

HIBERNATE & JPA COURSE

ASSOCIATIONS IN HIBERNATE & JPA



By the expert: Eng. Ubaldo Acosta



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

Hello, Ubaldo Acosta greets you. Welcome again. I hope you're ready to start with this one.

We are going to study the topic of Associations in Hibernate and JPA.

Are you ready? OK let's go!

MAPPING ASSOCIATIONS

- The associations in Hibernate and JPA are the same as those used in the theory of relational databases.

- ✓ 1 to 1
- ✓ 1 to Many or Many to 1
- ✓ Many to Many

- Hibernate and JPA support the relationships mentioned in the mapping files of each Entity class or through Java annotations.

HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

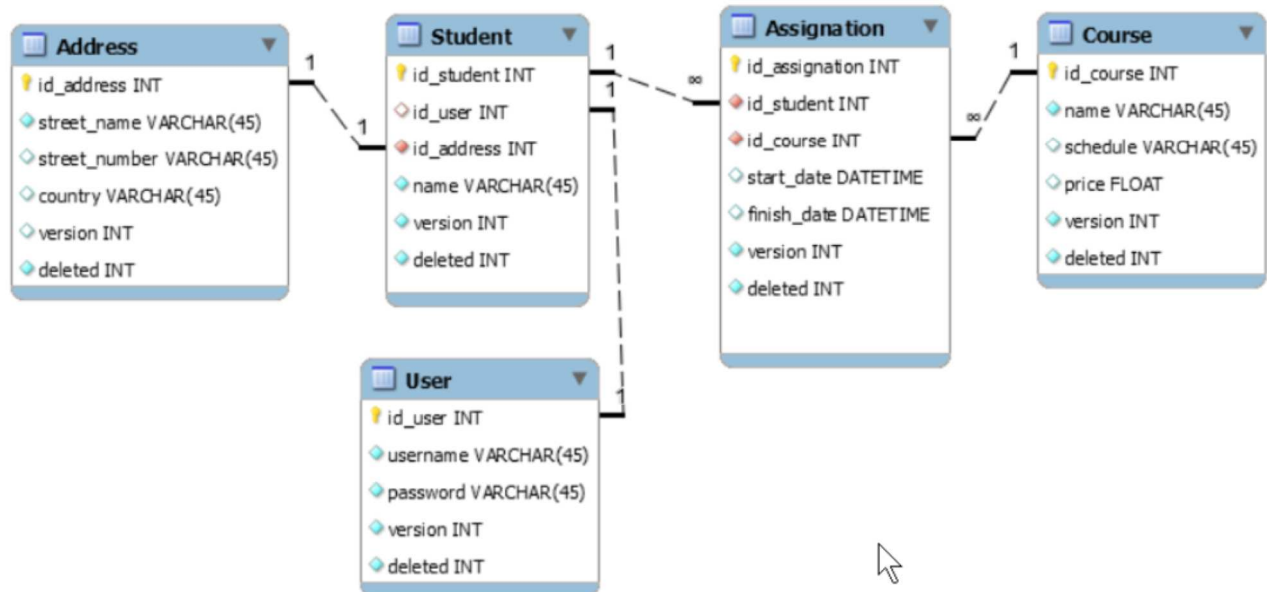
In this lesson we are going to review the mapping issue.

The associations in Hibernate and JPA are the same that are handled in the relational database theory, for example we will have the relation of 1 to 1, the relation of 1 to many or many to 1 and the relation of many to many.

Hibernate and JPA supports all the aforementioned relationships and we will configure this in our mapping file of each of the entity classes that we are managing or, more commonly, through Java annotations.

We will review in more detail the types of associations that we will be reviewing in this course.

