CT230 DATABASE SYSTEMS

PROBLEM SHEET 4: SAMPLE SOLUTIONS

REQUIREMENTS:

A database is to be created for a set of *underage football competitions* to keep track of players, teams, matches and results.

A number of competitions are run each year per age category and per gender. Associated with each competition is a unique ID, the name of the competition, the gender group (boys or girls), the age group (U12, U13, etc.) and the level (premiership, championship, division 1, etc.). A competition comprises matches between teams who play each over the course of the competition. A team can only be associated with one competition.

Each team has a unique ID, name, coach, gender type, age group, and home location (a pitch). Associated with each team are players who only play for one team. Details stored on each player are name, address, gender, date of birth, parent/guardian details (name and mobile) and any associated medical conditions. One player per team is chosen as the captain of the team.

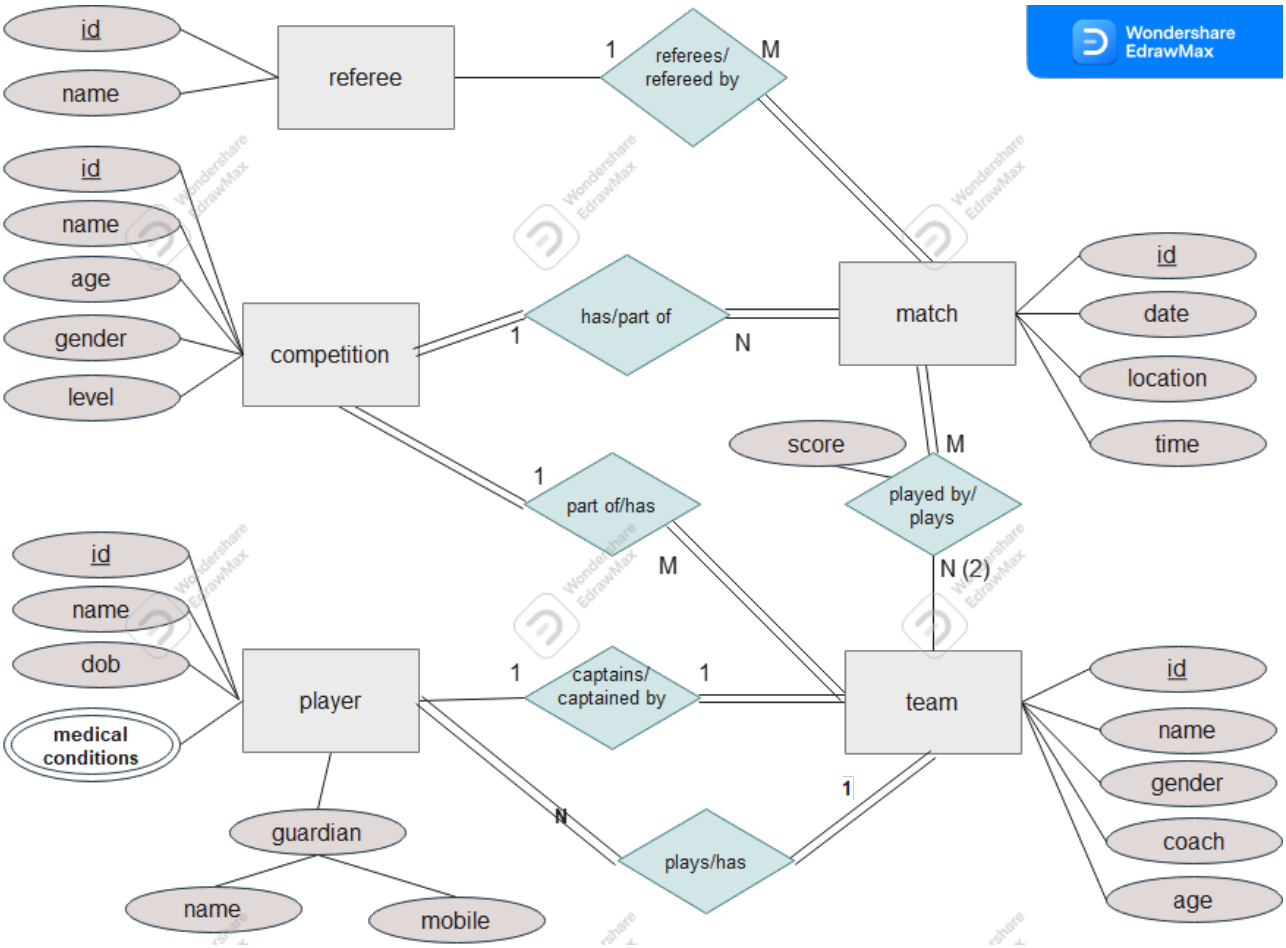
Matches, which have a unique ID, are played by two teams. Associated with each match is the date, time, location of the match and the score of each team involved in the match.

