Advanced Aggregates Exercise

Table of Contents

knowledge@outsystems.com

www.outsystems.com

Outline

In this exercise, we will create two different Aggregates, where we'll need to use multiple sources, calculated attributes, and even aggregation functions. These aggregates will need to respectively answer the following *requirements*:

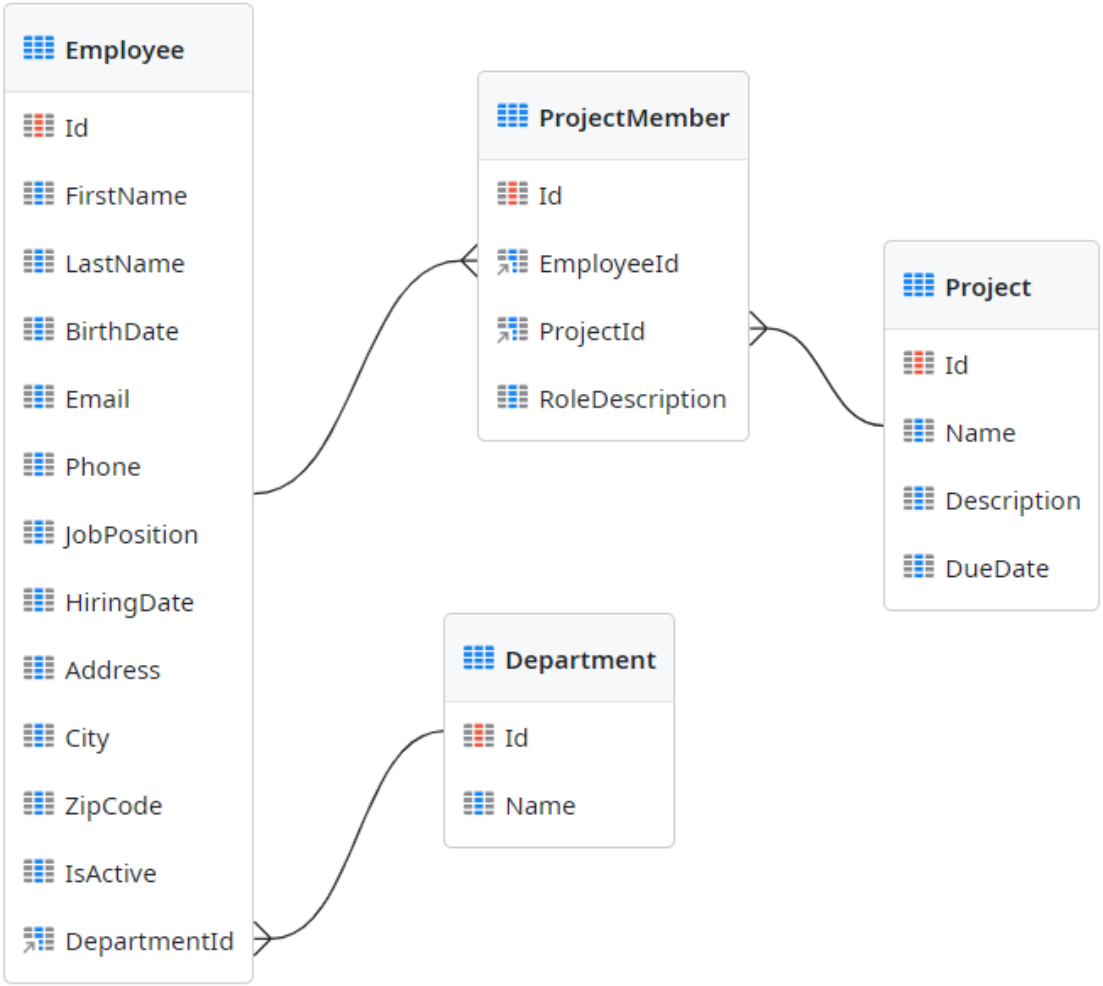
- List all the Employees and the department they belong to. This list of Employees should be sorted alphabetically and present the first and last name of the Employee in the following format: Last Name, First Name.
- List all the projects in the database. This list should give us all the name of the project, alongside the number of employees in the project, sorted from the project with more members to the one with less members.

Resources

This exercise has a Quickstart application already created. This application has everything needed to start the exercise. This quickstart application can be found in the Resources folder of this exercise, with the name **Advanced Aggregates Exercise.oap**.

Scenario

In this exercise, we will start from an existing application with one module. Inside that module, we have a data model already defined, with three Entities.



The module has the logic to import data from Excel to populate these Entities. So, when the application (and its module) is installed, the data will automatically be imported.

Finally, the module has a Server Action created, called AdvancedAggregates, where we will do most of our work. As mentioned above, we want to create two Aggregates, to give us very specific information about the Employees and Projects. As a recap, here is what we want to do with our Aggregates:

- List all the Employees and the department they belong to. This list of Employees should be sorted alphabetically and present the first and last name of the Employee in the following format: *Last Name*, *First Name*.

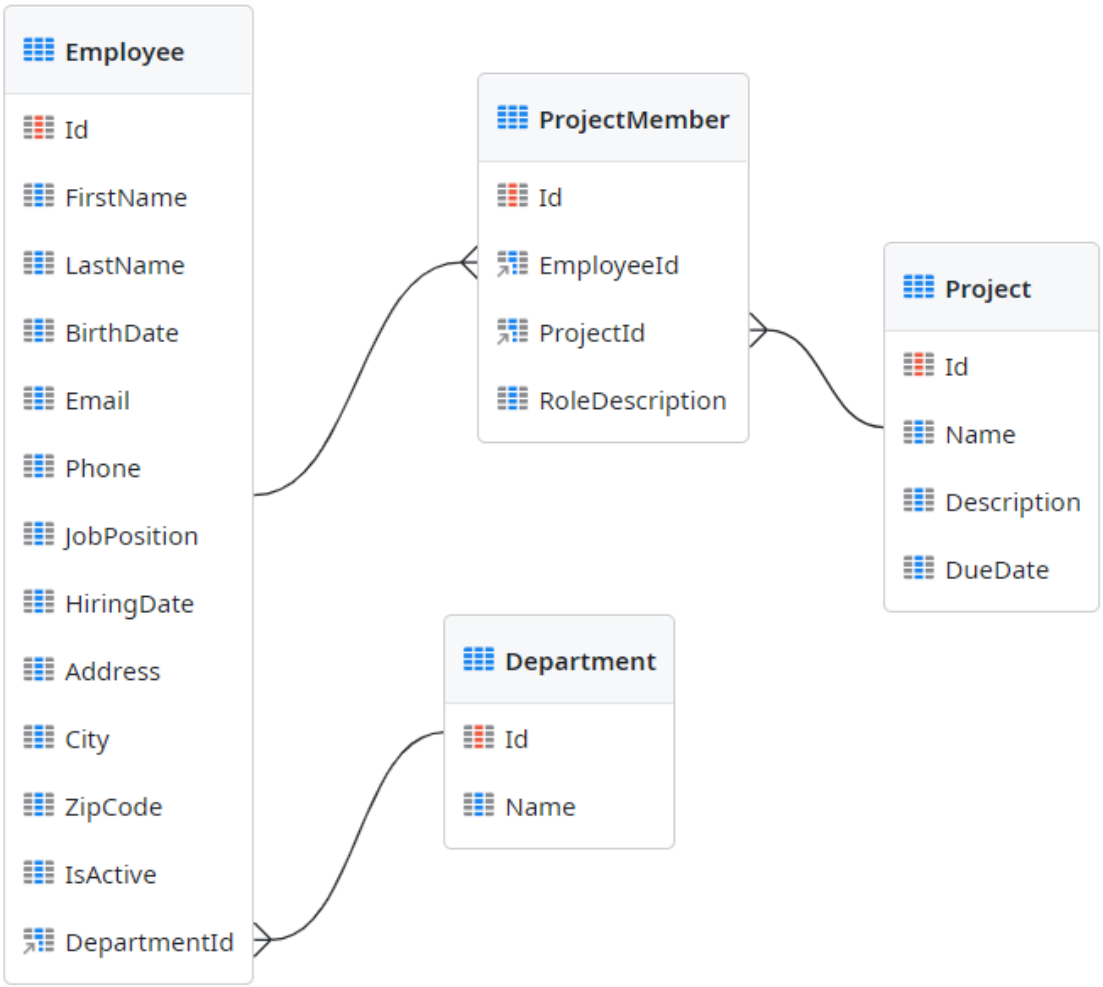
- List all the projects in the database. This list should give us all the name of the project, alongside the number of employees in the project, sorted from the project with more members to the one with less members.

