CECS 323 Term Project 3

Introduction

Please consult the general requirements for Term projects here.

Business Rules

You are running a coaching clinic.

1. For each player, we track their last name, their first name, their date of birth, and the date that they joined the clinic. We do not assume that the combination of first and last name is necessarily unique.
2. Several of our players have multiple talents, so they receive coaching in more than one sport. Each sport can have any number of players receiving coaching in that sport.
3. For each coach and assistant coach, we keep track of their first and last name, but we do not assume that the combination of first and last name is necessarily unique.
4. Each player has exactly one coach assigned them for each sport that we coach them in.
5. Both the coaches and the assistant coaches have one or more sports that they coach in, and a given sport could have several coaches who coach in that sport.
6. A given player will only be coached by a given coach for a given sport if that coach coaches in that sport.
7. There are assistant coaches as well, but they do not have players assigned to them.
8. Each **assistant** coach has one and only one coach as a mentor.

What to Turn In

All the deliverables required for term projects in general

Queries and sample data for the following

1. List each assistant coach's first name, last name, their mentor's first name, last name, and any sports that the two of them have in common. One row of output for each assistant coach[1]. List the assistant coaches sorted by last name, then their first name.
2. For each player, list their first name and last name, each sport that we coach them in, and the coach's first name and last name. One row of output for each player. List them by their last name, then their first name.
3. For each player Talent, list the player's age, the name of the sport, list all of the coaches who coach that sport, and the coach who is coaching that player in that sport.
4. Find the coach who is coaching the most players, and list the names of the players that they coach.

[1] You will need to use the MySQL group concat for this.

Denormalization

Remember that denormalization is a conscious decision to step away from a fully normalized model during the implementation phase. Usually, the design team makes this decision to improve the application's performance or to make navigation through the data easier. For this project, select **just one** place in your UML class model where you will denormalize. There are several types of denormalization that you can choose from for this assignment:

- Categorization roll up
    - All the associations and attributes of all the categories move into the generic parent.
    - You need to write triggers to make sure that any row of that generic parent does not have some associations/attributes from more than one of the original categories.
    - And you need triggers to ensure that if a given instance of the generic parent falls into a category, that **all** the appropriate associations/attributes have values.
- Migrating non-key attributes from one or more "ancestor classes" into a "descendant class".
    - Make sure that no one can change those migrated attributes/associations in the child.
    - Make sure that changes to those migrated attributes/associations in the parent are propagated as necessary so that they are in sync.
- Multi-valued attribute
    - Maintain both the junction table **as well as** the multi-valued attribute.
    - Only allow changes to the multi-valued attribute through inserts/deletes to the **junction table**.
- Merge the parent and child class into one table.
    - You will need to implement a trigger to make sure that the repeated values are consistent. For instance, if you merge State and Contact into one table, and the new table has state name and state code, then you must make sure that all rows in that table with a given state code also have the same state name.

The denormalization will **not** be reflected in the UML class diagram, but it **will** be reflected in the relation scheme diagram. Please be sure to put sufficient text in the relation scheme diagram to indicate what sort of denormalization you have opted and what you are denormalizing.

Trigger Code for Denormalization

Denormalization generally introduces redundancy, or a requirement for constraints that we can no longer capture directly in the database structure. Write the necessary trigger code to either make sure that the redundant data copies never get out of sync with each other, or that your rolled up categorizations do not violate the semantics of the categorization.

Triggers

Write triggers to ensure the following:

1. No player can receive coaching in more than three sports.
2. No coach can mentor more than three assistant coaches.
3. Players must be at least ten years of age before they can join our coaching clinic.