
Sportliga

Beschreibung, Klassendiagramm und relationales Modell
zur modellierung einer Sportliga

von
Seraina Tschuor & Lukas Schneider

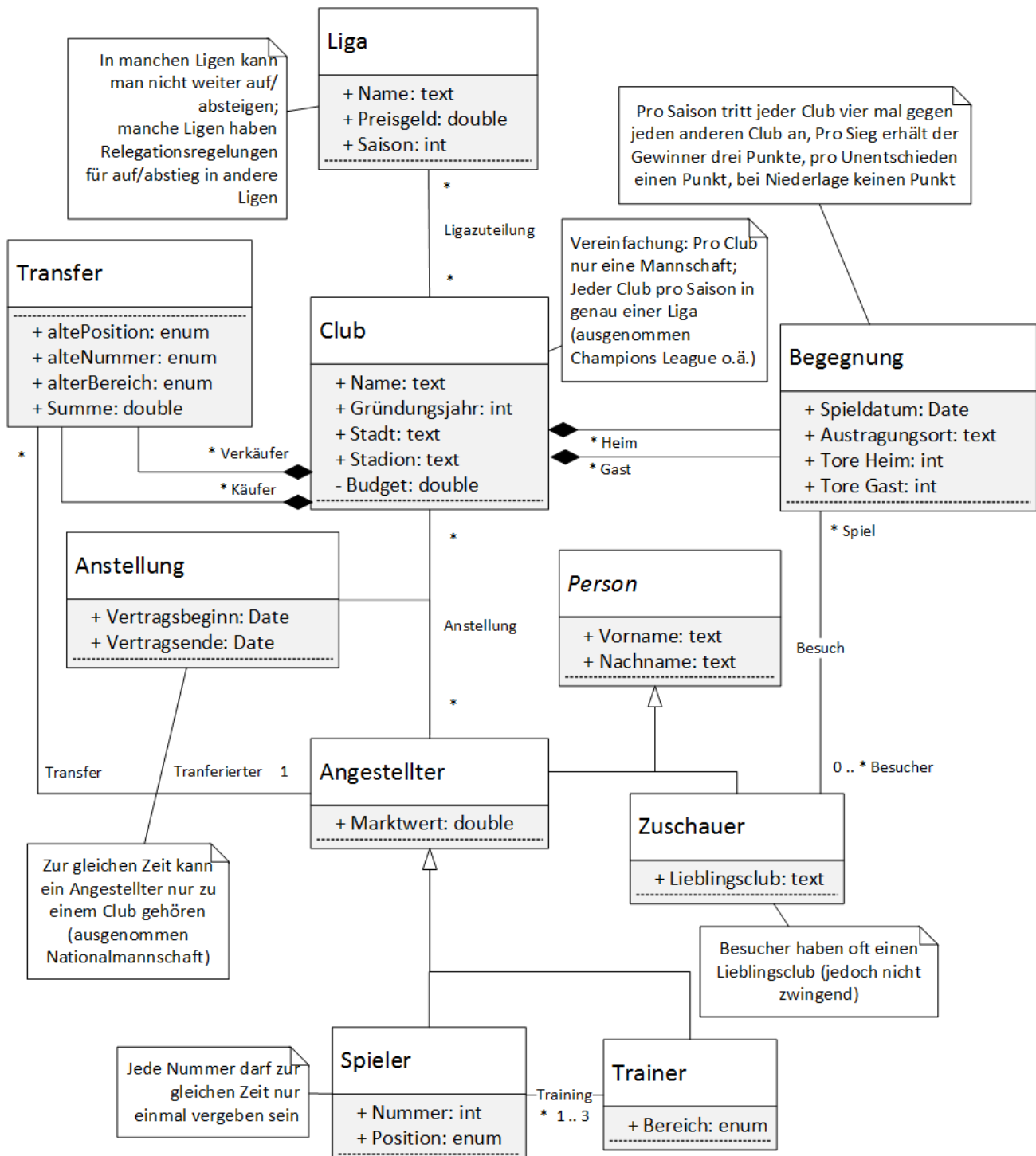
Systembeschreibung Sportliga

Das System modelliert Sportclubs, seine Angestellten, Spieler und Trainer, sowie die Beziehung von Angestellten zwischen Clubs in Form von Transfers und Begegnungen zwischen Clubs. Zuschauer besuchen diese Begegnungen. Diese Clubs spielen in einer Liga um einen Preis.

