

Semesteraufgabe

David Sindi
TGM

4/4/2016

FUSSBALL- VEREIN



Fussballverein Semesteraufgabe

Inhaltsverzeichnis

GitHub Link.....	2
Aufgabenstellung	2
Was gemacht werden muss	2
Aufgabe	2
Vorwort	3
Libpqxx.....	3
ERD	3
Datenbank.....	3
Verbindung.....	3

GitHub Link

<https://github.com/TgmSindl/Fussballverein>

Aufgabenstellung

Was gemacht werden muss

Aufgrund der positiven Kompetenzen, muss nur ein Teil der Aufgabe erfüllt werden. Fehlende Kompetenz QT muss erfüllt werden.

Aufgabe

Ein österreichischer Fußballverein braucht eine neue Datenbank, um zumindest die Personenverwaltung auf eine professionelle Basis zu stellen.

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.

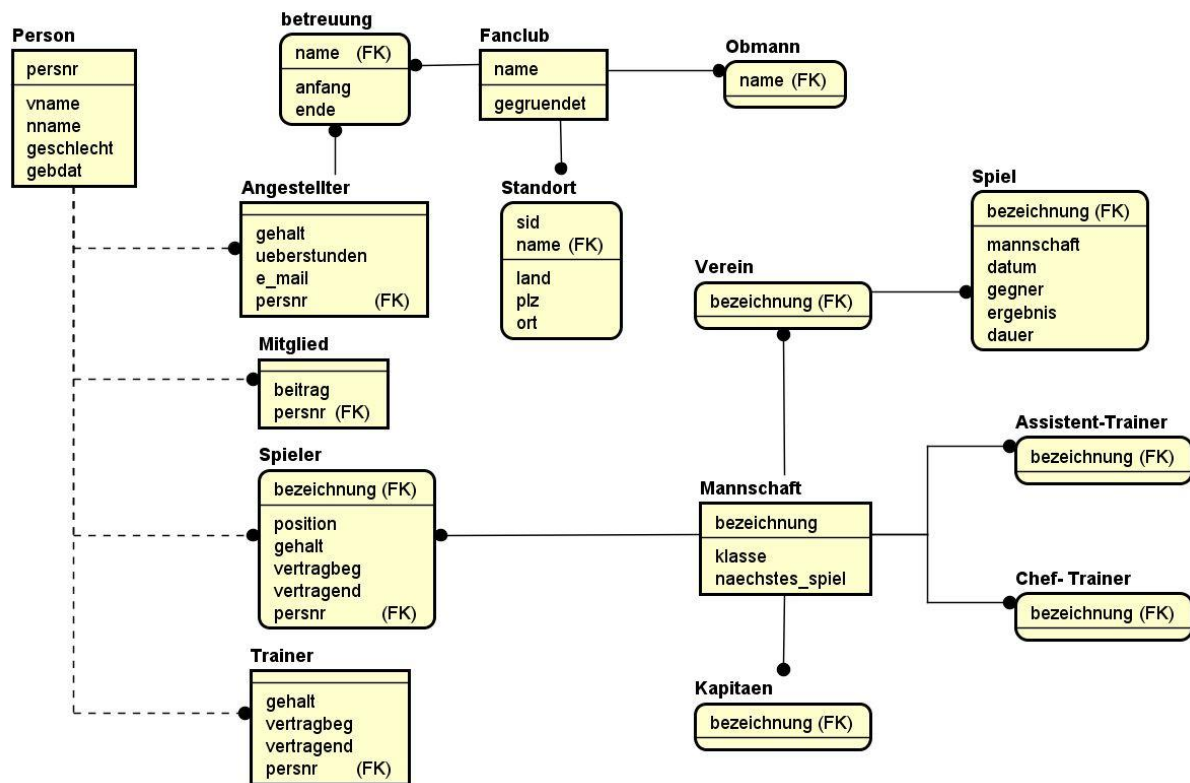
Implementieren Sie dabei eine transaktionelle, gesicherte Erstellung und Änderung von Spielen. Beachten Sie dabei, dass der Spielstand und die Spielzeit der einzelnen Spieler laufend und von mehreren Clients gleichzeitig aktualisiert werden könnte. Stellen Sie für die Eingabe der Spielerzeit und Spielstand 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.

