

Hi.  
  
This video is about first steps in SQL.  
  
The main purpose of this video is to introduce some elements of SQL so that you can connect this directly with the relational algebra you've seen.  
  
We will introduce SQL in a much more way in depth in one of the coming weeks.  
  
So, first step to SQL is to analyse what kind of language it is.  
  
So every database language usually has two components because we have two tasks whenever we deal with data.  
  
The one is we have to somehow first of all put the data somewhere, we have to store it, we have to say what the data is.  
  
So this is what's called the data definition language.  
  
And we have the data manipulation language that then allows us to change, update the data and to query the data.  
  
So why do we use SQL?  
  
Well, the good thing about SQL is that it's a very high level language.  
  
So it's not something where you could imagine a language that would ask you to say, show me the data that is stored in a certain address on the hard drive, which is somewhere located in the third floor of this building.  
  
But that wouldn't be a very good language because what if somebody moves the hard drive from the third floor to the fourth floor?  
  
So we want to be independent of all kinds of things as we've discussed already in the introduction to this class.  
  
Therefore we keep it as high level as possible.  
  
So what's the minimum information we need to know is we need to know what is the relations we are talking about and which part of the relations we need.  
  
So what we want to say is what we want, not how to get there.  
  
So we want to say show me for example, those movies.  
  
And then SQL has to figure out where to find those movies.  
  
And so that is kind of the paradigm here is that this SQL is a declarative language.  
  
We just say what we want and not necessarily how we get there.  
  
It's very different from languages you might know like Java, where you have to usually say exactly, store the value there, do that, do that, do that.  
  
Here we just say we want this, do it for us and then give me the result.  
  
So the database management system figures then out for you how to get to the data.  
  
So, and this also how to get there in a very fast way.  
  
There's a whole area of query optimization that we are probably not going to discuss.  
  
So what is now an SQL query for the sake of this video?  
  
It's something that has three components.  
  
A select component, a from component, and the where component.  
  
In the select component you specify what are the attributes that you want to see in the from component you specify which tables we are talking about.  
  
And in the where component you can say something about conditions.  
  
Let's have a look at an example.  
  
We want to know what the movies are that are directed by the director with an ID D20, whoever that is, we want to know what are the movies.  
  
So what is the query to do this?  
  
Well, we have to talk about movies, and our movie table is called film.  
  
So we select from film and then what movies?  
  
We just want to know the title in this case.  
  
So we just say select the title from film from which films where the director ID has to be D20.  
  
So we say where the director ID is equal to D20.  
  
We'll discuss in more depth what the conditions are that you can put in the where clause.  
  
But basically it's some kind of equality inequality with Boolean operators.  
  
And so you see already Also around the D20 there are single quotes, because if we talk about single kind of strings, then we have to put them in single quotes in SQL.  
  
More importantly, you see the three parts exactly as we discussed, the attribute, the table and the condition.  
  
And then in this case you get a result.  
  
The result in this case are just the titles, whatever.  
  
It starts with et, goes on with Jurassic park, and then comes to Raiders of the Lost Ark, and then goes on until infinity.  
  
And this is apparently the movies of the director.  
  
With ID20 you can figure out who that might be.  
  
In any case, the answer of that query is in fact again a relation in this case with that single attribute that we selected and with the titles of those rows where the director ID was equal to D20.  
  
So one thing I want to mention, because it's happening all the time, is that in the previous example we selected the title of the movie because we wanted to have only the title.  
  
Sometimes you just want to have all the information, for example about the movie, then you can just use star.  
  
So I'm just telling you about the star operator now, because it's coming up all the time, so whenever you don't care which attribute you want, you can just say star.  
  
So select star from film with directory ID is equal to D20 gives you again the information about the same movies, but now not just the title, but everything.  
  
Title, release year and so on.  
  
Now for a slightly more complicated example, an example that is called a multi relation query.  
  
So this is basically now not selecting just from one table, but from two tables.  
  
And the idea is now that we want to have the same information as previously, but clearly nobody wants to query for the director, which is called D20.  
  
But you want to query for the movies of a director with a real name.  
  
But because the information is stored in two different tables, we have to get the information of two different tables.  
  
So the rule of thumb here is really, whenever you want to say something about movies and you want to mention the director name, you have to somehow put into the from clause those tables that give you first of all the director name and it gives you the movie name.  
  
And then you have to combine those tables somehow.  
  
But in the first instance, you have to select all those tables that hold the information.  
  
If you don't do that, then there's no way that you can get to the information.  
  
So in this case, we want to have the movies by the director with the name Steven Spielberg.  
  
Now what we do is we select all the information.  
  
So we just say select star.  
  
And now because we talk about basically Steven Spielberg, we have to select also from the director table.  
  
So we have to select from film comma director, because you might get the information from two tables.  
  
And now what film director does in the relation world, so to speak, is it takes a cross product of these two.  
  
But we of course only want to combine movies with a director which has an ID for the director ID of that movie.  
  
Therefore, we need in the where clause to specify that we want to only have those rows where the film director ID is equal to the director director id.  
  
Because we want to only have the right matching combinations.  
  
So that is the one condition which we do to make sure that we only combine the right rows from films and directors.  
  
And then we also want to specify our condition that the name of the director should be Steven Spielberg.  
  
So this is second part and director name Steven Spielberg.  
  
And when we run this query, it will give us all the information about the movies where the director is Steven Spielberg.  
  
And again, it's important to remember selecting from several tables here.  
  
In this case, film director is just like forming the cross product.  
  
I hope you enjoyed this video.  
  
Thanks for listening.  
  
See you next time.