CS 240: Java Database Access with JDBC Transcript

[00:00:00] **JEROD WILKERSON:** Now that you know how SQL works and you understand the basic statements of SQL, It's time to learn how to execute SQL from a Java program.

[00:00:08] We do that with something called JDBC; that stands for Java Database Connectivity. That is a set of interfaces and classes that allow you to access relational databases and execute SQL against those databases.

[00:00:22] There are a set of steps that we follow in order to do that. We start by loading a database driver. A database driver is a set of classes usually written by the database vendor that allows you to access that code from Java.

[00:00:38] They will write a database driver. JDBC is mostly a set of interfaces that are provided, first by Sun and now by Oracle— now that Oracle owns Java. It's a set of interfaces that JDBC drivers implement.

[00:00:54] When you get a database driver, you're getting specific code that some database vendor—or it could be somebody else, but usually it's a database vendor—you're getting specific code that they wrote, but they wrote it to conform to the interfaces provided by JDBC.

[00:01:09] That tells you that you know what methods you can expect a call on those classes. You start by loading their driver into memory. That gives you the set of code that you need for a specific database.

[00:01:21] In our case, that'll be SQLite. Then you open a database connection to the database that you want to access. Then if you're doing transactions, you'll start a transaction.

[00:01:31] If you're not, if you're just doing a single statement and you don't need a transaction, you just skip that step. Then you execute your queries or updates using mostly the syntax you've already learned about SQL.

[00:01:44] Then you'll either commit or rollback. We're going to be doing this within try-catch blocks. If we are successful, if we get to the end of our try block, we'll commit.

[00:01:53] If we get an exception, we will rollback. The exceptions that you get when you're working with JDBC are called SQL exceptions. That's usually what we're watching for in a catch block.

[00:02:05] Once we do that, we'll close the database connection. We need to be careful to close the resources that we're using when we're doing JDBC. When we use connections, we close them when we're done.

[00:02:15] You'll learn about statements. When you're through with a statement, you close it. You basically close everything that you use when you're done because those are limited resources that you can run out of.

[00:02:24] In SQLite, not closing a connection will often cause database locks. You need to be careful with them.

[00:02:31] Now if we have any rows that we've inserted and we've given them autoincrement primary keys, think about what that might look like in a Java program.

[00:02:40] In a Java program, as you'll see, you'll generally have an object, and you want that to be represented as a row in the database. If that object has a bunch of instance variables, it's going to have an instance variable for the primary key.

[00:02:56] What if we're autoincrementing it? We don't know its value until after the insert in the database. We need to have a way that after we do that insert, we can pull the primary key back out of the database so we can put it in our Java object.

[00:03:08] That's often the last step. If we do that, we do it before we commit or close the connection. It would really be inserted right here in the steps.

Start slide description.

The following text appears on the screen.

Database Access from Java

- Load database driver
- Open a database connection
- Start a transaction
- Execute queries and/or updates
- Commit or Rollback the transaction
- Close the database connection
- Retrieving auto-increment ids

Omit transaction steps if you only need to execute a single statement.

End slide description.