

Hi.  
  
This video introduces a way of representing subtypes in our diagram.  
  
One more feature that can be useful and it quite frequently occurs also in practise.  
  
So what are subtypes?  
  
Well, for example, we talked already about vehicles.  
  
Of course, there are many types of vehicles like cars, buses, lorries and so on.  
  
And so we can say that we have a vehicle entity and this entity has sub entities, as I just mentioned, car, bus and lorry.  
  
And then we can somehow represent this in the diagram, hopefully.  
  
So one way of doing this would be just a simple minded way, just putting a big box vehicle and then small boxes inside which tell me that the car is an entity which lives inside the bigger vehicle entity.  
  
It's possible to do it like this, but it's in my opinion the best option.  
  
The question you might ask is why do we care at all?  
  
Because if we have a vehicle entity, we can just store everything.  
  
Well, here again we want to be as precise as possible in terms of what the structure of the data is.  
  
Because if we just store information about all different vehicles in one table, we might end up with a lot of null values, who knows?  
  
So it's better to really depict the proper structure of the data in your diagram to then really create the right shape of table.  
  
So vehicle was just one motivating example.  
  
Another one would be like super insubordinate types such as the following example.  
  
It is required to record the details in this example of researchers and investigators who supervise them.  
  
So we have researchers, we have the investigators who supervise researchers.  
  
They are all university employees and we want to depict this in a nice little diagram.  
  
So again, we could use this box trick where we just say there's a big box university employee and then there are two small boxes, investigator and researcher.  
  
And we can then draw within this big box a relationship between the investigator and the researcher.  
  
That tells me that this is a supervising relationship.  
  
But you already see what the problem, the slight problem here that we get in this notation with boxes is that we get this nesting of boxes and then suddenly we have boxes and relationships within those boxes and this can get easily a bit nasty.  
  
On the other hand, if you prefer to use this throughout your career, feel free to do so.  
  
I am going to introduce you to another notation for this class.  
  
But there are many different ways of doing this.  
  
So the thing I'm going to introduce you is enhanced entity relationship diagrams.  
  
And we are not going to discuss all the ways of enhancing the entity relationship model.  
  
I'm really just focusing on the sub entity relationship that I mentioned before.  
  
Such as between vehicle and car.  
  
So, for example, we have here one standard example again of employees.  
  
This is the big that we just saw also with this investigator and researcher.  
  
It's like the overarching entity.  
  
Now, one way would be to have a big box of employee and then say, what are the various types of employees in the enhanced entity relationship diagram is we just say we have an employee entity, we have this double line going down, which the double line here really indicates something like equality, ending with a circle.  
  
And in this circle we write D.  
  
The D here really means something like disjoint or disjunction, because we want to say it's exactly one of the following options.  
  
And then from the D we make lines to the entities engineer, technician and secretary.  
  
And then on these lines from the engineer to the D, we also indicate the subset relation.  
  
We say engineer has a line to this circle with a D, and then this line has an inclusion written on it.  
  
Similarly, the technician has the inclusion and the secretary has inclusion.  
  
And what this really means here in the enhanced entity relationship diagrams is that an employee is either an engineer, a technician or a secretary.  
  
Exactly one of those.  
  
So that inclusion is called specialisation.  
  
The D indeed is disjunction, and that equality means that we have total specialisation.  
  
Another option here would be that an employee could be also categorised in different ways.  
  
There's an employee that are either a salaried employee or an hourly employee.  
  
And then we have again a disjunction here between salaried and hourly.  
  
I don't have the double equality mark here, because it could be that the employee is currently neither getting a salary nor an hourly pay.  
  
Maybe it's just kind of away on sabbatical or something.  
  
For the sake of the argument, if you do not write this double equality sign here, it just means that employee can be one of the two, but there could be an employee that doesn't fit either of the two.  
  
And then we could also just have one sub entity which is kind of a certain subclass, nothing to do with disjunction, where we just have an employee entity and then manager entity.  
  
And we draw a line directly from the manager to the employee.  
  
And on this line we have again this kind of subset symbol, which just means that the manager entity is a sub entity of the employee entity.  
  
And then we can.  
  
All the examples that I discussed before, we can just put all in one diagram and then we see the full structure of we have the employee on the top and then we have the manager somewhere, and then we have salaried hourly.  
  
And we have that other disjunction with engineer, technician and secretary.  
  
And so this gives you this whole hierarchy of various entities.  
  
You could ask, why do we really care again?  
  
And so I gave you an answer previously by saying that we want to be really kind of faithful to the structure of the data.  
  
Perhaps not only that, we also want.  
  
It could be that some relationships, for example, with other entities only hold for the sub entities.  
  
So I don't know, I'm sure that the engineer might have a relationship with the kind of lab they are assigned to, which the secretary wouldn't have.  
  
So then, if we then want to represent this relationship on the employee level, we will get again, possibly a lot of null values.  
  
So if there are relationships which only hold for sub entities, it's much nicer to have the sub entities represented rather than just the whole thing in one big blob.  
  
We can also then in the whole thing have sub entities, which are in fact sub entities shared of several of the options.  
  
So we had the manager option, we had the salaried employee option, and we have the engineer option.  
  
So you could have an entity called Engineering Manager, which somehow is a sub entity of all the three.  
  
And then you get what is called the specialisation lattice.  
  
That is indeed a lattice means that we have this structure of things which could be kind of the join of several things, but then those things could have a common smaller element.  
  
So that is what is called in mathematics.  
  
There's this notion of a lattice and it looks a bit like this specialisation lattice.  
  
Another, maybe smaller example to practise this a bit is the following.  
  
We have this idea of a club where members may pay their fee monthly or annually.  
  
Some of those paying annually are office holders.  
  
Monthly payers have a registered bank account.  
  
So you see already we have members somehow as entities, they are distinguished between monthly or annually paying.  
  
And if they are monthly or annually paying, then there are different roles or relations that they can have there.  
  
So to put this into a diagram, we have somewhere that member entity which is now a disjunction between the members which pay annually or monthly, and the monthly entity has now a relationship with bank account.  
  
Because they have a registered bank account, it's a one to one relationship.  
  
And the annual payers, they have a sub entity called officeholders, because some of the ones which pay annually are officeholders.  
  
And this is the diagram that depicts exactly the situation which was described in this video.  
  
We discussed simple elements of the enhanced entity relationship diagrams, namely the ones related to subtyping in the next video, I'm going to explain how to then resolve those subtyping relationships in tables.  
  
Thanks for listening to this video.  
  
See you next time.