Modeling Data Relationships - How-to

Table of Contents

Outline

In this exercise lab, we will extend the current data model of the application by adding relationships between the Entities. By the end of the exercise, we want to make sure that:

- Every movie should have a genre and multiple movies can have the same genre.
- A Person can have multiple roles in the same movie, or in different movies. The database can also have multiple people with the same role, for instance, several Directors and several Actors.

Also, at this point we know that the application in the future will support new functionalities that allows users to rate a movie.

To support that, we need to add an extra Entity, **UserMovieRating**, which relates the users of the application with the movie they just rated. A user can rate multiple movies and a movie can be rated by multiple users.
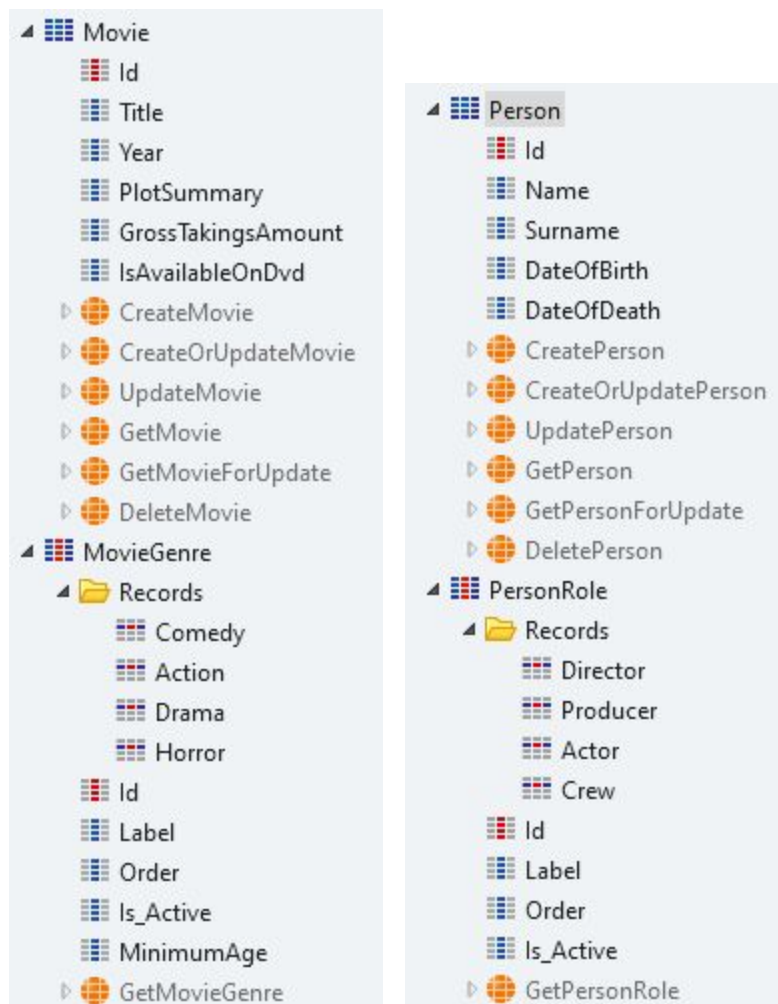
How to

In this section, we'll describe, step by step, exercise *4.4 - Entity Relationships*.

Recapping the Data Model

Let's start by recapping the data model we created two exercises ago.

1. Go back to the **OSMDb_Core** module and open the **Data** tab.

2. Expand the Movie and Person Entities as well as the MovieGenre and PersonRole Static Entities (if they are currently collapsed). Notice that all the Entities have a default attribute called **Id** that is colored red. This is what is called the Entity's Identifier**.**

3. Select the **Id** attribute of the Movie Entity and notice its **Data Type** is *Long Integer*

4. Notice that the **Is AutoNumber** property is set to *Yes*.

Movie Genre Relationship

In this section, we will prepare the data model to represent a one-to-many relationship between the MovieGenre and the Movie.
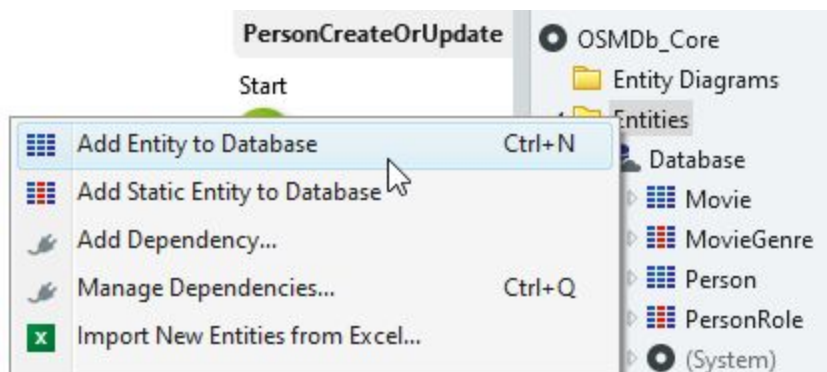
1. Right-click the Movie Entity and select **Add Entity Attribute**

2. Enter *GenreId* for the name of the attribute

3. Change the **Data Type** of the GenreId to *MovieGenre Identifier*

People in Movies

In this section, we will create a many-to-many relationship between the Person and the Movie. This will be done with a new Entity that besides the information regarding the movie and the person, will also have the role of the person in that movie.

