

Vortrag 0 Grundlagentheorie

Datenbanken

Lukas Wais

Codersbay

Version: 19. April 2023

Inhaltsverzeichnis

Einführung in Datenbanken

- Was ist eine Datenbank?

- Datenmanipulation

- Setup

- Datenbankentwurf - erste Einführung

Python

- Kurze Einführung für Java-Programmierer

Themen in diesem Kurs

1. Entity Relationship Diagramme
 2. Normalformen
 3. Datenmanipulation
 4. Relationale Algebra
 5. Fortgeschrittene Datenbankkonzepte
 6. Sicherheit von Datenbanken
 7. Zusätzliche Themen
- } Datenbankentwurf

Was ist eine Datenbank?

Was ist eine Datenbank? Und der Unterschied zu einer Tabellenkalkulation?

Definition

Datenbank

Eine Datenbank ist eine organisierte Sammlung von Informationen, die elektronisch in einem Computersystem gespeichert ist. Diese Informationen werden in einer strukturierten Weise dargestellt. Beispiele wären ein Graph oder eine relationale Struktur.

In den meisten Fällen werden Daten in Zeilen und Spalten innerhalb einer Tabelle gespeichert.

Der Unterschied zur Tabellenkalkulation

- ▶ Wie die Daten gespeichert werden
- ▶ Wer darauf zugreifen kann
- ▶ Größe und Skalierbarkeit

Datenbank Management System

DBMS

Eine Datenbank wird durch ein Datenbankmanagementsystem gesteuert. Die Daten und das DBMS zusammen werden oft einfach als Datenbank bezeichnet.

Unser DBMS der Wahl ist zu Beginn MariaDB

Example

Wir haben einen Student, wie können wir ihn in einer Datenbank speichern? Fragen, die man zuerst stellen sollte:

- ▶ Welche Eigenschaften habe ich?
- ▶ Wie kann ich einen Datensatz identifizieren?
- ▶ Habe ich unnötige Duplikate?

Example

Wir haben einen Student, wie können wir ihn in einer Datenbank speichern? Fragen, die man zuerst stellen sollte:

- ▶ Welche Eigenschaften habe ich?
- ▶ Wie kann ich einen Datensatz identifizieren?
- ▶ Habe ich unnötige Duplikate?

Vorname	Nachname	Studienkennzahl	Matrikelnummer
Micah	Obrien	033 551	811233
Alfie	Hyde	033 521	372435
Bibi	Banks	066 904	344957
Fredrick	Barrett	033 290	711778

CRUD Operationen

- ▶ Create
- ▶ Read
- ▶ Update
- ▶ Delete

SQL

Für die Bearbeitung von Daten in relationalen Datenbanken wird die Standardabfragesprache SQL verwendet. Sie wurde in den 1970er Jahren bei IBM, wobei Oracle einen wichtigen Beitrag leistete. Es gibt einen [SQL-Standard](#), der von den DBMS-Anbietern erweitert wird. SQL ist der menschlichen Sprache nachempfunden.

CRUD in SQL

CRUD	SQL
Create	INSERT
Read	SELECT
Update	UPDATE
Delete	DELETE

SQL Example

Was bewirkt die Abfrage?

```
SELECT * FROM students
```

SQL Example

Was bewirkt die Abfrage?

```
SELECT * FROM students
```

Es gibt jedes Element aus der Tabelle *students* aus.

SQL Example

Was bewirkt die Abfrage?

```
SELECT * FROM students
```

Es gibt jedes Element aus der Tabelle *students* aus.

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Was wir in diesem Kurs verwenden

- ▶ MariaDB als DBMS
- ▶ Podman; für das DBMS
 - ▶ podman-compose
- ▶ Python 3
 - ▶ Jupyter Notebooks
- ▶ Einige zusätzliche Technologien je nach Interesse

Was ist MariaDB?

MariaDB Server ist eine der beliebtesten relationalen Open-Source-Datenbanken. Sie wurde von den ursprünglichen Entwicklern von MySQL entwickelt und wird garantiert quelloffen bleiben. Sie ist Teil der meisten Cloud-Angebote und Standard in den meisten Linux Distributionen. ¹

¹Zitiert von <https://mariadb.org/> und aufgerufen am 04.10.2022

Wie kann eine Schule modelliert werden?

Einführung in Python

- ▶ Interpretiert
- ▶ High-level
- ▶ Einfache Syntax ohne Klammern und Semikolons
- ▶ Riesige Anzahl von Bibliotheken
- ▶ Jupyter Notebooks
- ▶ Häufig für wissenschaftliche Berechnungen verwendet → einfaches Plotten von Diagrammen

Hello World

```
print("Hello World")
```

Dynamische Typisierung

```
number = "two"  
number = number + 2 # NOT possible, number is not a string  
number = 2  
number = number + 2 # possible, the type changed dynamically
```

Funktionen for-Schleifen

```
def my_function(numbers):  
    for i in range(numbers):  
        print(i)
```

```
my_function(10)
```

Iteration von Listen

```
def my_function(elements):  
    for element in elements:  
        print(element)
```

```
fruits = ["apple", "pear", "cherry"]  
my_function(fruits)
```

Person-Objekt mit Konstruktor und str-Methode

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"{self.name}({self.age})"
```

```
p1 = Person("John", 36)
```

```
print(p1)
```

Der Parameter *self* ist ein Verweis auf die aktuelle Instanz der Klasse und wird für den Zugriff auf Variablen verwendet, die zu der Klasse gehören.

