JUST DO IT
HÖHERE ABTEILUNG FÜR
INFORMATIONSTECHNOLOGIE

tgm
Die Schule der Technik

Protokoll

Semesteraufgabe — Fussball

SEW & INSY
4AHITM 2015/16

Lorenz Breier

Version 0.1
Begonnen am 2016-03-16
Beendet am 1. April 2016

Note:
Betreuer: Prof. Dolezal, Prof. Borko

Inhaltsverzeichnis

1 Angabe

Ein österreichischer Fußballverein braucht eine neue Datenbank, um zumindest die Personenverwaltung auf eine professionelle Basis zu stellen. In dieser Datenbank werden folgende Daten verwaltet:

Jede Person ist identifiziert durch eine eindeutige Personalnummer (kurz "persnr"). Außerdem werden zu jeder Person folgende Informationen verwaltet: Vorname ("vname"), Nachname ("nname"), Geschlecht ("geschlecht") und Geburtsdatum ("gebdat"). Für "geschlecht" sind einzig die Eingaben "W", "M" und "N" gültig.

Jede Person, die am Vereinsleben beteiligt ist, gehört zu genau einer der folgenden 4 Kategorien: Angestellter, Vereinsmitglied (kurz "Mitglied"), Spieler oder Trainer. Für all diese Kategorien von Personen müssen zusätzliche Informationen verwaltet werden:
Von jedem Angestellten werden das Gehalt ("gehalt"), die Überstunden ("ueberstunden") und die E-Mail-Adresse ("e_mail") vermerkt.
Für jedes Vereinsmitglied werden ausständige Beitrage ("beitrag") abgespeichert.
Jeder Spieler hat eine Position ("position") (z.B. Tormann, Stürmer, etc.). Außerdem sind Gehalt ("gehalt") sowie Beginn ("von") und Ende ("bis") der Vertragsdauer abzuspeichern. Jeder Spieler ist zumindest einer Mannschaft zugeordnet. Die Spieler innerhalb einer Mannschaft müssen jeweils eine eindeutige Trikot-Nummer ("nummer") zwischen 1 und 99 haben.
Jeder Trainer hat ein Gehalt ("gehalt") sowie Beginn ("von") und Ende ("bis") der Vertragsdauer. Jeder Trainer ist genau einer Mannschaft zugeordnet.

Der Verein hat mehrere Mannschaften. Für jede Mannschaft sind folgende Informationen zu verwalten: eine eindeutige Bezeichnung ("bezeichnung") (wie 1. Mannschaft, Junioren, A-Jugend, ...), die Klasse ("klasse") sowie das Datum des nächsten Spiels ("naechstes_spiel"). Jede Mannschaft hat mindestens 11 Spieler, genau einen Chef-Trainer und genau einen Trainer-Assistenten. Beide sind als Trainer des Vereins registriert. Außerdem hat jede Mannschaft einen Kapitän (der ein Spieler des Vereins ist).

Der Verein hat mehrere Standorte. Jeder Standort ist identifiziert durch eine eindeutige ID ("sid"). Außerdem sind zu jedem Standort folgende Informationen verfügbar: Land ("land"), Postleitzahl ("plz") und Ort ("ort").

An jedem Standort gibt es 1 oder mehrere Fan-Clubs. Jeder Fan-Club hat einen für den jeweiligen Standort eindeutigen Namen ("name"), ein Gründungsdatum ("gegruendet") und einen Obmann ("obmann"), der Vereinsmitglied sein muss. Ein Vereinsmitglied kann von höchstens einem Fan-Club der Obmann sein. Die Fan-Clubs werden von Angestellten des Vereins betreut: Und zwar kann jeder Angestellte einen Fan-Club jeweils für einen gewissen Zeitraum betreuen. Dieser Zeitraum ist durch Anfangsdatum ("anfang") und Endedatum ("ende") bestimmt. Jeder Angestellte kann 0 oder mehere Fan-Clubs betreuen, und jeder Fanclub kann von einem oder mehreren Angestellten betreut werden.

Der Verein führt Aufzeichnungen über die absolvierten Spiele. Jedes Spiel ist gekennzeichnet durch die Bezeichnung der Mannschaft ("mannschaft") und das Datum ("datum"). Für jedes Spiel werden der Gegner ("gegner") und das Ergebnis ("ergebnis") eingetragen, das einen der Werte "Sieg", "Unentschieden" oder "Niederlage" haben kann. Außerdem werden zu jedem Spiel die beteiligten Spieler abgespeichert, wobei für jeden Spieler die Dauer Einsatzes ("dauer") als Wert zwischen 1 und 90 vermerkt wird.

2 Vorraussetzungen

2.1 Server starten

2.2 SSH - Verbindung

2.3 Postgres

Datenbank, User erzeugen, Rechte verteilen & Einloggen

```
1 CREATE DATABASE verein;
  CREATE USER manager WITH PASSWORD 'iderfrein';
  GRANT ALL ON DATABASE verein TO manager;
```

Connection :

```
psql -d verein -h localhost -U manager
```


3 Aufgaben

3.1 CREATE-Befehle

Schreiben Sie die nötigen CREATE-Befehle, um die vorgestellten Relationen mittels SQL zu realisieren. Dabei sind folgende Punkte zu beachten:

3.1.1 Die Datenbank soll keine NULL-Werte enthalten.

Dies lässt sich umsetzen, indem man am Ende des Attributes ein 'NOT NULL' anfügt.
Beispiel:

```
CREATE TABLE angestellter(
  persnr SERIAL PRIMARY KEY REFERENCES person,
  ueberstunden NUMERIC(3,0) NOT NULL,
  gehalt NUMERIC(8,2) NOT NULL,
  e_mail VARCHAR(50) NOT NULL,
);
```

3.1.2 Realisieren Sie folgende Attribute mit fortlaufenden Nummern...

...mit Hilfe von Sequences und Serial: "persnr"von Personen und "sid" von Standorten.

- Für das "persnr" - Attribut sollen nur gerade, 5-stellige Zahlen vergeben werden (d.h.: 10000, 10002, ..., 99998).

  Sequences kann man sich wie eine Zählervariable vorstellen, welche im Hintergrund mitläuft. Erstellt wird unsere Sequence so :

  ```
  CREATE SEQUENCE pid START WITH 10000 INCREMENT BY 2;
  ```

  Die 5 Stellen erhalten wir, indem wir uns an der Option **START WITH** bedienen.
  Für die geraden Zahlen helfen wir uns mit einem Trick, in dem wir mit Hilfe von **INCREMET BY** immer um 2 erhöhen.
  => da er nur immer um 2 erhöht, können nur gerade Zahlen kommen !

- Für das "sid" - Attribut sollen beliebige positive Zahlen (d.h. 1,2, ...) vergeben werden.
  Falls zwischen 2 Tabellen zyklische FOREIGN KEY Beziehungen herrschen, dann sind diese FOREIGN KEYs auf eine Weise zu definieren, dass es möglich ist, immer weitere Datensätze mittels INSERT in diese Tabellen einzufügen.

Tabellenverzeichnis

Listings

Abbildungsverzeichnis