Also associated with each match is a referee. Details stored on referees are the referee ID and name. One referee will referee many matches over the course of a competition.

TASKS:

1. Identify all entities from the description given, the primary keys associated with the entities and any other attributes of the entities.
2. Identify the relationships that exist between entities.
3. Using Chen's notation, draw the entities and relationships from Steps 1 and 2, by hand or using MS Visio or other drawing tool (not Word or PowerPoint).
4. Add the cardinalities to the relationships and check for any total or partial participation.

A. Sample Solution



Entities: referee, match, team, competition, player

Assumptions:

All players play for a team

Not all teams will always play a match

A score is associated with the relationship between a team and a match

All matches must have a referee, teams who are to play and teams must be part of the competition

Sample Solution for Part B: Mapping to Tables

Step 1. For each entity create a table R that includes all the **simple** attributes of the entity.

Step 2. For strong entities, choose a key attribute as primary key of the table.

5 strong entites – become 5 tables – picking better names for PK:

```
referee(refID, name

match(matchID, date, time, location

competition(compID, name, ageCategory

team(teamID, teamName, teamGender, teamAgeGroup, coachName,

player(playerID, name, dob, guardianName, guardianMobile
```

Step 3. not needed (no weak entities)

Step 4. For each binary 1:1 relationship, identify entities S and T that participate in relation. If applicable, choose the entity that has total participation in the relation. Include as foreign key in this table the primary key of other relation. Include any attributes of the relationship as attributes of chosen table.

1:1 relationship is "captains" with total participation on "team" side => put PK from player as FK in team – call the FK captainID which is a FK to player(playerID) to give:

```
team(teamID, teamName, teamGender, teamAgeGroup, coachName, captainID
```

Step 5. For each binary 1:N relationship, identify the table S that represents the N-side and T the table that represents the 1-side. Include as a foreign key in S the primary key of table T such that each entity on the N-side is related to at most one entity instance on the 1-side. Include any attributes of the relationship as attributes of S

There are four 1:N relationships:

- referee <referees> match => Include as a foreign key in match the primary key of table referee (refID):

  ```
  match(matchID, date, time, location, refID
  ```

- competition <has> match => Include as a foreign key in match the primary key of table competition

  ```
  match(matchID, date, time, location, refID, compID,
  ```

- competition <has> team => Include as a foreign key in team the primary key of table competition

  team(teamID, teamName, teamGender, teamAgeGroup, coachName, captainID, compID,

- player <plays> team => Include as a foreign key in player the primary key of table team

  player(playerID, name, dob, guardianName, guardianMobile, teamID

Step 6. For each M:N relationship, create a new table S to represent the relationship. Include as foreign key attributes in S the primary keys of the tables that represent the participating entity types – their combination will form the primary key of S. Also include in S any attributes of the relationship.

There is one M:N relationship between "team" and "match" and there is an attribute of the relationship => create new table and add primary keys of team and match to new table which become primary key and also are FKs back to their respective tables. Add score as attribute of table.

teamMatch(matchID, teamID, score)

Step 7. For each multi-valued attribute A of an entity S, create a new table R. R will include: an attribute corresponding to A and the primary key of S which will be a foreign key in table R. Call this K. The primary key of R is a combination of A and K

There is one multi-valued attribute which is the medical conditions of a player, so create new table with PK a combination of player ID and medical condition:

playerMedConds(playerID, medCond)

Therefore the full set of tables are:

referee(refID, name)

competition(compID, name, ageCategory)

match(matchID, date, time, location, refID, compID)

team(teamID, teamName, teamGender, teamAgeGroup, coachName, captainID, compID)

player(playerID, name, dob, guardianName, guardianMobile, teamID)

teamMatch(matchID, teamID, score)

playerMedConds(playerID, medCond)