How-To

In this section, we'll show you how to do this exercise, with a thorough step-by-step description. **If you already finished the exercise on your own, great! You don't need to do it again**. If you didn't finish the exercise, that's fine! We are here to help you.

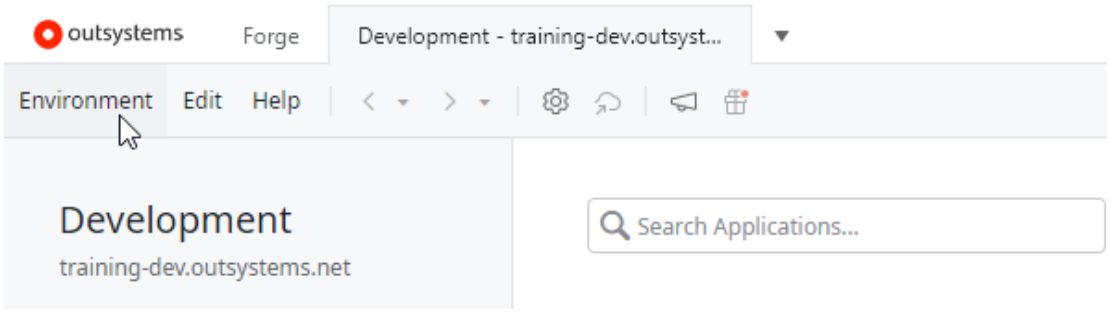
Getting Started

To start this exercise, we need to install the Quickstart file, **Advanced Aggregates Exercise.oap**. This file has three Entities created and all the logic to populate them with data.

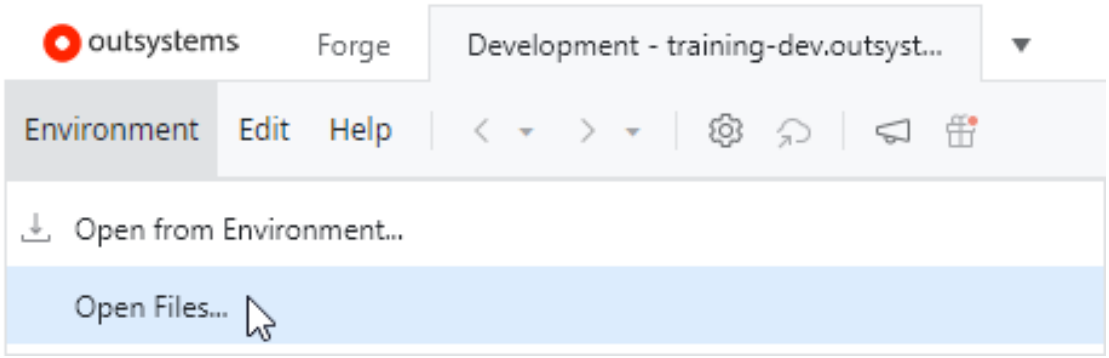


The first step that we need to take is to install the Quickstart application in our development environment. Before proceeding, you must have Service Studio open and connected to an OutSystems Environment (e.g. Personal Environment).

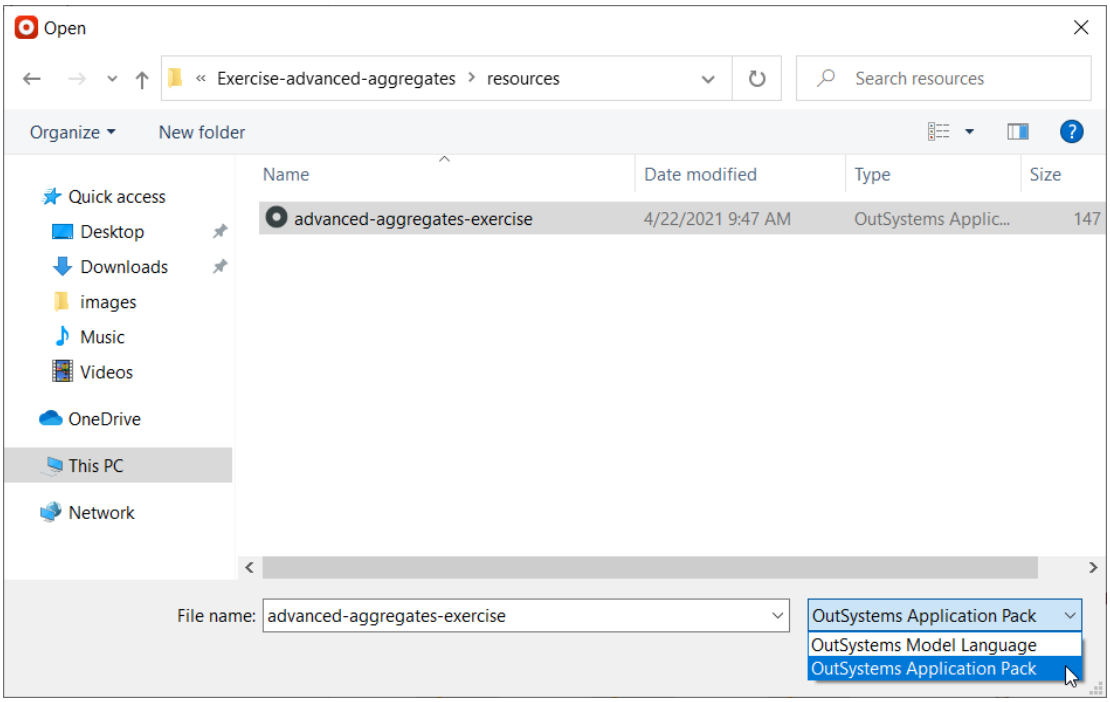
1) In Service Studio's main window, select the **Environment** menu on the top left.



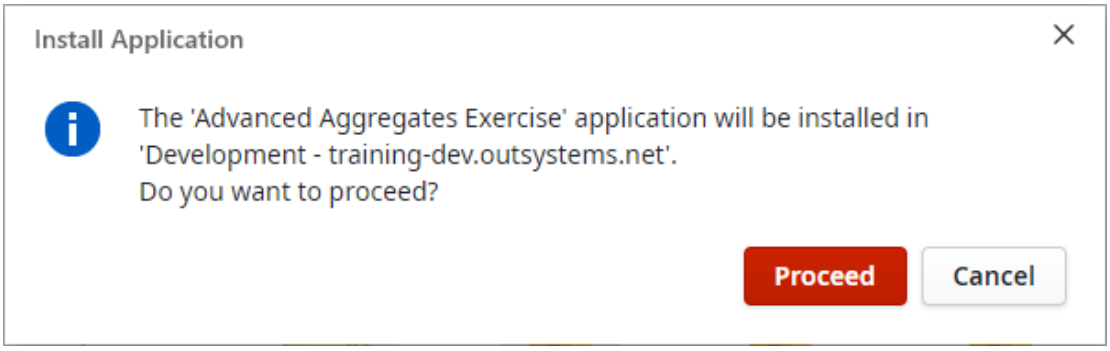
2) Select Open Files...



