CPSC 304: Introduction to Relational Databases

Tutorial #6: Java/JDBC Setup with Oracle

An Introduction to Running a Java/JDBC Program with Embedded SQL in Oracle

Due Date: Sunday, November 1, 2020 before 23:59 (midnight)

- Updated: Oct. 27 @ 21:05 – more information about screenshots
- Updated: Oct. 26 @ 18:30 – port forwarding information for Mac/Linux users getting a "Network Adapter" error

This is a hands-on exercise, and you will be awarded marks for completing the tutorial. Your submission will include a few screenshots of your work to show that you have successfully completed the tutorial. Canvas lets you upload multiple files, but you're welcome to put your results together into one file and submit that.

Screenshot Update: We've provided some examples of screenshots in this document. For other cases, such as where you have to show proof of the configuration and the code running locally, you can do one of the following for your submission: (a) show your IntelliJ configuration with your username and other credential information with the Oracle login, as well showing us a screenshot of running your code; or (b) show the terminal window where you SSH to, to create a tunnel into the remote server, and again show us a screenshot of some of your code/work that's running locally. For example, after logging in successfully in your local UI, your terminal or your console should display that Oracle connected successfully. Examples are below.

- **Late submission policy:** 25% penalty per day late

Introduction:

You are already familiar with using Oracle's SQL*Plus product on the undergraduate server, and you are able to create and load data into a sample database.

In this tutorial, we will be using Oracle again, but this time, we'll deal with *embedded* SQL programming. Some of you may be using this platform for your project.

Most of this tutorial involves observing how Java/JDBC, embedded SQL, and Oracle work together. You don't have to write your own code (save that for your project).

As you proceed, please note that, besides the instructions at the links below, you'll find troubleshooting information and other tips at https://www.students.cs.ubc.ca/~cs-304/resources/faq.html#java.

If you would like to use PHP/Oracle or PHP/MySQL, please wait for Tutorial #7 for us to cover it. However, you should still complete this tutorial to understand how to set up a project with

Java/JDBC and Oracle. It's also good to have this knowledge as a backup in case your alternative tech stack runs into complications. Besides, you are getting a mark for completing this exercise/tutorial, and you are adding more skills to your CS portfolio.

Feel free to get help from Piazza or the TAs' office hours when setting up this tutorial, whether with Java or later with PHP.

Oracle Code and Tables:

This tutorial involves a "branch" table—take your pick, either:

a) A **bank branch**, packaged together with the Java instructions at the setup page for these tutorial instructions in the section below entitled **Details**. The `.jar` file (i.e., the unzipped first `.zip` file) is important to run the application, and the second `.zip` file contains some code for you to explore). These `.zip` files are located here for you to download:

- https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/CPSC304-JavaDemoJar.zip
- https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/CPSC304-JavaDemo.zip

b) The **Motor Vehicle Branch** (MVB) tables can be found here:

- https://www.students.cs.ubc.ca/~cs-304/resources.html

If you plan to use the MVB branch application, then go ahead and create and load the MVB tables, using the SQL DDL scripts found on the above web page. To run the SQL using Oracle's SQL*Plus, see Tutorial #5. There are 3 scripts for MVB. Note that the above web page also has other links that may be useful to you.

Sample code and `.zip` files for the MVB branch application are located here:

- https://www.students.cs.ubc.ca/~cs-304/2018W1/tutorials/JDBC/branch.java
- https://www.students.cs.ubc.ca/~cs-304/2018W1/tutorials/JDBC/hb15.zip
- https://www.students.cs.ubc.ca/~cs-304/2018W1/tutorials/JDBC/classes12.zip
    - Download the code including the `hb15.zip` and `classes12.zip` (Oracle driver) files. Do not unzip them. `hb15.zip` is a package that contains routines for formatted output. The reason we need this package is because the standard Java 2 API does not provide, for a text-based interface, any specific library routines for

formatting output into straight columns. The only way that this can be achieved easily is through the use of a GUI. You can find documentation, a tutorial, and examples on how to use this package at the author's site. This program uses only `Format.printf()` and `Parameters.add()`. The comments in the sample program should be sufficient for you to understand how those methods are used.

