

### Program #3: JDBC

*Due Date: November 1<sup>st</sup>, 2018, at the beginning of class*

**Overview:** Embedding SQL within another programming language is nice for applications that require more complex calculations or manipulations than plain SQL can handle. Many DBMSes have add-ons that can be used for developing nice windowed applications, but even they may not be flexible enough to create the application you have in mind. Java's JDBC API is a common way for Java programs to exchange data with a DBMS.

Multiple times per year, the Florida Department of Education tests students in a subset of subjects, one of which is Biology 1, and reports a per-school summary of the results in Microsoft Excel `.xls` files. It's easy to get a spreadsheet program (such as Excel or LibreOffice) to output a spreadsheet in CSV (Comma Separated Value) format, which is a good format for reading into a program, and from there into a database.

**Assignment:** For this assignment, you will need to do the following, not necessarily in this order:

1. Get the Excel files for the State Report of Schools (not Districts!) Biology 1 EOC Assessments for the spring administrations from the years 2015, 2016, 2017, and 2018 from the [fldoe.org](http://fldoe.org) web site (see the **Data** section, below).
2. Using a spreadsheet program, convert their content to CSV files (or another text file format that appeals to you).
3. “Scrub” the files to make them consistent and suitable for importing into an Oracle database. (Again, see **Data**, below.)
4. Within your Oracle database, create four tables with an appropriate schema, one table for each year's data.
5. Move the results data from the files into the relations. (We expect that you will write a small Java program for this, but that's not a requirement.)
6. Write an application in Java 1.8 using JDBC that offers the user a menu of four questions that can be asked of the database of FL DOE data. The questions are:
  - (a) How many schools are listed with different school names in 2018 than they were listed in 2015? To be counted, a school must appear in both years.
  - (b) For a year given by the user, what are the district numbers, school numbers, and school names of the schools that have a strictly decreasing sequence of achievement level percentages, assuming the the percentages start with Level 5 and end with Level 1? For example, a school with percentages 34, 30, 20, 10, and 6 from Level 5 to Level 1 would be included.
  - (c) For a year and a district name given by the user, if the district has at least 10 schools in the report, list in descending order the five schools with the highest combined level 4 and 5 percentages. If the district has less than 10 schools, print “I'm sorry, but district [name] has just [#] schools.” Replace [name] and [#] with the given district name and quantity of schools.
  - (d) A question of your choice, subject to these restrictions: The question must use two or more of the four relations and must be constructed using at least one piece of information gathered from the user. If you have trouble thinking of something, envision yourself as a parent of a student and think about what you as a parent might like to be able to ask.

(Continued...)

**Data:** The Excel files with the Florida assessment data for the years 2015 through 2018 can be downloaded starting from this FL DOE web page:

<http://www.fldoe.org/accountability/assessments/k-12-student-assessment/results/>

You will need to perform some ‘scrubbing’ of the data. For example, we are interested in only the school data, not the metadata at the tops of the spreadsheets, so don’t try to store that. Another example: Many schools are missing all of the data except the numbers of students. Mark missing data with NULLs. You are likely to think of additional needed adjustments as you explore the data. These sorts of inconsistencies can be eliminated with a combination of creative (or brute-force!) editor skills, small data cleaning programs of your own creation, and/or basic SQL statements. It’s always tedious, but rarely difficult. (And see the Other Requirements and Hints section, below, too.)

**Output:** Your application is to display the output of a question in a clear, easy-to-read format (remember, you’re writing your own program; you are not restricted to SQL’s output format).

**Hand In:** You are required to submit a `.tar` file of your well-documented application program file(s) — including any code written to automate the data scrubbing and/or loading processes, although that code may be minimally documented — via turnin to the folder `cs460p3`. Name your main application program’s `main()` class `Prog3`, so that we don’t have to guess which file to compile, but split up your code over additional files as appropriate for good code modularity.

### Want to Learn More?

- The FL Dept. of Education K–12 Student Assessment Results page is:  
<http://www.fldoe.org/accountability/assessments/k-12-student-assessment/results/>
- Why Florida data and not Arizona data? Variety! I sometimes use the AZ data. See:  
<http://www.azed.gov/accountability-research/data/>

### Other Requirements and Hints:

- Because we will be grading your program on `lectura` using Oracle, it needs to run on `lectura` and use Oracle.
- If you wish to share any necessary data conversion and “scrubbing” chores with a few of your classmates, that’s fine. Stop collaborating when you start coding your JDBC application. In your documentation, be sure to credit those who helped you with the data organization. Please DO NOT post scripts, etc., on Piazza; we don’t want one person doing all of the dirty work for the entire class!
- It is OK to share query results on Piazza. Doing so can help you discover, for example, that your query isn’t finding everything it should be finding.
- Make certain that your database tables are accessible to us (by GRANTing us `SELECT` privileges) and that your relations are prefixed with “`yourNetID.`” in your application so that we can execute your program against your database, just as my tables for Homework #3 were accessible to you with the “`mccann.`” prefix. The form of the `GRANT` command is:  
  

```
GRANT SELECT ON tablename TO PUBLIC;
```
- Avoid the temptation to wait to start writing the JDBC code and your application program until you have all of the data loaded into tables. You can (and should!) create and populate small tables for testing purposes early in the development process.