

Article

[Robert Cemper](#) · Feb 23, 2018 2m read

M:N Relationship

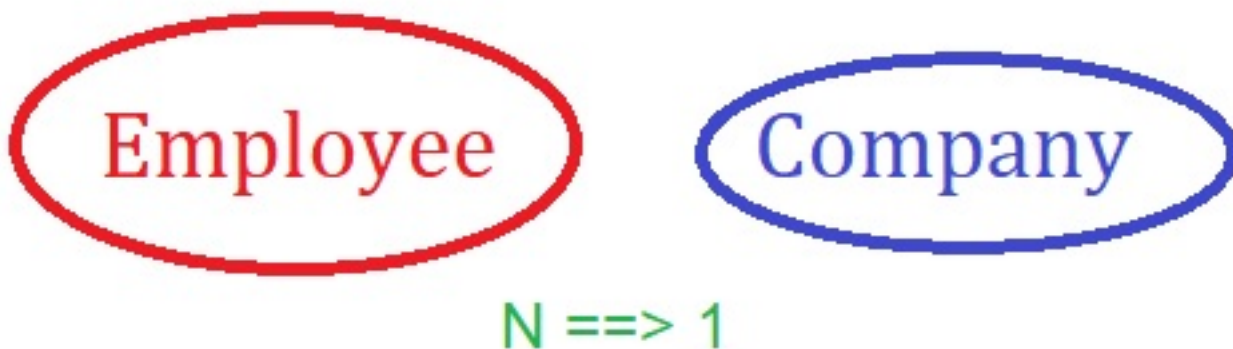
If you have worked with Caché Objects,
You know already all about Relationships (one:many , parent:child) ...
But you will not find a word on many:many relationships in the docs.

But I met the question quite often from new adopters of Caché objects:

"Is it possible to implement many:many relationships ?" YES - of course !

HowTo depends on the related tasks: There is a heavy and a lightweight solution.
Both have in common that they are not just out of the box and you have
to add some code to manage it.

Let's take an example based on SAMPLES namespace:

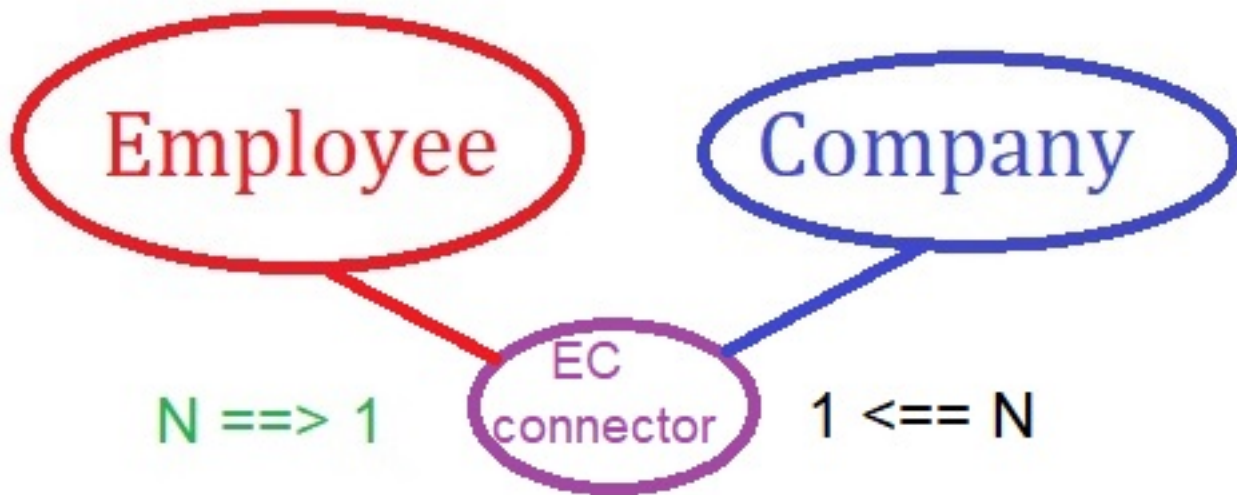


we have the case of an N:1 relationship

But how do you handle Employees with more than 1 Company ?

The heavy solution:

You add an additional persistent class with a one:many relationship to both sides.



The advantage: You can add administrative information as validity, various timestamps, ... and other stuff related to this "link".

But it is an additional persistent class with all pros and cons. So I feel this to be heavy.

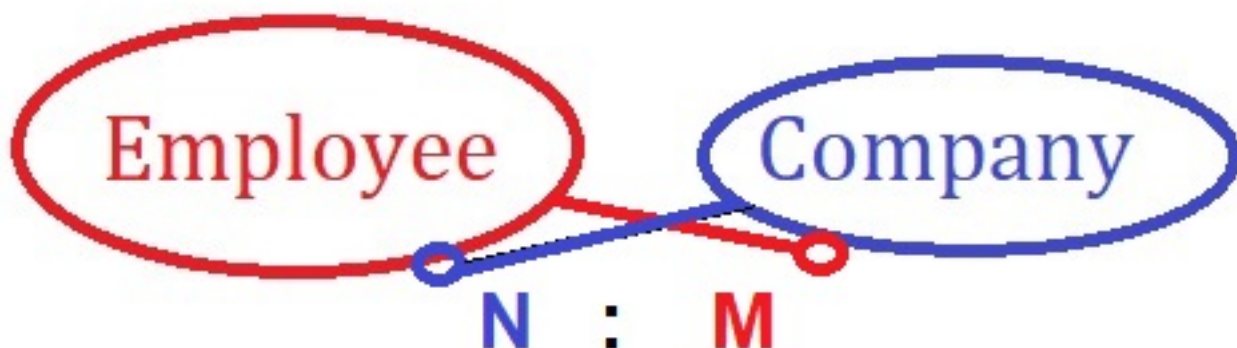
The class may look like this:

```
Class Sample.Jobs Extends %Persistent [Final]
{
Property Status As %Boolean;
Relationship Company As Sample.Company [ Cardinality = one, Inverse = Slot ];
Relationship Employee As Sample.Employee [ Cardinality = one, Inverse = Jobs ];
Index EmployeeIndex On Employee;
Index CompanyIndex On Company;
Index StatusIndex [Type = bitmap];
}
```

The lightweight solution:

If you are just interested if there is a relationship or not and nothing else then you can avoid the third class by using an array on both ends.

It's not more effort than the docking point of the Caché relationship but without "supervisor" in between.



all you need is

```
Class Sample.Employee Extends Person
{
Property Slot As Array of Company;
Class Sample.Company Extends %Persistent
```

```
{  
Property Slot As Array of Employee;
```

Using the related %Id() as key with the array prevents duplicates.

The last variant allows also you to let N or M be zero.
To express one sided termination too.

[#Beginner](#) [#Object Data Model](#) [#Caché](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/mn-relationship>