- Play around with the program, and study the code. The program uses Swing (a GUI package) to draw a login window for entering your Oracle username and password. It is a bad idea to hard code your password in a program because someone could read the source to obtain your password. It is also a bad idea to read the password from the command line. This is because you cannot turn off echoing in Java. You can find reference material and tutorial-like pages about Swing on the Web, using Google.

Summary: You can use either of the above 2 "branch" applications for this tutorial. MVB is the example traditionally used for the course, but recently, a bank branch with a Java .jar file have been introduced (that might be simpler for some students).

Details (useful for both "branch" applications, but more so for the bank branch):

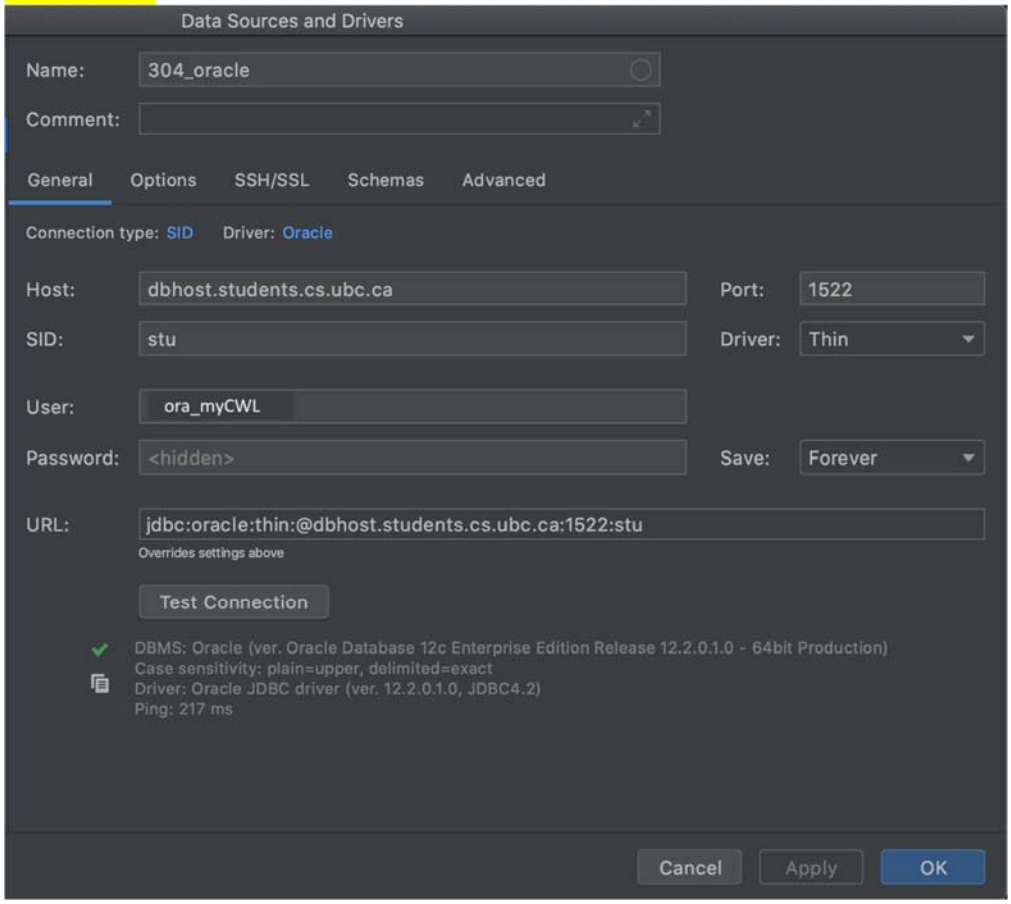
1. To get started, use the set-up instructions at https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/jdbc-java-setup.html. You should be able to run the sample project and log in using the GUI that appears. You can use the sample project to look at, and play with, the data you just populated into the database.
    - Note: Mac/Linux users who are encountering a "Network Adapter" error message, take note of the port forwarding instructions found at this location in the above document:
        - https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/jdbc-java-setup.html#port-forwarding