DIAGRAM ENTITY RELATIONSHIP EXAMPLE



www.globalmentoring.com.mx

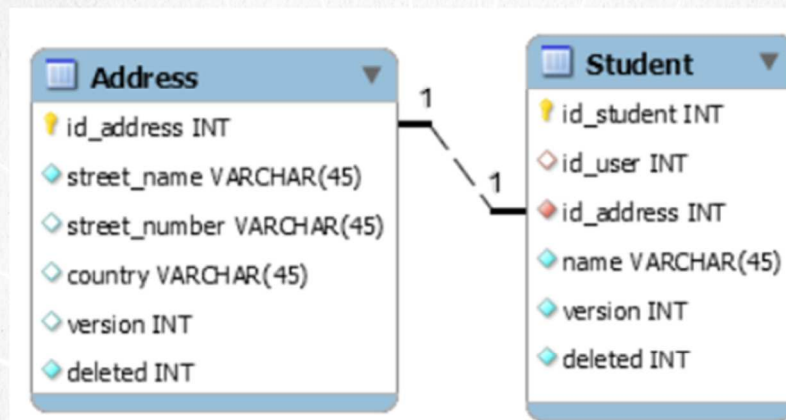
Throughout the course we will use the following Entity - Relationship diagram, which we use only as an example to be able to review the relationships and mappings mentioned above.

Although we could use more fields and more tables in the modeling, we have left the model as simple as possible so that we can focus on the relationships we will study throughout the course

We will review each of these relationships in more detail, but basically we have an Entity - Relationship diagram that presents a Student Management System for a School, for example, Global Mentoring students. The Web system that uses this database will be called the Student Management System or its acronym SMS and this is the Entity-Relationship scheme that we will use throughout the course.

EXAMPLE OF A 1 TO 1 RELATIONSHIP

- Let's see an example of a 1 to 1 relationship. A student has an Address.



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

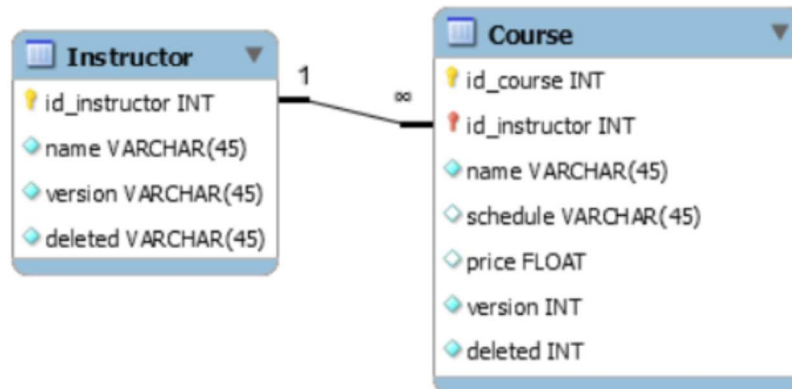
In this diagram we can see a relation of 1 to 1. As an example we are putting an Student has a related Address.

In this type of one-to-one relations the foreign key (speaking in terms of database) can be in any of the 2 tables, but in this case because from the Student entity we will retrieve the associated information of the Address, the most convenient is to place the foreign key of the address in the student's table.

This is an example that we will use later to map this association using Hibernate or JPA.

EXAMPLE OF A 1 TO MANY RELATIONSHIP

Example of relationship 1 to Many. An Instructor has many Courses.



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

Let's now review a 1 to Many relationship (1 to *).

In this case, we are exemplifying that an instructor can teach many courses, then because the instructor can teach one or more courses, we add the field of `id_instructor` as a foreign key in the **Course** table.

