

# Modeling Data

## Exercise: How to

<b>Outline</b>	<b>1</b>
<b>Resources</b>	<b>1</b>
<b>How to</b>	<b>1</b>
Create Entities	1
Bootstrap Data from Excel	8
Create Static Entities	10

# Outline

In this exercise lab, we will create the data model for the OSMDb app created in the previous exercise. At this stage, the data model will consist of two Entities, **Movie** and **Person**, and 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.

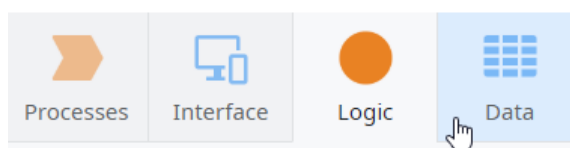
## How to

In this document, we will describe, step by step, the exercise 2 - *Modeling Data Exercise*.

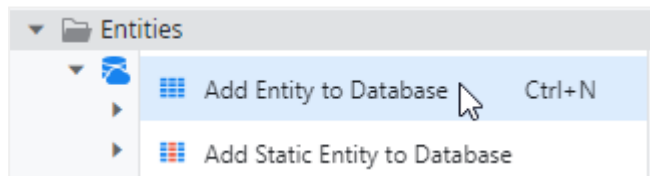
### Create Entities

Let's start by creating the two Entities: Movies and Person.

1. Create the *Movie* Entity with the following attributes: *Title*, *Year*, *PlotSummary*, *GrossTakingsAmount*, and *IsAvailableOnDvd*, with the first two being **mandatory**.
  - a. Click on the **Data** tab in the upper right corner of the workspace to switch to the Data elements.



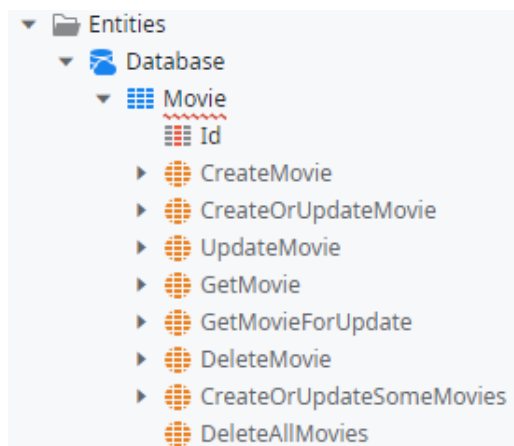
- b. Right-click on the **Entities** folder and select **Add Entity to Database**.



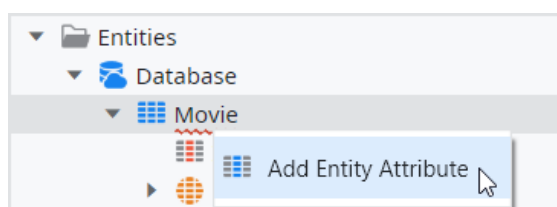
- c. Type *Movie* in the **Name** of the Entity.

A screenshot of the 'Movie Entity' configuration form. The form has a title bar with a blue icon and the text 'Movie Entity'. Below the title bar, there are three input fields: 'Name' (containing 'Movie'), 'Description' (empty), and 'Public' (set to 'No'). At the bottom right, there are two buttons: 'Indexes' and 'More...'. The 'Name' field is highlighted with a blue border.

- d. Expand the **Movie** Entity to see its attributes. There is an auto-number *Id* attribute and eight Entity Actions to provide typical Create, Read, Update and Delete (CRUD) functionality.



- e. Right-click on the Movie Entity and select **Add Entity Attribute**.



- f. Enter *Title* as the **Name** of the attribute. Notice that the error disappeared.
- g. Change its **Is Mandatory** property to *Yes*.

**Title**  
Entity Attribute

Name	<input type="text" value="Title"/>
Description	<input type="text" value="..."/>
Label	<input type="text" value="Title"/>
Data Type	<input type="text" value="Text"/>
Length	<input type="text" value="50"/>
Is Mandatory	<input type="text" value="No"/>
Default Value	<input type="text" value="Yes"/>

- h. Right-click on the **Movie** Entity and select **Add Entity Attribute** again.
- i. Set the **Name** of the attribute to *Year*. Notice the default **Data Type** is *Integer*. Make this attribute **Mandatory** as well.
- j. Add the *PlotSummary* attribute. Set its **Length** to 500 characters.

