outsystems

Modeling Data - Exercise

Table of Contents

Outline

In this exercise lab, we will create the data model for the OSMDb application created in the previous exercise.

The data model of this application will exclusively be created on the OSMDb_Core module, and will consist at this stage on two Entities, **Movie** and **Person**, and on two Static Entities, **MovieGenre** and **PersonRole**. These Entities will represent the movies (Movie) in the database and their genres (MovieGenre), as well as the cast and crew (People) and the role they play in the movies (PersonRole).

The movie genres we will use in this should be Comedy, Action, Drama and Horror, while the cast and crew roles used will be Director, Producer, Actor and Crew.

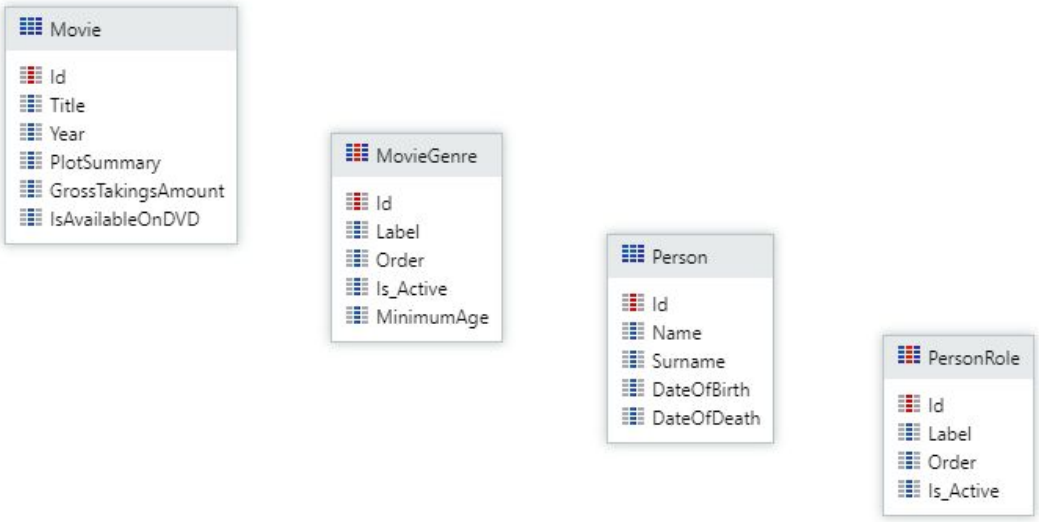
Resources

For this exercise, we will use two Excel files: *Movies.xlsx* and *People.xlsx*. These files can be found in the Resources folder of the Boot Camp materials.

Hands-on

In this exercise, we will create the data model of the OSMDb application. This data model will be created in the OSMDb_Core module and should consist of two Entities, Movie and Person, and two Static Entities, MovieGenre and PersonRole.

The data model should look like the following screenshot



In the Movie Entity, the **Title** and **Year** should be set as mandatory in the Entity's properties. Also, make sure to set a proper **Length** to the **PlotSummary** attribute, since it's a longer text. In the Person Entity, the **Name**, **Surname** and **DateOfBirth** should be mandatory.

The MovieGenre Static Entity should have an extra attribute regarding the minimum age for a person to watch a movie of that particular genre. The genres that we will have in the database, plus the respective minimum age are:

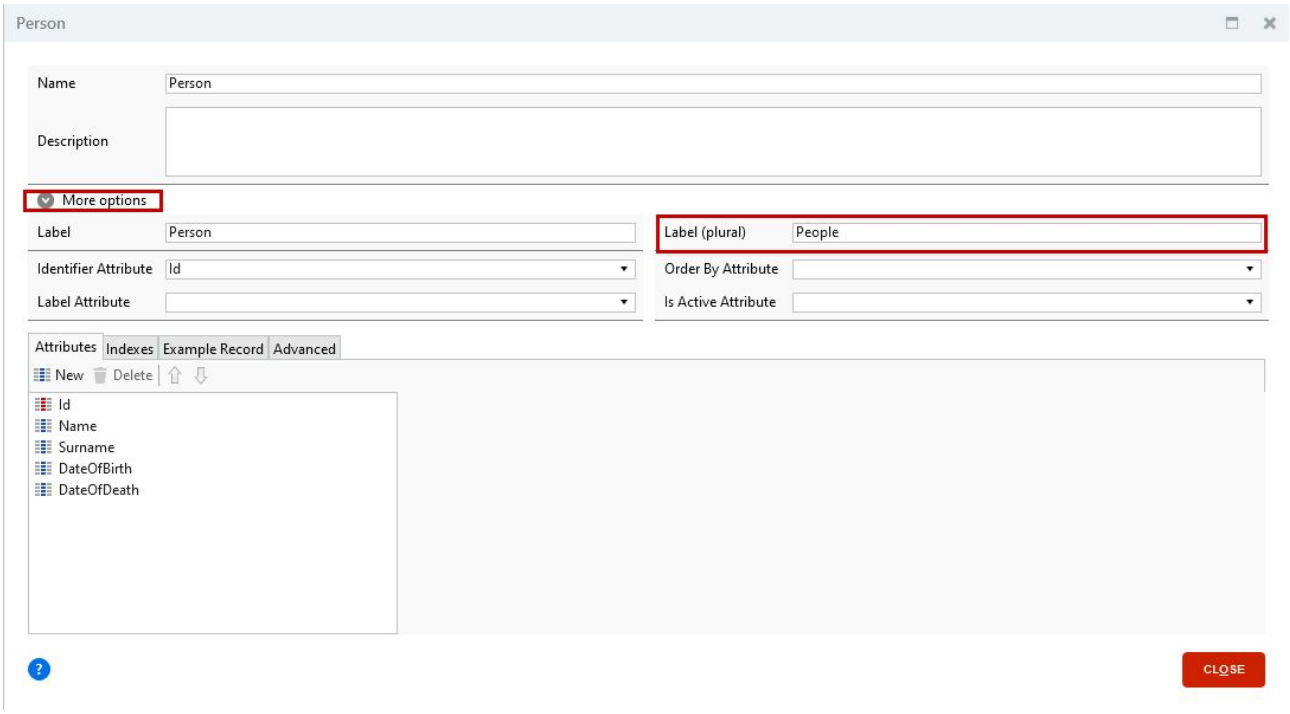
- Comedy: 6
- Action: 12
- Drama: 16
- Horror: 18