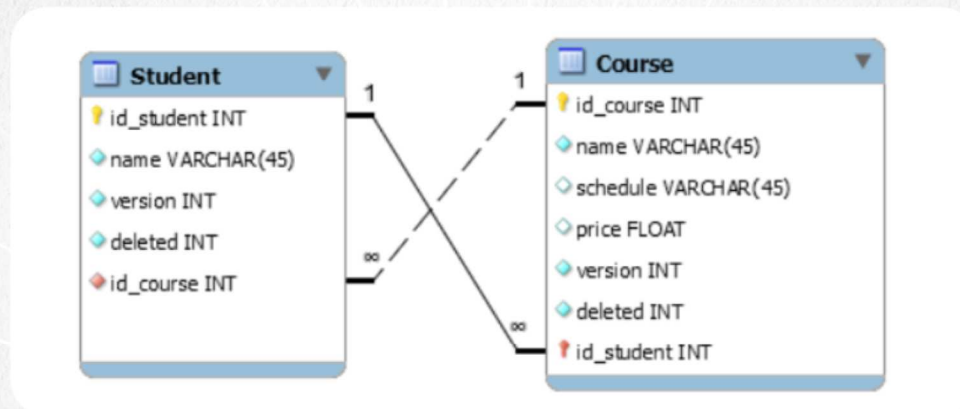
Then, from the foreign key of the **Instructor** we will be able to request all the courses that this instructor is teaching.

This is one of the most used relationships in the design of relationships in databases, so it is vitally important to correctly apply the concepts of mapping that we will see later.

This relationship will not be worked on in the final entity-relation scheme, but it remains as an exercise to add it and work it on the project

EXAMPLE OF RELATIONSHIP FROM MANY TO MANY

Example of relationship Many to Many. A Student has many Courses, and a Course has many Students.



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

Here is an example of many to many (* to *).

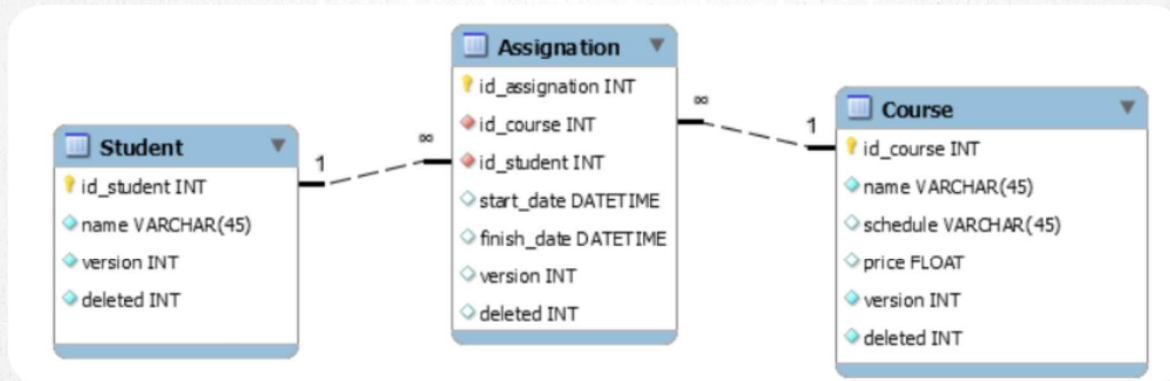
Here we can see that a Student can be related to a Course, but in turn a Course can have many Students.

This type of Many-to-Many relationships can also be represented with Hibernate or JPA, however this type of relationship is better to apply the concept of database normalization to simplify this type of relationships and instead of having many-to-many relationships, we have One to Many or One to One relationships.

If we do not normalize this type of relationship we can fall into circular references, because a Student can have a Course, but a Course can have the same Student. To correct, this we will normalize (simplify) this relationship, so that it remains as a one-to-many or one-to-one relationship. Let's see how to do this.

NORMALIZATION OF MANY TO MANY RELATIONSHIP

Example of normalization of the Many to Many relationship to a One to Many relationship. A student has an Assignment and a course has a student assigned. Assignment is the transitive table used to break the Many-to-Many relationship and now have only One-to-Many relationships.



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

We can see the normalization of the previous relationship, in this case the Student table that was directly related to the Course table.

