CPSC 431, Database and Application

Homework #3, Using MySQL

Purpose: Reinforce Basic Database Design, Creation, Manipulation and PHP Integration Concepts

You are a PHP and now a database developer continuing your work to develop a web application project that keeps track of players' statistics for our Cal State Fullerton Basketball team started in Homework #2. One of the functionalities offered by the application is for users to enter players' game data (playing times, scores, assists, rebounds etc.) after each game is played. In Homework #2, you persisted players' name, address, and game data information using text files on the server.

Here in Homework #3, you migrate from text files to a MySQL relational database. Just as you had two text files (TeamRoster.txt and Statistics.txt) in Homework #2, you will now have two tables in your relational database (TeamRoster and Statistics).

We also update the web application's home page here in Homework #3. We merge the Team Roster and Statistics tables into a single table populated with information from both database tables mentioned above. We also add the capability for a user to ask questions and receive answers about the team and players.

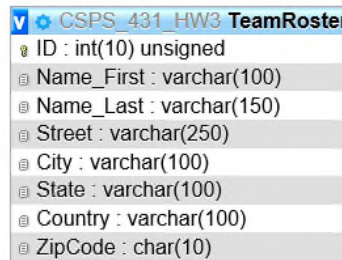In this assignment, you are to demonstrate your understanding of:

1) Setting up a MySQL database for use on a website, including
   a. Creating a database
   b. Setting up users and privileges
   c. Introducing the privilege system
   d. Creating database tables
   e. Creating indexes
   f. Choosing column types in MySQL

2) Integrating a database into your PHP scripts, including
   a. Querying a database from the web using the basic steps
   b. Setting up a connection
   c. Querying the database
   d. Retrieving the query results
   e. Disconnecting from the database

   f. Putting new information in the database
   g. Using prepared statements

3) The Structured Query Language (SQL) and its use in querying databases, including
   a. Inserting data into the database
   b. Retrieving data from the database
   c. Joining tables

4) The advantages of using a database instead of a flat file, such as
   a. faster access to data
   b. easily queried to extract sets of data that fit certain criteria
   c. concurrent access mitigation
   d. random access to your data.
   e. built-in privilege systems

by performing the task listed below and then submitting this homework's deliverables through TITANium by the due date.

Task 1: Design, create, and initialize your database schema. You are to write and submit the DDL script that does this. Name this script file *hw3_ddl.sql*. The following outline describes the contents:

```
1. drop   database if exists     <your database name>;
2. create database if not exists <your database name>;

3. drop user if exists <your database user's name>;
4. grant select, insert, delete, update, execute on ...

5. use <your database name>;

6. CREATE TABLE TeamRoster ...
7. INSERT INTO TeamRoster ...

8. CREATE TABLE Statistics ...
9. INSERT INTO Statistics ...
```

Limit your script to completing the 9 queries in this outline.

Your design should include:

1. Columns as shown

2. `ID` columns are primary keys and values are automatically assigned

3. `Statistics.Player` is a foreign key referencing `TeamRoster.ID`

4. If a player is deleted from `TeamRoster`, then all statistics about that player in `Statistics` shall also be deleted.

5. `TeamRoster.Name_Last` is a required value

6. `TeamRoster.ZipCode` must be validated against the following rules
   a. 5 digits, not all are zero and not all are nine,
   b. optionally followed by a hyphen and 4 digits, not all are zero and not all are nine.
   Hint: use the3 regular expression `(?!0{5})(?!9{5})\d{5}(-(?!0{4})(?!9{4})\d{4})?`

7. Accessing the column `Name_Last` and the pair of columns `Name_Last` & `Name_Last` occurs frequently and therefore should be indexed, and the combination of `Name_Last` & `Name_Last` should be unique

8. `Statistics.Player` is a required value

9. `Statistics.PlayingTimeMin`, `PlayingTimeSec`, `Points`, `Assists`, and `Rebounds` should default to 0.