The PersonRole Static Entity should have the following records:

- Director
- Producer
- Actor
- Crew

These Entities should be set to **Public** and exposed with **read only permissions** in their respective properties area, since they will be used in the future in the OSMDb module.

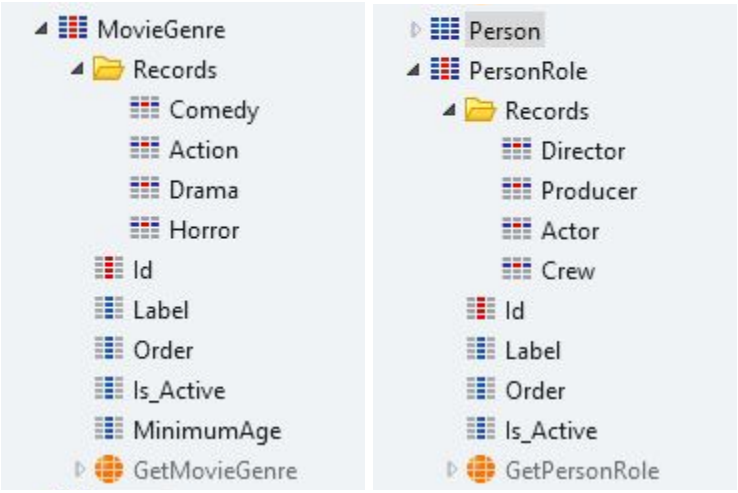
Double-click the Person Entity to open the Entity Editor. In the **More options** section edit the **Label Plural** field to *People*.



Populating the Entities with Data

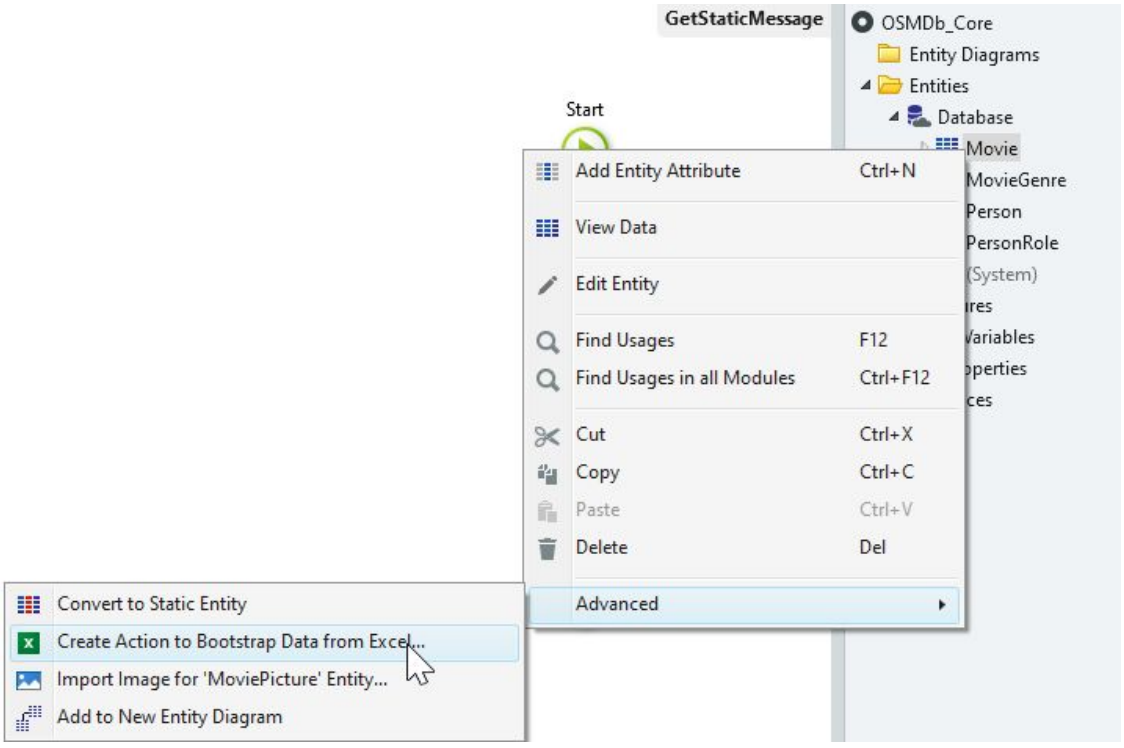
The Entities that were just created are empty. So, now we need to populate them with data.

