

Jonathan Flessner
16-March-2014
CS 275 Final Project

General Outline

My database project is comprised of tables to represent some aspects of Major League Soccer (MLS). MLS is the professional soccer league in the United States made up of 19 teams from around the country, with 3 more added in the coming seasons. With 19 teams, up to 30 players on each team, other free agent players, coaches, and the stadiums the teams use, there is a wealth of data to be categorized and analyzed. The league has been around since 1996 and is rapidly gaining popularity, especially among the younger demographic. My database has 4 main entities: teams, players, coaches and stadiums. I have two other tables that track the relationships between coaches/players and teams. Both players and coaches often have been associated with more than one team. So these tables track their former teams.

Database Outline

The database has 4 tables: teams, players, coaches, stadiums, playerTeams, and coachTeams. The playerTeams and coachTeams tables are many to many relationship tables.

The players table holds the name of the player, their date of birth, and their current team. The name of the player must be unique. In the case of two players with the same names, a middle initial or other identifier should be used. The primary key of the table is an auto incrementing id. The players team serves as a foreign key referencing a team name in the teams table. This means that the team that the player is being inserted into must already exist in the teams table. If a player is a free agent (is not currently under contract with an MLS side), they should be added to the FA team in the table.

The coaches table is nearly identical in every way to the players table with the obvious exception that it holds coaches instead of players. Also, importantly, while there may be many players on one team, there is only one coach per team. Because of that, the team field in coaches is unique while it is not in the players table. It should also be noted that the same person can be in the players and coaches table. One example is Pablo Mastroeni, who is the current head coach of the Colorado Rapids, but who played for the Los Angeles Galaxy last season. He is currently set as a free agent in the players table and also as the coach of the Rapids. It is up to the database managers' discretion when to remove a player from the players table.

The teams table holds the entire name of the team (examples: Seattle Sounders, Real Salt Lake, Chivas USA). The reasoning for this is that unlike in other American sports leagues where each team has a clear city and mascot, the line is blurred with many MLS sides (see: Sporting KC, Real Salt Lake, Chivas USA). The database holds the teams division (also called conference) as either East or West. This value is allowed to be null to allow for the addition of teams like New York City FC, Orlando City SC and the Miami team, which have been officially announced, but are not yet playing in the league. The year a team was founded and the year the team joined MLS are stored as four digit ints that are allowed to be null. The name of the teams' home stadium is required. The number or titles a team has won, US Open Cups (crowns the best team in the United States, includes lower divisions and semi-pro sides, not just MLS teams), MLS Cups (given for winning the playoffs), and Supporters Shields (given to the regular

season champion) are all tracked as two digit ints. Teams must have a unique name. The primary key for this table is an auto-incrementing id.

The stadiums table gives information over the stadiums that the teams play in. The primary key is an auto-incrementing id. The name, city and home team of the stadium are all required. Do note that a stadium can have two teams call it home. A good example is the LA Galaxy and Chivas USA who both play at the StubHub Center in Carson, CA. For this reason the stadium name is not unique so that you can create multiple instances of the same stadium. However, the team is unique as there shouldn't be multiple instances of the same team in the stadium table. The team attribute in the stadiums table is a foreign key that references the name of a team in the teams table. Because of this, you must always add the team first, before adding the stadium. The average attendance for the last season is also included in the table as an int.

Lastly, there are the two many to many tables, playerTeams and coachTeams, these tables are essentially identical, again with the obvious exception that one stores the former teams for players and the other stores former teams of coaches. Each table has a player/coach id (pid,cid respectively) and a team id (tid) that combined serve as the primary key. Each column is a foreign key that references the id of the player/coach in their respective tables. The team id references the team id in the teams table as a foreign key. Because of all that, there are no nulls allowed in these tables and the player/coach and team all need to exist before being entered into these tables. Do note that while this table is designed to only hold the former teams of a player/coach, it is still possible to enter in the player/coaches current team. The reason for this is that a current team can also be a former team. See the example of Dwayne De Rosario who is currently with Toronto FC who he also played for from 2009-2011. Therefore, it is up to the manager of the database to ensure accurate information.