

Hi.  
  
This video will be about many to many relationships.  
  
So far we've seen entities and relationships and how to identify them from the system description.  
  
We also started to draw diagrams.  
  
And while we did so, you might have been confused about one little fact.  
  
When I changed the diagram, when I resolved one of the many to many relationships.  
  
And I want to tell you exactly how this works in this video.  
  
I'm also going to tell you because we in the last example removed some redundancy from the diagram.  
  
I want to warn you of not to remove too many relationships, because even though they might seem redundant, they might not be.  
  
So this is all about how to handle many to many relationships in the diagram.  
  
So first of all, let's think back how we did this.  
  
In this bus company example, we had this relationship called passes through, which was between towns and stages.  
  
And it was many towns were related to many stages.  
  
So it was many to many.  
  
How did we resolve it?  
  
We introduced a new entity in the middle called town stop.  
  
And then the stages were related to the town stops and the town stops were related to the stages.  
  
You can see that indeed what we did, we were taking one many to many relationship in the diagram, placed a new entity and put two new one to many relationships to that entity from the previous entities in the many to many relationship.  
  
Now the one thing to take care of here is the optionality.  
  
So the optionality of that new entity in these two new relationships is always obligatory.  
  
And the optionality in these two new relationships of the old two entities that were already there stays the same.  
  
So you have to make sure.  
  
But you can see that on the slide for this relationship.  
  
And just try to remember that the optionality for the existing entities will stay the same and the new entity will have obligatory relationships to the two new ones.  
  
Okay, so now that we've seen how to resolve many to many relationship, I want to put one more warning about diagrams.  
  
Some relationships might seem to be redundant even though they aren't.  
  
Let's look at the following example.  
  
So here we have in fact three many to many relationships in a triangle between supplier parts and projects.  
  
So a supplier can supply many parts and the part can be supplied by many suppliers.  
  
A part can be used in many projects and the project can use many parts and the project can be supplied by many suppliers and the supplier can supply many projects.  
  
So we have three many to many relationships.  
  
It looks a bit redundant already, because you could argue if you just remove one still all three of them are connected.  
  
Now first of all, before we discuss why they are not redundant.  
  
Let's see what we would do in the first instance, just by what we've just learned.  
  
We could just turn those three many to many relationships into in one to many relationship by introducing three new entities in the middle.  
  
So that is just what we've seen.  
  
Now we could resolve those and we would end up with many to one relationships.  
  
But this looks even more redundant.  
  
Now we have many more relationships all in the triangle.  
  
Can't we remove some of those?  
  
And this brings me to that notion of a fan trap, which can be dangerous in this context.  
  
So if we were to remove for example, the relationship between suppliers and the direct relationships between supplier and project, what could happen?  
  
Well, there could be suppliers which supply the same part and that part could be used by several project.  
  
And from this information now you cannot read any more of which supplier supplies to which project because.  
  
Because two suppliers supply the same part and that part is used by three different projects.  
  
So how do you figure out now which supplier is going to be supplying to which project?  
  
So we just lost information.  
  
So this relationship wasn't redundant.  
  
So what you could do is you could just stick with the previous solution where you have resolved the many to many relationships with one to many relationships.  
  
And that's fine and there's no redundancy.  
  
However, you could also do one more trick that I want to show that we are not going to use very often.  
  
But in this example it would make things much better, namely by just using ternary relationships.  
  
So we just plug in.  
  
Instead of having three new entities, we have just one entity.  
  
This case we call it contract.  
  
And then one too many relationships from all the other three entities, from part, from supplier and from project.  
  
And then we have just introduced one more entity and three one to many relationships.  
  
And this would be the best solution in this case.  
  
But it's just important to remember there might be cycles on the diagram, but that doesn't necessarily mean that these are kind of redundant relationships.  
  
Okay, so thanks for listening, see you next time.