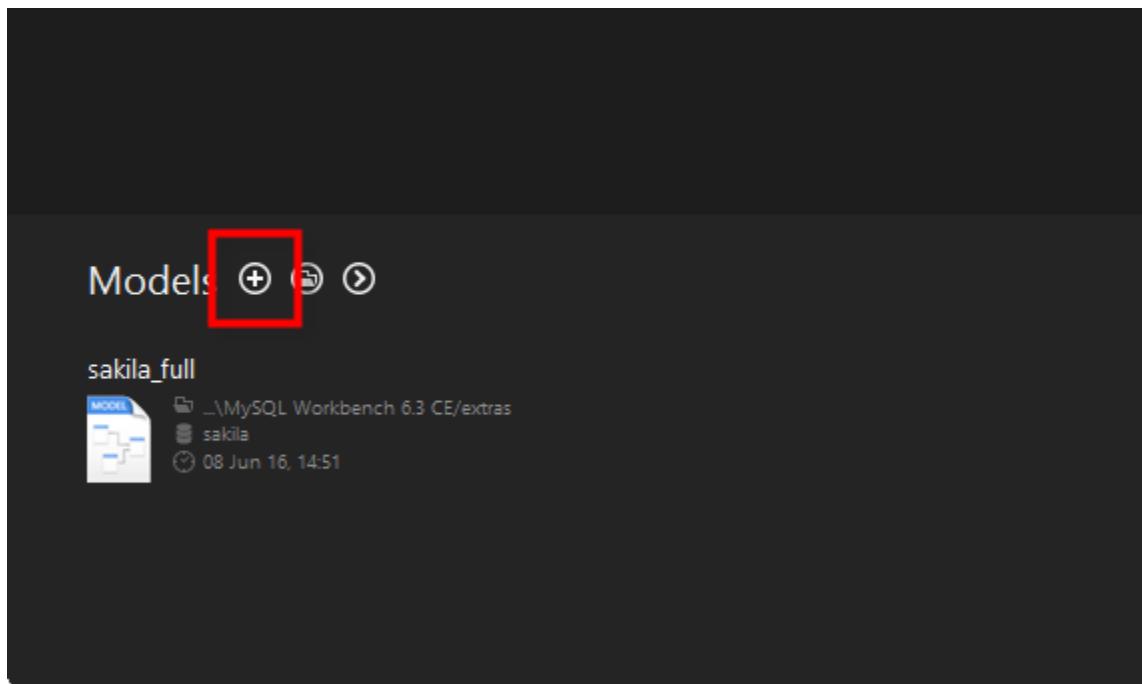
The following holds true about members and payments

- A member can only have one account but can make a number of payments.