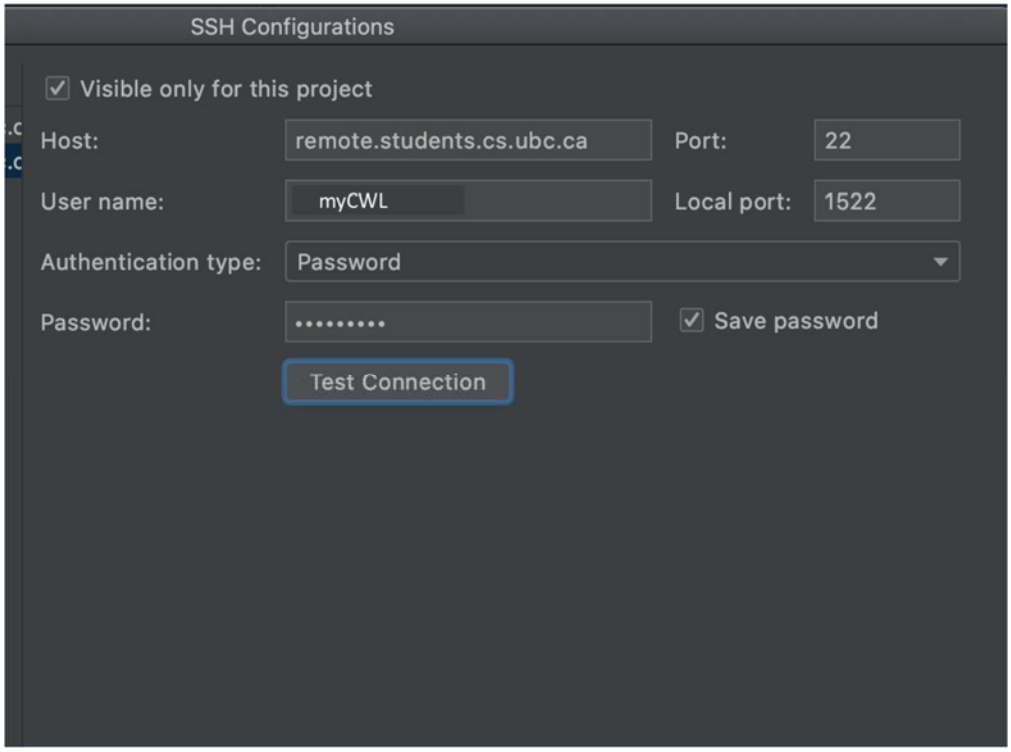
2. Read the **Looking Through the Code** web page at https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/jdbc-java-looking-through-code.html to develop an understanding of what is happening inside the code. This can help you get a high-level overview of what JDBC is and how it is used to connect a Java application to an underlying database.

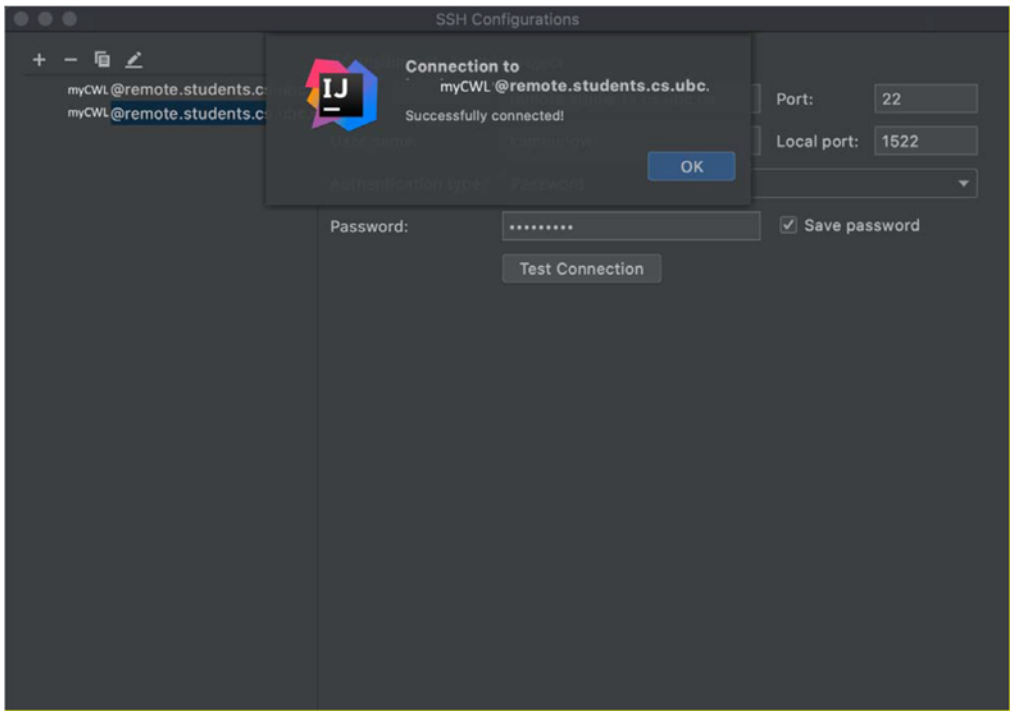
3. Some extra information including material about **result sets** is found in https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/jdbc-java-result-set-extras.html, and there are miscellaneous topics in https://www.students.cs.ubc.ca/~cs-304/resources/jdbc-oracle-resources/jdbc-java-misc.html.

