FUSSBALLVEREIN

Jakob Grieshofer

Version 1.0

Inhaltsverzeichnis

1 Einführung

Dieses Protokoll beschreibt die Vorgangsweise bei der Aufgabe „Fussballverein“.

1.1 Kompetenzen

1) Eigenschaften und Architekturen von Datenbanksystemen
   „können den Begriff "Transaktion" erklären und die Voraussetzungen für eine korrekte Abarbeitung nennen sowie die Problematiken bei parallel auftretenden Transaktionen aufzeigen und diese in Fehlerklassen kategorisieren“

2) Abfragesprachen
   „können komplexe Abfragen für konkrete Problemstellungen entwickeln und optimieren“
   „können den Aufbau von Sichten erklären sowie deren Vor- und Nachteile nennen“

3) Datenbankanwendungen
   „können standardisierte Datenbankschnittstellen installieren und konfigurieren, um aus gängigen Programmiersprachen mit einem „Datenbanksystem kommunizieren zu können“
   können die Einsatzgebiete von datenbankseitiger Programmierung evaluieren und solche Anwendungen entwickeln“
   „können Anwendungen mit Datenanbindung entwickeln“

4) Anwendungsentwicklung
   „können Anwendungssysteme unter Verwendung von Nebenläufigkeit entwickeln“
   „können einfache Schnittstellen zur Kommunikation zwischen Anwendungen entwerfen und implementieren“
   „können umfangreiche Client-Server Anwendungen entwickeln“

1.2 Voraussetzungen

- Postgresql 9.5
- Qt
- Linux VM
- Java

1.3 Aufgabenstellung

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

* Die Datenbank soll keine NULL-Werte enthalten.
* Realisieren Sie folgende Attribute mit fortlaufenden Nummern mit Hilfe von Sequences: "persnr" von Personen und "sid" von Standorten. Für das "persnr"-Attribut sollen nur gerade, 5-stellige Zahlen vergeben werden (d.h.: 10000, 10002, ..., 99998). 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.

2.) Schreiben Sie INSERT-Befehle, um Testdaten für die kreierten Tabellen einzurichten. Jede Tabelle soll zumindest 100000 Zeilen enthalten. Sie dürfen die Wahl der Namen, Bezeichnungen, etc. so einfach wie möglich gestalten, d.h.: Sie müssen nicht "real existierende" Fußballer-Namen, Länder, Städte, etc. wählen. Stattdessen können Sie ruhig 'Spieler 1', 'Spieler 2', 'Land 1', 'Land 2', 'Stadt 1', 'Stadt 2', etc. verwenden. Sie können für die Erstellung der Testdaten auch entsprechende Generatoren verwenden!

3.) Schreiben Sie die nötigen DROP-Befehle, um alle kreierten Datenbankobjekte wieder zu löschen.

Lösen Sie die folgenden Probleme mittels SQL:

S1.) (Fan-Club Betreuung) Wählen Sie "per Hand" die Personalnummer eines Angestellten aus Ihren Testdaten aus. Schreiben Sie eine SQL-Anfrage, die jene Fan-Clubs ermittelt, die dieser Angestellte im Moment nicht betreut. Geben Sie zu jedem derartigen Fan-Club die Standort-ID und den Namen des Fan-Clubs aus.
Bemerkung: Ein Fan-Club wird von einem Angestellten im Moment nicht betreut, wenn entweder der Angestellte diesen Fan-Club überhaupt nie betreut hat oder wenn das heutige Datum (= sysdate) außerhalb des Betreuungszeitraums liegt. Vergessen Sie nicht, jene Fan-Clubs zu berücksichtigen, die von überhaupt keinem Angestellten betreut werden (dieser Fall sollte zwar laut Datenmodell nicht vorkommen. Die Einhaltung dieser Bedingung wird aber vermutlich vom Datenbanksystem nicht überprüft)!

S2.) (Die eifrigsten Angestellten) Schreiben Sie eine SQL-Anfrage, die den Nachnamen und die Personalnummer jener Angestellten ausgibt, die im Moment sämtliche Fan-Clubs betreuen. Ordnen Sie die Nachnamen alphabetisch.
Bemerkung: Passen Sie die Testdaten so an, dass diese Anfrage zumindest zwei Angestellte liefert.

S3.) (Spielereinsätze) Geben Sie für alle Spiele des Jahres 2015 jeweils alle Spieler und

die Dauer ihres Einsatzes aus, d.h.: Gesucht sind alle Tupel (mannschaft, datum, vorname, nachname, dauer), mit folgender Eigenschaft:

"mannschaft" ist die Bezeichnung der Mannschaft, die gespielt hat.

"datum" ist das Datum, an dem das Spiel stattfand.