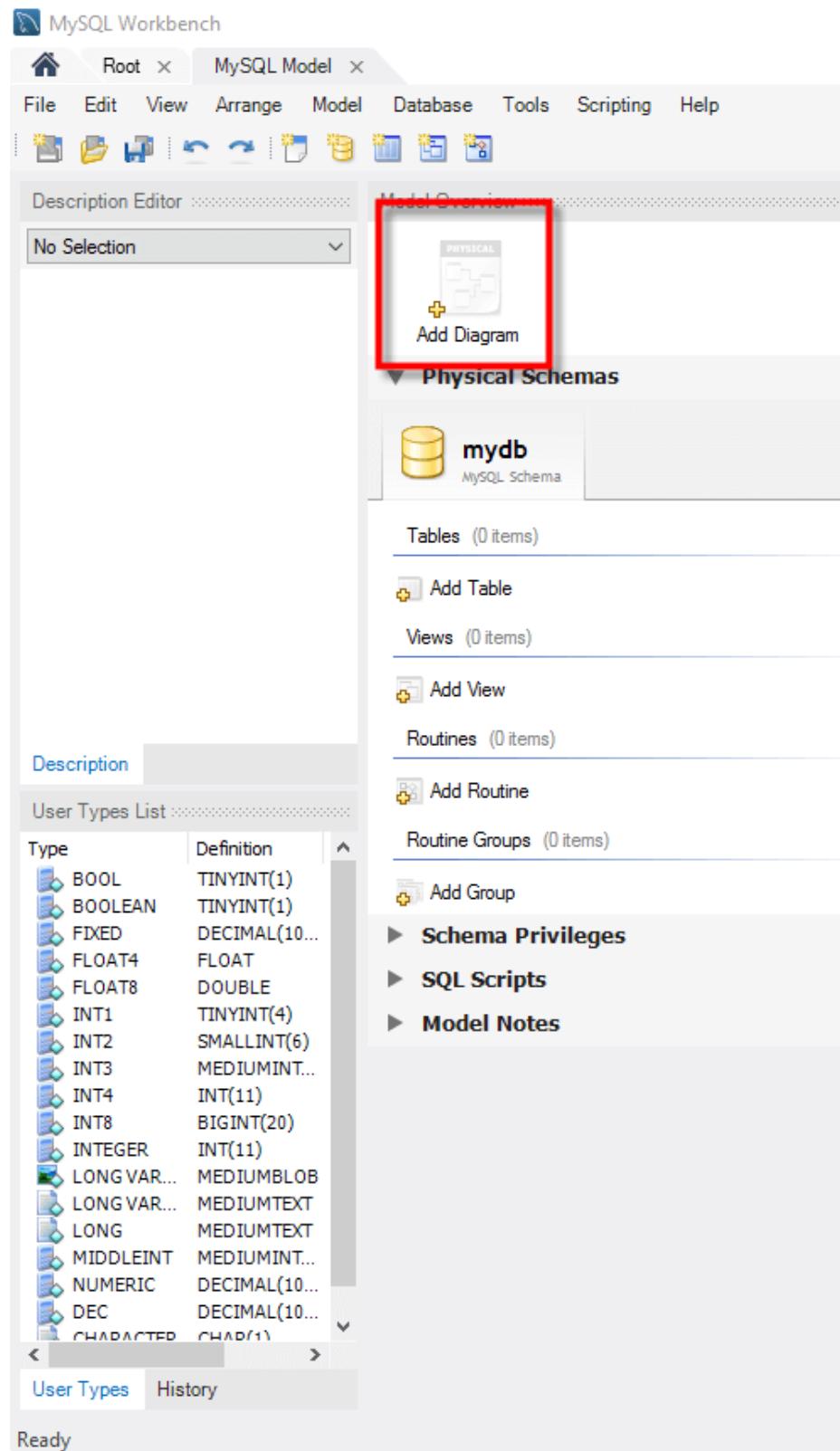
We can deduce from this that the nature of the relationship between members and payments entities is one-to-many.

