Final Project

Step 3: 10 points *+ 5 bonus points*

A. Create Database and Tables:

Translate the relational schema you developed in Step 2 into an actual database. Use MySQL's Workbench or Shell and the SQL query language to create your database and the corresponding tables. Place the constraints (e.g., default values, foreign keys) you find appropriate - **Important notes:**

1- In Step 2, there were some requirements guiding the minimum number of entity sets and relationship sets for your database. These are the minimum requirements because of the limited time, however, as the designer and developer, you may decide that your database requires more entity sets and relationship sets.

2- In Step 2, you submitted an ER diagram and corresponding relational schemas – as the design process of any database is an iterative one, you can add/remove/update the diagram and schemas to fulfill the implementation phase.

3- In Step 2, you submitted a number of queries - as we discussed more SQL operators and techniques, you can update/modify the submitted queries to fulfill the implementation phase.

> As you decide on which programming language to use (e.g., Java, Python, JS) and the type of your application (e.g., desktop, web, mobile), you should first take some time to make sure you can connect with your MySQL database and test simple queries (e.g., you may test using one of the databases you created in assignment3 or assignment4).
>
> - Check this early so you can decide on the language and the type.
>
> - The Java tutorial we discussed assumes your database runs locally (on the same device that hosts the Java application).

B. Prepare and Insert Records:

Create a simple "***Interactive***" application (GUI-based or web-based) that connects to the created database to insert tuples and run queries. Prepare the data tuples/rows to be inserted into your database as follows:

1. You can ***(Recommended option)*** populate your database with real data. Given your ER diagram, you will figure out which attributes are essential (this can be two, three, or four attributes) and try to find real data to cover these attributes.

- Search online for data (usually in .csv, .txt, or .excel format) and use them ***through your app*** to build the insert queries to fill the different tables.
- You may not find a data set that fully covers all attributes of all tables; such data is expected to cover only the important attributes.

2. If you don't have real data to use or you couldn't find a data set that fully covers all attributes of all tables (as discussed above), your app should generate random data for the uncovered attributes - The generated data should make sense as much as possible (e.g., don't generate negative ages for employees or students with GPA > 4.0). Take the following ***Football Games*** table from a football database as an example:

GameID	Team Name	Opponent Team	Match location	Match date	Result

To prepare data tuples/rows for this table, search the internet and find real data covering the Team Name, Opponent Team, and Match Date attributes. In this case, your application should then randomly generate data for the rest of the attributes (GameID, Match location, and Result). The GameID can be an integer that autoincrements every time you insert a record; the Result attribute can be a random integer from 0 to 10, the Match location can be a random choice from a list like (Paris, London, Cairo, Moscow, Dubai).

3. If finding real data from external online resources or generating meaningful random data using will be a complex problem, you may use online tools (e.g., https://www.mockaroo.com/) to generate random data into files, then ***use your application*** to read these files, line by line, and build the corresponding insert queries.

- From the data tuples you prepared, ***your app*** can now populate (via SQL insert command) the database with these tuples. Each main table in your design is expected (***if possible***) to have at least 500 tuples.

C. Run Queries and Print Results:

Your app handles user interactions. Display a nice name for your application, along with a short welcome message describing the goal of your database. ***You can design how your users may interact with your database the way you find appropriate, however, there are few important points:***

- The user should not write SQL queries through your GUI to be executed.
- You may allow the user for further input after selecting a query (e.g., you might have a generic query like "*What are the top 10 products in the year of X?*" – you may then ask the user for the value of X, the user then may enter 2022 for you to run the query)
- After the application executes a query, the app should allow the user to select another query or exit the application.

***here is an example of an interactive program*:**

Your application may display the different queries you prepared in step2 (query number and short description) and ask for user's input (user enters '8' and clicks enter, your application executes the 8th query). Your application will display the results the way you find appropriate on the GUI.

Your application *might (not essential)* also allow the user to select one of the following options:

- *Option1 - "Display the results here!"* In this case, your application will display the results the way you find appropriate on the GUI.
- *Option2 – "Save the results to a file!"* In this case, your application will save the results to a file on the desktop, use the file name you find appropriate that makes each file unique (e.g., the query number + the time).

Submission:

1. Directly to the folder titled Step 3 under the D2L Assignments tab, submit the file(s) you developed and one PDF structured as follows:
   1. On the first page, place you name and a short description of your idea (from Step1)
   2. On the second page, mention the type of data you used:
      i. did you use real data (yes or no), which tables and attributes are covered with such real data set.
      ii. did you use programming language to generate random data, which tables and attributes are covered with such random data.
      iii. did you use online tools to generate random data, which tables and attributes are covered with such data set.
   3. On the third page, one screenshot of your running application to show if you developed a GUI, or web application along with the menu of queries for user interaction. **Do not take screenshots with a camera.**
   4. On the following page(s), for each of the 8 queries, take screenshot(s) for the SQL query you developed as well as the output. **Do not take screenshots with a camera.**
   5. On the last page, list the edits/modifications/adjustments you made (if any) to the ER, Relational Schemas, or Queries you submitted in step2.

2. If you work in a team, just one member should submit the PDF file on D2L.

3. Check the due date on D2L. You can submit your assignment within 24 hours after this due date to be graded out of 75% of the assignment's grade. After this grace period, your late submission will not be accepted.