Assignment # 6 - SQL Project (Part 2): From ER Design To Implementation - Paper Review Database - JAVA Application

(Due: 5/18/2019 @ 11:59PM) [90 Points]

So far, in this project you have taken an existing ER Diagram for a “paper reviews” database and implement it in a relational database. This database was initially designed in Assignment 3’s problem # 3.34 (see the assignment’s key for details).

In this part of the project you will create a simple command line, Java application to interact with your database.

Part A: Setting Up Your Java Development Environment

Before getting started, you must set up your Java development environment. Note, if you already have a Java development environment setup feel free to skip this step.

Setting up your environment includes installing the Java JDK (version 8 or 9) and installing an integrated development environment (IDE) such as Eclipse.

Please note, if you are comfortable writing and debugging your Java application via a simple text editor and the command line you do not need to install an IDE. If you have another prefered Java IDE such as IntelliJ or Netbeans feel free to use that instead of Eclipse.

*** The remaining sections assume you are using Eclipse. The steps are similar for other IDEs etc but the details will vary.*

To install Eclipse and the Java JDK follow this tutorial. It has instructions for Linux, OS X and Windows:
https://www.ntu.edu.sg/home/ehchua/programming/howto/EclipseJava_HowTo.html

Part B: Create A Java Project In Eclipse

A project in Eclipse is essentially a folder containing all the source code and other files you need to build your application. Let's start by creating a project for your program.

1. Open Eclipse 2. From the File menu select New → Java Project
2. From the File menu select New → Java Project
3. Give your project a name, like "CS623PaperReviewProject." You can use the default values for all the other settings. Make note of the project folder in the "Location" box; you will need to remember this later.
4. Press Finish to create the project
5. Eclipse might ask you whether you want to switch to the Java perspective. If so, say Yes.
6. You should see an empty project.

Part C: Installing The JDBC Driver To Connect To Your MySQL Database

In this step, we install the Connector/J driver so your program can connect to MySQL. Please make sure you download the version that is compatible with your version of MySQL.

1. Go to this web page: http://www.mysql.com/downloads/connector/j/
2. Press the download button next to the "Platform Independent (Architecture Independent), ZIP Archive" version. (Or the TAR version if you prefer; it doesn't really matter.)
3. The next page will ask you to create an account. Instead, click the little blue link on the bottom that says "No thanks, just start my download."
4. Open the .zip file you just downloaded, and look for a file inside named something like mysql-connector-java-xxxx-bin.jar. Copy this file into your project directory (from the Location box in step 3 of the last section).
5. Now that we have the driver, we need to tell your project about it. Go back to Eclipse, right click on the project, and select Build Path → Add External Archives…
6. Select mysql-connector-java-xxxx-bin.jar and press Open
7. Now we're ready to configure the project for your copy of MySQL and run it!

Part D: Let's Write Some Code!

First, create a new java class called PaperReviewDriver.java in the java project you created. Note, this is the class where you will implement and test the methods that implement your database queries.

To create a new Java class:

- Right click on the project name and select **New->Class** from the context menu.
- When the **New Class** dialog appears, enter "PaperReviewDriver" as the class name and check the box **public static void main(String[] args)**.
- Click on the **Finish** button.

To run your program:

- Right click on the file with the ***main*** method, and from the context menu select the **Run->Java Application** menu option.
- The program runs. Any console output is directed to a window at the bottom of the screen.

JDBC code examples:

- https://www.tutorialspoint.com/jdbc/jdbc-sample-code.htm
- https://examples.javacodegeeks.com/core-java/sql/preparedstatement/java-sql-preparedstatement-example/

Next, in the above class, create java methods that implement the queries below using SQL. Implement a method per query. Note, all of your queries should be parameterized queries if they are taking in input such as an Id etc. For each method print the query's result set to the console. If no results were returned, print a message that indicates no results were found etc.

Please follow the standard Java naming conventions listed in the "Basic Naming Conventions Java" document located in "Documents > Course Materials" in blackboard.

Queries:

- Get a submitted paper's details by the author's Primary Key. The query should return the following data (columns): Paper.Id, Paper.Title, Paper.Abstract, Author.EmailAddress, Author.FirstName, Author.LastName
- Get all reviews for a paper by the paper's Id, where the paper was recommended to be published. The query should return the following data (columns): All columns from the Review table.
- Get a count of all papers submitted.
- Create a new paper submission. Remember this includes creating new records in both the Author and Paper tables.
- Try and Delete the first "Author" row in your Author table by the author's id. Did you receive an error? If yes, print to the console the error you received. Also note in your message why the query failed. If it didn't fail, print a message explaining why you were able to delete the row.

Lastyly, in the main method, call each of the methods you have created. Make sure all results from your queries are being printed to the console.

Submit Your Work

[IMPORTANT] I will be testing / running all java applications against my local PaperReviews database. All code must compile and run properly. Applications that fail to compile will not be graded. Please see the course syllabus for details regarding how coding assignments are graded.

- Create a screencast that shows you running your java application. It should show all of your implemented queries running and hence printing their results.
- Create a new folder in the github repository you created for Part 1 of this project. Add all of your .java source code file(s) to this new folder.
    - Please do NOT submit the project from your IDE.
    - Add a second README file to the new folder which explains your application design, the queries it runs as well as instructions on how to run your application.
- If you updated your database schema or added additional database objects please make sure new or updated SQL scripts are in your repository.
- Submit the link to your screencast video and the link to your project's github repository.