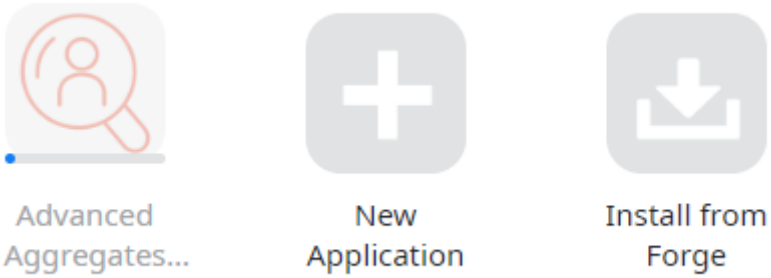
3) In the following dialog, change the file type to OutSystems Application Pack (.oap), find the location of the Quickstart and open the file named **Advanced Aggregates Exercise.oap**.



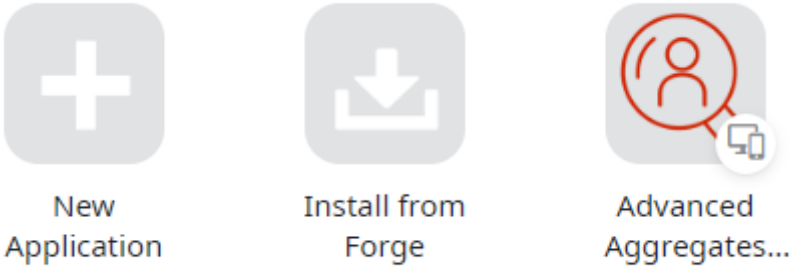
4) In the new confirmation dialog, select Proceed.



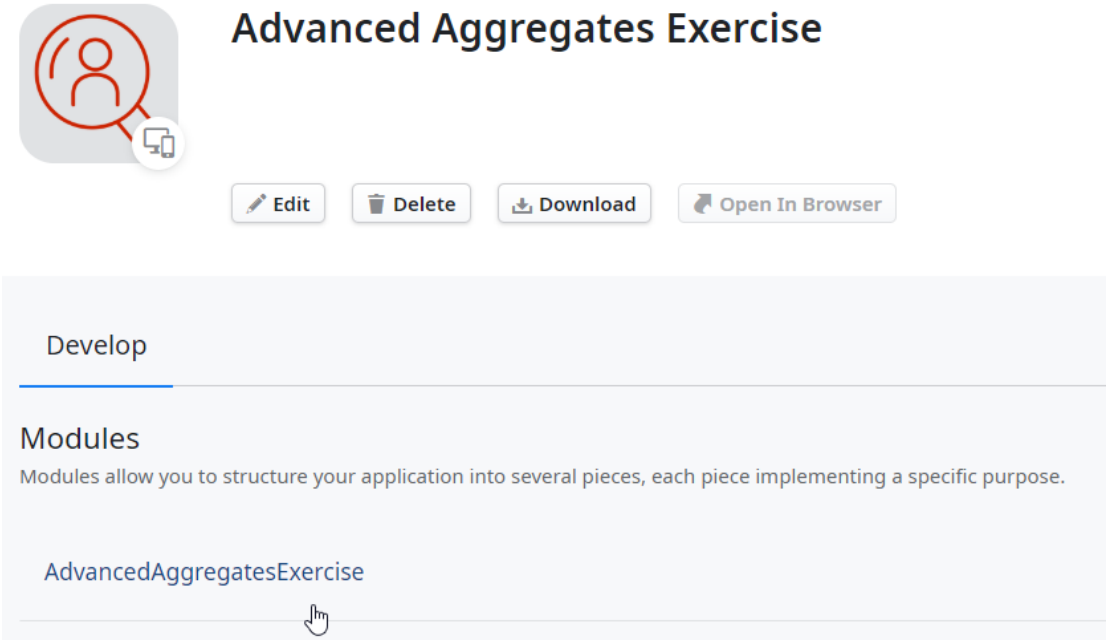
5) The application will begin installing automatically. When it's finished, we're ready to start!



6) Open the application in Service Studio.



7) The application has only one module. Let's open it!

