Project 1: VolleyballDB

CMPE 321, Introduction to Database Systems, Spring 2024

Due: 18.03.2024 Monday, 23:59

1 Introduction

Last year, our national volleyball team, which became the European champion on the 100th anniversary of the Republic of Türkiye, made us all proud. Of course, the best team in the world cannot stay away from technological developments and must systematically manage its players' data. A well-designed database system is essential to ensure the reliability, consistency, and security of such systems. Turkish Volleyball Federation needs your help with this database design!

Your task in this project is to create the conceptual design and logical design for a database based on the given requirements, taking into account various constraints that may arise during the implementation phase. These constraints **may** include issues such as conflicting contract periods of coaches or players playing in wrong positions. However, we know that not every constraint can be included in the ER diagram nor in the logical design. In this case, you are expected to include and discuss them at the end of the PDF files for the corresponding parts. In a follow-up project this semester, you will implement the database you designed in this project and an application program that uses that database. By the end of these two projects, you will have gained valuable experience in designing and implementing databases and in the practical application of SQL, while also understanding the importance of data integrity and system reliability in the real world.

2 Project Description

In this project, you will design a match and team management system for Türkiye's national volleyball federation. You will manage different types of teams, such as: A-team or U19, U21 etc., along with players, coaches, match sessions and TV channels for which the teams have signed a contract this season. You will begin with a detailed description of the content. Then you will need to systematically go through parts of the standard database design process as you learned about in class, including conceptual and logical design. Our database should contain the following information:

1. **User** includes the following attributes; *username, password, name, surname.* Each user has a unique username. Also each user is either a player or a coach or a jury.

(a) **Players** additionally have attributes of *date_of_birth*, *height*, *weight*, *position_list* and *team_list.* A player must be registered with at least one team.

A player can be registered to different types of teams, but of course cannot play in matches where there are time conflicts. In addition, a player can play in different positions. There must be at least one position in which the player can play. However, in a match, he/she can only play in a position within the positions he/she can play (is registered). There are no age-team restrictions.

(b) **Coaches** additionally have *nationality* information. Each coach **must** have **only one** *nationality.* That is, dual citizenship is not allowed, and being in an agreement with more than one team at a time is not possible for coaches. Also, each team **must** be directed by a unique coach.

(c) **Juries** additionally have *nationality* information. Each jury **must** have **only one** *nationality.* That is, dual citizenship is not allowed. They rate match sessions. Every match has an assigned jury, and that assigned jury rates a match session only once. Juries can not rate matches that they are not assigned to. They can not edit/change their ratings.

2. **Position** includes the following attributes: *position_ID, position_name.* position_id must be unique. Normally, a player can play in different positions, but in a match, he/she has to play in just one position.

3. **Team** includes the following attributes: *team_ID, team_name, coach_username, contract_start, contract_finish, channel_ID.* team_ID must be unique, and each channel_ID is associated with a unique channel_name. Each team has an agreement with just one TV channel. Each team is led by just one coach. contract_start and contract_finish are used for coach-team agreements. A coach cannot direct more than one team at the same time. A TV channel can have agreements with more than one team.

4. **Match Sessions** include the following attributes: *session_ID, team_ID, played_player_username_list, stadium_id, stadium_name, stadium_country, time_slot, date, assigned_jury_username, rating.* Each stadium_ID is associated with a unique stadium_name. The session_ID must be unique. We are only interested in the teams of our federation. So no opponent information or details are necessary.

    - No two match sessions can overlap, both in terms of stadium and playing time.

    - There are four time slots for each day.

    - The duration of the match is closely related to the time slots. The time_slot attribute determines the starting time of the match, and the end time is determined by the duration. Each match has a duration of 2 time slots. (For example, if a match starts at time slot 2 and has a duration of 2, then the stadium is reserved for that match during the following time slots: [2, 3]).

- Each stadium_id corresponds to a physical location. Hence, stadium_name and stadium_country depend solely on the stadium_id.
- The entire team does NOT have to be in the "player list" for a match session.
- A jury can rate the same match only once.
- Each match will be rated by a jury assigned to that match.

2.1 Sample Data

The tables are given as examples to provide sample data for you. You can find the ".xslx" files attached in the Moodle page of this assignment. Please note that you can and have to make your own design (e.g., design and create your own tables) according to the description and constraints specified in the previous section. You can mix, divide or combine any tables you find, as long as the information is not lost.

2.2 Part 1: Conceptual Database Design

Your task in Part 1 is to perform the Conceptual Database Design (or ER Design) – draw ER diagrams to capture all the information, following the approach described in the lectures. While there are many ER model variants, for this project, we expect you to use the ER notation from the textbook and lectures.

To receive full points for this part, you need to identify all the entity sets and relationship sets in a reasonable way. We expect there to be multiple correct solutions since the ER design is subjective. Your goal should be to capture the given information and constraints reasonably. For the entity set names, relationship set names, and attribute names that you will be using in your ER diagram, you can use the ones we have provided in Section 2.1. You can use underscores, spaces, numbers, uppercase, or lowercase letters to construct those names. It is required to use the features of ER modeling that you have learned in the lectures, including participation constraints, key constraints, weak entities, class hierarchy, and aggregation whenever necessary. We provide you with sample data in Section 2.1, since these may help you understand the problem and the requirements better.

You must use computer-based / online drawing tools such as Lucidchart and diagrams.net. Handcrafted diagrams will not be accepted.

Include a discussion at the end of the report for this part, indicating the constrains that you were unable to represent in your ER design.

2.3 Part 2: Logical Database Design: Mapping ER Design to the Relational Model

For the second part of the project, your task is to convert the ER diagrams into relational tables, based on the set of simple rules as described in the textbook and in the lectures. You are required to write SQL DDL statements that create the relational tables of your database. You should specify all the integrity constraints such as primary key, foreign key, Unique, and NOT NULL. Please also define the general constraints using the CHECK construct.

Include a discussion at the end of the report for this part, indicating the constrains that you were unable to represent in your relational design.

Note: For this part, you must use MySQL Server, others are not allowed.

3 Submission

This project can be implemented either individually or as a team of two people. You are free to change teams in the upcoming projects.

You are required to submit reports for Part 1 and Part 2 in PDF format. Place your two PDF files in a folder named with the names and surnames of the team members separated by underscores (e.g. Egemen_Isguder_Arzucan_Ozgur). Zip the folder for submission and name the .zip file with the same name. Submit the .zip file through Moodle. Only one of the team members should make the submission. Please, do not make multiple submissions.

The files must follow the specific naming procedure:

- Student1Name_Student1Surname_Student2Name_Student2Surname.zip
    - Student1Name_Student1Surname_Student2Name_Student2Surname (This is a folder)
        - part1.pdf
        - part2.pdf

4 Late Submission Policy

We will accept late submissions, however;

- One day late (even one minute will be considered a day late) would mean -30 points penalty.
- Two days late (even one minute and a day will be considered two days late) would mean -60 points penalty.
- Moodle will close after two days. No other submission method will be accepted.

5 Academic Honesty

Please read carefully the academic honesty part of the syllabus as we give utmost importance to academic honesty.