

## Zugriff auf eine SQL-Datenbank per Access-Frontend

Eine Datenbank verwendet heute so gut wie jeder. Dabei ist eine elegante Verwaltung und Benutzung der Daten schon selbstverständlich. Kein Anwender möchte per ssh-Zugriff im Terminal oder mit phpMyAdmin die Datenbank verwalten. Somit gehört die Entwicklung eines *Frontends* zu den Aufgaben eines Datenbank-Frontendentwicklers.

Der vorliegende Beitrag zeigt, wie du von einem Access-Frontend aus auf die MySQL-Datenbank auf dem Internetserver zugreifen und mit den Daten so arbeiten kannst, als würden diese auf dem heimischen Rechner liegen. Die benutzte Sprache ist *Python*. Die Schritte sind jedoch mit jeder anderen Hochsprache möglich. Wir starten zuerst mit dem Zugriff auf eine Datenbank, die auf demselben Rechner liegt, wie der an dem der Client arbeitet. (localhost)

### Python und SQLite

SQLite (SQL - Structured Query Language) ist ein dateibasiertes Datenbanksystem, welches in vielen Endgeräten und in Standardbrowsern (Safari, Chrome, Firefox) verwendet wird. Es ist ein relationales Datenbanksystem und basiert auf dem SQL-92 Standard (Quelle: Wikipedia.) Der große Vorteil ist in der einfachen Verwendung von SQLite. Es braucht kein Datenbanksystem installiert werden, die Speicherung der Daten erfolgt in einfachen Textdateien und die API ist standardisiert. So besitzt Python eine API, welche per `import sqlite3` verfügbar gemacht wird.

Der folgende Quelltext gibt einen Einblick in die Benutzung von SQLite.

```
1 import os, sys, sqlite3
2
3 # Existenz feststellen
4 if os.path.exists("universitaet.db"):
5     print("Datei bereits vorhanden")
6     sys.exit(0)
7
8 # Verbindung zur Datenbank erzeugen
9 connection = sqlite3.connect("universitaet.db")
10
11 # Datensatz-Cursor erzeugen
12 cursor = connection.cursor()
13
14 # Datenbanktabelle erzeugen
```

```
15 sql = "CREATE TABLE professoren(" \
16     "Name TEXT," \
17     "PersNr INTEGER PRIMARY KEY," \
18     "gehalt REAL," \
19     "Rang TEXT," \
20     "Raum INTEGER)"
21 cursor.execute(sql)
22
23 # Datensatz erzeugen
24 sql = "INSERT INTO professoren VALUES('Socrates'," \
25     "'2125,3500','C4',226)"
26 cursor.execute(sql)
27 connection.commit()
28
29 # Datensatz erzeugen
30 sql = "INSERT INTO professoren VALUES('Russel'," \
31     "'2126,3750','C3',320)"
32 cursor.execute(sql)
33 connection.commit()
34
35 # SQL-Abfrage
36 sql = "SELECT * FROM professoren"
37
38 # Kontrollausgabe der SQL-Abfrage
39 # print(sql)
40
41 # Absenden der SQL-Abfrage und Empfang des Ergebnisses
42 cursor.execute(sql)
43
44 # Ausgabe des Ergebnisses
45 for dsatz in cursor:
46     print(dsatz[0], dsatz[1], dsatz[2], dsatz[3], dsatz[4])
47
48 # Datensatz entfernen
49 sql = "DELETE FROM professoren WHERE PersNr = 2125"
50 cursor.execute(sql)
51 connection.commit()
52
53 # Ausgabe der Tabelle
```

```
54 sql = "SELECT * FROM professoren"
55 cursor.execute(sql)
56
57 for dsatz in cursor:
58     print(dsatz[0], dsatz[1], dsatz[2], dsatz[3], dsatz[4])
59
60 # Verbindung beenden
61 connection.close()
```