Examples of Deliverables:

For IntelliJ:





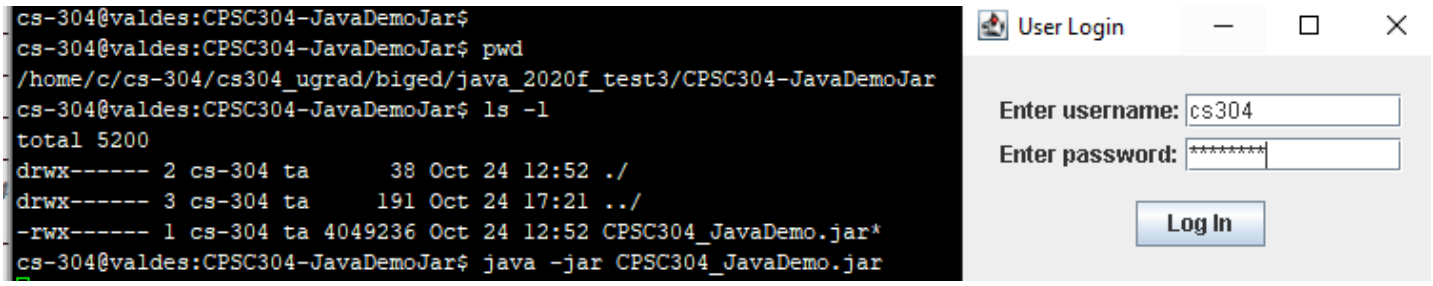


Examples of deliverables if you are using the bank branch application from the command-line:

1. Hand in screenshots of at the following items, taking care to identify on your screenshots that it is your account, and not your friend's. Most of the following is based on "Method 2" found in the **Details #1** section's link a few paragraphs above this one:

    - **Screenshot 1**: (It is OK to have multiple screenshots for this, or to produce the equivalent results if you're using some environment other than the undergrad machines and command line input. The screenshot that follows is just an example. Again, you might be using the bank branch (Bank.java) application; that's fine.)
        i. Include your userid in your screenshot (so that we know that it's you, and not your friend), and show your current directory (Unix `pwd` command, Windows Explorer, or equivalent).
        ii. Show the unzipped `.jar` file in your directory/folder.
        iii. Begin running the program up to the point where you enter your userid and password, but don't click Log In before you capture a screenshot. For example, my userid is "cs304" but yours will be your own Oracle userid (ora_*userid*).
            - By the way, some students with older XManager software have reported that the "Enter password" field will reject their valid password, but if they type it into Word or Notepad first, and then copy-and-paste it into the "Enter password" field, it works—strange. XManager Version 6 doesn't seem to have this problem. This is one

of the troubleshooting tips you'll find at https://www.students.cs.ubc.ca/~cs-304/resources/faq.html#sql-plus.



