MIS 372: Lightboard

SPEAKER 1: Hopefully, you've had a chance to read chapter five before we enter into today's lecture. And what we're looking at today is-- how do we deal with what are called many-to-many relationships?

And the example we have to exemplify this is to track which countries won which medals in various summer Olympics for both men and women in field hockey. So it's somewhat specific to keep this problem somewhat small.

But, much like we've done other problems, we need to look at this statement and figure outt-- how do we break it down into pieces that are manageable?

So we need to keep track of countries or teams that we have competing in the Olympics. We've got the countries. We also need to keep track of the Olympics. What year was it? What city was it held in? And from those two combinations of things, we need to keep track of-- how successful were these different teams?

So if we started with just the country, and over here we had the Olympics, we would have what is known as a many-to-many issue. We've got many countries that each participate in the Olympics. There's three medals that are given out for each competition. And each Olympics has many countries that are in it.

And if we just started with those two, we'd have nowhere to store, to keep track of the medal information. So we need to have a table that is in between these guys that we're going to call results. There might be other names that we would give this, but we're going to keep track of the results in between these.

And the results, each result is for a given Olympic. Each Olympic might have many results. And each country will have one result for each Olympics, and each result belongs to each given country.

So let's break this down and take a look at what we're keeping track of. And when it comes to the attributes in this example, I'm not going to put up here the data types and what's required. I'm just going to focus on primary keys, foreign keys, and then the relationships of the entities.

And the reason for that is, with the limited space I have here, I would run out of it in order to communicate everything I would need to in a perfect entity relationship design. That doesn't mean we don't need to do that. We'll mention what we would have on there.

So when it comes to the country, we might have a country ID as our primary key, so just some number to keep track of what country it is. We would have the country name. And this would be a text field that would keep track of the name of the country.

And that's about all that we really need to keep track of for country. So we can go ahead and finish off our box here and call that our entity. Then, the next one that we're going to deal with is the Olympics. We'll come to results here in a minute.

So each Olympics we're going to say is held in a given year. We're only dealing with summer Olympics. So I could call this the primary key all by itself. And then I might want another city and the country where this is held. And go ahead and call that entity done as well.

And from there, we need to keep track of both men and women. So that's going to tell me a piece of information that I need to know in the results. I need to know the results if it's a men's event or a women's event. And I also need to keep track of the medals.

So how is this going to work? Well, first thing I'm actually going to do for the results is I'm going to put the country ID in and I'm going to put the year in, and together right now those two things make up the primary key. But there may be more.

Because this is in the middle of what was two many-to-manys, this is referred to as a bridge table. And whenever you have a bridge table, at the very least your primary key is made up of the primary keys of their parents.

So these guys are also foreign keys, so let's go ahead and note that. Running out of space here. They are foreign keys pointing back to these other guys.

We also need to keep track of the gender. And we need to keep track of what medal the country, the year, and the gender got in the given Olympics.

So if we were to look at what might also be needed as part of the primary key, we would also have to include the gender, because it is possible that both the men's and the women's field hockey teams from the United States got medals in the exact same year. They might have even gotten the same medal.

So to uniquely identify this record, I need to know the results based on gender for a country and for a year and for the gender.

So then I merely just connect these with lines. Because this was many-to-many, by default this is automatically going to look like this on the maximum cardinality side of things. But why is that?

Well, each country has many results. I could get several years of having different Olympics. But if I know the results in this one table, I'm going to be able to identify just the one country from that.

Same thing with the Olympics. If I know the results, I know what year it was from this table. And that's going to identify only one year over here, which means I got my lines backwards.

And if with my Olympics each Olympics has many results, so there's three medals for those given Olympics. It's likely on the minimum side I would do something like this, that a country, I'm going to keep track of one that has never had any results. So it's optional in the results side. The minimum cardinality is zero.

The Olympics. I could initially set up an Olympics and not have any results yet. So I might have a zero there. But if I have results, they'd have to be for a given year.

So at the end of the day, this is what our solution looks like if we're trying to keep track of this particular problem.

I encourage you to take a look at that and see if that makes sense and begin to work some of these problems that you see in chapter five on your own to gain an understanding of how to do this type of modeling.