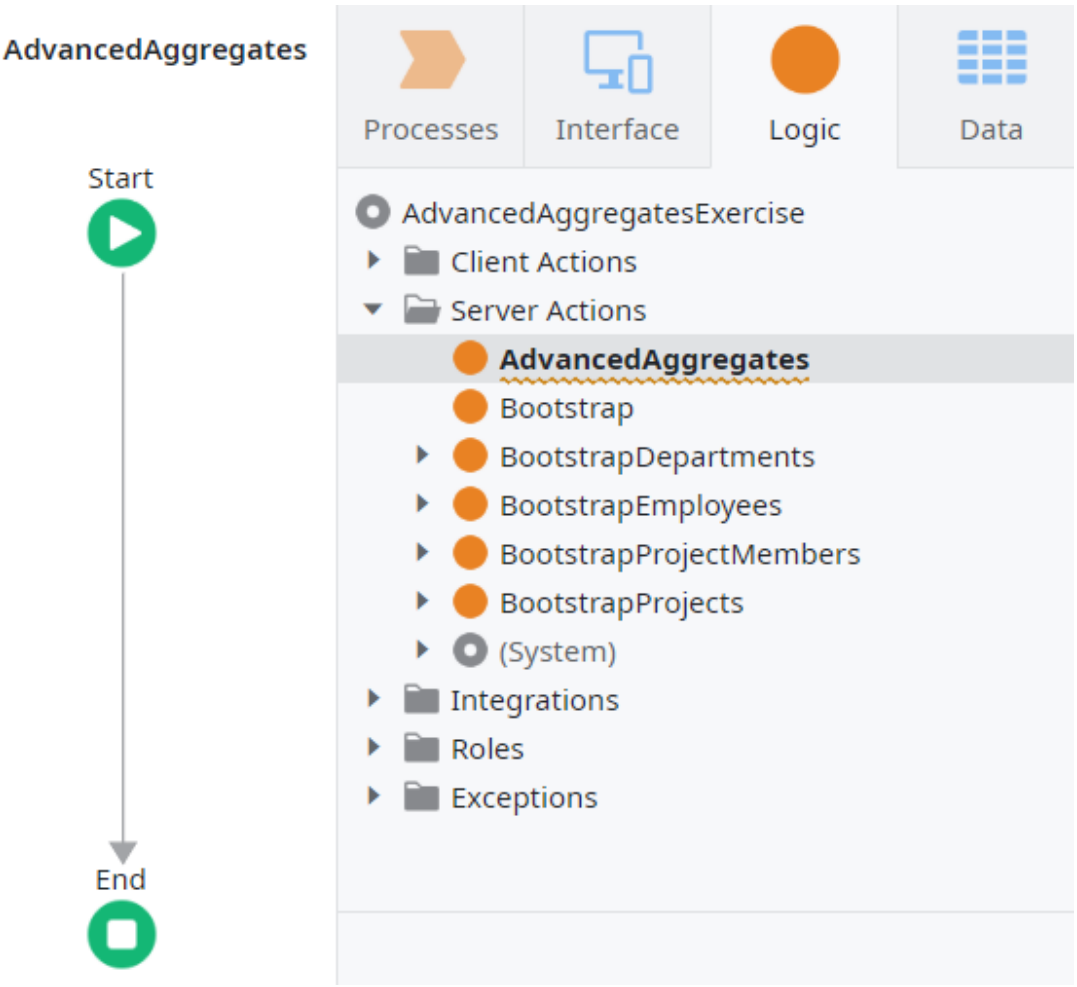


List all the Employees with Department and Full Name

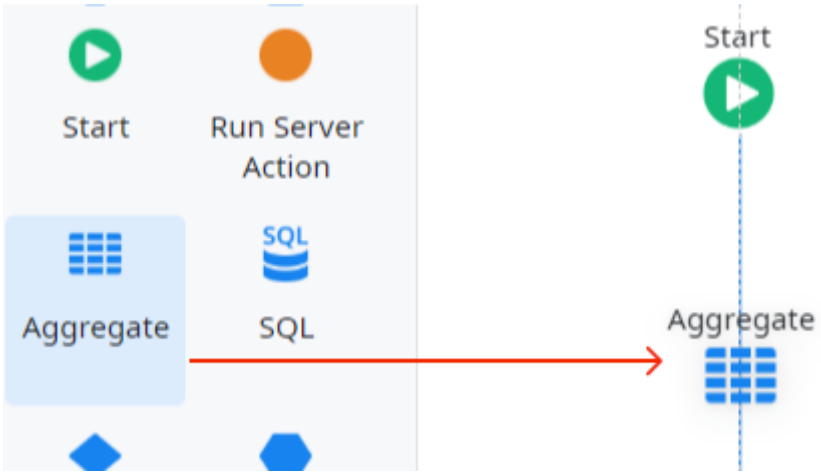
In this section, we'll create the first Aggregate to fulfill the first requirement: list all the Employees and the department they belong to.

The list of employees should be sorted alphabetically and present the employee's first and last name, with the following format: *Last Name, First Name*.

1) The module already has the AdvancedAggregates Server Action created.

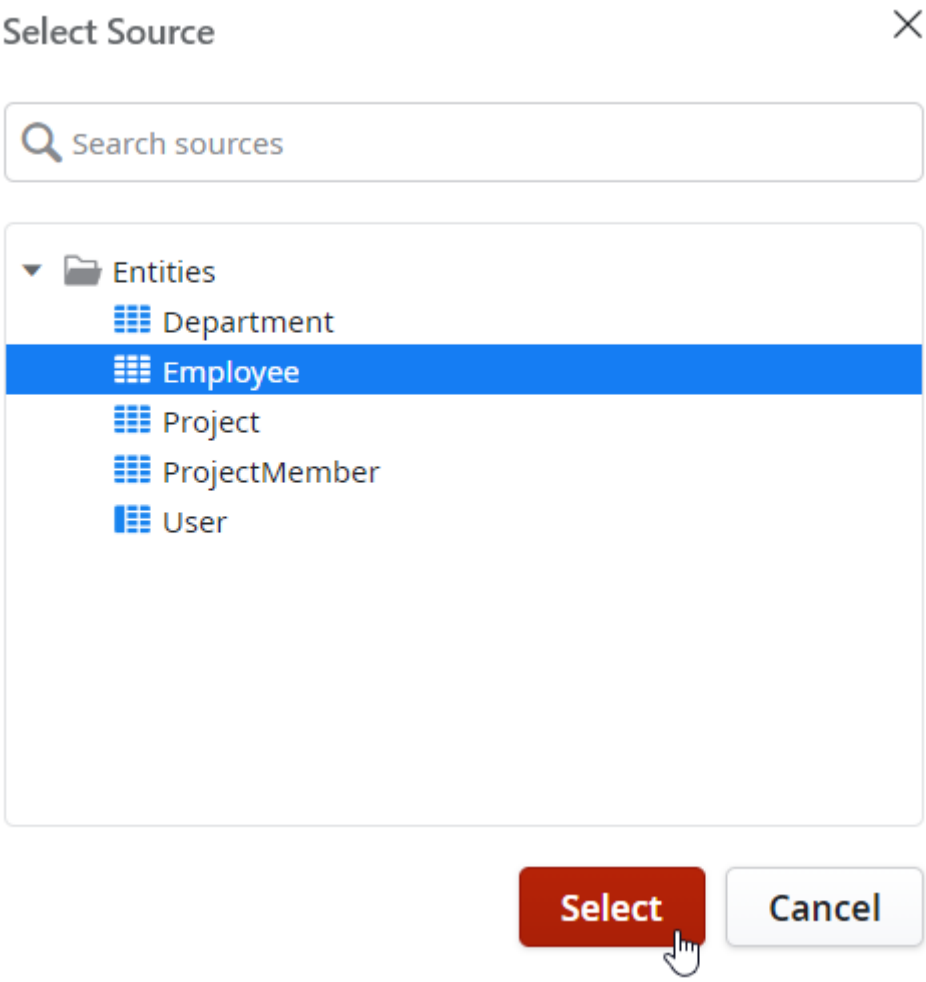


2) From the toolbox to the left, drag an Aggregate and drop it in the flow, between Start and End.

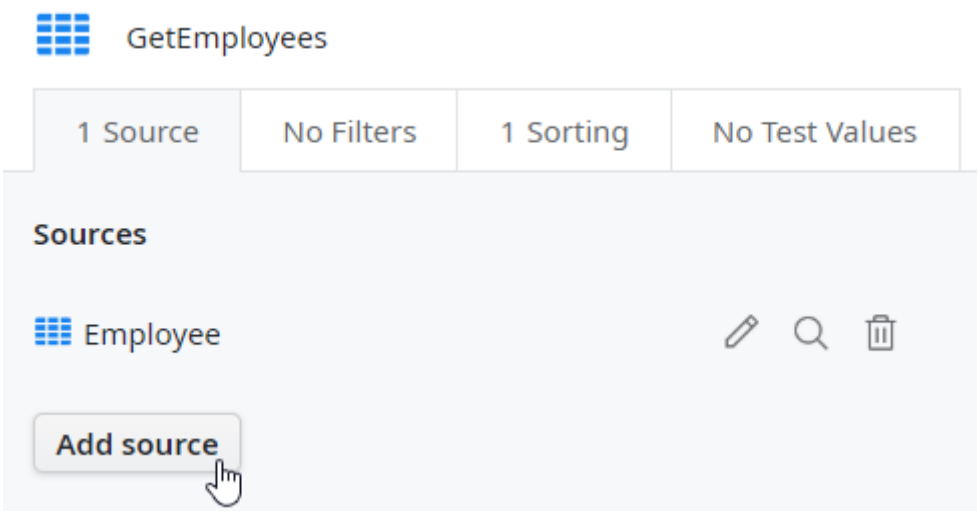


3) Double-click the Aggregate to open it.