## Python und lokaler MySQL-Server

Dieser Fall ist sehr gut in den Dokumentation von Python aufgeführt. Benötigt wird lediglich das Modul *mysql.connector*, welche per `pip install mysql-connector-python` installiert werden kann.

```
1 import mysql.connector
2
3 mydb = mysql.connector.connect(
4     host="localhost",
5     user="dbuser",
6     password="informatik"
7 )
8
9 mydb.database = 'unidatenbank'
10 print(mydb) # Kontrolle erfolgreiche Kontaktaufnahme
11
12 # SQL-Befehl ausfuehren
13 cursor = mydb.cursor()
14 SQLBefehl = 'SELECT * FROM Professoren'
15 cursor.execute(SQLBefehl)
16
17 # Durchlaufen der Ergebnisse
18 row=cursor.fetchone()
19 while (row!=None):
20     print(row[0], row[1])
21     row = cursor.fetchone()
22
23 # Ende der Verarbeitung
24 cursor.close()
```

Alternativ kann man auch mit dem Modul MySQLdb arbeiten.

## Python und entfernter MySQL-Server

Diese Variante kommt immer dann zum Tragen, wenn die Daten und somit die Datenbank auf einem anderen Rechner arbeitet als der Client. Das kann z. B. ein DB-Server im Internet sein, oder auch ein Server im lokalen Netzwerk, z. B. einer Firma. Auch hier zeigt wieder das Beispiel (Quelltext unten, wie man den Zugriff realisiert. Eines ist jetzt zu beachten. Der Zugriff auf die Datenbank muss für den DB-User und auch der Datenbank selbst autorisiert werden. Ein DB-User wird in der Datenbank standardmäßig nur für den Zugriff von *localhost* eingerichtet. Weiterhin ist der Datenbank mitzuteilen, dass sie nicht nur Anfragen vom *Looback-Interface* (*127.0.0.1*) entgegennimmt. Im einzelnen ist folgendes durchzuführen:

1. Deaktiviere in der Datei `/etc/my.cnf` die Zeile `bind-address = 127.0.0.1` durch `#`
2. Gib dem DB-User das Recht von jedem Rechner auf den DB-Server zuzugreifen: `GRANT ALL ON <dbname>.* to '<dbuser>'@'%' IDENTIFIED BY '<password>' WITH GRANT OPTION;`  
(Die eckigen Klammern sind nicht miteinzugeben und die jeweiligen korrekten Bezeichner einzusetzen. Das '%' bedeutet Zugriff von jedem anderen Rechner ist erlaubt. )
3. `FLUSH PRIVILEGES;` (Der Inhalt des Speichers wird zur Datenbank übermittelt. )

```
1 import mysql.connector
2
3 mydb = mysql.connector.connect(
4     host="85.13.143.90", #Achtung! Ohne http://
5     user="d035c1e0",
6     password="informatik"
7 )
8
9 mydb.database = 'd035c1e0'
10 print(mydb) # Dient nur zur Kontrolle der erfolgreichen
11             Kontaktaufnahme
12
13 # SQL-Befehl ausfuehren
14 cursor = mydb.cursor()
15 SQLBefehl = 'SELECT * FROM Professoren'
16 cursor.execute(SQLBefehl)
17
18 # Durchlaufen der Ergebnisse
19 row=cursor.fetchone()
```



```
19 while (row!=None):
20     print(row[0], row[1])
21     row = cursor.fetchone()
22
23 # Ende der Verarbeitung
24 cursor.close()
```

- 1.1 Erklären Sie, warum dieses Programm funktioniert. Gehen Sie auch auf die auskommentierte Zeile 5 ein. (Hinweis: Die Methode `ToString()` macht das, was man hinter dem Namen vermuten könnte und ist für diese Aufgabe eher irrelevant.) ( /4 BE)
- 1.2 Neben den einfachen Datentypen (Werttypen) gibt es noch Collections. Zwei wichtige Vertreter sind Arrays und ArrayLists. Stellen Sie beide Datentypen im Vergleich (Gemeinsamkeiten und Unterschiede) dar. ( /6 BE)
- 1.3 Der abgebildete untere Quelltext zeigt die Verwendung von Arrays. Die Methode `CopyTo(Int32, Char[], Int32, Int32)` kopiert eine angegebene Anzahl von Zeichen von einer Startposition eines Arrays an eine Zielposition in einem zweiten Array. Die Methoden `ausgeben(char[])` und `ToCharArray()` machen das, was der Name vermuten lässt. Begründen Sie, warum in der letzten Zeile die Ausgabe `False` beträgt. Welche Vorschläge können Sie unterbreiten, dass zu korrigieren? (Ohne Implementierung!) ( /5 BE)

```
1 static void Main(string[] args){
2     char[] array1 = new char[args[0].Length];
3     args[0].CopyTo(0, array1,0,args[0].Length);
4     ausgeben(array1);
5     Console.Write("Bitte_gib_ein_Wort_ein:");
6     string eingabe = Console.ReadLine();
7     char[] array2 = eingabe.ToCharArray();
8     ausgeben(array2);
9     Console.WriteLine("Vergleichen:");
10    Console.WriteLine ("{0}",array1 == array2);} //Ausgabe False!!
```

