CS3431-A19: Project Description
Phase 3: Simple Database Application

Due Date: Friday, Oct. 4 at 11:59pm.
Late Policy: 10% off until Saturday, Oct. 5 at 11:59pm

Teams: The project is done in the same teams of two as in the previous project phases.
Submission: One teammate will upload a zipped file containing the pdata**X**.sql and pf**X**.java (**X** should be replaced with your project team number – for example, Team 8 would be pf8.java) file to Canvas. Although you may develop the code using either Eclipse or IntelliJ, **the TA will compile your file on a CCC machine so make sure your file compiles and runs on the CCC machines**! Any comments or assumptions that you have, you can include them in a separate Word or PDF file. **Also make sure to use the same interface, filename, and class name or we will not be able to grade your project and your grade will be 0**!

Description:
For the first two phases of this term project, you were a database developer in charge of creating the database component that application developers would use. For this final phase of the project, you will get experience from the other side - you are now a Java application developer who will access and update an external database.

Part 1: Populate the Database
Beginning with the project3start.sql file, enter data for the tables and include these SQL commands in the pdata**X**.sql file where X is your team number.

a. 20 patients
b. 5 PCPs
c. 5 surgeons
d. 3 OP doctors
e. 20 surgeries, one for each of the 20 patients
f. fill any additional tables that were created for the relationships

Note that project3start.sql has some differences from project2start.sql such as the Boolean for isBoardCertified. Sequences are also different. For the doctor, patient, and operation primary keys, create a sequence that starts with your team number times 1000 and increment by 10. You will NOT create a sequence for Organ because OP doctors may have organs with the same Organ ID. If you are team 17, the first value for physicianID would be 17000 and your SQL commands will be saved in a file named pdata17.sql. Do not include the starter code in your submission. Do include the create sequence and insert commands in your submission.

Part 2: Java Database Program
You are required to write a java program that accesses and modifies the database, and prints results on the screen. Your program will be the interface to perform some simple functionality over the database. **Pay careful attention to the naming of your file and class so we can run them and give you credit for the project phase!**

Create a program named "pf**X**.java" where **X** is your team number. The program will always take two parameters, e.g., username and password to connect to the database. (Pass them as parameters such that the TAs can easily set them as needed without recompilation).

0. When your program is executed without any additional arguments, e.g.,

```
> java pfX <username> <password>
```

Then the program should output a message with the following options, and then terminate:

```
1 – Report Patient Information
2 – Report Primary Care Physician Information
3 – Report Operation Information
4 – Update Patient Blood Type
5 – Exit Program
```

If your program is executed without enough parameters, the program should output a message indicating that the user name and password should be included as parameters.

1. When the program is executed with an argument 1 as follows:

```
> java pfX <username> <password> 1
```

The program now enters the “Report Patient Information” mode. The program should print out the following line:

```
Enter Patient’s Healthcare ID: <and wait for user’s input>
```

When the user enters the healthcare ID, the program should execute a query of the Patient table and print on the screen the following location information and then terminate.

```
Patient Information
Healthcare ID: …
First Name: …
Last Name: …
City: …
State: …
Birth Date: …
Blood Type: …
```

2. When the program is executed with an argument 2 as follows:

```
> java pfX <username> <password> 2
```

The program now enters the “Report Primary Care Physician Information” mode. The program should print out the following line:

```
Enter Primary Care Physician ID: <and wait for user’s input>
```

When the user enters the physician ID, the program should execute a query and print on the screen the following Primary Care Physician information and then terminate.

```
Primary Care Physician Information
Full Name: …                  (this should display the first and last name)
Physician ID: …
Specialty: …
Medical Facility: …
```

3. When the program is executed with an argument 3 as follows:

   ```
   > java pfX <username> <password> 3
   ```

   The program enters the “Operation Information” mode. The program should print out the following line:

   ```
   Enter Operation Invoice Number: <and wait for user’s input>
   ```

   When the user enters the operation invoice number, the program should execute a query of the Operation and related tables and print on the screen the following information:

   ```
   Operation Information
   Invoice Number:  …
   Operation Date: …
   Surgeon Full Name: …
   Board Certified?: …
   Patient Full Name: …
   Blood Type: ...
   City: …
   State: …
   ```

   Then the program terminates.

4. When the program is executed with an argument 4 as follows:

   ```
   > java pfX <username> <password> 4
   ```

   The program now enters the “Update Patient Blood Type” mode. The program should print out the following line:

   ```
   Enter the Patient’s Healthcare ID: <and wait for user’s input>
   Enter the Updated Blood Type: <and wait for user’s input>
   ```

   Then your program should update the blood type in the Patient table and then the program terminates. Now if you execute option 3 again with the same healthcare ID, you should get the updated blood type.

   Note 1: **the user entering an input should NOT have to enter single quotes as part of the input**. For example, if a blood type is being input by the user, they should be able to type AB instead of 'AB' in order for the program to work.