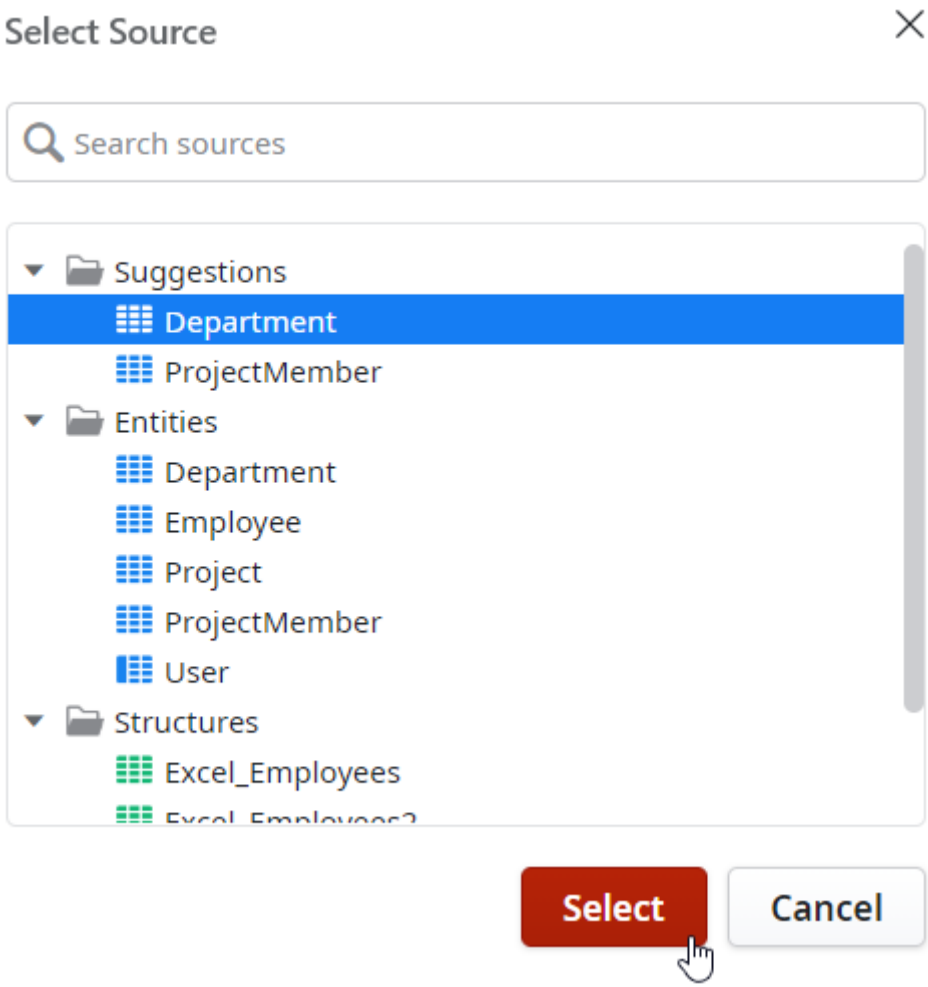
4) Click the Aggregate editor, and then select the **Employee** entity.



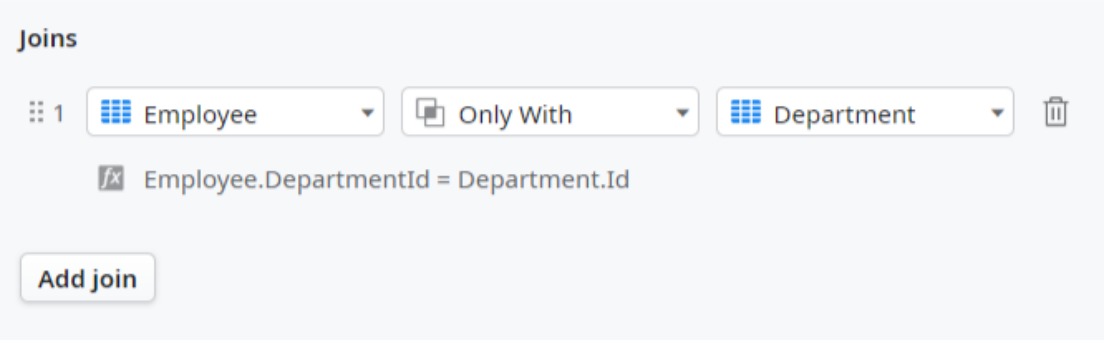
5) Click **Add Source** to add the second Entity to the Aggregate.



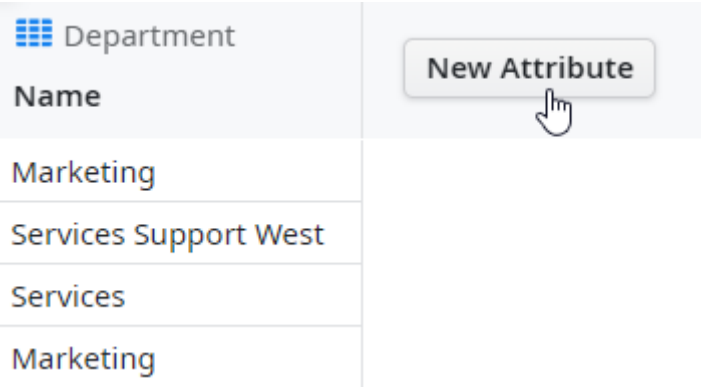
6) Select the **Department** entity and click **OK**.



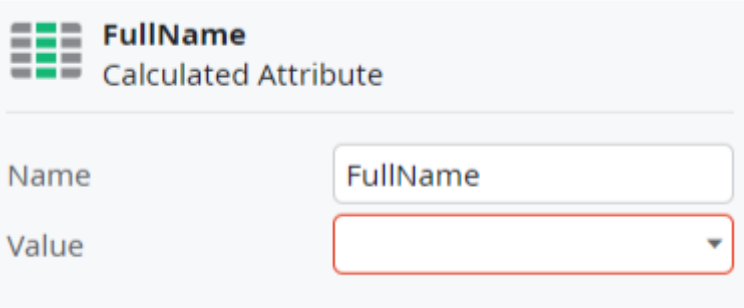
7) Notice that the Join between the two Entities was created as **Only With** automatically. Since all Employees have a Department, we will leave this as is.



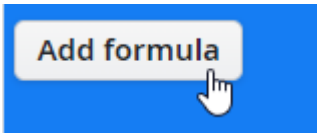
8) In the aggregate editor on the right, click **New Attribute**. You may need to scroll a bit to the right.



9) Select the **Attribute1** name and set the name of the calculated attribute to *FullName*.



10) On the new column, click **Add formula** to open the Expression Editor. You may need to scroll down a bit in the Aggregate previewer to find the option.