10. `Statistics.PlayingTimeMin` must be in the range of 0 .. 40 (a game is 40 minutes long)

11. `Statistics.PlayingTimeSec` must be in the range of 0..59 (60 seconds in a minute)

12. Initialize the contents of your TeamRoster table as shown:

ID	Name_First	Name_Last	Street	City	State	Country	ZipCode
100	Donald	Duck	1313 S. Harbor Blvd.	Anaheim	CA	USA	92808-3232
101	Daisy	Duck	1180 Seven Seas Dr.	Lake Buena Vista	FL	USA	32830
102	Mickey	Mouse	1313 S. Harbor Blvd.	Anaheim	CA	USA	92808-3232
103	Pluto	Dog	1313 S. Harbor Blvd.	Anaheim	CA	USA	92808-3232
104	Scrooge	McDuck	1180 Seven Seas Dr.	Lake Buena Vista	FL	USA	32830
105	Huebert (Huey)	Duck	1110 Seven Seas Dr.	Lake Buena Vista	FL	USA	32830
106	Deuteronomy (Dewey)	Duck	1110 Seven Seas Dr.	Lake Buena Vista	FL	USA	32830
107	Louie	Duck	1110 Seven Seas Dr.	Lake Buena Vista	FL	USA	32830
108	Phooey	Duck	1-1 Maihama Urayasu	Chiba Prefecture	Disney Tokyo	Japan	*NULL*
109	Della	Duck	77700 Boulevard du Parc	Coupvray	Disney Paris	France	*NULL*

13. Initialize the contents of your Statistics table as shown:

ID	Player	PlayingTimeMin	PlayingTimeSec	Points	Assists	Rebounds
17	100	35	12	47	11	21
18	102	13	22	13	1	3
19	103	10	0	18	2	4
20	107	2	45	9	1	2
21	102	15	39	26	3	7
22	100	29	47	27	9	8

Notes

1. *You are expected to write this DDL script. DDL exported from MySQL (or IDEs running on top of MySQL, such as phpMyAdmin) will not be accepted. The first step in grading your homework is reviewing and executing this file. So consider, if this file is not accepted your entire project will be rejected.*

2. *Field validation is usually implemented using a CHECK constraint. Earlier versions of MySQL parse but ignore CHECK constraints. If your version of MySQL does not process CHECK constraints you will not be able to test this capability.   However, you are expected to code these constraints. My version of MySQL does process CHECK constraints, so I will be looking for them.*

3. *When using regular expressions, you may have to escape the backslash. That is, to use "\d" you may have to code it as "\\d".*

Task 2: Update the home page to merge the Team Roster and Statistics tables into a single table populated with information from both database tables mentioned above. Once your database is created and initialized, your homepage should look like this:

Description:
1. Every player is listed once sorted by last name and then first name
2. The player's address is display as shown
3. The Games Played column is the count of entries in the Statistics table for that player
4. The columns under Statistic Averages are just that. These are the average Time on Court, Points scored, Number of Assists and Number of Rebounds per game per player

Notes

1. You are to re-use your Address and PlayerStatistic classes from HW2. You may need to incorporate comments and fixes

2. Modify your processAddressUpdate.php and processStatisticUpdate.php files to insert data into your database instead of your text files.

3. Fill in the PHP blocks in home_page.php

4. Use a left join to find all the players with and without related statistics

5. Use the AVG aggerate function to find averages and the COUNT aggerate function to count the number of games played

- *While not technically correct, it's okay for this exercise to average the Statistics.PlayingTimeMin column independent of the seconds Statistics.PlayingTimeSec column*

6. Group and order by the player's first and last name to aggregate the averages

7. Construct Address and PlayerStatistic objects with averaged data retrieved from the database to validate and format information, such as Lastname, Firstname and min:sec formats.

Task 3: Run your program and take screen shots showing your table, and one or two more as you add records to the database. Archive (zip, tar, whatever) everything into a single file and submit:

- Your screenshots
- All your .php files.
    - Address.php
    - home_page.php
    - PlayerStatistic.php
    - processAddressUpdate.php
    - processStatisticUpdate.php
- You DDL Script
    - hw3_ddl.sql.

Note: you can point your browser to http://99.31.220.177/cpsc431/hw3/home_page.php to run my baseline point-of departure program. This might provide a sense of how it should look and feel.