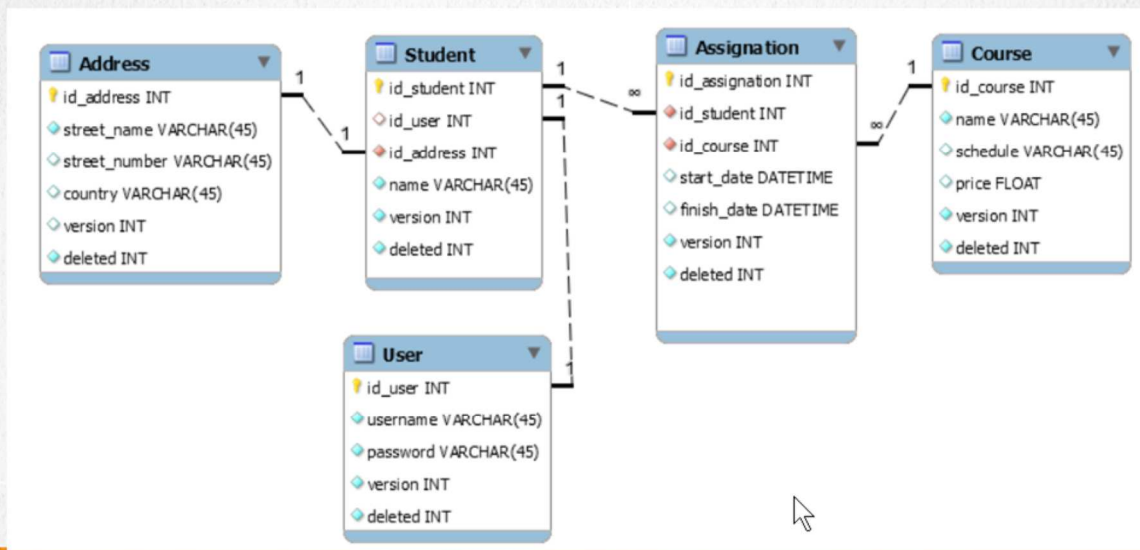
It is no longer directly related, but we have now put an intermediate table, this intermediate table is known as a transitive table and what we are doing is converting the Many to Many relationship to a One to Many relationship of Students to Assignment and a One to Many relationship from the Course table to the Assignment table.

The primary keys of `id_student` and of `id_course` add them as foreign keys to the Assignment table and in turn, by combining them, it becomes a primary key composed for the Assignment table, these are some of the practices that can be used in Hibernate or JPA. However, a simple primary key can also be added to the Assignment table and thus not handle composite primary keys. Either option is feasible.

Even though we could have handled a Many to Many relationship directly with Hibernate or JPA, normalizing this type of relationship allow us to manage more easily One-to-Many or One-to-One relationships, and therefore easier to maintain.

In addition, once we have generated a transitive table for the Student and the Course, we can add certain attributes of the new relation, for example, a Student when taking a Course can be at a certain time, among other types of attributes.

REVIEW OF THE ENTITY RELATIONSHIP - COMPLETE DIAGRAM



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx

Observe the relationship structure scheme of the SMS system (Student Management System), which is the database that we will be working on throughout the course. This entity-relationship scheme has the most common relationships that we will find when mapping Entity classes using Hibernate or JPA.

We have a Student table, which is related to an Address table, which will keep the Student's address data, this relationship is of the One to One type. A Student can have only one Address.

A student may also have assigned certain courses and a course has several students, for this we have created the Assignment table, as we discussed earlier when we denormalized the relationship of many to many among Student and Course, we can see that in this transitional Assignment table we have the relation of a Student to many Assignment and a Course also to many Assignment.

The User table will allow us to have the data to access the SMS system, so in this Users table we will have the username and password.

Each table has version and deleted fields, these fields can be used for very specific Hibernate issues, and also to know if a record is active or not. These fields are optional, but it is advisable to have a strategy of how the deleted records will be handled. If they are really going to be removed from the database, or they will only be marked as deleted. So it is the decision of the System Architect to decide this strategy. Here we will handle the simplest one.

This is the relationship entity scheme that we will be working on throughout this course and it is important to analyze and study it in detail because the Hibernate or JPA relationships that we are studying are going to be based on the relationships of this relationship entity scheme.

ONLINE COURSE

HIBERNATE & JPA

By: Eng. Ubaldo Acosta



HIBERNATE & JPA COURSE

www.globalmentoring.com.mx