- Ein Club besteht aus seinen Angestellten, wobei diese sich aufteilen in Spieler und Trainer. Jeder Club hat einen Namen, eine Stadt, ein Stadion, ein Gründungsjahr und ein Budget. Jeder Club hat nur eine Mannschaft (vereinfachung) welche in genau einer Liga spielt.
- Alle Spieler eines Clubs werden von einem bis drei Trainern trainiert, wobei jeder einen bestimmten Spezialbereich hat. (Wann welcher Trainer welchen Spieler trainiert ist nicht modelliert) Jeder Spieler hat eine bestimmte Nummer, welche zur gleichen Zeit nur einmal vergeben sein darf.
- Angestellte bekommen ein Gehalt. Dieses Gehalt wird in einem Vertrag festgelegt, wenn der Vertrag ausläuft kann dieser entweder verlängert werden oder der Angestellte kann gegen Zahlung einer Ablösesumme zu einem anderen Club transferieren (Der Angestellte kann natürlich auch den Vertrag einfach auslaufen lassen).
- Jeweils zwei Clubs treten an bestimmten Daten in Spielen gegeneinander an. Einer der Clubs fungiert als Gastgeber, in seinem Stadion wird das Spiel ausgetragen. Die Bewertung findet nach der Drei-Punkte-Regelung statt.
- Diese Spiele finden an einem bestimmten Datum statt. Zuschauer haben oft einen Lieblingsclub, d.h. ein Club zu dessen Spielen sie regelmässig gehen.
- Diese Spiele werden von einer Liga mit einem bestimmten Namen veranstaltet, die Ergebnisse der Spiele führen zu Platzierungen, welche zu Abstiegen oder Aufstiegen führen. Die Platzierung wird über die Punkte entschieden, bei Punktgleichstand entscheidet die Tordifferenz.
- Am Saisonende erhält der Erstplatzierte Club ein Preisgeld und steigt je nach Liga auf. Die oder der letztplatzierte(n), je nach Liga, steigen ab.

Das System sollte möglichst international kompatibel sein. Da die Regeln je nach Liga über die Zeit hinweg ändern können, sollten diese nicht in der Datenbank modelliert sein. Diese Regeln (und auch Saisondaten) müssen in den Zugriffen formuliert sein.

UML Klassendiagramm



Relationales Modell

```
Ligen(LigaId INT,
      Name TEXT NOT NULL,
      Preisgeld DOUBLE NOT NULL,
      Saison INT NOT NULL)
```

```
Clubs(ClubId INT,
      Name TEXT NOT NULL,
      Stadt TEXT NOT NULL,
      Stadion TEXT NOT NULL,
      Budget DOUBLE NOT NULL,
      Gründungsjahr INT NOT NULL)
```

```
Ligazuteilungen(LigaZuteilungsId INT,
                Club REFERENCES Clubs(ClubId),
                Liga REFERENCES Ligen(LigaId))
```

```
Angestellten(AngId INT,
             Vorname TEXT NOT NULL,
             Nachname TEXT NOT NULL,
             Marktwert DOUBLE NOT NULL,
             Nummer INT,
             Position TEXT,
             Bereich TEXT)
```

```
Anstellungen(AnstellungsId,
            AngId REFERENCES Angestellten,
            Club REFERENCES Clubs(ClubId),
            Vertragsbeginn DATE NOT NULL,
            Vertragsende DATE NOT NULL)
```

```
Transfers(TransferId INT,
          Käufer REFERENCES Clubs(ClubId),
          Verkäufer REFERENCES Clubs(ClubId),
          Transferierter REFERENCES Angestellten(AngId),
          Summe DOUBLE NOT NULL,
          altePosition ENUM,
          alteNummer INT,
          alterBereich ENUM)
```

```
Begegnungen(BegegnungsId INT,
            Heim REFERENCES Clubs(ClubId),
            Gast REFERENCES Clubs(ClubId),
            Spieldatum DATE,
            Austragungsort NOT NULL,
            ToreHeim INT,
            ToreGast INT)
```

```
Zuschauer(ZuschauerId INT,
          Vorname TEXT,
          Nachname TEXT,
          Lieblingsverein REFERENCES Clubs(ClubId))
```

```
Zuschauerverteilungen(ZuschauerVerteilungsId INT,
                      Spiel REFERENCES Begegnungen(BegegnungsId),
                      Zuschauer REFERENCES Zuschauer(ZuschauerId))
```