## 2. Aufgabe: (Programmanalyse) ( /20 BE)

- 2.1 Das folgende Beispiel enthält mehrere syntaktische und logische Fehler. Finden und erläutern Sie diese Fehler (mind. 5)! ( /8 BE)
- 2.2 Sie haben nun alle Fehler gefunden. Was liefert das Programm? ( /5 BE)
- 2.3 Die Ketten von Einsen sollen nach ihrer Anzahl aufsteigend sortiert werden. Dabei soll jeweils eine Null zwischen den Einserketten stehen. Einserketten mit gleicher Anzahl sollen hintereinander durch Null getrennt stehen. Beispiel: 0111000111101111101111000000111 ~> 11101110111101111011111. Lösen Sie das Problem. (Die Darstellungsweise für Algorithmen ist Ihnen überlassen. Nutzen Sie die für Sie geeignetste.) ( /7 BE)

```
1 using System;
2 using System.Collections.Generic;
3
4 namespace WasWeissIchDenn
5 {
6     class WasWeissIchDenn_Class
7     {
8         static void Main(string[] args)
9         {
10             string b = "0111110001110111111011111";
11             string a = '0';
12             int zeichenfolge = 0;
13             char werte = b.ToCharArray();
14
15             foreach (var item in werte){
16                 if (item == '1' && item = a){
17                     zeichenfolge++
18                 }
19                 a = item;
20             }
21
22             if (zeichenfolge <= 1) {
23                 Console.ReadLine("Wahr: □{0}", zeichenfolge)
24             }
25             else{
26                 Console.WriteLine("Falsch: □{0}", zeichenfolge);
27             }
28         }
29     }
30 }
```

### 3. Aufgabe: Rekursion und Iteration ( /20 BE)

Eine interessante Rekursion ist das sogenannte Collatz-Problem:

$$f(n) = \begin{cases} 1 & , \text{ falls } n=0 \\ 1 & , \text{ falls } n=1 \\ f(\frac{n}{2}) & , \text{ falls } n \text{ gerade} \\ f(3n+1) & , \text{ falls } n \text{ ungerade} \end{cases} \quad n = 0, 1, 2, \dots$$

3.1 Definieren Sie den Begriff Algorithmus. Welche Eigenschaften kennzeichnen Algorithmen?

( /4 BE)

3.2 Berechnen Sie einige Funktionswerte und begründen Sie damit, dass Rekursion vorliegt.

( /5 BE)

3.3 Überführen Sie die Funktion in Pseudocode. ( /5 BE)

3.2 Gegeben ist ein iterative Methode für die Berechnung von Potenzen. Implementieren Sie eine rekursive Methode in C#. ( /6 BE)

```
1 using System;
2
3 namespace potenz
4 {
5     class Program
6     {
7         static void Main(string[] args){
8             Console.WriteLine("Hello World!");
9             Console.WriteLine(f1(5,3));
10        }
11
12        public static int f1(int b, int n){
13            int erg = b;
14
15            for (int i=1; i<n; i++){
16                erg*=b;
17            } //Ende der for-Schleife
18            return erg;
19        } //Ende der Methode f1
20    }
21 }
```

### Bewertung:

	Aufgabe 1	Aufgabe 2	Aufgabe 3		
maximal erreichbare BE	15	20	20	55	%
davon erreicht:					

Verstöße gegen die sprachliche und formale Korrektheit werden mit \_\_\_\_\_ Punkten berücksichtigt.





---

**erreichte Punktzahl:** \_\_\_\_\_ /55 BE    **Notenvorschlag:** \_\_\_\_\_ Punkte

Berlin, 18. Juni 2022