Pair Assignment: MySQL + Java

Due Nov 7 by 11:59pm **Points** 100 **Submitting** a website url

Pair Programming Assignment

Read the following if you are not familiar with pair programming: **PairProgramming.pptx (https://wwu.instructure.com/courses/1500739/files/81104107/download?download_frd=1)**

You are supposed to work together at the same time and coordinate this, online or face2face but at the same time.

Instructions

For this assignment, you will be using the existing Employees database (**employees.sql.zip (https://wwu.instructure.com/courses/1500739/files/81104085/download?download_frd=1)**) that you have been using to construct your queries (assignments **SQL 1** and **SQL 2**). This time around we want to allow the manager to access some of the database but without having to teach them SQL. Typically that would be the time to develop a web interface or better yet a RESTful API but this is still a database class, so, nope.

Instead, we will be building a command line interface (CLI) for our HR manager and pretend that they would be using such an interface to manage the employee database. You will be using Java for this part of the assignment and its JDBC connector to access MySQL. There is a handful of IDEs (e.g., IntelliJ) or you can just work everything through vim, nano, etc. There is plenty of online tutorials for Java and MySQL (e.g., **https://www.vogella.com/tutorials/MySQLJava/article.html (https://www.vogella.com/tutorials/MySQLJava/article.html)** or **https://personal.ntu.edu.sg/ehchua/programming/java/jdbc_basic.html (https://personal.ntu.edu.sg/ehchua/programming/java/jdbc_basic.html)**). Notice that in this example the credentials are hard-coded in the application as opposed to an external file.

The JDBC driver can be found here: **https://dev.mysql.com/downloads/connector/j/ (https://dev.mysql.com/downloads/connector/j/)**

The CLI will need to do the following functions (example output is included):

Important:

- **Anything in between <> means fill in the blanks and remove the <>. E.g., <first_name> becomes Michael**
- **All input is passed as command line arguments (NOT user input).**

java main show employees department <department name> # printing the employee table as a tsv for a particular department

```
<empid> <first_name> <last_name>

<empid> <first_name> <last_name>

<empid> <first_name> <last_name>
```

java main add employee <first_name> <last_name> <dept_name> <birthdate> <gender> <salary> # adding a new employee to the database

birthdate format = YYYY-MM-DD

Note: to_date in salaries should be set as the employees current salary based on the database convention: 9999-01-01

```
Employee <first_name> <last_name> added!
```

java main delete employee <empid> # deleting an employee from the database

```
Employee <first_name> <last_name> deleted!
```

or

```
Employee with id <empid> does not exist .
```

java main show salaries sum # displaying the total salary payouts for the latest year for all employees

```
$900000
```

Logistic Instructions

- Your program/solution must be developed in a git version control repository (gitlab.cs.wwu.edu). The name for your repository can be anything but files must be placed on the root directory of the repository. Set the visibility settings on **Private** and add my account (**tsikerm and stovalc**) as a **Maintainer** to your repository. See **Git Quick Guide** for more info.
- There is a "checkpoint" deadline of this assignment upcoming **two weeks before it is due**. At that point you will need to have set up your repository and submitted its URL on canvas.
- An automated testing suite (ATHINA - **A**utomated **T**esting **H**omework **I**nterface for **N** **A**ssignments) will assist in verifying that your application is compliant with the project requirements. Once you submit the location of your files, it will test the repository and submit 80% of your grade depending on the outcome of the tests. With every new commit to your repository, it will re-evaluate and submit a new grade. This is meant to give you immediate feedback and multiple opportunities to correct your code and get full points for the assignment.
- I will also test your program by reading your source code and evaluating that everything is in order.
- Your main file must be main.java but you are allowed to have as many java (and other files) in your git repository. In other words, modularize your code any way you like.
- Your main script name should be **main.java**. If you are getting a jdbc not found error you can pass terminal parameters as (replace with where your jdbc mysql jar file is):

  -
    ```
    CLASSPATH=$CLASSPATH:/usr/share/java/mysql.jar
    export CLASSPATH
    ```
- You will need to load the connection string from the code below from a file called **credentials.txt** (add this file to .gitignore, I'll be loading my own credentials file locally):

  ```
  connect = DriverManager.getConnection("jdbc:mysql://localhost/feedback?user=sqluser&password=sqluserpw");
  ```

In the text file you only need to add this portion (typing another variant of the connection string): `jdbc:mysql://localhost:3306/employees?user=user&password=password&useSSL=false`

Make sure that your program eliminates any trailing new line (\n) characters that may have been added to the text string.

Note: If your password contains non URL compatible characters, encode it (**https://www.url-encode-decode.com/** **(https://www.url-encode-decode.com/)**)

Sample code to get you started:

```
import java.sql.*;
import java.io.File;
import java.util.Scanner;

class main {
    public static void main(String[] args) throws Exception {
        //your code here
    }
    //other functions here
```

```
}
```