Now lets create EER model using MySQL Workbench

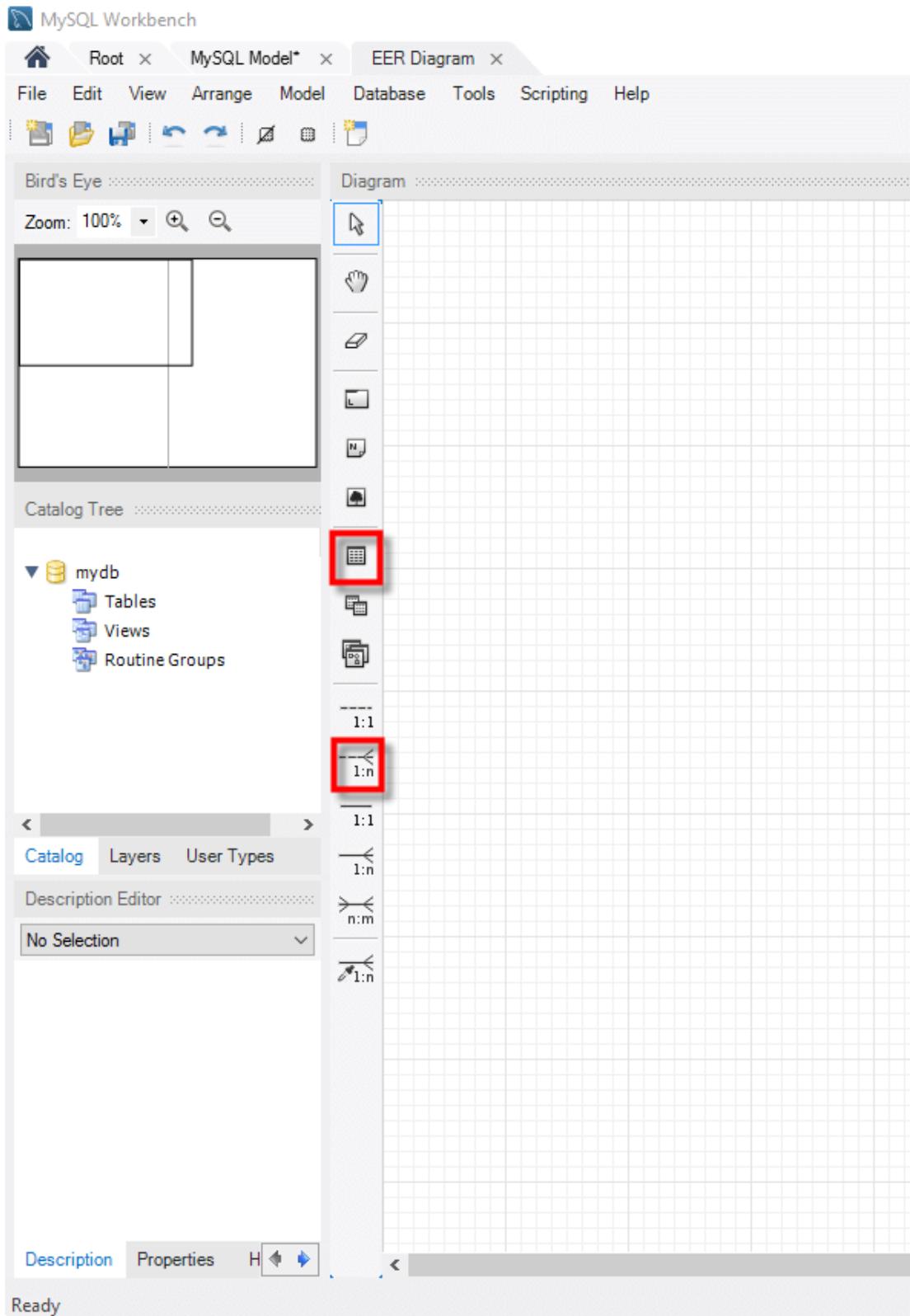
In the MySQL workbench , Click - "+" Button



Double click on the Add Diagram button to open the workspace for ER diagrams.



Following window appears



Let's look at the two objects that we will work with.

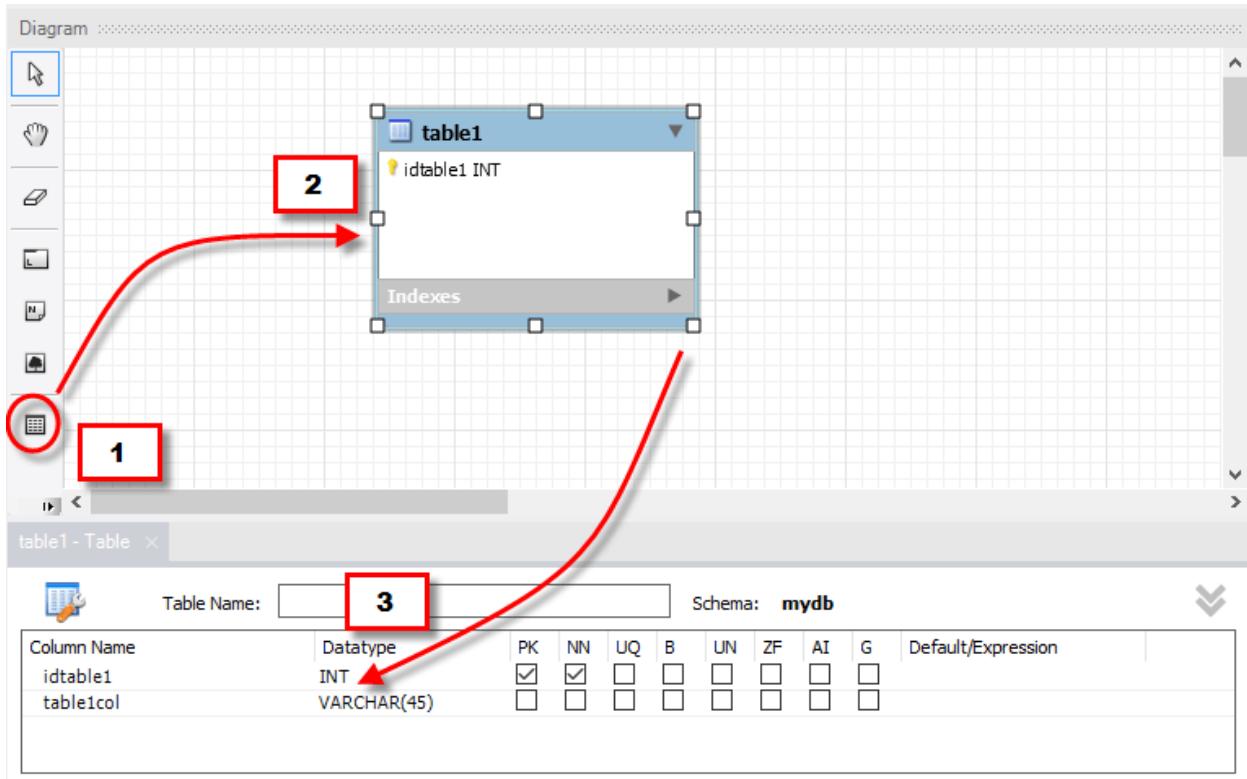
-  The table object allows us to create entities and define the attributes associated with the particular entity.
-  The place relationship button allows us to define relationships between entities.

