COMP421: Database Systems
Project: Database Design and Data Modelling

Part 1: E/R Schema and Relational Translation
Due: Feb 01, 12:00 pm

Justin Sun	260707684
Dominic Wener	260871052

Specification

The database should keep information about ticket purchases from clients for specific matches. Unpurchased tickets must be shown as available, while sold tickets must still be visible but not available for purchase. Tickets sold are for designated seats of a participating stadium during a given date, which will be linked to the unique match a ticket is sold for. The stadium linked to a valid ticket has a visible location, unique name and capacity. Seats that can have assigned tickets to them will have a unique seat number within its stadium and will have a section number. Tickets have a variable price that could be dependent on the combination of the match and seat it is assigned for. Some tickets may end up not being sold.

Clients may purchase any number of available tickets that they want. Clients may create an account in the application used to sell the tickets, for which an email, a name and a password is needed. If a Client does not desire an account, they can buy tickets as a Guest. Only a valid email would be required for the transaction. Clients may not buy tickets after creating an account. Purchased tickets may be for any of the upcoming matches and the desired tickets may be for any of the valid seats in the stadium. A client may purchase multiple tickets for different seats for the same match.

Assumptions

- It is possible that the *name* of a player, a coach or a referee is not unique. These name attributes (cname, pname, rname) should not be NULL to ensure a minimal amount of interesting data appears when users query them.
- Although player numbers should be unique within a team, they will not be unique across multiple teams.
- Special roles for members of a given Team should be considered unique, such as “Head Coaches”, “Goalies”, etc. However, they will have substitutes in case they need to be replaced under special circumstances, so the attribute fields could possibly not be unique in a team.
- The status attribute of a team keeps track of their performance, whether they are still competing or keeps the value of when the team was eliminated (i.e. either shows the current match round the team is competing in or shows the round when the team was eliminated). For example, if a team is eliminated in the quarter-finals, status would show ‘Quarter-Finals’.

- Referees can have any role in any match.

- We assume Stadiums that are allowed to host Matches and are tracked in our database will all have unique names (attribute sname is unique).

- National associations must "oversee" ONE team and Teams must be overseen by ONE Association (ONE-to-ONE Key constraints).

- WEB URLs are guaranteed to be unique (if valid) based on the nature of their purpose. Thus, we can safely use them as primary keys.

- The *score* attribute of Matches assumes the assigned relationships between matches and teams will be as follows: "Team1 - Team2". Therefore, the score would have the same format: "Team1_Score - Team2_score".

- A Match doesn't initially have assigned tickets, until tickets are created and posted for sales. A Match could also remain without any assigned tickets given extraordinary circumstances. However, we should assume they will usually have assigned tickets.

- Clients can create accounts to the database application that will be used to interface ticket sales. They would thus exist in the database as clients, even if they do not purchase tickets. If they do purchase tickets, then those tickets will be associated with the client account.

- Clients that buy tickets as "Guests" (i.e. without creating an account) will exist in the database with only an email, other attributes will be NULL. Tickets will be linked to the client email.

- By the nature of email addresses, the Email attributes of Clients will be unique values, assuming the email given by the Client is a valid email. We can assume the application interfacing sales will do some kind of email validation before a purchase or an account creation.

- There should only be one head referee per unique Match instance.

- The database only keeps track of Players, Coaches, Referees and Clients and doesn't add to the database the other types of people involved in the proper development of the events such as administrators, coordinators, etc.

Restrictions

- Matches that have the round attribute value of "Group" can end in a tie, thus it is possible that no goals were scored in the entire match (i.e. final score of 0 to 0). In this case, the "Owned" relationship between Matches and Goals captures the correct restrictions of a match having 0 to many goals (regular line). However, matches from the knockout type rounds cannot end in a tie, thus there has to be at least one goal scored in the game. The "Owned" relationship between Matches and Goals doesn't capture the appropriate restrictions anymore, of a match having at least 1 to many goals (participation constraint).

- In the context of the application requirements, clients can create an account to purchase tickets or can buy tickets without an account (i.e. clients can purchase as "Guests"). The ER Schema doesn't capture the idea of client accounts, but the existence of an account can be inferred by looking at the attribute values of a Client entity. In other words, only the email attribute is required NOT NULL, while name and password can be NULL if a Client doesn't have an account.