**PlotSummary**  
Entity Attribute

Name	<input type="text" value="PlotSummary"/>
Description	<input type="text" value="..."/>
Label	<input type="text" value="Plot Summary"/>
Data Type	<input type="text" value="Text"/>
Length	<input type="text" value="500"/>
Is Mandatory	<input type="text" value="No"/>

**Note:** The Length property determines the maximum length of the text attributes in terms of the number of characters. This means the plot summary of the movie will only have 500 characters at most.

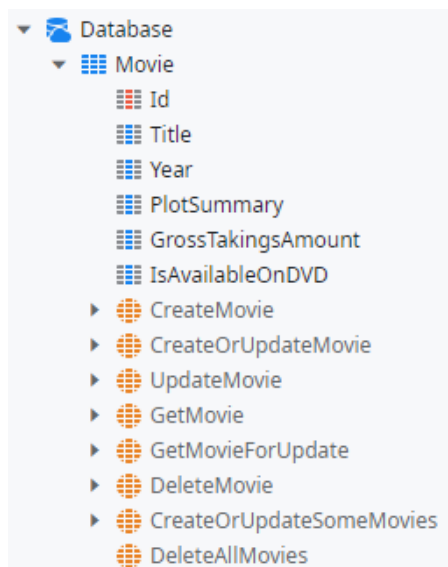
- k. Add the *GrossTakingsAmount* and *IsAvailableOnDVD* attributes. Notice the default **Data Types** are correctly set to *Currency* and *Boolean*.

- l. Select the GrossTakingsAmount attribute and change its **Label** property to *Gross Takings*.

The screenshot shows the configuration form for the 'GrossTakingsAmount' Entity Attribute. The form includes fields for Name, Description, Label, Data Type, and Length. The 'Label' field is highlighted with a red rectangular box and contains the text 'Gross Takings'. The 'Data Type' is set to 'Currency'.

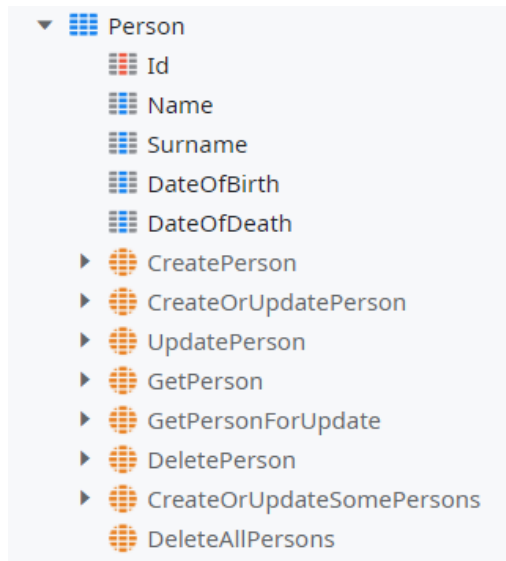
**Note:** The Label property is useful to set how we want the attribute to be identified in the UI when used, for instance, in a Table. This way, we are shortening the Label and you will see the results in the UI in a later exercise.

- m. The Movie Entity should look like this:



2. Create the *Person* Entity with the following attributes: *Name*, *Surname*, *DateOfBirth*, and *DateOfDeath*, with the first three being **Mandatory**.
  - a. On the **Data** tab, create a new Entity and name it *Person*.
  - b. Add the following attributes to the Entity: *Name*, *Surname*, *DateOfBirth*, and *DateOfDeath*. Did ODC Studio guess the Data Types correctly from the attribute name? **Always double-check!**

- c. Set *Name*, *Surname*, and *DateOfBirth* as **Mandatory**.
- d. The Person Entity should look like the following image:



- e. Select the Person Entity, then click on the button **More...** in the properties editor to launch the **Entity Editor**.