The **members**' entity will have the following attributes

- Membership number
- Full names
- Gender
- Date of birth
- Physical address
- Postal address

Let's now create the members table

- 1.Drag the table object from the tools panel
- 2.Drop it in the workspace area. An entity named table 1 appears
- 3.Double click on it. The properties window shown below appears



Next ,

1. Change table 1 to Members
2. Edit the default idtable1 to membership_number
3. Click on the next line to add the next field
4. Do the same for all the attributes identified in members' entities.

Your properties window should now look like this.

Members - Table

Column Name	Datatype	PK	NN	UQ	B	LIN	ZF	AT	G	Default/Expression
membership_number	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
full_names	VARCHAR(150)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
gender	VARCHAR(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
date_of_birth	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
physical_address	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
postal_address	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
contact_number	VARCHAR(75)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
email	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Table Name: **Members** Schema: mydb

Column Name: Data Type: Expression:

Collation: Table Default

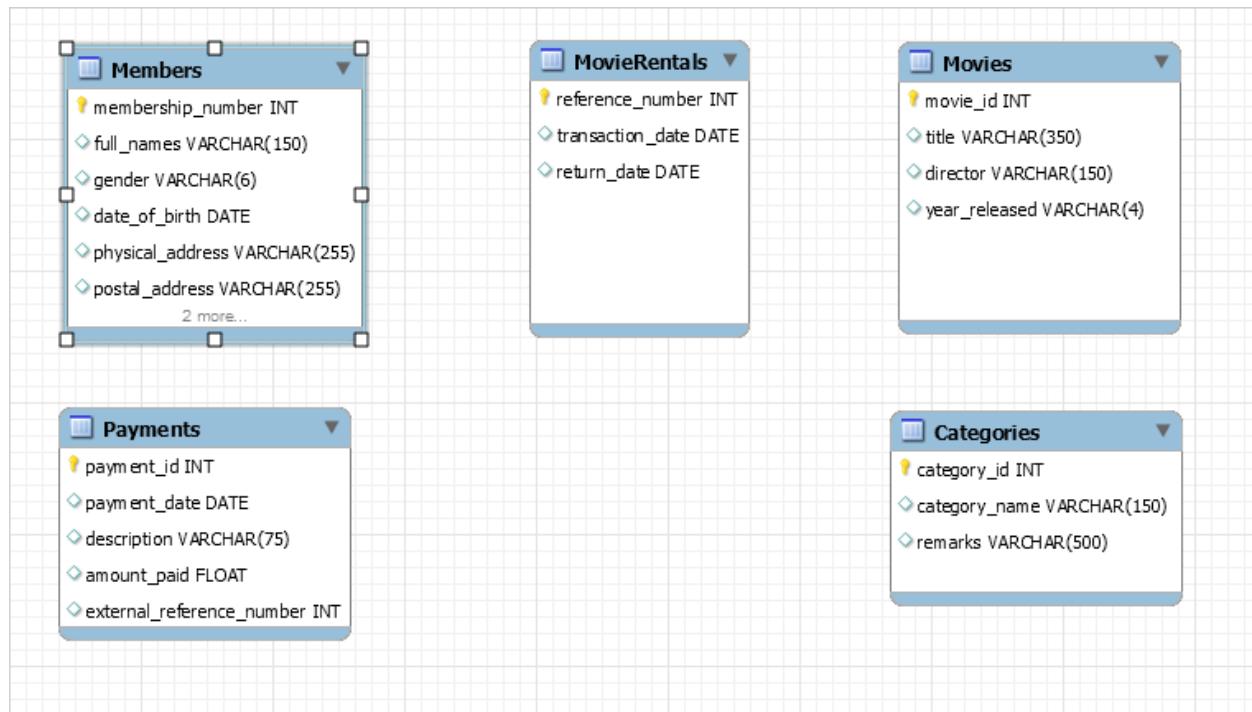
Comments:

