

CS 235 - Homework 10

Deadline:

Due by **11:59 pm** on **Wednesday, December 2, 2015**.

How to submit:

Submit your files using `~st10/235submit` on nrs-projects, with a homework number of 10, by the deadline shown above.

Purpose

To provide practice with JDBC.

Important notes:

- **If you missed the Week 13 Lab**, be sure to set up your nrs-projects account as described in the Week 13 Lab Exercise handout (posted along with this homework handout) so that you can run JDBC from your nrs-projects account!
- The `dragon` table has been created on the Oracle `java` account, and has had information on a number of dragons inserted into it.
- If you have an HSU Oracle account and would rather use it than the `java` account for any of this homework's problems, feel free to do so. If you don't, and would like one, e-mail me accordingly as soon as possible and I will request one for you.
 - If you do use your own account, you will want to run the posted `dragon.sql` SQL script to create and populate the `dragon` table. You can do so by creating file `dragon.sql` on nrs-projects, and then typing, from the nrs-projects prompt:

```
sqlplus
```


entering your HSU username and password when prompted,
and then, at the `SQL>` prompt, running:

```
start dragon.sql
```


Then you can type `exit` to exit `sqlplus`.
 - You are certainly welcome to add additional dragons to your copy of the `dragon` table if you wish.
- Follow the Java coding standards mentioned in previous homework handouts and discussed in class.
- Note that Java applications with graphical user interfaces are expected to be structured in the way demonstrated in the in-class examples (as in `ButtonTest.java`)
 - ...but using appropriate additional helper methods in the `JPanel` subclass is fine, as is using additional classes. IF you use additional public classes, be sure to submit those as well!
- It is possible that some of your programs may be posted to the course Moodle site.

Problem 1

As a warm-up: write a Java command-line application `IntroDragons.java` that starts by printing a descriptive introductory message to the screen including your name, and then simply queries and prints to the screen the `dragon_class` and `dragon_name` for each dragon in the `dragon` table,

- printing one dragon's information per line, and
- separating the `dragon_class` and the `dragon_name` by a space, a dash, and a space

The SQL query string you should use for this is:

```
"select dragon_name, dragon_class "
+ "from dragon "
+ "order by dragon_class, dragon_name"
```

Submit your resulting `IntroDragons.java`.

Problem 2

There are a number of statistics for each dragon in the `dragon` table. To slightly-more-safely query for these for particular dragons, write a Java command-line application `AllAboutDragon.java` that:

- expects one or more command-line arguments, each expected to be the (possible) name of a dragon
 - what if there are no command-line arguments? Complain and exit.
- starts by printing a descriptive introductory message to the screen including your name
- then uses a `PreparedStatement` to query all of the dragon information for each command-line-argument dragon name,
 - and for each attractively outputs all of the data for that dragon.

(NOTE that the dragon name, species, and class are all strings, and the rest of the dragon values are all integers.)

 - What if a dragon with that name is NOT found?

Then the `ResultSet` instance's `next()` method will return `false` when it is first called -- in this case, you should print a message to the screen, including the not-found dragon's name, saying no dragon with that name was found.

The SQL query string you should use for this is:

```
"select dragon_name, dragon_species, dragon_class, dragon_attack, "
+ "    dragon_speed, dragon_armor, dragon_firepower, "
+ "    dragon_shot_limit, dragon_venom, dragon_jaw_strength, "
+ "    dragon_stealth "
+ "from dragon "
+ "where dragon_name = ? "
```

Submit your resulting `AllAboutDragon.java`.

Problem 3

Think of something else you'd like to do with this `dragon` table, and write a Java application that uses JDBC to help to do it. It should output or somehow display your name when it runs, and it should include `Dragon` somewhere in its class name (so I can tell it is your Problem 3 answer...!)

Some suggestions:

- It could be as simple as outputting to the screen all of one of the characteristics of all of the dragons;
- It could let the user enter some limit value, and it prints all of the names of dragons higher than (or lower than) that limit for some characteristic or characteristics;
- It could let user enter a dragon class, and list all of the names of dragons in that class;
- It could perform some computation of a dragon's values for a given dragon;
- It could use command-line arguments or interactively-entered information;
- It could allow the user to enter a new dragon into the `dragon` table;
- It could allow a user to enter two dragon names, and some characteristic(s?), and it would determine which would "win" in a contest;
- ...etc.!

This can be a command-line application, or it can be a GUI application if you want to try that; I will be posting (in the next few days) ideas for how you might be able to run JDBC from a computer other than nrs-projects.

NOTE that I would be happy to (try to!) send you SQL statement string(s) for your program, but you need to request them by e-mail SOONER rather than later, and no later than 10:00 pm on Tuesday, December 1st.

Submit your resulting `.java` file (or files, if what you do uses more than one Java class).