CS482/502 Database Management Systems I

Project

Deadline for the first phase: 11:59pm Sept. 26, 2017
Deadline for the whole project: 11:59pm Oct. 31, 2017

Requirements

- You should form **a group that consists of 3 people** to work on this project. Each group just needs to submit one copy of your program.
- You need to finish the tasks stated in the following two phases.

Scenario

Assume that you get the following relational schemas and the constraints for the basketball team at NMSU.

- Player (ID: integer, LoginID: varchar(16), Name: varchar(64), Password: varchar(8), Birthday: date, Address: varchar(128), Email: varchar(32), PhoneNumber: char(10), PlayPos: varchar(16))
  Constraints: (1) Name should NOT be NULL. (2) PlayPos should only take values from {"point guard", "shooting guard", "small forward", "power forward", and "center"};
- Manager (ID: integer, LoginID: varchar(16), Name: varchar(64), Password: varchar(8), Birthday: date, Address: varchar(128), Email: varchar(32), PhoneNumber: char(10))
  Constraints: (1) LoginID and Name should NOT be NULL.
- Staff (ID: integer, Name: varchar(64), Birthday: date, Address: varchar(128), Email: varchar(32), PhoneNumber: char(10), Title: varchar(16))
  Constraints: (1) Name and Title should NOT be NULL.
- ManagerCertificate (ManagerID: integer, CertificateId: int, Certificate: BLOB)
  - Foreign key: ManagerID references Manager(ID)
- Stats (PlayerID: integer, Year: char(4), TotalPoints: integer, ASPG: integer)
  - Foreign key: PlayerID references Player(ID)

  Constraint: (1) TotalPoints and ASPG should be non-negative.
- Training (TrainingName: varchar(16), Instruction: varchar(256), TimePeriodInHour: integer)
  Constraint: (1) Instruction and TimePeriodInHour should NOT be NULL.
- AssignTraining (PlayerID: integer, ManagerID: integer, TrainingName: varchar(16))
  - Foreign key: PlayerID references Player(ID)
  - Foreign key: ManagerID references Manager(ID)
  - Foreign key: TrainingName references Training(TrainingName)
- Game (GameID: integer, Date: date, Result: varchar(16), PlayingVenue: varchar(256), OpponentTeam: varchar(32))
  Constraint: (1) Date, Result, PlayingVenue, and OpponentTeam should NOT be NULL. (2) Result should only take values from {"Win", "Lose", "Tie"};
- Play (PlayerID: integer, GameID: integer)
  - Foreign key: PlayerID references Player(ID)
  - Foreign key: GameID references Game(GameID)

Phase 1 (20 points)

- (15 points) Implement the database using MySQL. Write proper SQL statements to create the required tables in the database and to set all the required constraints (e.g., primary key, foreign keys, and value constraints). Put all the SQL statements into "CreateDB.sql".
- (5 points) Insert 5 records into each table. Put all the insertion SQL statements into "DataDB.sql".

Note: All your submitted SQL statements should run correctly (syntactically and semantically) at the MySQL server dbclass.cs.nmsu.edu. If an SQL statement returns any error, that SQL statement is graded as **ZERO** (i.e., no partial score).

Phase 2 (80 points)

Create a small web application using PHP to manage the above database.
Note: You have some flexibilities to design this web application. If you are not clear on the requirements, please discuss with the TA.

Your program should contain a main file named index.php. From this main file, a user should be able to go to the following sub-systems. In your project, you are required to implement **three sub-systems**.

(i) (25 points) Player management sub-system. In this sub-system, a player should be able to

   A. (5 points) **Request a new account**: if this player does not have an account yet. The requesting account interface should be able to let the player (1) request a login id and password and (2) reset his/her password for a given login id.

   B. (2 points) **Log in**: if a player already has an account, he/she should be able to login using this interface.

   C. (6 points) **View:** After a player logs in, this player should be able to view his/her information, i.e., the system should show the information in the *Player* relation and other related information including player stats, assigned training. Please design your interface in a user-friendly manner.
   − In viewing his/her information, the player should be able to see his/her trainings and who set the trainings. However, he/she cannot change the training.

   D. (10 points) **Edit**: After a player logs in, the player should be able to edit his/her information. Please design your interface in a user-friendly manner.

   E. (2 points) **Logout**: a player should be able to logout.

(ii) (55 points) Manager administration sub-system. In this sub-system, a manager should be able to

   A. (5 points) **Create a new account** if this manager does not have an account yet. The creating account interface should ask the manager to input his/her login ID, password, name, email, and rest of the information in the *Manager* table. This function should check the *Manager* relation and the *Player* relation to make sure that the login ID does not exist.

   B. (2 points) **Login**: if a manager has an account already, he/she can login and use the following functionalities in this sub-system.

   C. (4 points) **View**: After a manager logs in, the manager should be able to *view* his/her information which include name, email, and rest of the information in the *Manager* table. The manager should also be able to download **one** or **several** of his/her certificates. Please design your interface in a user-friendly manner.

D. (7 points) **Edit:** After a manager logs in, the manager should be able to *reset* his/her password, *edit* his/her information which include name, email, and all the other information in the *Manager* relation. The manager should also be able to upload a new certificate or re-upload an existing certificate (to replace an old one). Please design your interface in a user-friendly manner.

E. (5 points) **View Players**: After a manager logs in, he/she should be able to list all the players' information and see all the detailed information about each player. −− In listing the players, the player should be ordered according to their names.

F. (8 points) **View, add, update, and delete trainings**: a manager should be able to see all the type of trainings, add new training, update a training, or delete an existing training. When deleting a training, the training can be deleted only if it is not assigned to any existing player.

G. (6 points) **View and assign trainings to players**: a manager should be able to see all the trainings assigned to a player and see who assigned it. A manager should also be able to add new trainings for a player.

H. (8 points) **View, add, and update games**: a manager should be able to see all the past and upcoming games, add new games, or update.

I. (5 points) **Assign players to games**: a manager should be able to assign and re-assign players to existing games.

J. (4 points) **Approve players' log-in requests**: a manager can check and approve any request from a player for a new login ID and password, or reset a password for a player with an existing login ID.

K. (1 point) **Logout**: a manager should be able to logout.

Handing in & Grading:

- You have to hand in your assignment electronically (through Canvas). Printed copies are not accepted. Combine and compress all the source files (sql, php, and other related files) to a .zip file with name proj-(your banner ids). See course syllabus for policies on late submission and plagiarism.
- Demonstration. You should demonstrate your project to your TA within **one week after the due date of this project** (inclusive). We will create a Google Sheet to decide the final times for each group's demonstration.