Storage: Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

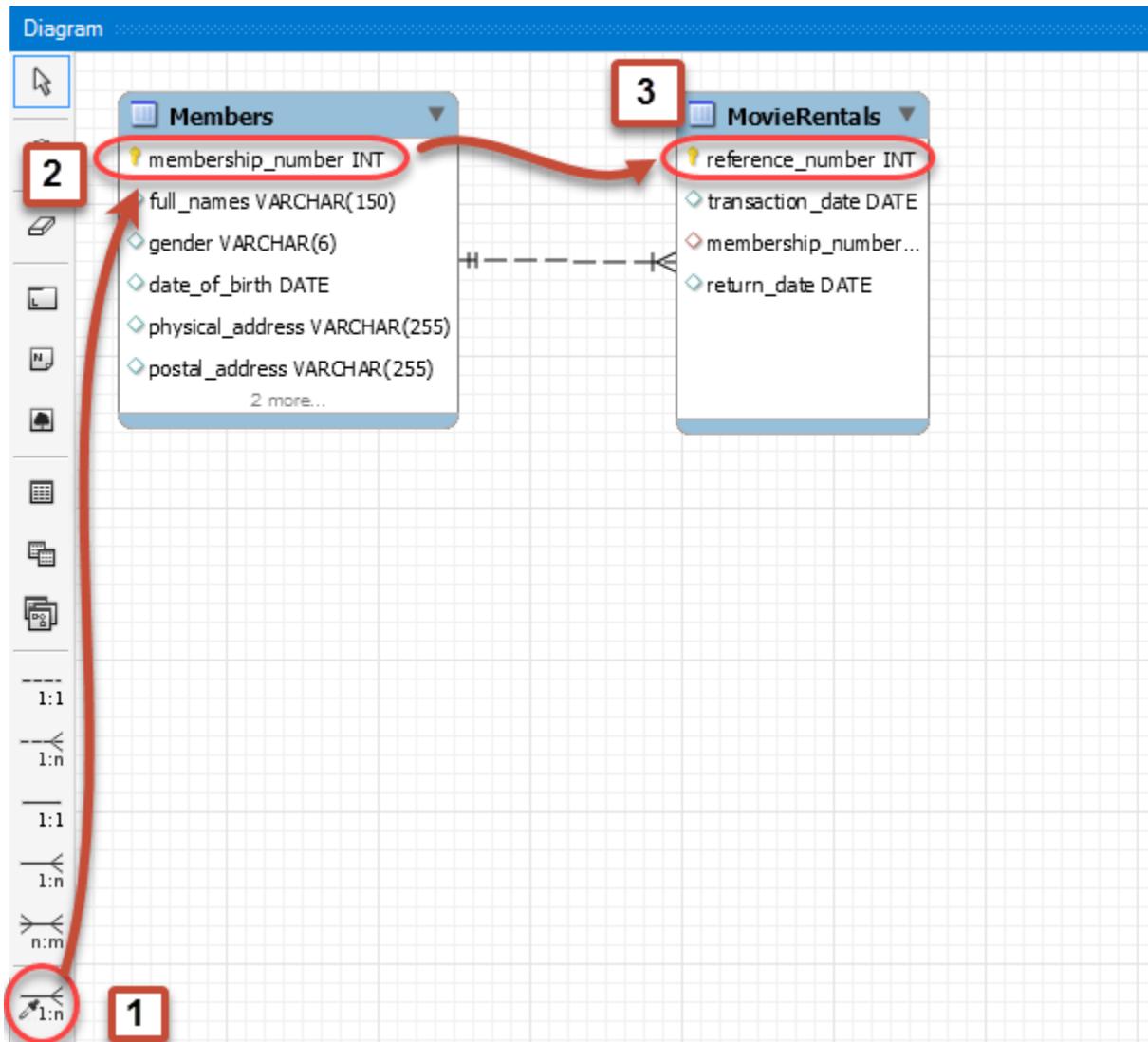
Repeat the above steps for all the identified entities.

