CS 240: Executing Queries with JDBC Transcript

[00:00:00] Once you have AJ D BC connection, you can use that connection to execute a query queries are the way we pull data out of the database or in other words, read the data from the database.

[00:00:10] So here um we are going to read all of the books out of our book table.

Start visual description. The professor demonstrates how to read all of the books from the book table in a database using JDBC. The professor shows the creation of an ArrayList to store the books and explains the importance of using reference types as interfaces for flexibility. End visual description.

[00:00:16] Usually when we do that, uh we're doing that from some Java program.

[00:00:20] And so what we really want to do is we want to read something out of the database and represent that as objects in our program.

[00:00:26] So if we're going to read all of the books out of the books table, we need a place to put them.

[00:00:31] So we'll start by creating an array list of books.

[00:00:35] So here we have our array list of books and just as kind of a side note, it's a good practice to have your reference types be um interfaces whenever you can.

[00:00:48] So we're using the array list class, but the reference type is a type list um that just makes things more flexible.

[00:00:55] So if I ever wanted to change my code and use a different kind of list, I could just change it here and I wouldn't have to change anything else.

[00:01:02] OK. So here I have my, my list of books and the first thing you want to do is create your SQL statement.

[00:01:09] That's why we had you learn SQL first before you learn JDBC.

Start visual description. The professor demonstrates the process of creating an SQL statement to select columns from the book table. The professor shows how to prepare the statement using the connection, execute the query, and retrieve the results in a ResultSet object. The professor explains how to iterate through the ResultSet and extract data to create book objects. End visual description.

[00:01:13] So you need to create an SQL statement. So here you can see this is a select statement that can select these columns from the book table.

[00:01:22] So it'll, it'll select all the rows once I have that I need to prepare that statement, so you can use your connection to prepare the statement.

[00:01:31] So we call connection do prepare statement past the SQL string that gives me back a prepared statement.

[00:01:38] And now with that prepared statement, I can call execute query and that will actually invoke the SQL on the database and return the result.

[00:01:47] So that gives me the result in a result set object.

[00:01:51] So a result set object in in database terms that's called a cursor.

[00:01:55] So a result set object is basically a list of the, of the results from the query.

[00:02:03] Um And it has a pointer in it and that pointer points above the first row.

[00:02:08] So when I do this, I have my results set. So, within a while loop, I can iterate through all of the rows that I just retrieved.

[00:02:16] So within that y loop, since the pointer points above the first row I call results set dot Next, now that will have me pointing at the first row and then I can pull all of the values for the attributes out of that result set row.

[00:02:30] So there are actually two ways I can do that so I can do it by index.

[00:02:35] So here I’m saying result set now that I’m on the F on a row, I say use my reference and call the get in method.

[00:02:43] You have to know what the data types are of the of the values that you just received retrieved from the database.

[00:02:49] So um the first column is the ID column that’s an in.

[00:02:54] So I call result set dot get 11 thing you have to be careful of that.

[00:02:59] That’s a little bit confusing or tricky.

[00:03:01] This is about the only place in Java where something is one based instead of zero based with everything else.

[00:03:07] Um Things start the first element of an array, for example, is element zero but with queries, the first column is column one.

[00:03:18] And the reason for that seems strange.

[00:03:20] But the reason for that is because that’s the way they’re following the database, the SQL convention, not the Java convention for that.

[00:03:27] OK. And these column numbers refer to the numbers from the select statement.

[00:03:32] So it doesn’t have anything to do with the order in which the column, the columns appeared in the create table statement.

[00:03:39] It’s just what order were they were these in the select statement? That was executed. So, the first thing in the select statement is ID, so column one in the row is the ID column. And since that’s an in I call, get in that returns the in um value from that row.

[00:03:58] And you can see that I do that for all the other columns as well.

[00:04:00] So I, I get the title of the author, the genre and the category ID.

[00:04:04] So now I have all the data, and I can use that data to create a book object.

[00:04:09] So I construct a book object with that data and then I add that to my books list.

[00:04:14] And also, I’m doing that in a wild loop.

[00:04:16] So when this wild loop is through executing, I’ll have a list of all the books from that row.

[00:04:21] Um You typically need to deal with SQL exceptions whenever you’re doing anything with JD BC, just about anything can throw an SQL exception.

[00:04:30] So here, I’m just showing that we’re catching the SQL exception, and we would do something with that.

[00:04:36] I forgot to mention that there is actually another way to specify the attributes when you’re processing a result set.

[00:04:44] So I showed you the way to do it by index, but you can also do it by, by string um name.

[00:04:52] So I could, instead of saying get in one, I could say get in and then specify ID as a string.

[00:05:00] And that would also work.

[00:05:02] So either way, either way is, is perfectly fine for processing a result set.