On the other hand, the Static Entities already have the data in the database, since we added the corresponding records for the MovieGenre and PersonRole manually.

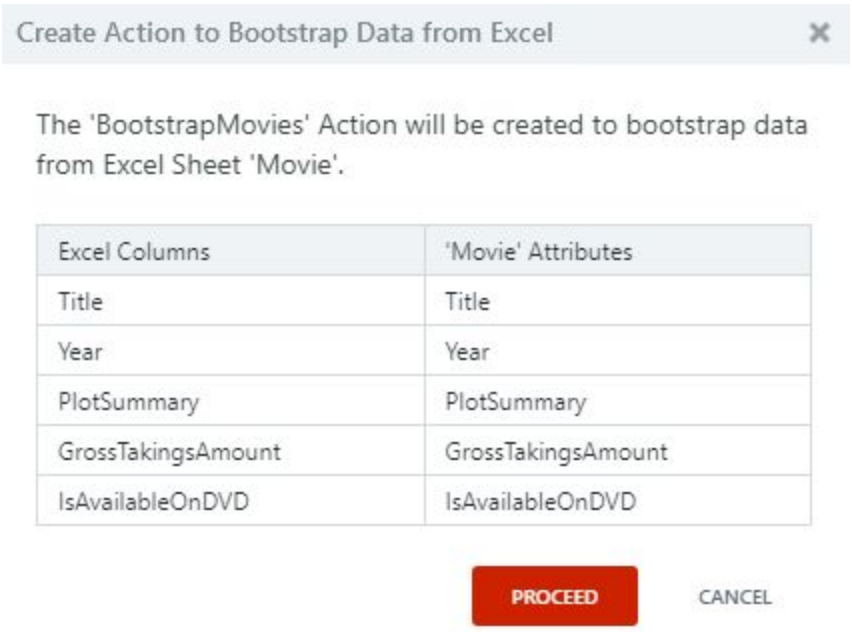


For the Movie and Person Entities, use the Excel files in the resources folder to bootstrap data from the files to the Entities.

To bootstrap the data, we need to select the respective Entity, in this example the Movies, right-click on it and select the option *Advanced->Create Action to Bootstrap Data from Excel...*



Then, we should select the respective Excel file, in this case the Movies.xlsx file, and move on with the process. We will get to a dialog that matches the columns in the Excel file with the attributes in the Entity. This is done automatically by the platform.

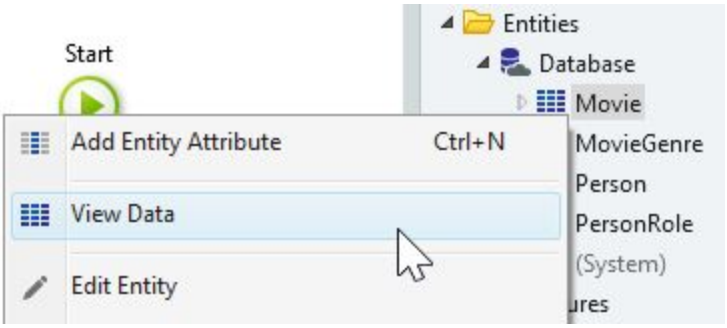


In this dialog, we **need to make sure that both columns are properly filled** with the names of the columns / attributes. Otherwise, the bootstrap will ignore the unmatched columns and the respective Entity attributes will not be populated with data. This matching is done per name and by type, meaning that the column name in the Excel file must have the same name of the Entity attribute, and the content of the column must match the Entity attribute's data type.

If everything looks ok, just like in the previous screenshot, we can **Proceed**.

NOTE: The logic for fetching the data from the Excel file and adding it to the database is created in the Action **BootstrapMovies,** under the Logic tab. It checks if any Movies currently exist. If not, it imports the Movies from the Excel spreadsheet and creates a Movie in the database, for each row in the spreadsheet. The Excel file will be saved inside the module, in the Resources folder under the Data tab. This Action runs when the module is published.

To make sure this worked, we can right-click on the Entity again and select the option *View Data*.



Now that the Movie has data, let's repeat the process for the Person Entity.