

Basics of database systems

Project – Database design

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2022

TABLE OF CONTENTS

TABLE OF CONTENTS.....	1
1 DEFINITION.....	2
2 MODELING	3
2.1 Concept model	3
2.2 Relational model	3
3 DATABASE IMPLEMENTATION	5
4 DISCUSSION.....	6

1 DEFINITION

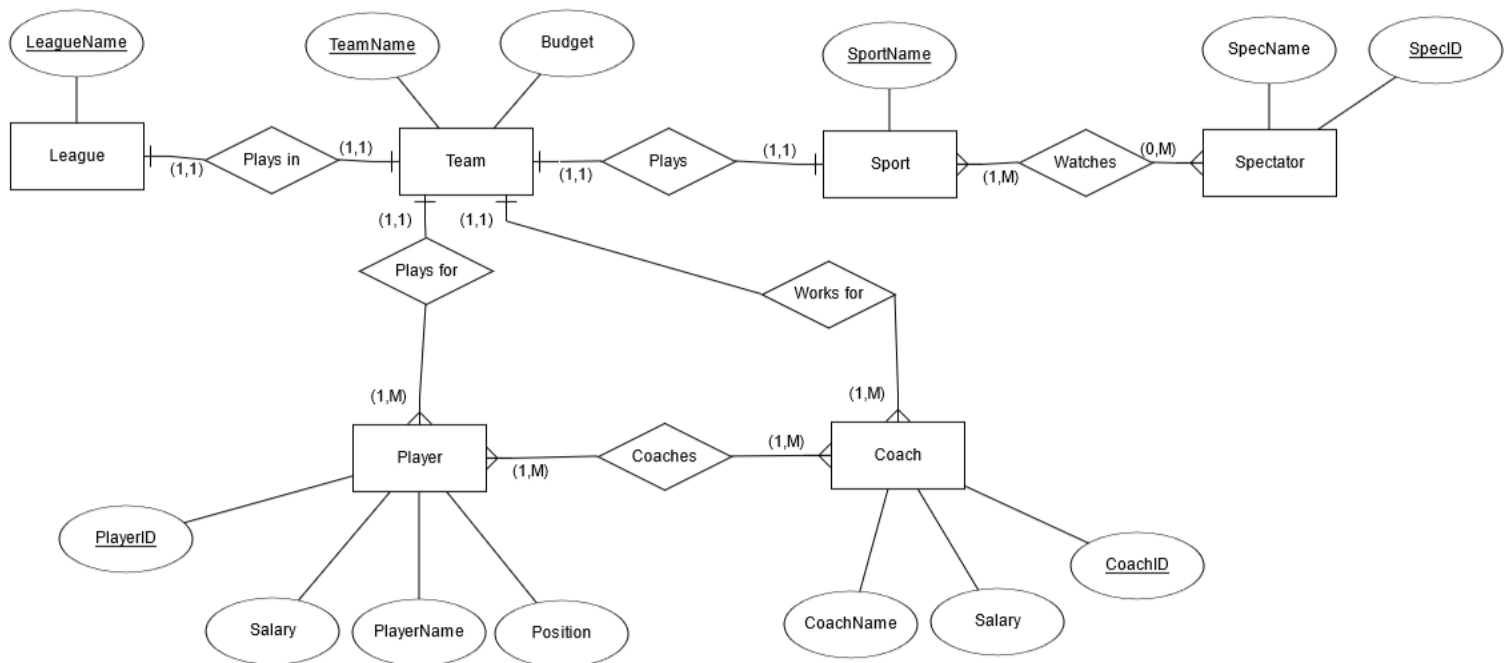
This database is developed for the HR personnel of any sport team or organisation in order to keep track of the team's players, coaches and the fanbase. All relevant information regarding the personnel is stored such as names, salary, team, league, sport, and player – coach relationships. All personnel are distinguishable by their individual id numbers in the case of players or coaches that share the same name. In addition, the fanbase of any given sport is monitored, and individual fans are identified by their id numbers. The database also allows for some light accounting regarding the personnel salaries in relation to the team budget.

The database is intended for quick and easy use mostly within the HR department of the target organisation to improve decision making in recruitment processes and salary negotiations, and as such the implementation should be very lightweight and streamlined. The database is only accessible to people in administrative roles within the organisation. All authorised personnel are allowed to alter the tables, queries, views and create new data and structures.