"vorname" und "nachname" beziehen sich auf einen Spieler, der bei diesem Spiel zum Einsatz kam.

"dauer" gibt die Dauer des Einsatzes (in Minuten) dieses Spielers bei diesem Spiel an.

S4.) (Spieler-Ranking) Geben Sie für jeden Spieler den Vornamen und Nachnamen sowie die Gesamtdauer ("gesamtdauer") der von ihm bei Spielen im Jahr 2015 geleisteten Einsätze aus. Vergessen Sie nicht, jene Spieler des Vereins zu berücksichtigen, die im Jahr 2015 bei keinem einzigen Spiel mitgespielt haben (d.h. gesamtdauer = 0). Ordnen Sie die Ausgabe in absteigender Gesamtdauer. Bei Gleichheit der Gesamtdauer sollen die Spieler in alphabetischer Reihenfolge (zuerst des Nachnamen, dann des Vornamen) sortiert werden.

S5.) (Der fleißigste Spieler) Geben Sie den Vornamen und Nachnamen jenes Spielers aus, von dem die unter b) berechnete Gesamtdauer am größten ist, d.h.: dieser Spieler ist bei Spielen im Jahr 2015 insgesamt am längsten im Einsatz gewesen. Falls sich mehrere Spieler den ersten Platz teilen (d.h. sie kommen auf die gleiche Gesamtdauer), dann sollen diese in alphabetischer Reihenfolge (zuerst des Nachnamen, dann des Vornamen) geordnet werden. Der Fall, dass im Jahr 2015 überhaupt kein Spiel stattfand, darf ignoriert werden.
Bemerkung: Berücksichtigen Sie bei Ihren Testdaten die Situation, dass sich zumindest 2 Spieler den ersten Platz teilen.

S6.) Schreiben Sie CREATE und DROP Befehle für eine View, die alle Informationen über Trainer aus der Personen- und Trainer-Tabelle zusammenfügt, d.h.: sowohl die allgemeinen Personendaten (Personalnummer, Vorname, Nachname, Geschlecht und Geburtsdatum) als auch die Trainer-spezifischen Informationen (Gehalt sowie Beginn und Ende der Vertragsdauer). In Summe ist also folgende View erforderlich:

Trainer_view (persnr, vname, nname, geschlecht, gebdat, gehalt, von, bis).

Datenbankclient (Java/C++) und DB-Connector (JDBC/libpqxx):

Schreiben Sie einen Client, der eine Datenbank-Verbindung herstellt. Realisieren Sie eine GUI (JavaFX/Qt), die das einfache Ändern (CRUD) der Spieler des Vereins erlaubt. Verwenden Sie dabei auf jeden Fall eine Tabelle (TableView, QTableView), die auch eine grafische Veränderung der Datensätze erlauben soll.

Ermöglichen Sie die gleichzeitige Verbindung von mehreren Clients auf die Datenbasis. Implementieren Sie dabei eine transaktionelle, gesicherte Erstellung und Änderung von

Spielen. Beachten Sie dabei, dass der Spielstand und die Spielzeit der einzelnen Spieler laufend und von mehreren Clients gleichzeitig aktualisiert werden könnte. Stellen Sie für die Eingabe der Spielerzeit und Spielstand eine einfache grafische Möglichkeit zur Verfügung. Verwenden Sie dabei Transaktionen bzw. Locks und entsprechende programmtechnische Mittel um Inkonsistenzen zu vermeiden. Definieren Sie dabei für die einzelnen Informationen (Spielerzeit, Spielstand) eigene Threads.

2 Ergebnisse

2.1 Postgresql 9.5

Anfangs gab es Schwierigkeiten da immer ein Authentifizierungsfehler auftrat. Außerdem konnte ich nicht apt-get update ausführen da ein Medienwechsel auf cdrom verlangt wurde.

```
Medienwechsel: Bitte legen Sie das Medium mit dem Namen
 »Debian GNU/Linux 8.2.0 _Jessie_ - Official amd64 DVD Binary-1 20150906-11:13«
in Laufwerk »/media/cdrom/« ein und drücken Sie die Eingabetaste (Enter).
```

Nach kurzer Recherche habe ich herausgefunden dass in sources.list eine Zeile auskommentiert werden musste.

```
#deb cdrom:[Debian GNU/Linux 8.2.0 _Jessie_ - Official amd64 DVD Binary-1 20150906-11:$
```

Allerdings konnte ich trotzdem keine gescheite Installation machen.

