JDBC Exercise

Follow these steps to connect a Java program to your postgresQL database and run some queries from Java. See the other side for the most common errors students make.

1. Log on to dbsrv1.cdf.utoronto.ca
2. Create a directory where you’d like to do this work and cd to it.
3. Get a copy of a file, which defines a small dataset (jelly-beans.sql) and some java code (Example.java) that you will use today:

   ```
   cp ~csc343h/winter/public_html/in_class/w6/code/* .
   ```

(You really do need that period at the end.)
4. Start postgresQL:

   ```
   psql csc343h-<your username>
   ```
5. Import jelly-beans.sql into your database:

   ```
   \i jelly-beans.sql
   ```
6. Use \d to see what is defined and a SELECT query to examine the contents of table guesses. Then use \q to exit psql.
7. Get your bearings in the code. Find the parts where it:

   (a) Establishes a connection to the database.
   (b) Runs a query to find all the guesses by kids under 10, and iterates through the results to print them.
   (c) Builds a query to find all the guesses by a particular person, but with a placeholder for the person’s name. The program then substitutes in a name entered by the user and goes on to run the query and print the results.
8. Edit Example.java to replace dianeh with your cdf userid on these two lines:

   ```
   url = "jdbc:postgresql://localhost:5432/csc343h-dianeh";
   conn = DriverManager.getConnection(url, "dianeh", ";
   ```
9. Compile the Java code:

   ```
   javac Example.java
   ```
10. Run the compiled code:

    ```
    java -cp /local/packages/jdbc-postgresql/postgresql-9.4.1212.jar: Example
    ```

    The path to the jar file is hard to type. Tip: Use filename completion! (And you really do need that colon.)
11. The provided example runs a query on the database, but it doesn’t change anything. In this step, go back to accessing the database again through postgresQL as you did in step 4. You can skip step 5 this time because you don’t need to import the table again. Use a SELECT statement to see that the database has not changed. Now use an INSERT statement to add a new guesser to the table as number 12.

12. Use a DELETE statement to remove the new guesser that you just added.

13. **Helpful Hint:** Before you start this next step, open a second window on cdf. Leave the postgresQL prompt open in the first window, since we will be coming back to examine the database after doing the next step. Add to Example.java so that it inserts guesser number 12. You don’t need to ask the user for the name or the number of guesses, just hardcode these values into your code. Compile and run your updated Java code. See below for common errors. The last error, where you already have a tuple with number 12, might surprise some of you. If you correctly used DELETE to remove this tuple, why is it still there? This will happen if you run your program twice and may even happen if your program failed on an earlier run. Depending on when the failure happened, the tuple may have already been inserted. Repeat step 12 to delete the tuple again and then rerun your program.

14. Once you have successfully run your Example code, use a SELECT statement from the postgresQL prompt to confirm that the new guesser was added. If you didn’t already experience the error discussed above, rerun your program and see it now. Play around with editing and viewing the database from the postgresQL prompt in between runs of your program.

**Common errors**

- If you forget the colon:

  ```
  wolf% java -cp /local/packages/jdbc-postgresql/postgresql-9.4.1212.jar Example
  Error: Could not find or load main class Example
  ```

- If you try to run the code on any machine other than dbsrv1:

  ```
  wolf% java -cp /local/packages/jdbc-postgresql/postgresql-9.4.1212.jar: Example
  SQL Exception.<Message>: Connection refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.
  ```

- If you try to use an executeQuery statement to do an insert. You want to use executeUpdate instead.

  ```
  SQL Exception.<Message>: No results were returned by the query.
  ```

  This is an error that you could see inside psql so doesn’t belong with this list of JDBC-specific errors.

- If you try to use your Example.java program to insert a tuple for guesser number 12 when there is already one in the table:

  ```
  SQL Exception.<Message>: ERROR: duplicate key value violates unique constraint "guesses_pkey"
  Detail: Key (number)=(12) already exists.
  ```