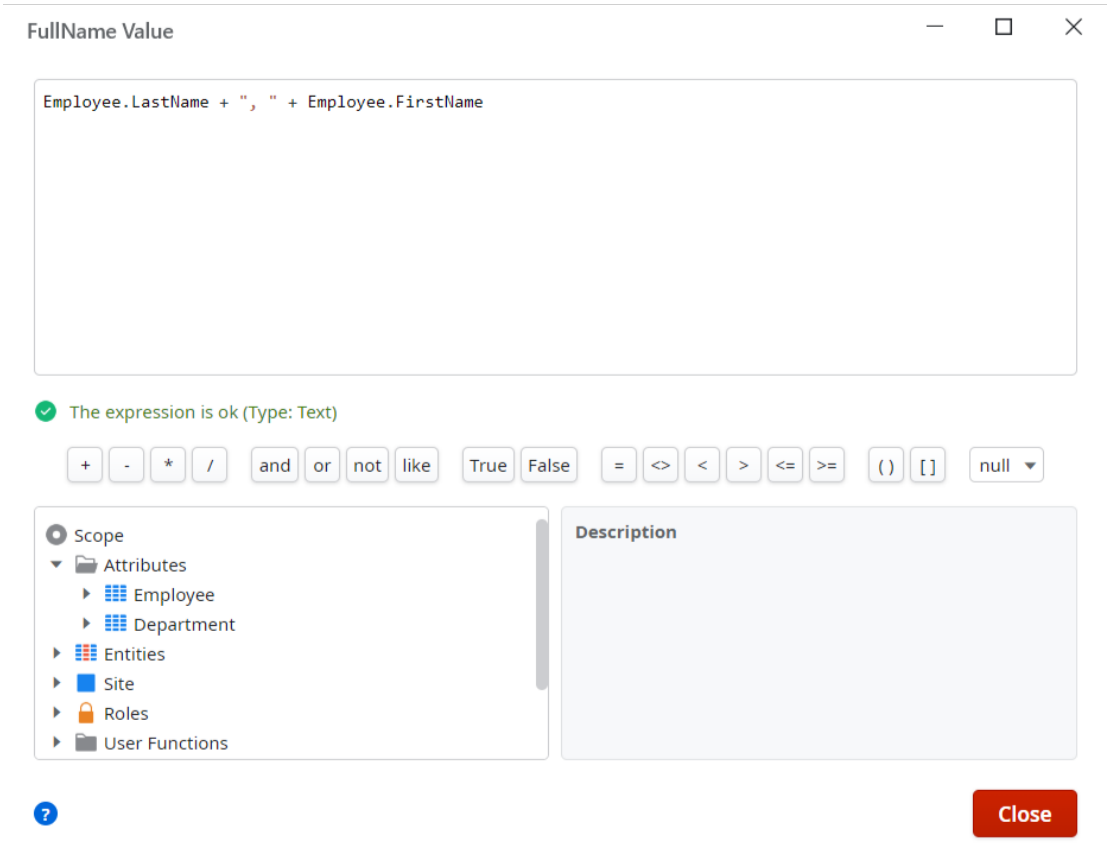


This can also be done using the *Value* property.

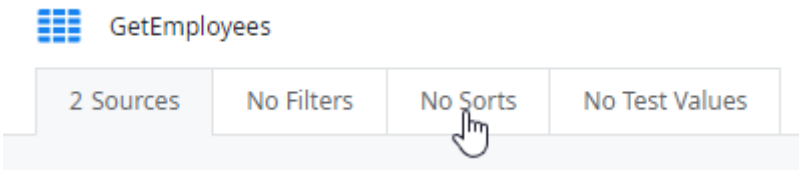
11) Set the expression to:

```
Employee.LastName + ", " + Employee.FirstName
```

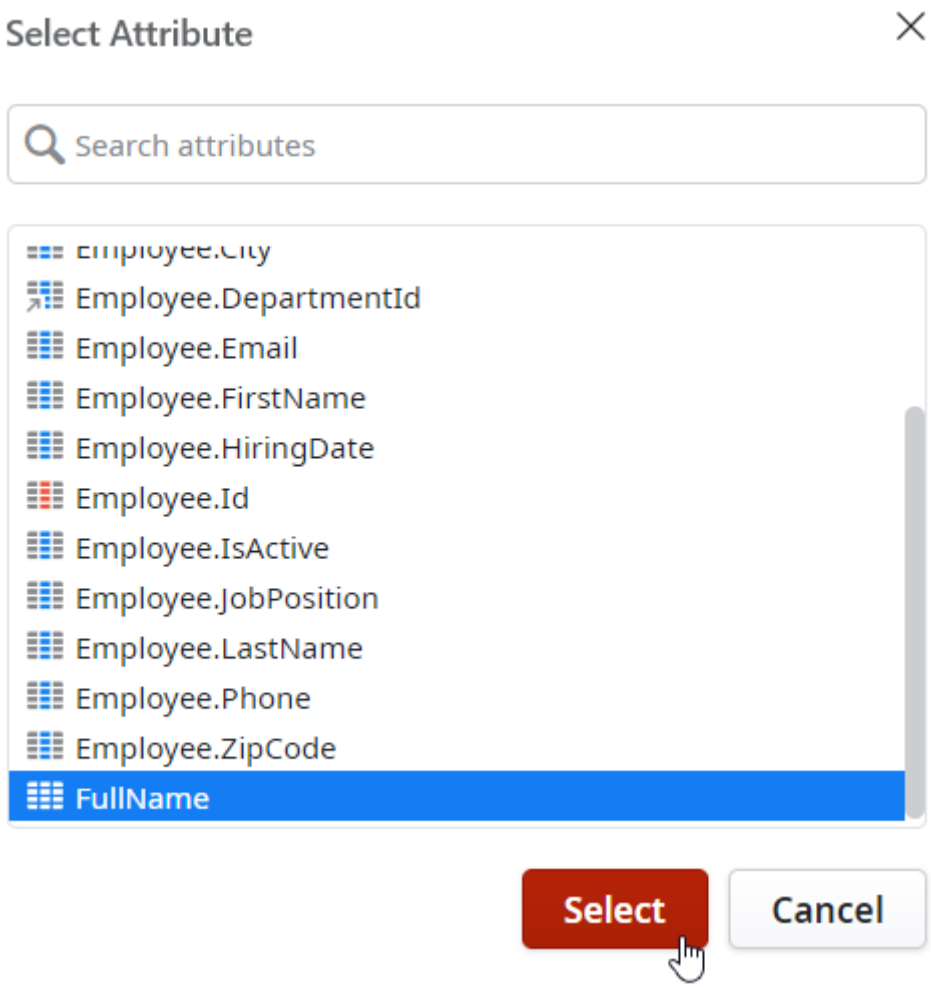
This will make sure that this column will return the Employee's full name, with the last name appearing before the first name, separated by a comma.



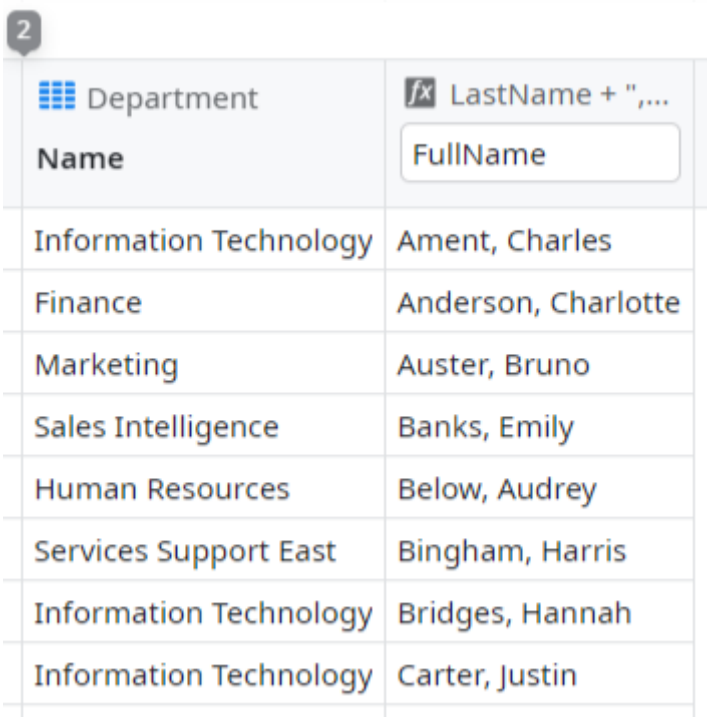
12) Click the **Sorting** option on the top of the Aggregate.



13) Click **Add Sort** and choose the **FullName** attribute.



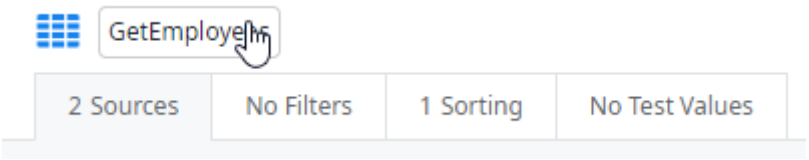
14) In the Aggregate previewer, the results should appear sorted by the Employee's full name.