1. In the Data tab, right click the Entities folder and select **Add Entity to Database**.

2. Enter *PersonMovieRole* for the name of the Entity.

3. Right-click the PersonMovieRole Entity and select **Add Entity Attribute**. Set its Name to *PersonId* and its **Mandatory** property to *Yes.*

4. Create two new **mandatory** attributes: *MovieId* and *PersonRoleId*.

5. Verify that the types for these attributes were inferred correctly:

> **PersonId:** *Person Identifier*

> **MovieId:** *Movie Identifier*

> **PersonRoleId:** *PersonRole Identifier*

6. Since we will be using the PersonMovieRole Entity on our UI module, change the **Public** property to *Yes* and the **Expose Read Only** to *Yes*.

7. The **PersonMovieRole** Entity should look like this
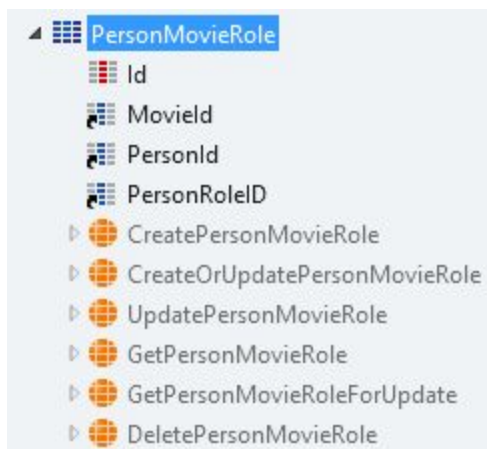
User Ratings

In this section, we will create a many-to-many relationship between a User and the Ratings it can give to a movie. This will also require a new Entity.

1. Right-click the **Entities** folder and select **Add Entity to Database**. Enter *UserMovieRating* for the name of the Entity.

2. Add a new *UserId* attribute to the UserMovieRating Entity. Leave its **Mandatory** property as *No*. Verify if the **Data Type** was set to *User Identifier*.

3. Add a new *MovieId* attribute to the UserMovieRating Entity. Set its **Mandatory** property as *Yes*. Verify if the **Data Type** was set to *Movie Identifier*.

4. Add a new *Rating* attribute to the UserMovieRating Entity. Set its **Mandatory** property to *Yes* and set its **Data Type** to *Integer*.

5. Since we will be using this Entity on our UI module, in the properties editor, change the **Public** property to *Yes.*

6. Your Entity should look like this:
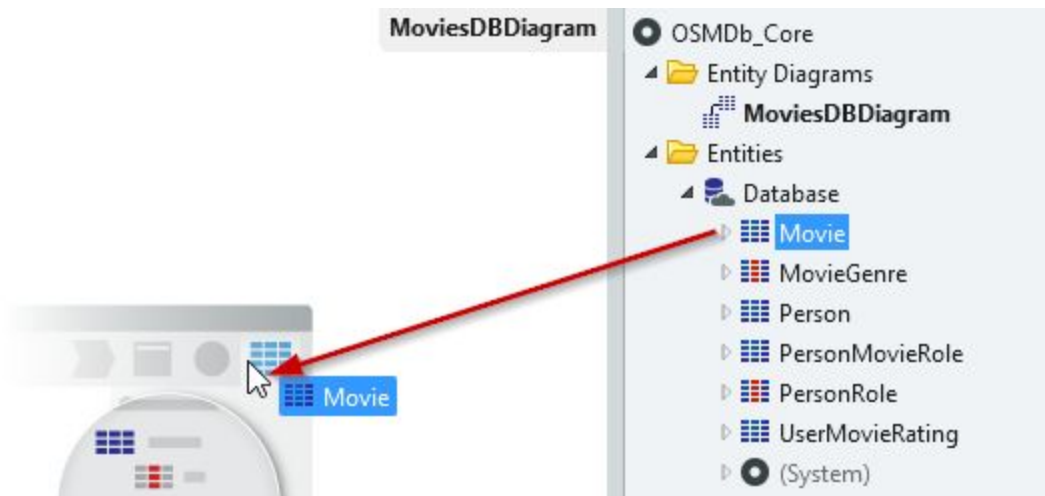

Entity Diagram

To have a visual representation of the data model, we need to create an Entity Diagram with all the Entities in it.

