Assignment 7 Database design with JDBC

Objectives

- Normalizing data for creation in an application
- Creating tables in the database for use by Java.
- Creating objects in Java that match normalized data
- Reading and writing objects to the database.
- Lay groundwork for the project

Instructions

Follow the instructions below to normalize data, create their tables in PDBORA19C, insert data, and create the associated Java code.

Database

Data

For this assignment you will be working with a table of data representing the courses taken by computer science students in the 420.B0 program. You can find a PDF of the information in CourseList420B0.pdf. Note, the PDF contains a link to the same information on Dawson's website.

Although the document refers to Terms by number, i.e. Term 1, Term 2, etc. add additional information about what semester this term occurs. I.e., Fall, Winter, Summer. Be careful to consider how this affects normalization.

Normalize

Normalize this information. Draw an Entity Relationship Diagram (ERD) for the final normalized data. Ensure you follow the standard normalization process.

To create the ERD you can use any tool you like as long as it is legible. Some free solutions: https://app.diagrams.net/, https://www.lucidchart.com/pages/ (requires sign up).

Create the tables

Create a new SQL file called assignment7.sql. In this file, write the create table statements for creating the normalized version of the database. Ensure you handle constraints (PRIMARY KEY, FOREIGN KEY) as necessary.

Insert Sample Data

In assignment7.sql, write the insert statements to fill out your tables with the necessary data. Add the course Programming I course and Programming II course. Note, you do not have to transcribe all the courses from the PDF, just the ones asked.

Creating the Java Application

Setting up Java Project

Create a new Java project called ca.dawsoncollege.assignment7. Add the JDBC dependency to your Maven pom.xml file. See assignment 5 for a refresher about how this is done.

Adding data with objects

Create new classes for the tables you have made above. Ensure they have all the fields necessary to build the object and can be constructed. For each class add the following method:

- Create a method called addToDatabase(Connection conn), which takes in an SQL connection. It should add the current object to the database. Note this is different from assignment 5 as the object itself will know to add itself to the database provided it is given a connection.
- Objects should override their toString() method to allow it to be printed out

Create a new class called **CourseListServices**. This class will contain the code necessary to connect to the database, as well as other service methods.

- Create a constructor that takes in the Username and password for the database. Create a connection to the database that will be used by **CourseListServices** after construction. This means the connection must not be stored in a static variable.
- Create a method called **Close**, it must close the connection opened when the object was constructed.
- Create a method called **addCourse**, it should take all information necessary to add a new course. Note, this should be a user-friendly method, where people can read off the PDF document and provide the information.
    - The method must create the corresponding Java objects you created above and use their methods to add information to the database.

Testing the objects

Inside the App class, which should have been created by default and contains a main method perform the following. In the main method prompt the user to provide their username and password. Use the **CourseListServices** to add the Programming III course to the database. Indicate to the user that the course has been added by printing it to the console.

Deliverable

Submit a zip called assignment7.zip. It must contain assignment7.pdf that contains your ERD diagram. Additionally, submit the SQL Script file assignment7.sql. Be sure your script is runnable from start to finish (i.e., it drops tables, indexes, etc. at the start) and has the necessary comments to be well understood. Finally include your Java code for assignment7. Ensure it is well structured, compiles, and runs without errors.