2

Department Name	LastName + ",... FullName
Information Technology	Ament, Charles
Finance	Anderson, Charlotte
Marketing	Auster, Bruno
Sales Intelligence	Banks, Emily
Human Resources	Below, Audrey
Services Support East	Bingham, Harris
Information Technology	Bridges, Hannah
Information Technology	Carter, Justin

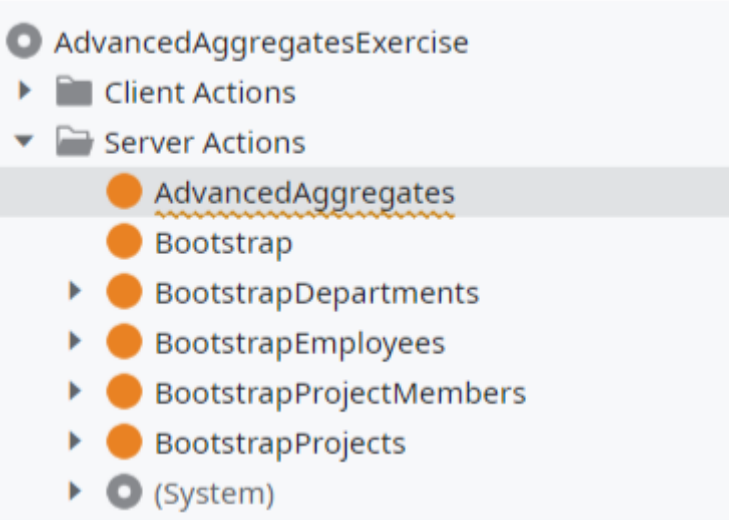
15) Change the name of the Aggregate to *GetEmployeesOnlyWithDepartments*.



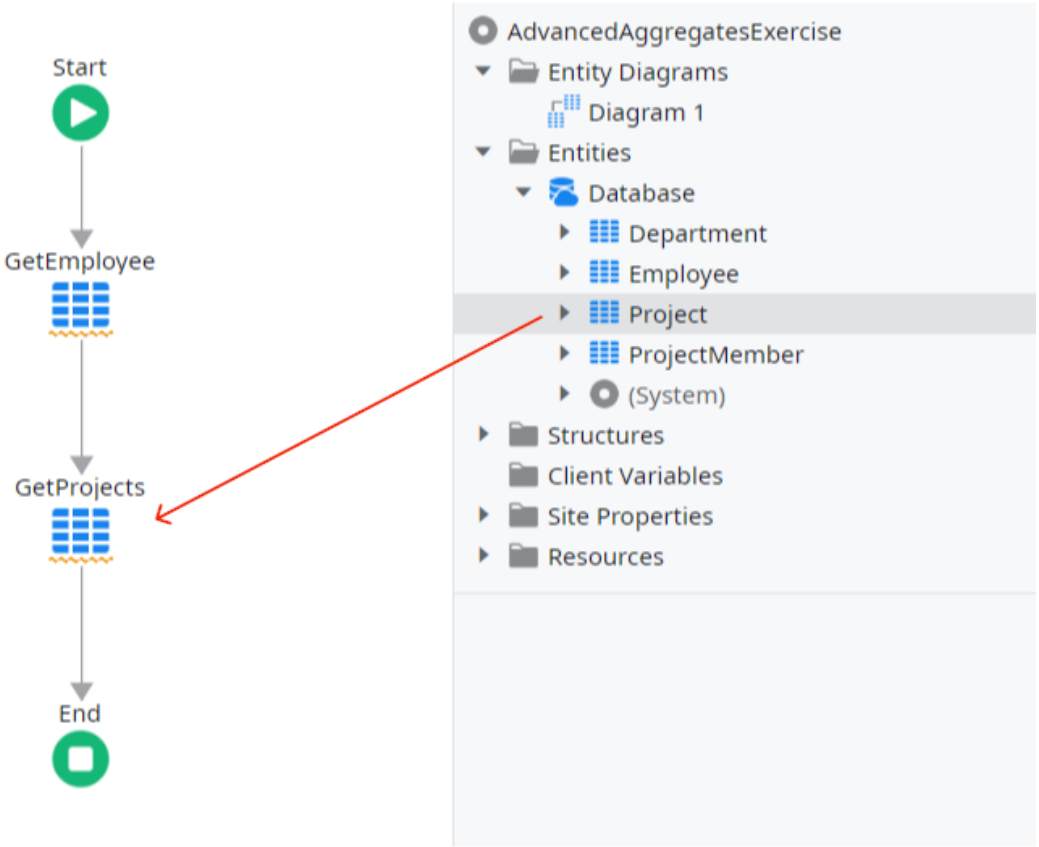
List all the Projects with Number of Employees per Project

In this section, we'll create the second Aggregate, to list all the projects in the database. This list should give us all the project's names, along with the number of employees in the project, sorted from the project with more members to the one with less members.

1) Return to the **AdvancedAggregates** Server Action, by double-clicking it on the right in Service Studio.

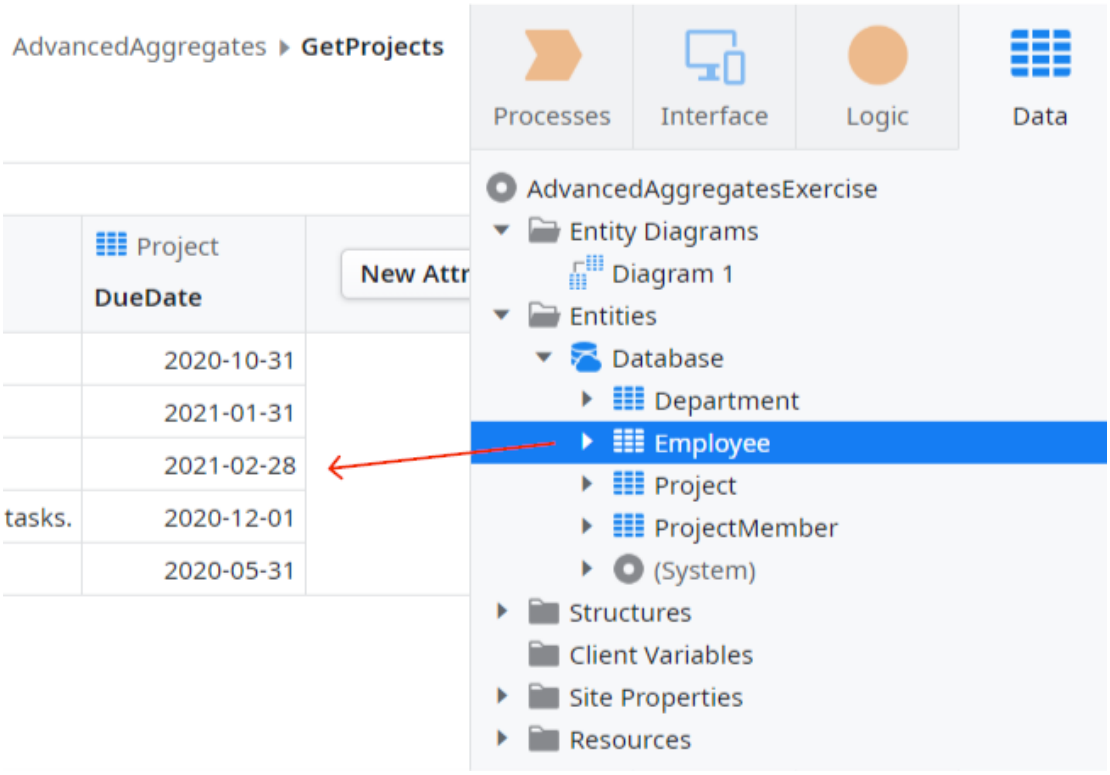


2) From the Data tab, drag the Project Entity and drop it between the existing Aggregate and the End.