- **Screenshot 2**:
  i. Begin running the program.
  ii. Show screenshot(s) for the results to show us that it's your account. Yours should be similar in content to the following sample screenshot (on the next page, but feel free to use your own data):

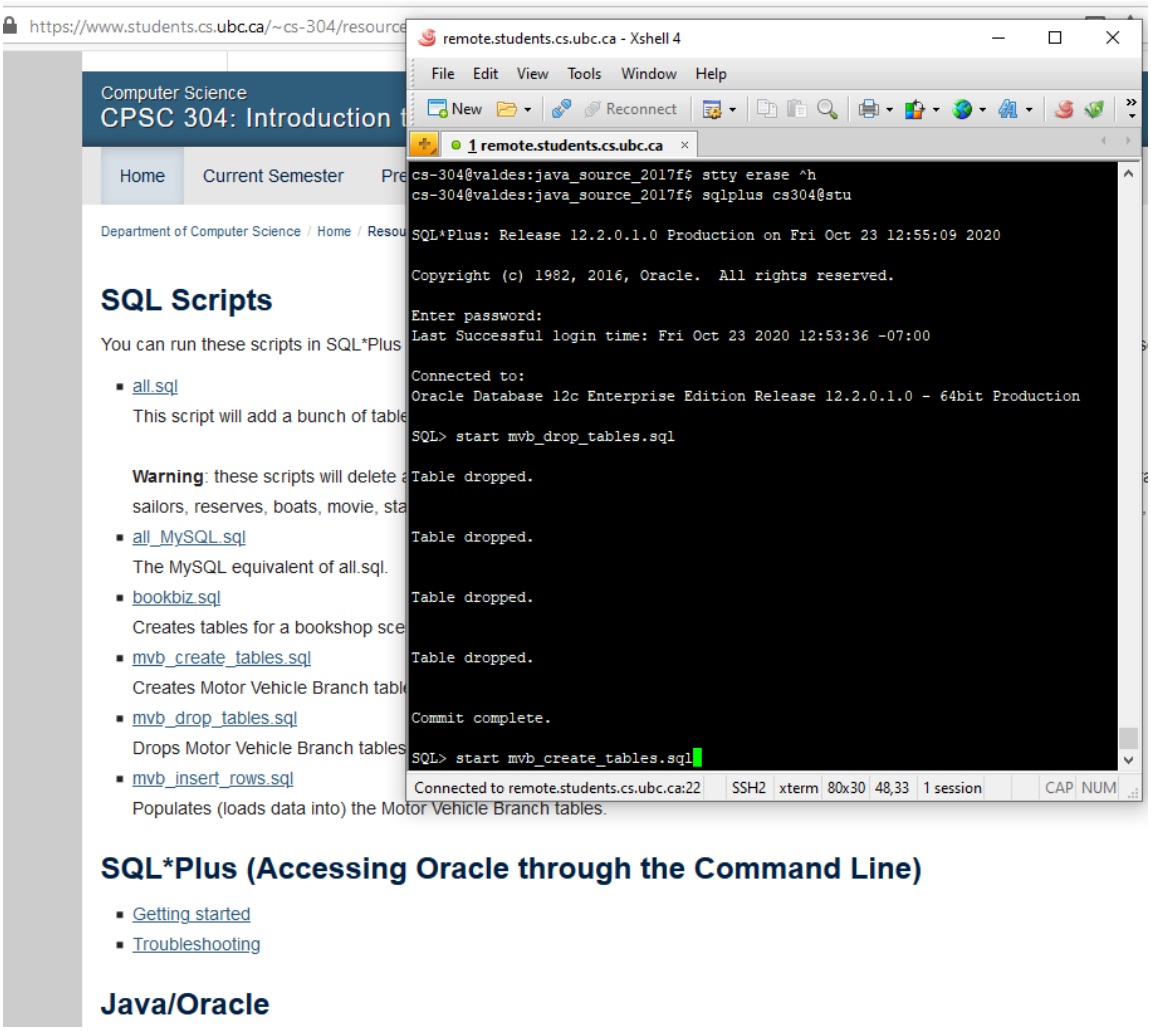
```
cs-304@valdes:java_2020f_test3$ cd CPSC304-JavaDemoJar/
cs-304@valdes:CPSC304-JavaDemoJar$ java -jar CPSC304_JavaDemo.jar

Connected to Oracle!
If you have a table called Branch in your database (capitialization of the name
does not matter), it will be dropped and a new Branch table will be created.
If you want to proceed, enter 1; if you want to quit, enter 2.
1

1. Insert branch
2. Delete branch
3. Update branch name
4. Show branch
5. Quit
Please choose one of the above 5 options: 1

Please enter the branch ID you wish to insert: 1234
Please enter the branch name you wish to insert: South Hill
Please enter the branch address you wish to insert: 5995 Prince Albert
Please enter the branch city you wish to insert: Vancouver
Please enter the branch phone number you wish to insert: 5551234

1. Insert branch
2. Delete branch
3. Update branch name
4. Show branch
5. Quit
Please choose one of the above 5 options: 4

1         First Branch        123 Charming Ave    Vancouver      1234567
2         Second Branch       123 Coco Ave        Vancouver      1234568
1234      South Hill          5995 Prince Albert  Vancouver      5551234

1. Insert branch
2. Delete branch
3. Update branch name
4. Show branch
5. Quit
Please choose one of the above 5 options: 5

Good Bye!
cs-304@valdes:CPSC304-JavaDemoJar$
```

