

Hi.  
  
We've seen database management systems and I've mentioned that the main distinction between those is the underlying data model.  
  
In this video we are going to clarify what a data model is.  
  
I'm going to give you a brief history of data models and list a few data models that are actually in use currently.  
  
First of all, what is a data model?  
  
And so the answer to this question is it's just a representation of data.  
  
It's a notation for describing the data.  
  
Data.  
  
A description consists of three things.  
  
We need to have a way of saying how do we represent data?  
  
Then how do we manipulate data?  
  
We need to know what are the operations on data.  
  
And finally we need to know how do we maintain constraints on the data.  
  
So what are possible constraints on data?  
  
And the most common models that exist at the moment, I would say are the relational model, the semi structured data model and the object oriented model.  
  
All of the data models have certain commonalities.  
  
So the way we structure data is usually in three the entities, which are like the objects about which we want to store data, the attributes which are the things that describe those entities, and then finally the relationships.  
  
This is like the connections between the various entities.  
  
And we will come back to this in more detail.  
  
But just to already get you into the idea of thinking of the world in entities, attributes and the relationship between entities.  
  
So let's make a step back and let's go to the history of data models.  
  
The first data model used probably in the 60s was the so called hierarchical database model.  
  
So what happened there is that the database has a tree structure and all the entities are in fact arranged in a tree.  
  
So you have an example on the slide with the customer and their orders, and so the customer is the root of the tree and then as a child he has one order and then the order has an order, has various order items as a child, and so on.  
  
So if you want to access the data in such a tree, you have to exactly know where to go in the tree to find where the item is that you're looking for.  
  
You have to know what is the structure of the tree, which is not obvious to somebody who didn't really programme this tree.  
  
That is one thing which is not ideal about it.  
  
The other thing about the hierarchical model is that it has to be really a tree.  
  
And you can easily come up with an example where you would want to have two things, like in this case, an invoice item and an order item, referring to the same part, where you would want to have one child of two different parents.  
  
Which would violate the tree structure.  
  
So this rigidness about being a tree is not good.  
  
Therefore people realise this, this is too rigid.  
  
There's a problem here.  
  
They went to something which is called the network model, where we just said, okay, it's basically like the hierarchical model.  
  
We still arrange the data somehow in some kind of graph like thing.  
  
Now, not tree like, but graph like.  
  
And in order to query the data, you have to know where to look in the graph and how the things are arranged on the graph.  
  
So it's better than the hierarchical model, because now we are not anymore forced to have a tree structure.  
  
But it has still some issues in terms of logical independence, because we have to exactly know how the database was arranged, which items can be reached via which path in the graph.  
  
So it's better than the hierarchical model, but still not good in terms of the independence conditions that we saw in previous video.  
  
And then we move forward to the 70s where there was some seminal work in 1970, to be precise, by Kot, and he proposed what's called the relational model.  
  
And this is the thing that was originally considered to be too theoretical, too abstract.  
  
Nobody is going to use it, but nowadays, basically everybody is using it.  
  
And so what is the relational model now?  
  
We forget about all these graphs and trees.  
  
We just think about the data as every entity is just by itself a table.  
  
Everything in the relational model is a relation.  
  
And so we have a table here on the slide.  
  
What is the table?  
  
Well, you know, all tables think of a spreadsheet.  
  
The columns of the tables correspond now to the attributes and the rows of the tables correspond exactly to the various entities that are described using these attributes.  
  
And we are going to discuss the relational model in more detail.  
  
Now you might ask, where did these trees and the graph structure go?  
  
So there's something smart if you want to now connect tables.  
  
This is done via what is called foreign key constraints.  
  
We will see this in detail.  
  
It's nicely abstract and very flexible.  
  
And we don't have to think about, do we allow trees or graphs.  
  
It's just exactly.  
  
You can present the relationships that you want.  
  
So in the relational model, everything is a table.  
  
All the things are modelled using tables.  
  
Each table has what is called a schema.  
  
So the schema is just the name of the table and the name of all the attributes in the table.  
  
And a database.  
  
So what do we call a database now?  
  
A database is just collection of tables or relations, and the schema of the database is just the schema of all the relations in the database.  
  
Together we will Discuss the relational database model in more detail once we discussed what a relation is next week.  
  
For now, I want to say that relational model was extremely successful, but there are other things that were very successful.  
  
For example, since the 90s there was an emergence in what's called semi structured data.  
  
And semi structured data, funny enough, goes a bit back to the origins of data models, because things there are again quite tree structured and there's no strict schema, there's some kind of schema language, but it's very flexible.  
  
So the motivation of the data now is indeed again modelled as trees or graphs, which has some complications.  
  
But on the other hand, you allow any kind of tree and any kind of graph, whereas in a relational data model you only say that this is how the relation looks like, and then every row in the relation has to satisfy the schema.  
  
In the semi structured world you do not have this.  
  
And so there the data is kind of very flexible in the sense that we do not specify what the entries, what the shape of each item is, so to speak.  
  
Things are arranged in trees.  
  
So a database could be thought of in this case just as a collection of trees, a forest if you wish, or a collection of graphs.  
  
And the motivation there is we want to be as flexible as possible because we want to share data.  
  
So it's not for nothing that it was emerging since the 90s, because as you might know, since the 90s also we had the Internet emerging.  
  
If you have the Internet, you have sources of data worldwide and you want to say, okay, I'm now sending you whatever data I have and I want to have this done in a very easy fashion.  
  
If I want to share relational data, I have to specify exactly what the structure is of the data that I send you.  
  
And merging of two different relational databases can be a nightmare.  
  
So for heterogeneous data coming from different sources, semi structured data was perfect.  
  
That's why this kind of data became more and more popular.  
  
And we have also other data models nowadays, which would be the object oriented data model that I don't want to discuss in detail, but you can imagine that people are programming in object oriented languages all the time.  
  
So what you want is some kind of data model that fits very well into these languages.  
  
And this is how the popularity of this kind of databases can be explained.  
  
Linked data is a way of, it's a kind of movement of trying to turn the whole Internet into a database and it's growing.  
  
So basically what you're describing is the data that is available of the Internet in a way such that it can be queried.  
  
The language of doing this is called rdf.  
  
It's a fascinating topic.  
  
We are not going to discuss it in detail, but if you want, sure we can give you the necessary links to relevant papers if you are interested.  
  
In summary, I have here one slide which shows you the various data models that we discussed.  
  
So we have the relational one, the semi structured one, we have linked data and RDF and object oriented and you can here read for yourself what are the structures involved, how do we store there the things and which database management systems are available to deal with this data.  
  
I hope you enjoyed this video.  
  
Thanks for listening.  
  
See you next time.