

Datenbanken

Proseminar Abgabe 3

Sommersemester 2015

In diesem Teil des Projektes wird eine Anwendung erstellt, welche die Datenbank aus dem vorigen Projektteil anfragt und verändert.

1 Aufgabestellung

Erstellen Sie eine einfache Anwendung,

- (i) entweder als eigenständige Anwendung mit graphischer Benutzeroberfläche,
- (ii) als Web-Anwendung, oder
- (iii) als Kommandozeilenapplikation.

Die “Kommandozeilenvariante” ist sicherlich jene mit dem geringsten Aufwand, jedoch gibt es Bonuspunkte für die Implementierung einer grafischen Oberfläche.

Die Anwendung sollte auf jeden Fall das **Abfragen, Einfügen, Ändern** und **Löschen** von Daten erlauben.

Daten abfragen Der Benutzer soll die Möglichkeit haben Auswahlkriterien einzugeben und sich das entsprechende Resultat anzeigen zu lassen. Mindestens ein Auswahlkriterium soll unsterstützt werden.

Hinweis

Die Angabe des Schlüssels als einziges Suchkriterium ist nicht ausreichend.

Daten einfügen/ändern Es sollte die Möglichkeit geben, neue Daten einzufügen und bereits gespeicherte Daten nachträglich zu verändern.

Daten löschen Es soll die Möglichkeit geboten werden, Daten aus dem Ergebnis einer Suchanfrage auszuwählen und in weiterer Folge zu löschen.

Hinweis

Es ist **nicht** nötig, dass alle Daten in der Datenbank abgefragt bzw. verändert werden können. Es reicht, wenn ein Beispiel von jeder Art (abfragen, einfügen/ändern, löschen) programmiert wird.

2 Technologie und Testumgebung

Wenn Sie die Anwendung auf einem Rechner vorführen, den Sie selbst mitbringen, können Sie eine beliebige Programmiersprache und Programmierumgebung verwenden. Sie sind allerdings dafür verantwortlich, dass Ihre Anwendung während der Demo funktioniert.

Sollten Sie keinen eigenen Rechner für die Abschlusspräsentation zur Verfügung haben, melden Sie sich bitte beim Betreuer Ihrer Proseminargruppe *bevor* Sie diesen Teil des Projektes bearbeiten (bitte rechtzeitig vor dem Präsentationstermin). Es gibt dann nämlich Restriktionen bezüglich Programmiersprache und Programmierumgebung.

3 Abgabeformat

Die Abgabe erfolgt via E-Mail an dbabgabe@cosy.sbg.ac.at mit dem Betreff **PS Datenbanken <teamname>**.

Abzugeben sind folgende Dateien:

1. **README.TXT** beschreibt, welche Systemvoraussetzungen Ihre Anwendung erfordert (Betriebssystem, Programmiersprache, installierte Pakete, etc.) und wie Ihre Anwendung compiliert und gestartet wird.
2. Source Code der Anwendung.

Die Datei **README.TXT** und der Source Code werden in eine ZIP Datei mit dem Namen **<teamname>_3.zip** gepackt, wobei **<teamname>** durch den Namen Ihres Teams zu ersetzen ist, z.B. **a3_3.zip**, **b4_3.zip**.

Die Einteilung in Teams finden Sie auf der [Website des Proseminars](#).

Achtung: Verspätete Abgaben oder Abgaben, die das oben angegebene Format nicht beachten, werden zurückgewiesen und nicht bewertet.

4 Demonstration während des Proseminars

Zusätzlich zur Abgabe stellen Sie Ihre Anwendung auch während des Proseminars vor.

Sie haben **6 Minuten** Zeit, die Funktionen Ihre Anwendung zu präsentieren. Anschließend werden sie noch kurz zu Ihrer Anwendung befragt. Durch die Präsentation sollten folgende Punkte klar werden:

- Die Grundoperationen sind implementiert und funktionieren (siehe Anforderungen).
- Welche Funktionalitäten bieten die verschiedenen Operationen (z.B. welche Suchkriterien können angewandt werden)?

- Wie komplex sind die SQL Anfragen, die sich hinter den Formularen verbergen (z.B. ein Join ist nötig, Einfügeoperation erfordert die Aktualisierung mehrerer Tabellen).
- Eventuelle fortgeschrittene SQL Konzepte, die Sie verwendet haben, z.B. Transaktionen, Indexe, etc. Dabei sollten Sie deren Sinnhaftigkeit in Ihrer Anwendung kurz begründen.

Es ist notwendig, dass **alle Gruppenmitglieder anwesend** sind. Jedes Gruppenmitglied muss sich an der Präsentation angemessen beteiligen und in der Lage sein, Fragen zu allen Teilen des Projektes zu beantworten.

Achtung: Bitte halten Sie sich streng an den Zeitrahmen. Wenn Ihre Zeit verstrichen ist, werden Sie unterbrochen und die ersten 6 Minuten werden bewertet. *Es wird sehr empfohlen, die Demo vorher zu proben!*

5 Bewertungskriterien

Zusätzlich zur korrekten Funktion Ihrer Anwendung fließen folgende Kriterien in die Bewertung ein:

- *Funktionalität:* Welche Operationen wurden implementiert? Welche Möglichkeiten bietet die Anwendung für die Suche von Daten, die angezeigt oder verändert werden sollen?
- *Komplexität der Anwendung:* Komplexere Szenarien entstehen zum Beispiel, wenn das Ergebnis einer Anfrage einen Join erfordert oder das Einfügen/Ändern/Löschen von Daten Aktualisierungen in mehreren Tabellen erfordert.
- *Demo:* Qualität (Klarheit, Verständlichkeit, Vollständigkeit) der Demonstration während des Proseminars.

Ein *Bonus* wird für die Verwendung fortgeschrittener SQL Konzepte vergeben (sowie die Implementierung einer grafischen Benutzeroberfläche). Beispiele fortgeschrittener SQL Konzepte sind Transaktionen, die mehr als einen SQL Befehl umspannen, um die Sicherheit der Schreiboperationen zu gewährleisten oder die Verwendung von Indexstrukturen, um die Ausführungszeit für Anfragen zu erhöhen. *Die Höchstpunktzahl kann ohne Bonus erreicht werden.*

Beispiel einer Datenbankverbindung in Python

Das folgende Beispiel illustriert, wie man Python verwenden kann um eine Verbindung zur Datenbank aufzubauen und eine einfache Anfrage abzusetzen (auf Basis der bekannten NAWI Datenbank aus der Vorlesung). Im Speziellen benutzen wir das Python Modul `psycopg2` (welches einfach mit `pip install psycopg2` unter Linux installiert werden kann):

```
import psycopg2
import pprint

conn_str = ""
conn_str += 'host=localhost ' # Name des Rechners
conn_str += 'dbname=rktest ' # Name der Datenbank
conn_str += 'user=rkwitt ' # Benutzername
conn_str += 'password=''' # hier: kein Passwort

# Verbindung zur Datenbank aufbauen
conn = psycopg2.connect(conn_str)

# Wir holen uns den Cursor
cursor = conn.cursor();

# ... und setzen unser erstes SELECT ab
query_str = "SELECT * FROM Projekte"
cursor.execute(query_str)

# Anschliessend geben wir die Resultate aus
while True:
    records = cursor.fetchone()
    if records is None:
        break
    else:
        pprint.pprint(records)
```

Natürlich soll dies nur als Beispiel dienen. Ein Beispiel zur Datenbankverbindung in JAVA können Sie den Vorlesungsunterlagen entnehmen.