- And that is it. Hand your work in by uploading it to Canvas Assignments. You can put together your work in a single document (e.g.,. PDF, Word, .jpg, etc.) if you wish. In any case, we'll remind the TAs to check for any additional uploads beyond the most recent one; Canvas maintains these.

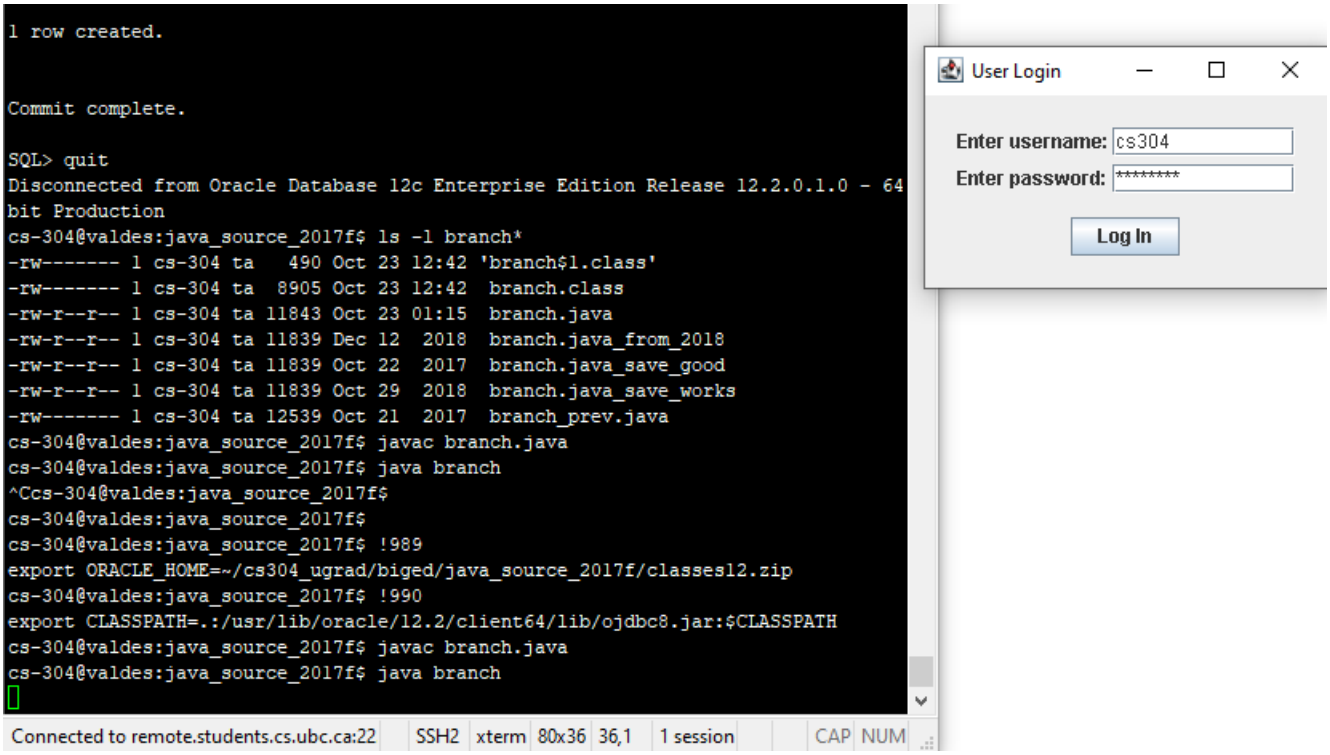
Deliverables if you are using the MVB branch application:

