

CS 336 Homework 3

1. Create a database in MySQL for representing students, departments, and classes.

We have the following departments: Bio, Chem, CS, Eng, Math, Phys. Each department has a name, and a campus (Busch, CAC, Livi, CD).

Students must have a first name, last name, and 9-digit ID number. The ID is unique among all students. Students may also have one or more majors and one or more minors in some department. A student may be currently taking some number of classes, and may have taken some number of classes. A course a student has taken will have a grade assigned (A, B, C, D, or F).

Each class has a name and number of credits (3 or 4). For example, “Introduction to Whale Songs” might be 3 credits. You can create your own class names.

Please use these table and field names:

- `Departments(name, campus)`
- `Students(first_name, last_name, id)`
- `Classes(name, credits)`
- `Majors(sid, dname)`
- `Minors(sid, dname)`
- `IsTaking(sid, name)`
- `HasTaken(sid, name, grade)`

2. Populate your database with reasonable random data for 100 students. For example, you should have roughly equal numbers of students in each year (see 3(b) below).

The campus for each department can be randomly selected and doesn’t have to match reality (e.g., CS could be on CAC, Eng could be on CD, etc.).

3. Create a Java application that allows users to query the database. You should use JDBC to connect to the database. You should support the following queries and operations:

- (a) Search students by name. This should match any students where the search string is a substring of either the first or last name (case insensitive).

- (b) Search students by year (Fr, So, Ju, Sr), where a student's year depends on how many credits they've completed (with a grade higher than F).

Year	#credits
Fr	0 – 29
So	30 – 59
Ju	60 – 89
Sr	90+

- (c) Search for students with a GPA equal to or above a given threshold.
GPA is not stored in a table, but can be calculated based on the set of classes a student has taken. Point values for letter grades are A = 4, B = 3, C = 2, D = 1, and F = 0. Then if a student took 5 3-credit classes and got grades of A, B, B, C, and D, and took 3 4-credit classes with grades of B, C, and F, their GPA would be

$$\frac{3(4 + 3 + 3 + 2 + 1) + 4(3 + 2 + 0)}{27} \approx 2.19$$

where 27 is the number of credits for the classes taken. Note that the number of credits here includes courses where the student got an F, unlike in 3(b).

- (d) Search for students with a GPA equal to or below a given threshold.
(e) For a given department, report its number of students and the average of those students' GPAs.
(f) For a given class, report the number of students currently taking it. Also, among students who've taken the class, show the number of students who've gotten each letter grade.
(g) Execute an arbitrary SQL query. You may want to also use the `ResultSetMetaData` class to help with printing the result.

For students that match a query, it should print them in the following format:

```
2 students found
Smithers, Jessica
ID: 384571348
Major: Math
Minor: Phys
GPA: 3.825
Credits: 65
```

```
Smith, Alan
ID: 478237289
Majors: Bio, Eng
GPA: 3.6
Credits: 42
```

Your application should take as command-line parameters a URL, username, and password. The URL will be given in the format server:port/dbName.

Example

```
$ java UniDB localhost:3306/uniDB root rootpass
Welcome to the university database. Queries available:
1. Search students by name.
2. Search students by year.
3. Search for students with a GPA >= threshold.
4. Search for students with a GPA <= threshold.
5. Get department statistics.
6. Get class statistics.
7. Execute an arbitrary SQL query.
8. Exit the application.
Which query would you like to run (1-8)?
1          <-- user input
Please enter the name.
smith      <-- user input
... print student matches ...
Which query would you like to run (1-8)?
2          <-- user input
Please enter the year.
Fr         <-- user input
... print student matches ...
Which query would you like to run (1-8)?
3          <-- user input
Please enter the threshold.
3.5        <-- user input
... print student matches ...
Which query would you like to run (1-8)?
4          <-- user input
Please enter the threshold.
3.5        <-- user input
... print student matches ...
Which query would you like to run (1-8)?
5          <-- user input
Please enter the department.
bio        <-- user input
Num students: 23
Average GPA: 3.7
Which query would you like to run (1-8)?
6          <-- user input
```

```

Please enter the class name.
Calculus      <-- user input
17 students currently enrolled
Grades of previous enrollees:
A 5
B 12
C 18
D 3
F 2
Which query would you like to run (1-8)?
7             <-- user input
Please enter the query.
SELECT * FROM classes WHERE credits = 3
name    credits    <-- column headers, tab separated
Calculus    3       <-- row data, tab separated
Oceanography 3      <-- row data, tab separated
Which query would you like to run (1-8)?
8         <-- user input
Goodbye.

```

Submission

Please write a README file describing how you generated your random data and loaded it.

In a zip file, please include your main application, the README, and any other relevant code you used (e.g., for data generation).