- We assume the application will have many more features implemented for browsing and purchasing, such as a "shopping cart" for bulk purchases and credit card authentication. However, these features are not captured as required data in the ER schema.
- The number of teams in a Group is upper bounded by 4, which cannot be shown in the participation relation between Groups and Teams. This limit should be enforced given the rules of the tournament. Similarly, the number of Players per team, the number of seats per stadium (and consequently the number of tickets per match), the number of Matches and the number of Groups
- From the required structure of a valid match in the tournament, exactly two teams must be assigned to each unique scheduled match. The data captured in the "Scheduled" relation assumes (i.e. is dependent) on respecting that each mid value should appear exactly twice in the "Scheduled" relation table in the database. This restriction cannot be captured in the participation constraint, although it does capture the requirement that a match cannot have no teams scheduled to play in it.

Data Duplications/Redundancies

Goal differentials (Attribute)

Count of goals scored based on the relationships with Matches entities – The attribute needs to be updated after every match, adding the count of goals for the team and subtracting the count of goals scored against. This count could be found by looking at the total score of each match a team participates in, but would be complex to fetch this information for each query, thus the attribute created for the Teams entities.

Status (Attribute)

By looking at all the matches, we could deduce the rankings and the current status of a team in the tournament. For the sake of the potential abundant queries looking to see how teams are doing in the tournament, a redundant tournament status attribute is added to the Teams entity set to track this information.

Scorer Name (Attribute)

Attribute to the "Goals" entity set makes for easier queries about: players that scored goals in a specific match; total goals for a player; total goals for a team. By using the relationship "Scored-by" between Goals and Players, this information could be deduced. However, for simplicity of queries about detailed match information, this redundant attribute is created.

Groups (Entity Set)

As we can expect many queries about the standings during the group round, the Group entity set was added. The relationship "Grouped" from this set to Teams captures the points each team has with regards to the group they belong to. A "Group" attribute would be possible in the Teams entity set, but would require queries to trace back to many Match entities to deduce the standings of the group stage.

Relational Translation

Teams(country, status, goal_differential, web_URL, gname)

web_URL foreign key referencing NationalAssociation

web_URL NOT NULL

gname foreign key referencing Groups

gname NOT NULL

Grouped(country, gname, points)

country foreign key referencing Teams

gname foreign key referencing Groups

Groups(gname)

National Associations(web_URL, aname, country)

country foreign key referencing Teams

country NOT NULL

Players(pid, pname, number, position, DOB, country)

country foreign key referencing Teams

country NOT NULL

Coaches(cid, cname, DOB, role, country)

country foreign key referencing Teams

country NOT NULL

Referees(rid, rname, country, experience)

Stadiums(sname, capacity, location)

Matches(mid, match_length, score, start_time, round, date, sname)

sname foreign key referencing Stadiums

sname NOT NULL

Scheduled(country, mid)

country foreign key referencing Teams

mid foreign key referencing Matches

Refereed(rid, mid, role)

rid foreign key referencing Referees

mid foreign key referencing Matches

Played(pid, mid, y_cards, r_card, specific_position, time_in, time_out)

pid foreign key referencing Players

mid foreign key referencing Matches

Goals(gid, pid, mid, penalty, occurrence_order, scorer_name, time)

pid foreign key referencing Players

pid NOT NULL

mid foreign key referencing Matches

mid NOT NULL

Seats(seatNumber, sname, sectionNumber)

sname foreign key referencing Stadiums

sname NOT NULL

Clients(email, name, password)

Tickets(date, seatNumber, sname, mid, price, purchase_status)

seatNumber foreign key referencing Seat

sname foreign key referencing Stadiums

mid foreign key referencing Matches

Buys(email, date, seatNumber, sname, mid)

email foreign key referencing Client

date foreign key referencing Tickets

seatNumber foreign key referencing Seat

sname foreign key referencing Stadiums

mid foreign key referencing Matches

*Note: No redundancies can be removed by combining relations