

Hi.  
  
This video is about an extended example of going from the diagram to the logical design of our tables.  
  
So this is an example where we have a database of a library.  
  
I'm not now going to the details of how we actually arrive at the diagram, but let's look at the diagram and see whether everything makes sense in the library.  
  
What we have is book as an entity.  
  
We have a borrower entity of the people who are borrowing books.  
  
For each books we have copies.  
  
So copy is a separate entity of books, because there are books and there could be several copies of this book, which we have to keep track of.  
  
Then there are publishers of books and there are authors, and between those there are many relationships.  
  
For example, publisher can publish books and one publisher can publish many books.  
  
And usually a book has only one publisher.  
  
So it's a one to many relationship.  
  
And I would say that it's obligatory on both, because every book needs a publisher.  
  
And if you're not having published any book, it's hard to believe that you can be called a publisher.  
  
Similarly, we have a relationship between books and authors.  
  
Slightly different is now that the book can have several authors, and so we have a many to many relationship.  
  
One author can have several books and a book can have several authors.  
  
Again, I would think it is obligatory on both because, well, you need, as a book you need an author, and as an author you need a book.  
  
Now we have books and copies.  
  
This is a really classical example of a one to many relationship, because one book can have many copies, but you definitely can't be the copy of many books.  
  
So it's a one book to many copies relationship.  
  
And this is again obligatory on both.  
  
There can't be any book without copy.  
  
Now we have the relationship between books and borrowers.  
  
Books and borrowers.  
  
Well, one book can be borrowed by several people.  
  
Because now you might protest, how is it possible that several people can borrow a book?  
  
But it can happen because we might have several copies and of course a borrower can borrow several books.  
  
So it's a many to many relationship.  
  
But I would call this optional on both.  
  
And then finally we have one between borrower and copy, which is the loan relationship.  
  
And it's again optional or non obligatory on both sides.  
  
So this is a scenario.  
  
Now we want to turn this into tables.  
  
First of all, how do we start with this?  
  
Well, we take all of the entities and we create the tables for those entities.  
  
So what we create are what I called skeleton tables.  
  
So with skeleton, I mean that we do not yet write down all the attributes we're just writing down the name of those tables and the identifiers.  
  
At this stage we are only interested in the identifiers.  
  
And then once we have created all the tables we need, we just fill out the attributes.  
  
So we have five entities, so we get five skeleton tables.  
  
We have the borrower, the copy, the author, the book and the publisher.  
  
And with the various identifiers in this case, I just created what we called surrogate keys or surrogate identifiers.  
  
For borrower has a borrower number.  
  
We have for the copy, we have an accession number, author has an author number, book has an ISBN number, and then the publisher has some kind of publisher code.  
  
These are the identifiers.  
  
So the first question is now how do we deal with the relationship, loan and reservation?  
  
So remember we have loan is a one to many relationship which is obligatory on both between borrow and copy.  
  
And reservation as well is a many to many relationships between borrower and book.  
  
So we have this.  
  
How do we deal with these two relationships?  
  
First, before we deal with them, let's just backtrack on one step.  
  
Note that they look really redundant in a way, because if you have already the relationship between borrower and copies, but they are not, because clearly the reservation is something different from the loan.  
  
So just think it through.  
  
You might be tempted to think that they are redundant.  
  
There's no redundancy.  
  
Turn them into tables.  
  
Well, to do this we just look at the instructions at our famous diagram where we see that in both of these cases, one to many and many to many optional.  
  
In both we do the same thing.  
  
We introduce new tables to represent the relationship and post the identifiers on both sides of the relationship into the new table.  
  
So this was relationship, reservation and loan.  
  
And so we now get two new tables, one called loan and one called reservation.  
  
And the loan one has the borrower number and the accession number.  
  
And the reservation number has the borrower number and the ISBN number corresponding to the identifiers of both sides of the relationship.  
  
Let's turn to another relationship in the diagram.  
  
It's the relationship between books and copies.  
  
It was one to many obligatory on both sides.  
  
How do we turn this into a table?  
  
Again, we simply go back to the diagram to our how to deal with relationships diagram, where we have a one to many relationship which is obligatory on both how to deal with it.  
  
Post the identifier of the one table to the many table.  
  
So the identifier of the one table is the book.  
  
So it's the ISBN number.  
  
So we just copy it over to the Copy to the copy basically of the book.  
  
So what we get then is a table where we have copy has now accession number and an ISBN number as an additional attribute.  
  
Similarly, we want to deal with the relationships written by in publishers.  
  
Publishers is a one to many relationship that we again resolve by copying the identifier of the one of the publisher to the many site.  
  
So then we get the publication publisher code as an attribute of a book.  
  
So this is the one to many relationship.  
  
And for written by which is many to many.  
  
The only way of resolving a many to many relationship is usually introducing a new table.  
  
We get the written by table where we copy the identifiers of both sides, the author number and the ISBN number.  
  
And this marks exactly which author wrote which book.  
  
So then we ended up with these tables.  
  
We have 1, 2, 3, 4, 5, 6, 7, 8 tables.  
  
Now these are just the identifiers at the moment.  
  
So to finish it all off, the logical design Olaf, we just add the other attributes that we just discarded so far.  
  
For example, for the borrower, we will have a borrower name and a borrower address.  
  
For the book we get the title date and the publisher code we got from the publisher, we get for the copy we get an ISBN number and the price and so on.  
  
So basically we just add attributes wherever needed.  
  
And this is the final step set of tables that we have in our logical design.  
  
You can think this through.  
  
I hope we didn't forget to represent any relationships.  
  
Thanks for watching this video.  
  
See you next time.