1. Hand in screenshots of at the following items (it is OK to use command-line, SSH, GUI, IntelliJ, your favourite environment, etc; we're flexible), taking care to identify on your screenshots that it is your account, and not your friend's. The example we've provided here uses a text-based approach with SSH, and deals with the motor vehicle branch.

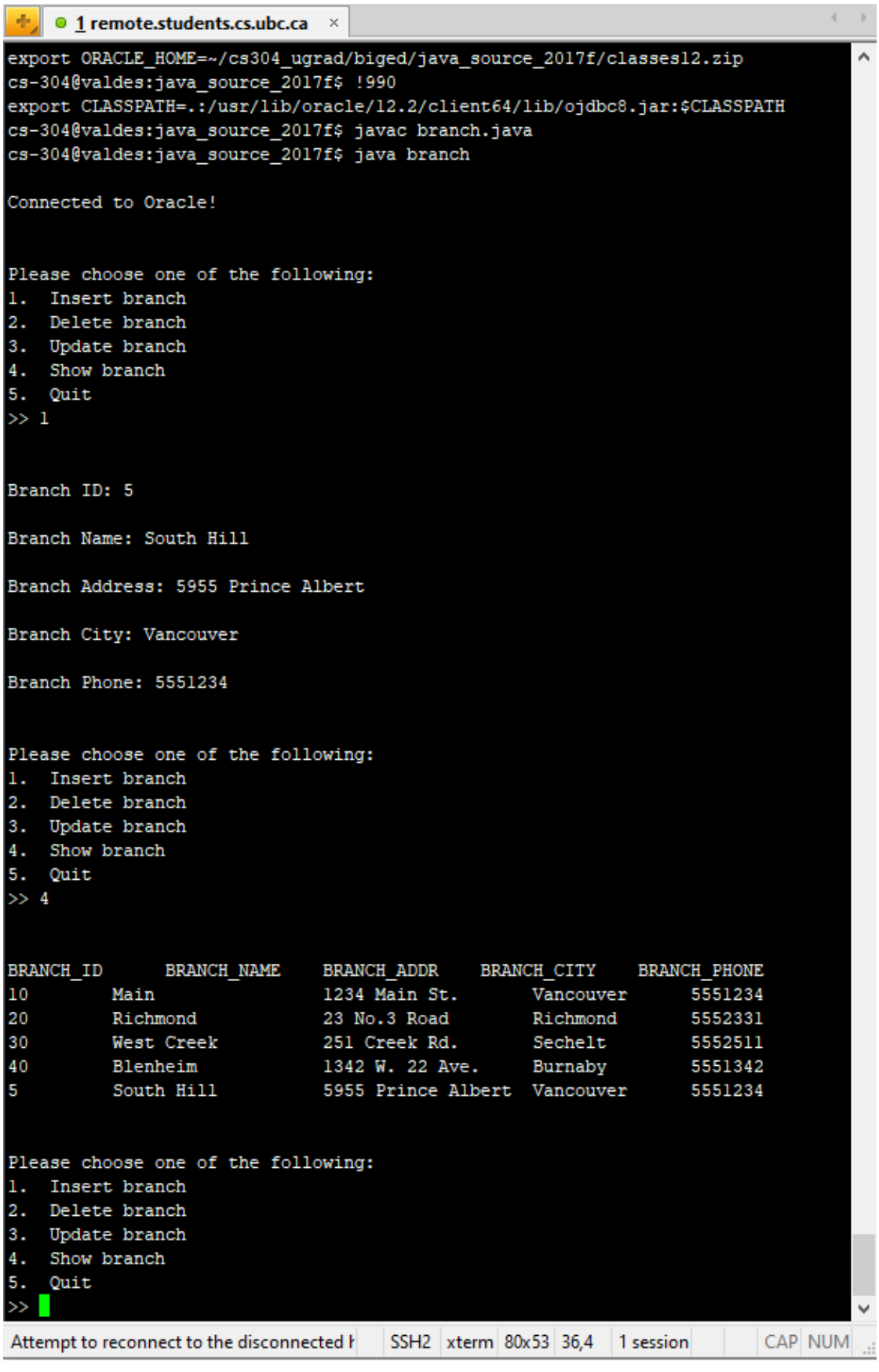
    - **Screenshot 1**: Show that you drop/create the tables (a short confirmation of either the drop or the create is fine … I'm including the web page of where I found the scripts; you don't have to show the web page, and you don't have to show all the confirmation steps—a few are fine). If using SSH, note that `stty erase ^h` is a useful command to avoid all the backspace problems, especially if you make as many typing mistakes as I do.