- f. In the Entity Editor dialog, expand the **More Options** section and change the **Label (plural)** property to *People* and click the **Close** button.

Person ×

Indexes

This entity has no indexes

[New index](#)

[What are indexes?](#)

Example record

Id	Name	Surname	DateOfBirth

More Options ⌵

? Close

Person ×

Id	Name	Surname	DateOfBirth

More Options ⌵

Update Behavior

Changed Attributes ⌵

☐ Expose Process Events

Label

Person

Label (plural)

People

Identifier Attribute

Id ⌵

Order By Attribute

⌵

Label Attribute

⌵

Is Active Attribute

⌵

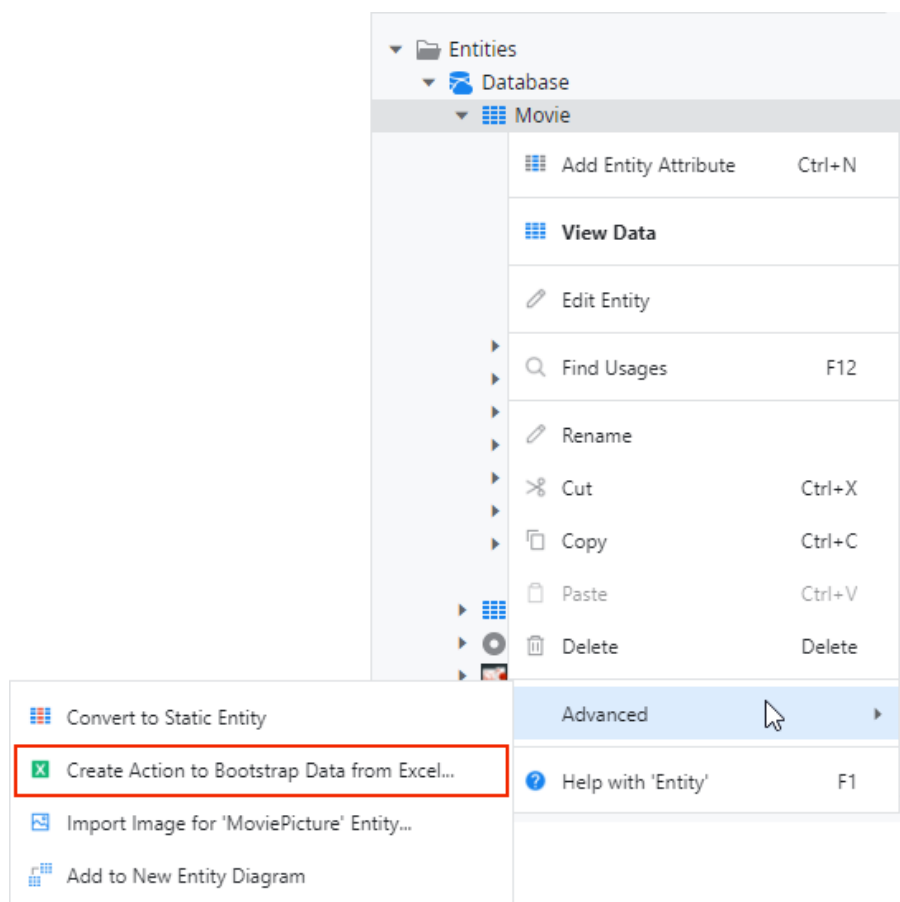
? Close

**Note:** Remember when we changed the Label of the Gross Takings Amount? Here, we're doing something similar, but for the cases where the plural of Person will be used in the UI.

## Bootstrap Data from Excel

Now that we have the two Entities created, we will populate them with data bootstrapped from two Excel files.

1. Populate the Movies Entity with data by bootstrapping data from the Excel file **Movies.xlsx**.
  - a. Right-click on the Movie Entity, select **Advanced**, then the option **Create Action to Bootstrap Data from Excel...**



- b. Browse to the Resources folder and select the **Movies.xlsx** Excel file.