Your diagram workspace should now look like the one shown below.

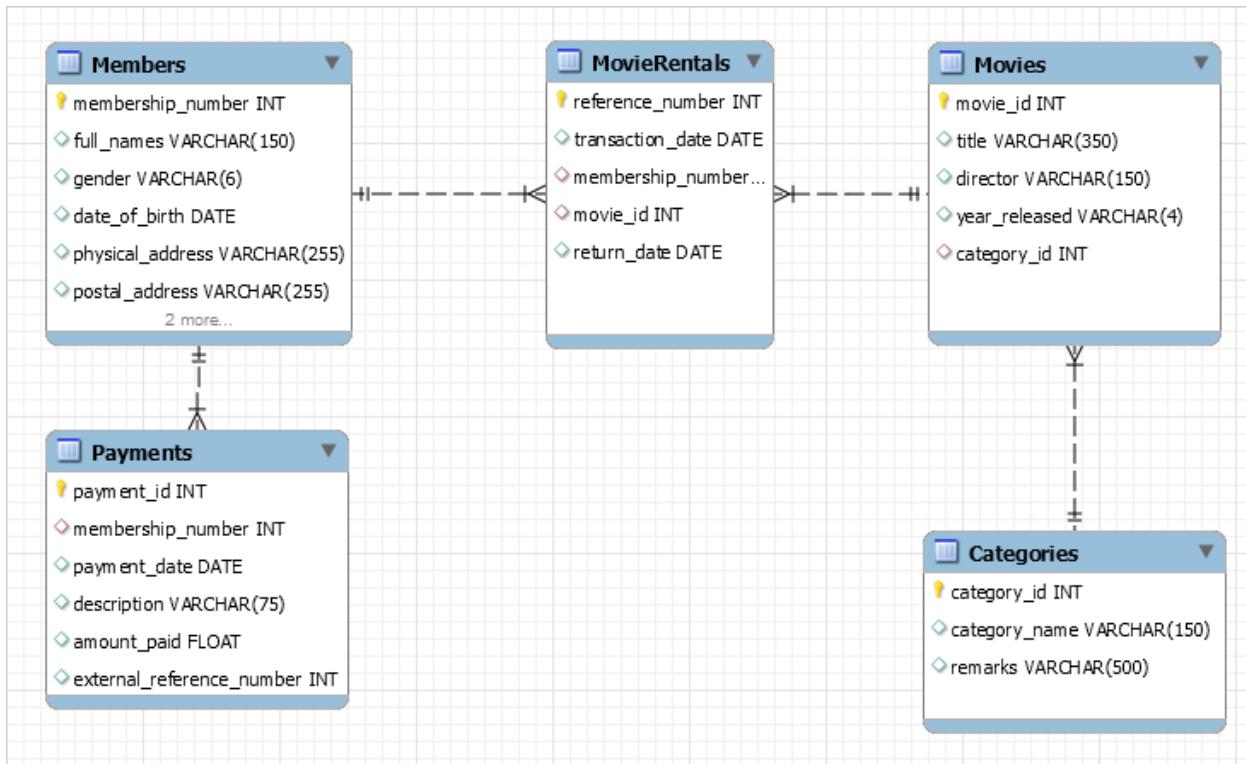


Lets create relationship between Members and Movie Rentals

1. Select the place relationship using existing columns too
2. Click on membership_number in the Members table
3. Click on reference_number in the MovieRentals table



Repeat above steps for other relationships. Your ER diagram should now look like this -



Summary

- The full form of ER is Entity and Relationships Diagrams. They play a very important role in the database designing process. They serve as a non-technical communication tool for technical and non-technical people.
- Entities represent real world things; they can be conceptual as a sales order or physical such as a customer.
- All entities must be given unique names.
- ER models also allow the database designers to identify and define the relations that exist among entities.

The entire ER Model is attached below. You can simply import it in MySQL Workbench