The following database queries must be implemented: (1) Print the information of a specific player. (2) Print teams' players (3) Print teams coaches (4) Print all sports (5) Print all leagues (6) Print all teams (7) Print all players of a given league

2 MODELING

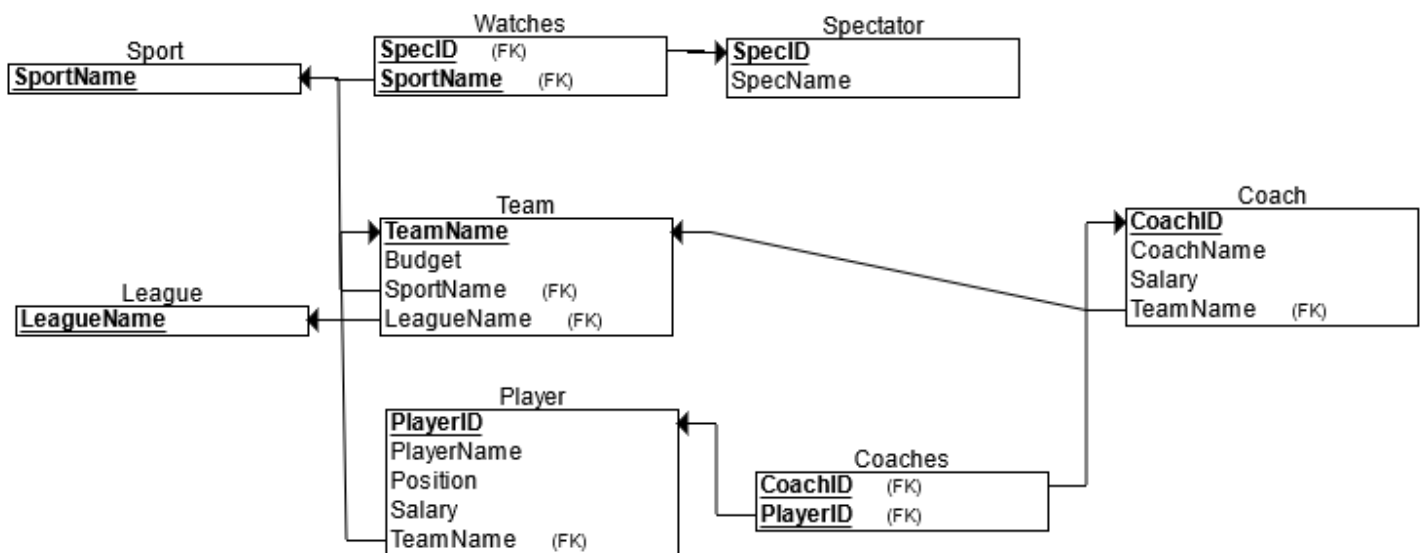
2.1 Concept model



2.2 Relational model

1. Transforming entities
 - a. Only strong entities exist in this ER-model so relations formed based on this
 - b. Key attributes were used to make primary keys (underlined)
2. Transforming attributes
 - a. Only simple attributes were used
3. Transforming relationships
 - a. One to one relations
 - i. One team plays in one league and one team plays one sport
 - ii. Foreign keys under Team-relation
 - b. One to many relations
 - i. Team can have multiple players, so the foreign key (TeamName) goes to the player side
 - c. Many to many relations

- i. Coaches often train multiple players, but often a individual player might also have many coaches. Similarly, a spectator might watch many sports and often sports also have many spectators.
- ii. In these situations, a linking relation is formed, and a composite primary key is formed based on the primary keys of the tables that formed the composite table
- iii. In this case those tables were named 'Coaches' and 'Watches'



The definitions of value ranges were implemented in the SQL-commands (see attached file createTables.sql) but were left out of the above illustration to avoid clutter.

3 DATABASE IMPLEMENTATION

In general:

During the transformation I noticed that I had given ambiguous column names to several entities especially regarding the Id's. This was corrected to reflect the Id's corresponding entity (e.g., ID \rightarrow PlayerID). In addition, I chose to implement leagues and sports as single column entities, as I found that they serve their purpose as so-called look-up tables for certain functionalities of my python-based user interface. When programming the UI, I noticed that the SQL-queries that demand user input are prone to errors regarding case sensitivity. So, I decided to add a clause that renders the team, coach, league and player names case insensitive in the creation phase of the tables. I also added indexing to the Player table to speed up the search queries, due to it potentially having a large amount of data points. Cascading was also implemented when updating or deleting.

Implementation:

I wrote a python-based UI to illustrate the database functionalities as requested in the project description. First, the program initialises the database and some testing data is then created. The UI has 9 functionalities ranging from general printing of categorical information such as team names to some more specific queries that require user input, like printing a certain players information. In addition, the user can update the salary information of a player. Finally, the salary information of all players can be visualised as a bar chart using the bokeh library.

4 DISCUSSION

This project was done to demonstrate the basic functionalities of an SQLite database and as such, cyber security issues regarding specific user authorisations to parts of the database were not taken into consideration and were not implemented. For example, in a real-world implementation a member of another team's administrative staff would obviously not have the access to personnel information of other teams.