- **Screenshot 2**: It is OK to have multiple screenshots for this, or to produce the equivalent results if you're using some environment other than the undergrad machines and command line input. The screenshot that follows is just an example.
    i. Include your userid in your screenshot (so that we know that it's you, and not your friend), and show your current directory (Unix `pwd` command, Windows Explorer, or equivalent).
    ii. Show the `branch*` files in your directory/folder.
    iii. Compile your `branch.java` program.
    iv. Give evidence that you compiled the program successfully (e.g., show the latest timestamp in your file list).
    v. Begin running the program.
    vi. Show the pop-up window for sign-on with your userid displayed; but, before clicking Log In, take a screenshot.
    vii. Show screenshot(s) for all of the above. Yours should be similar to the following sample screenshot (my userid is cs304; yours will be your Oracle ora_*userid*), and note that I have some extra files like branch.java_* which you can ignore):



- **Screenshot 3** (this will be a *before* and *after* screenshot of inserting a new branch and showing the results)
    i. An example follows, but can use your own data if you wish:


1 remote.students.cs.ubc.ca
export ORACLE_HOME=~/cs304_ugrad/biged/java_source_2017f/classes12.zip
cs-304@valdes:java_source_2017f$!990
export CLASSPATH=.:/usr/lib/oracle/12.2/client64/lib/ojdbc8.jar:$CLASSPATH
cs-304@valdes:java_source_2017f$ javac branch.java
cs-304@valdes:java_source_2017f$ java branch
Connected to Oracle!
Please choose one of the following:
1. Insert branch
2. Delete branch
3. Update branch
4. Show branch
5. Quit
>> 1
Branch ID: 5
Branch Name: South Hill
Branch Address: 5955 Prince Albert
Branch City: Vancouver
Branch Phone: 5551234
Please choose one of the following:
1. Insert branch
2. Delete branch
3. Update branch
4. Show branch
5. Quit
>> 4
BRANCH_ID BRANCH_NAME BRANCH_ADDR BRANCH_CITY BRANCH_PHONE
10 Main 1234 Main St. Vancouver 5551234
20 Richmond 23 No.3 Road Richmond 5552331
30 West Creek 251 Creek Rd. Sechelt 5552511
40 Blenheim 1342 W. 22 Ave. Burnaby 5551342
5 South Hill 5955 Prince Albert Vancouver 5551234
Please choose one of the following:
1. Insert branch
2. Delete branch
3. Update branch
4. Show branch
5. Quit
>>
Attempt to reconnect to the disconnected h
SSH2 xterm 80x53 36,4 1 session
CAP NUM

- And that is it. Hand in your work by uploading it to Canvas Assignments. You can put together your work in a single document (e.g.,. PDF, Word, .jpg, etc.), if you wish. In any case, we'll remind the TAs to check for any additional uploads beyond the most recent one; Canvas maintains these.