- c. In the new dialog, ensure that the column's headers names from the Movies.xlsx excel file (**Excel Columns**) match the names of the Movie Entity attributes (**'Movie' Attributes**), then click on the **Proceed** button. Otherwise, click on the **Cancel** button and fix the names or data types of the mismatched Movie Entity attributes. Then, redo the previous steps. Your Entity attributes **must match in name and type** what is in the Excel file.

Create Action to Bootstrap Data from ...

The 'BootstrapMovies' Action will be created to bootstrap data from Excel Sheet 'Movie'.

Excel Columns	'Movie' Attributes
Title	Title
Year	Year
PlotSummary	PlotSummary
GrossTakingsAmount	GrossTakingsAmount
IsAvailableOnDVD	IsAvailableOnDVD

?

Proceed

Cancel

2. Repeat the previous steps and populate the Person Entity with data from the Excel file **People.xlsx** by bootstrapping it. Don't forget to verify if the values in **Excel Columns** match the ones in **'Person' Attributes** before you click on **Proceed**.

Create Action to Bootstrap Data from ...

The 'BootstrapPeople' Action will be created to bootstrap data from Excel Sheet 'Person'.

Excel Columns	'Person' Attributes
Name	Name
Surname	Surname
DateOfBirth	DateOfBirth
DateOfDeath	DateOfDeath

?

Proceed

Cancel

3. Publish the app to save the changes.



**Note:** By default, the bootstrap data is made when the app is published. So, when you publish the app at this point, the data will be automatically fetched from the Excel files and the respective records will be created in the database.

4. If you want to preview your data, just double-click on the Entity. You will see on the main window of ODC Studio a preview of the data. If you don't see it immediately, just give it a couple more seconds, since the bootstrap can still be running.

The screenshot shows the ODC Studio interface. The main window displays a table titled 'Movie Data' with columns: Id, Title, Year, and PlotSummary. The table contains 7 rows of movie data. The sidebar on the right shows a tree view of the database structure, including 'OSMDB', 'Entity Diagrams', 'Entities', 'Database', and 'Movie'. The 'Movie' entity is selected, showing its attributes: Id, Title, Year, PlotSummary, GrossTakingsAmount, IsAvailableOnDVD, CreateMovie, and CreateOrUpdateMovie.

Id	Title	Year	PlotSummary
19	Dune: Part Two	2023	Paul Atreides unites with Chani and the Fremen while seeking
18	Dune	2021	Paul Atreides arrives on Arrakis after his father accepts the ste
17	Spider-Man: Beyond the Spider-Verse	2024	After leaving off from Spider-Man: Across the Spider-Verse, the
16	Don't Worry Darling	2022	In the 1950s, Alice and Jack live in the idealized community of
15	Avengers Endgame	2019	After the devastating events of Avengers: Infinity War (2018), t
14	Avatar 3	2024	Sequel of Avatar 2 (2022). The plot is unknown.
13	Black Panther Wakanda Forever	2022	A sequel that will continue to explore the world of Wakanda ar

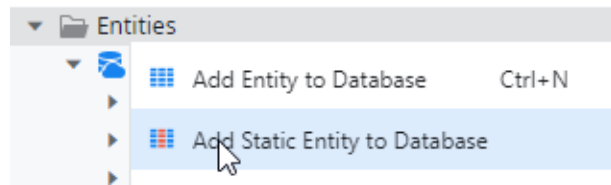
## Create Static Entities

Now that we have the two Entities to represent the movies and people, it's time to create the two Static Entities.

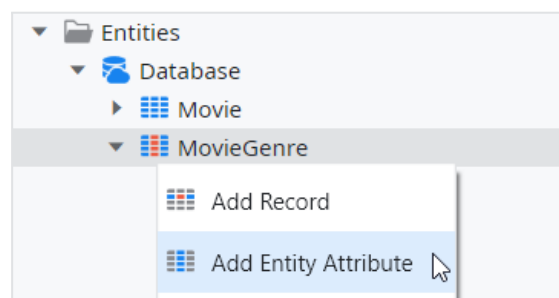
The first Static Entity, **MovieGenre**, represents the several genres that a movie can have: Comedy, Action, Drama, and Horror. The genres will have a minimum age associated with them, with the Comedy having a minimum age of 6 years old, action a minimum age of 12 years old, Drama requires an age of 16, and horror an age of 18. The second Static Entity, **PersonRole**, represents the several roles that a person can have in a movie: director, actor, producer, or a member of the crew.