1. In the elements area of the Data tab, right click the **Entity Diagrams** folder and select **Add Entity Diagram**.
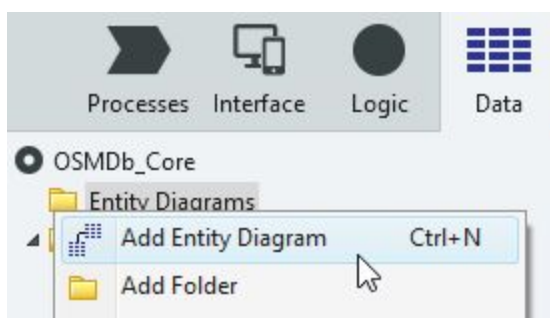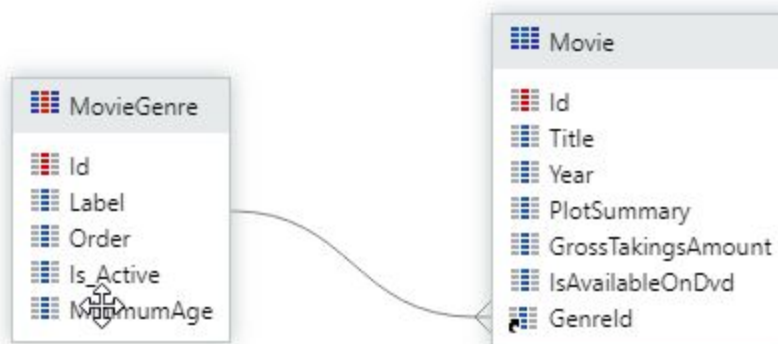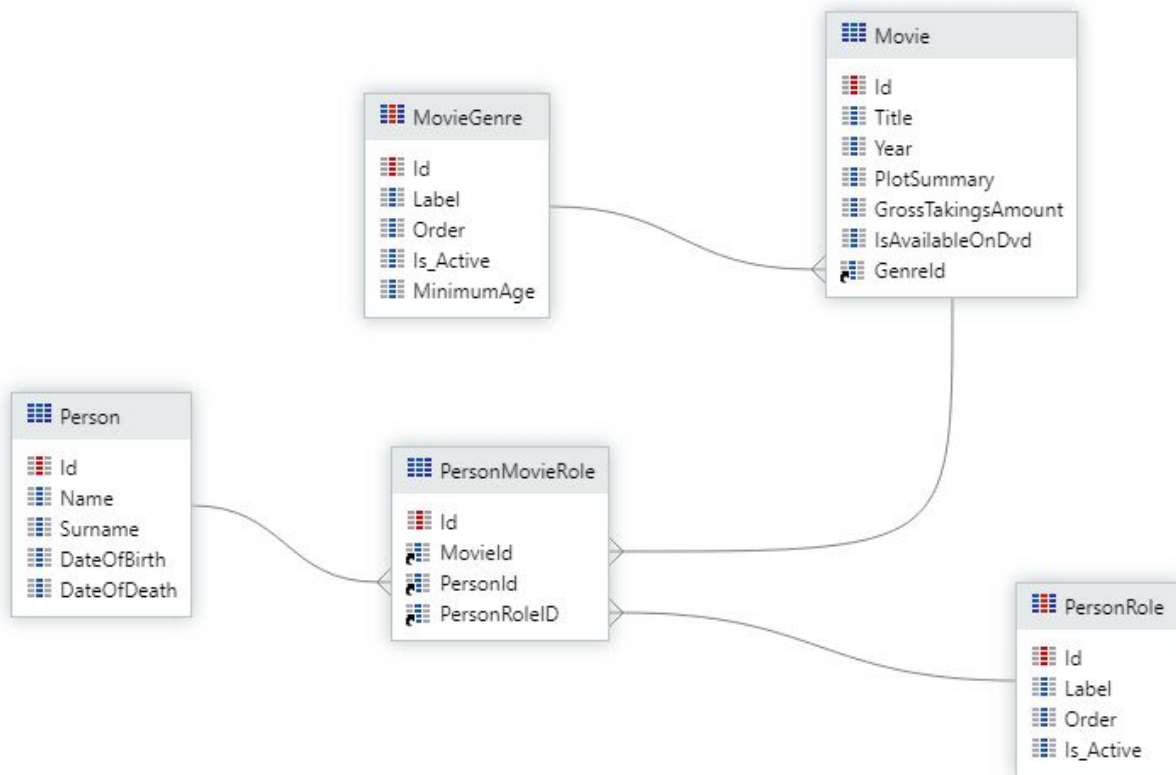

2. Name this diagram *MoviesDB Diagram*.

3. Drag and drop the **Movie** Entity into the *MoviesDB Diagram* canvas.

4. Drag and drop the **MovieGenre** Static Entity to the diagram. Notice the connector that links the **GenreId** attribute to the MovieGenre Static Entity highlighting the one-to-many relationship.

5. Add the **Person** Entity and **PersonMovieRole** and **PersonRole** Static Entities to the Entity Diagram. The Diagram should look like this:

6. Publish your module to finish!