Introduction to Database Systems Exercise 5: Database Design

Fall 2021 – Björn Þór Jónsson

The goal of this exercise is to get you started on ER design. Note that this case study is too extensive for you to finish within the two-hour exercise class! But it is a good exercise, so it may be worthwhile to complete more of this exercise than you can do in the exercise session alone.

Case Description

Disclaimer: The description of this project is entirely fictional. I am not aware of any movement to combine these two competitive sports into one.

A new sports federation with plans to conquer the world has approached you to design their new database. This federation, the Program-Dance Federation, or PDF for short, is planning to establish a new sport, the Program-Dance. This sport combines programming and dancing into a single sport, which gracefully combines one of the most demanding mental activities (dancing) with one of the most demanding physical activities (programming – why else would you be sweating?) into a single, hugely exciting event. During these PDF tournaments, dance performance will be monitored using a game computer, while an automated code testing system will be used to judge the programming skills.

The PDF have already hired consultants to interview potential users, coming up with the following requirements for the database, but they need you to design their database by creating an ER-diagram and transforming it into SQL tables.

Notes

The database design should follow the requirements definition strictly. For example, ID columns should only be used where specifically mentioned. Where ID columns are mentioned, however, you can assume that they are a key.

There are requirements, however, which cannot be specified in the ER-diagram. Some of these can be addressed in the database creation script, while others may require more advanced techniques, such as triggers. Part of the exercise is to realize and note such restrictions on the ER-diagram. We are not asking you to create triggers in this exercise.

Requirements

The following requirements were determined through interviews with PDF:

1. The athletes are called danes (short for dance-nerds). For each dane, an ID must be stored, along with their name, nationality, gender and degrees. Gender can be stored as either 'M' or 'F'.
2. Danes can have multiple degrees, but for each degree of a particular dane the database must store the level (e.g., "bachelors", "masters" or "doctoral"), subject, institution awarding and year awarded. Each dane can only have one degree per subject and level.
3. Clubs are identified by their name and nationality (you could thus, e.g., have two clubs with the same name in different countries, but not within the same country). To simplify the database design, however, an ID must also be stored.
4. Each dane can be a member of multiple clubs. For each membership, the start and end dates must be registered. Furthermore, a dane can leave and rejoin a club, in which case all the different registration intervals must be stored.
5. Competitions in Program-Dance take place as tournaments. Tournaments are held in a specific venue at a specific date, and are always held by one specific club. For each tournament, a unique ID is stored, as well as the tournament's name.
6. PDF employees have an ID and a name. A single PDF employee always monitors each tournament.
7. A tournament consists of multiple leagues, which may or may not be simultaneous. Each league has an ID, time (they all run on a single date which is stored with the tournament), duration, number, name, and gender. Note that the combination of league number and gender is unique within each tournament.
8. A dane can participate in one or more leagues of a tournament. At the end of the league, the dane is awarded a final rank within the league. Note that ties are allowed in the final ranking of danes within the league.
9. For leagues, the gender can be 'M', 'F' or 'X' (for mixed). A dane should only be able to participate in leagues of the same gender as the dane, or mixed leagues.
10. A club (later) pays for the participation of a dane in a league. The fee is determined using a complex formula based on club and dane rankings (rankings are discussed below), so the fee amount must be stored. (We can assume that the club will prevent danes from registering in multiple simultaneous leagues, but the database design should not prevent it.)
11. Each league consists of multiple problems. Each problem can be used in multiple leagues, including leagues of different tournaments. For each problem, we must store ID and name. Both are unique across all problems.

12. For each problem, we must also store the name of the dance step program involved and the text of the programming assignment. Each problem is written by one or more PDF employees.
13. For each problem, we must also store multiple test cases. Each test case has a name and an order indicator (both are distinct but for one particular problem), as well as (optional) input text and (mandatory) output text.
14. A dane solves a problem as part of the participation in a league. (The same problem could be solved by the same dane multiple times.) The database must store the resulting dance score and the resulting programming score; both are integers on a scale from 0 to 100. These scores are subsequently used to determine the rank of the dane within the league, as mentioned above).
15. The PDF maintains many rankings, which are updated once each year. Each ranking has an ID, name and start year. The names of the rankings are unique.
16. Each ranking is either an "individual" ranking (exclusive) or a "club" ranking. For individual rankings, the gender and age group must be stored. For club rankings, gender is not required, but ranking can be national (in which case the nationality of the ranking is stored) or (again, this is an exclusive or) global.
17. A dane is represented in exactly one individual ranking at any time, and the current rank within the ranking is stored.
18. A club is represented in exactly two rankings at each time; a national club ranking and the global club ranking. As with danes, the current rank must be stored.
19. Sponsors have an ID, name and nationality. Each sponsor can only sponsor one club. For each such sponsorship the text of the latest deal and the total accumulated amount must be stored.
20. Each sponsor can sponsor multiple danes, however, but here it is sufficient to store the total accumulated amount of the sponsorship.

Outcomes

If this was an actual project, we would be looking for three outcomes:

a) An ER-diagram. The diagram must follow the notation presented in the PDBM book, as amended in lectures.

b) A text file with SQL DDL commands to create the corresponding database tables.

c) A listing of a) assumptions made that were not clearly specified in the project description; b) design decisions made when converting the ER-diagram to tables; and c) requirements which are neither represented in the ER-diagram nor the SQL DDL commands.