1. Create a Static Entity called *MovieGenre* with the following records: *Comedy*, *Action*, *Drama*, and *Horror*. This Entity should have an additional attribute, called *MinimumAge*, with the following values for the records: 6, 12, 16, and 18.

- a. Still in the Data tab, right-click on the Entities folder and select **Add Static Entity to Database**.



- b. Type *MovieGenre* as the **Name** of the Static Entity.
- c. Notice that some attributes were automatically created. However, we want to create a new one to represent the minimum age. Right-click on the MovieGenre Entity and select **Add Entity Attribute**.



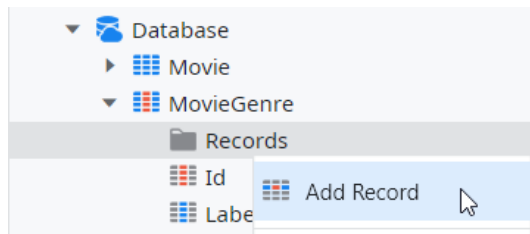
- d. Set its **Name** to *MinimumAge* and change the **Data Type** to *Integer*.

A screenshot of the 'MinimumAge' Entity Attribute configuration form. The form has the following fields and values:

- Name: MinimumAge
- Description: (empty)
- Label: Minimum Age
- Data Type: Integer (selected from a dropdown)
- Is AutoNumber: No
- Is Mandatory: No (selected from a dropdown)
- Default Value: (empty)

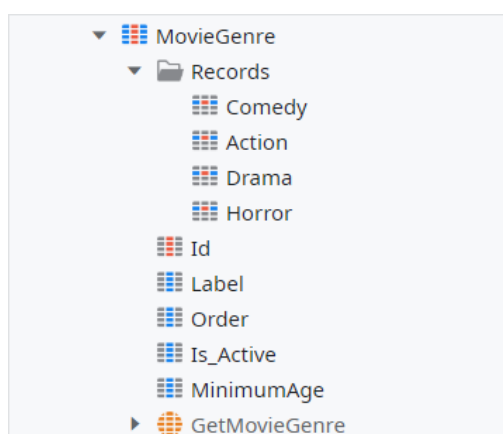
Now it's time to add the records to the Static Entity. Don't forget that it's only possible to add records to a Static Entity at design time, and not during runtime.

- e. Right-click on the Static Entity or the Records folder and select **Add Record**.

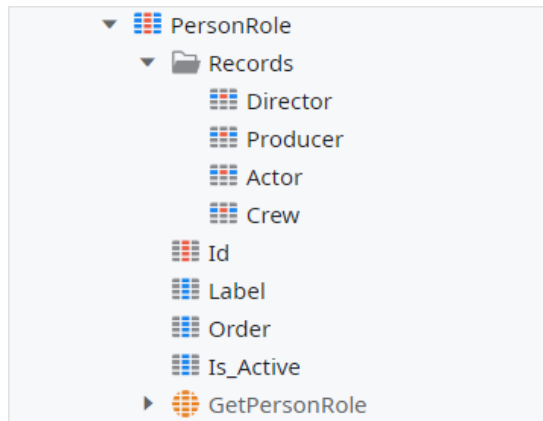


- f. Type *Comedy* for the identifier of the Record and set the **MinimumAge** to 6.

- g. Do the same for the *Action*, *Drama*, and *Horror* records with their respective **MinimumAge** values: 12, 16, and 18.
- h. The Static Entity should look like the one below:



2. Create a Static Entity called *PersonRole*, with the following records: *Director*, *Producer*, *Actor*, and *Crew*.
  - a. Create a new Static Entity and set its **Name** to *PersonRole*.
  - b. Add four Records to the Static Entity: *Director*, *Producer*, *Actor*, and *Crew*.
  - c. The Static Entity should look like this



- d. Click the **1-Click Publish** button to publish the module.



And you're done with this exercise! For now, there's no UI to show the data in the app, but we will deal with that in a future exercise.