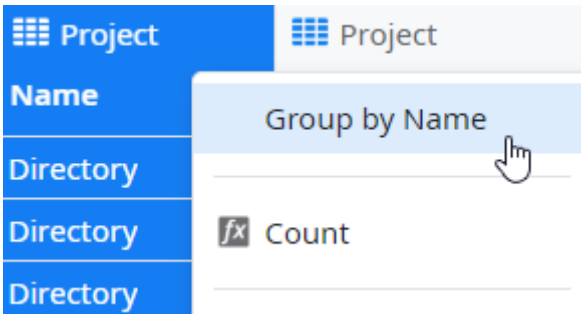
3) Double-click the **GetProjects** aggregate to open it.

4) Drag the **Employee** Entity and drop it in the Aggregate, to add it as a Source.

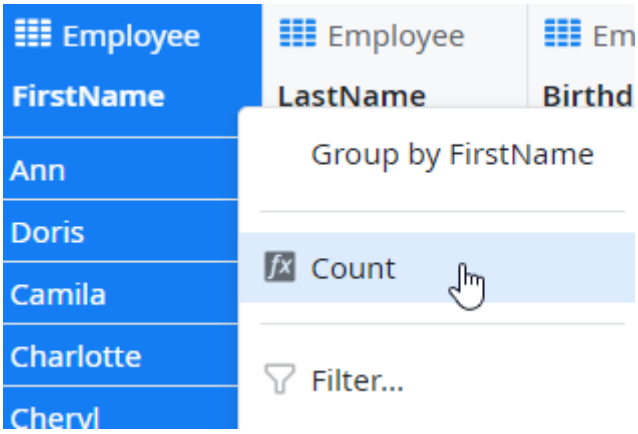


Notice that the ProjectMember Entity was also added, since it's a part of the many-to-many relationship between the two Entities.

5) Right-click the **Name** attribute of the Project Entity, and select the option **Group By Name** to group the results by Project.

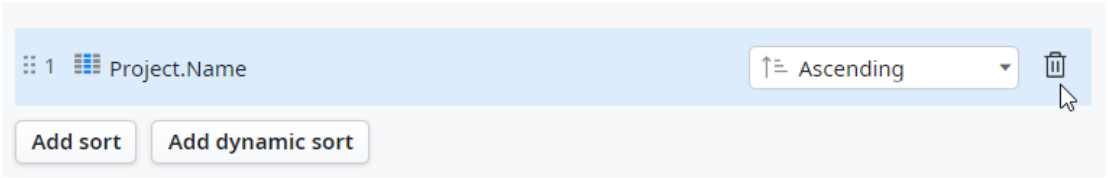


6) Right-click the **FirstName** attribute of the Employee Entity,select **Count** to determine how many employees we have per project.

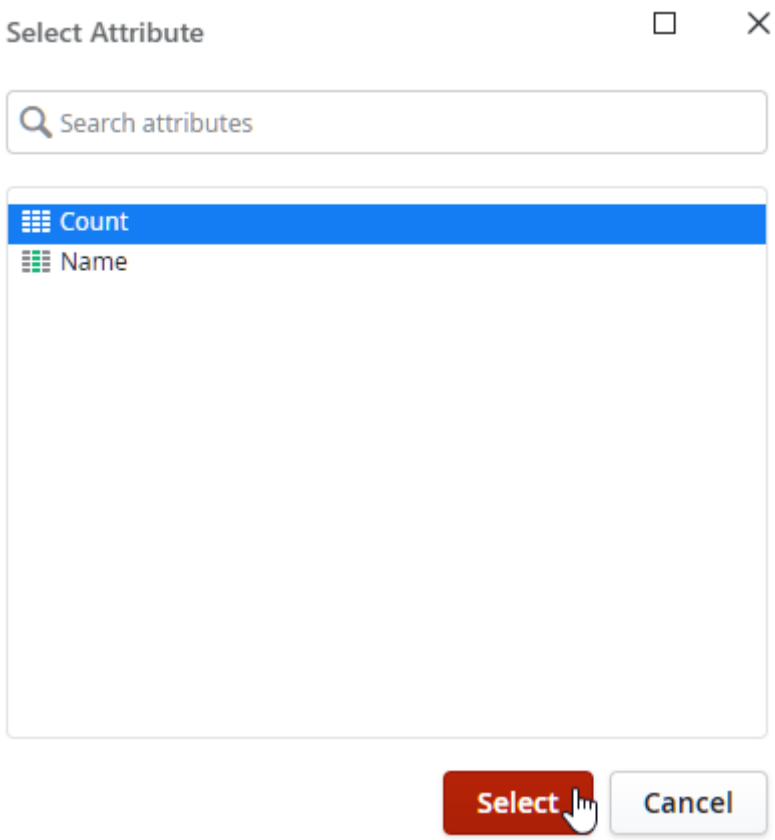


Note: Alternatively, the count can also be done by the EmployeeId attribute, being more efficient for the database.

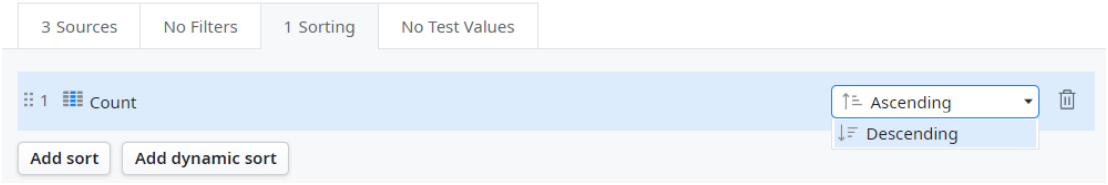
7) Open the Sorting tab, and delete the Sorting by Name.



8) Click **Add Sort** and choose the **Count** attribute. Click **OK** to close the window.



9) Change the Sorting to **Descending**, meaning from the projects with higher number of employees down.



10) The Aggregate previewer should display the results sorted by Count, in a descending order.

Group of Name Name	Count of F... Count	Project Description	Project DueDate
Task Manager	15	Web application for customers to manage their tasks. This application will allow people to submit tasks, set priorities and manage the teams working on those tasks.	2020-
		Web application for customers to manage their tasks. This application will allow people to submit tasks, set priorities and manage the teams working on those tasks.	2020-
		Web application for customers to manage their tasks. This application will allow people to submit tasks, set priorities and manage the teams working on those tasks.	2020-
		Web application for customers to manage their tasks. This application will allow people to submit tasks, set priorities and manage the teams working on those tasks.	2020-
		Web application for customers to manage their tasks. This application will allow people to submit tasks, set priorities and manage the teams working on those tasks.	2020-
		Web application for customers to manage their tasks. This application will allow people to submit tasks, set priorities and manage the teams working on those tasks.	2020-
		Remaining results hidden	
Secret Mobile B2C	13	Top secret B2C mobile application. This project is need to know basis.	2021-
		Top secret B2C mobile application. This project is need to know basis.	2021-
		Top secret B2C mobile application. This project is need to know basis.	2021-
		Top secret B2C mobile application. This project is need to know basis.	2021-
		Top secret B2C mobile application. This project is need to know basis.	2021-
		Top secret B2C mobile application. This project is need to know basis.	2021-
		Remaining results hidden	
		Web application for managing travel requests.	2020-
		Web application for managing travel requests.	2020-
		Web application for managing travel requests.	2020-

11) Publish the module to the server to save the work.