Vorwort

Libpqxx

Da Libpqxx nicht auf Windows installationsfähig ist, musste es auf einem UNIX System installiert werden und da QT die Library von Libpqxx braucht für die Verbindung zu postgres, muss QT ebenfalls im selben UNIX System installiert werden.

ERD



Datenbank

Verbindung

User mega, pw zerleger, dieser User hat alle Rechte über die Datenbank erhalten mittels GRANT ALL PRIVILEGES TO mega

Datenbank erstellen

```
CREATE DATABASE fuessball;
```

Zugriff auf die neue Datenbank mittels
/c fuessball;

Table

```
DROP TABLE spieler;  
CREATE TABLE spieler(  
    spielernr INTEGER,  
    vname VARCHAR(30),  
    nname VARCHAR(30),  
    mannschaft VARCHAR(1),  
    gebdat DATE,  
    PRIMARY KEY (spielernr));
```

Daten in den Table speichern

```
INSERT INTO spieler (spielernr, vname, nname, mannschaft, geschlecht,  
gebdat)  
VALUES (1, 'Sepp', 'Depp', 'Mannschaft', 'm', '1956-10-11'),  
VALUES (2, 'Gepp', 'Dapp', 'Mannschaft', 'm', '1966-10-11'),  
VALUES (3, 'Fepp', 'Dipp', 'Mannschaft', 'm', '1976-10-11'),  
VALUES (4, 'Lepp', 'Dopp', 'Fannschaft', 'm', '1986-10-11'),  
VALUES (5, 'Wepp', 'Dupp', 'Mannschaft', 'm', '1996-10-11'),  
VALUES (6, 'Nepp', 'Dadd', 'Kannschaft', 'm', '1956-10-11'),  
VALUES (7, 'Cepp', 'Doqq', 'Mannschaft', 'm', '1959-10-11');
```

QT

Creator

Im Creator kann man nachdem man ein neues Projekt erstellt hat, im mainwindow.ui (im Unterordner Formulardateien), die GUI entwerfen. Man benötigt dafür geringe Kenntnisse, über Layouts, da alles gekennzeichnet ist. Buttons, Layouts müssen nach dem erstellen, ein Wert zugewiesen werden.

Problem

Da die Libpqxx einen unbekannten Fehler hat, der nicht wirklich herausfindbar ist, wurde das Programm, ohne testen geschrieben. Problembeschreibung hat keine Lösungen im Internet zum Vorschein gebracht.

Mainwindow.cpp

```
#include "mainwindow.h"  
#include "ui_mainwindow.h"  
#include <pqxx/pqxx>  
#include <iostream>  
#include <string>  
  
using namespace pqxx;  
using namespace std;  
MainWindow::MainWindow(QWidget *parent) :  
    QMainWindow(parent),  
    ui(new Ui::MainWindow)  
{  
    ui->setupUi(this);  
}  
MainWindow::~MainWindow()  
{  
    delete ui;  
}
```

```
/* pushButton integrieren und festlegen was bei drücken, passieren soll */
void MainWindow::on_pushButton_clicked()
{
    char * sql;

    try{
        connection C("dbname=fussball user=postgres password=postgres \
        hostaddr=127.0.0.1 port=5432");
        if (C.is_open()) {
            ui->label->setText("Datenbankverbindung erfolgreich");
        } else {
            ui->label->setText("Datenbankverbindung fehlgeschlagen");
            return;
        }

        /* Auswählen von welcher Table was genommen werden soll */
        sql = "SELECT * from spieler";

        nontransaction N(C);
        result R( N.exec( sql ) );

        /* Welche Einträge angezeigt werden sollen */
        for (result::const_iterator c = R.begin(); c != R.end(); ++c) {
            cout << "spielernr = " << c[0].as<int>() << endl;
            cout << "vname = " << c[1].as<string>() << endl;
            cout << "nname = " << c[2].as<string>() << endl;
            cout << "mannschaft" << c[3].as<string>() << endl;
            cout << "gebdat" << c[4].as<string>() << endl;
        }
        cout << "erfolgreich" << endl;
        C.disconnect ();
    } catch (const std::exception &e){
        cerr << e.what() << std::endl;
        return;
    }
}
```