Dann wurde mir ein Tutorial zur Installation empfohlen mit dem diese reibungslos funktionierte. http://www.gab.lc/articles/install_postgresql_9-5_debian_ubuntu

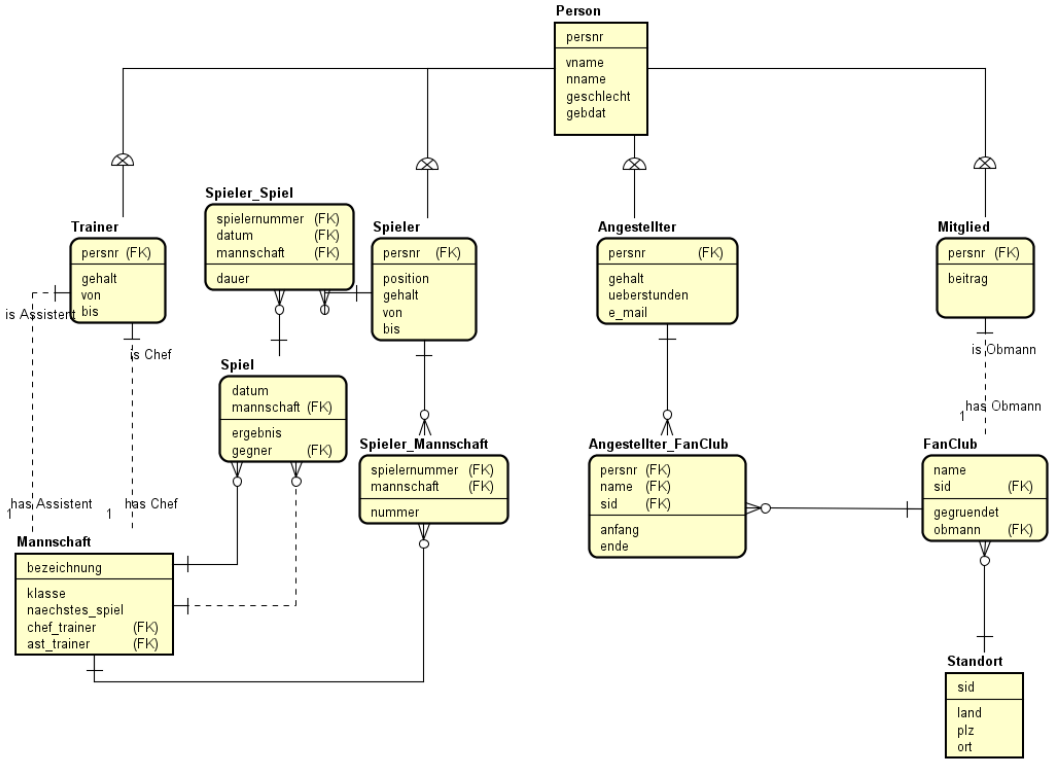
Das Erstellen eines Users als Superuser, Erstellung der Datenbank mit dem besagten User als Owner und das geben aller Rechte auf diese:

```
create user fussball with password 'fussball';
alter user fussball with superuser;
create database fussballvereinv3 with owner fussball;
grant all privileges on database fussballvereinv3 to fussball;
```

2.2 ERD

Das ERD wurde mit Astah erstellt.

Ich konnte ich in Astah die Kapitäns - Relation nicht darstellen, es wurde einfach kein neuer FK erstellt. Somit sollte im folgenden ERD noch eine Non-Identifying Beziehung zwischen Spieler (parent) und Mannschaft sein, und in der Mannschaft ein FK kapitaen.



2.3 Creates

Diese wurden grundsätzlich aus dem ERD exportiert. Allerdings mussten noch händisch Änderungen durchgeführt werden.

- Die Kapitaens – Beziehung (Begründung unter 2.4)
  ALTER TABLE Mannschaft ADD CONSTRAINT FK_Mannschaft_2 FOREIGN KEY (kapitaen)
  REFERENCES Spieler (persnr);
- Die Sequenzen
  Erstellung:
  Sequenz für persnr:
  CREATE SEQUENCE seq_incr_two INCREMENT BY 2 START WITH 10000;
  Diese ist für die sid:
  CREATE SEQUENCE seq_auto_incr INCREMENT BY 1 START WITH 1;

  Zuweisung:
  persnr INT NOT NULL DEFAULT nextval('seq_incr_two'),
  sid INT NOT NULL DEFAULT nextval('seq_auto_incr'),

2.4 Inserts

Ich habe einen eigenen Generator, der die Inserts in ein .txt File schreibt, gemacht.

Er funktioniert einwandfrei bis auf 2 Fehler:

Es kann beim generierten Datum einen 29.02. geben, dies ruft wiederum Fehler hervor.
Eine mit math.random() generierte persnr kommt manchmal doppelt vor z.B.: 2 mal der selbe Spieler in Mannschaft, was allerdings nicht geht und einen Fehler ausgibt.

Im Allgemeinen kann man das allerdings vernachlässigen da es ein geringer Prozentsatz ist.