Datenspeicherung, Datenbanken

Ab Kapitel 9.1, 9.2 in Tigerjython: Datenbanken und SQL

- Textdateien als Datenspeicher
- Übungen

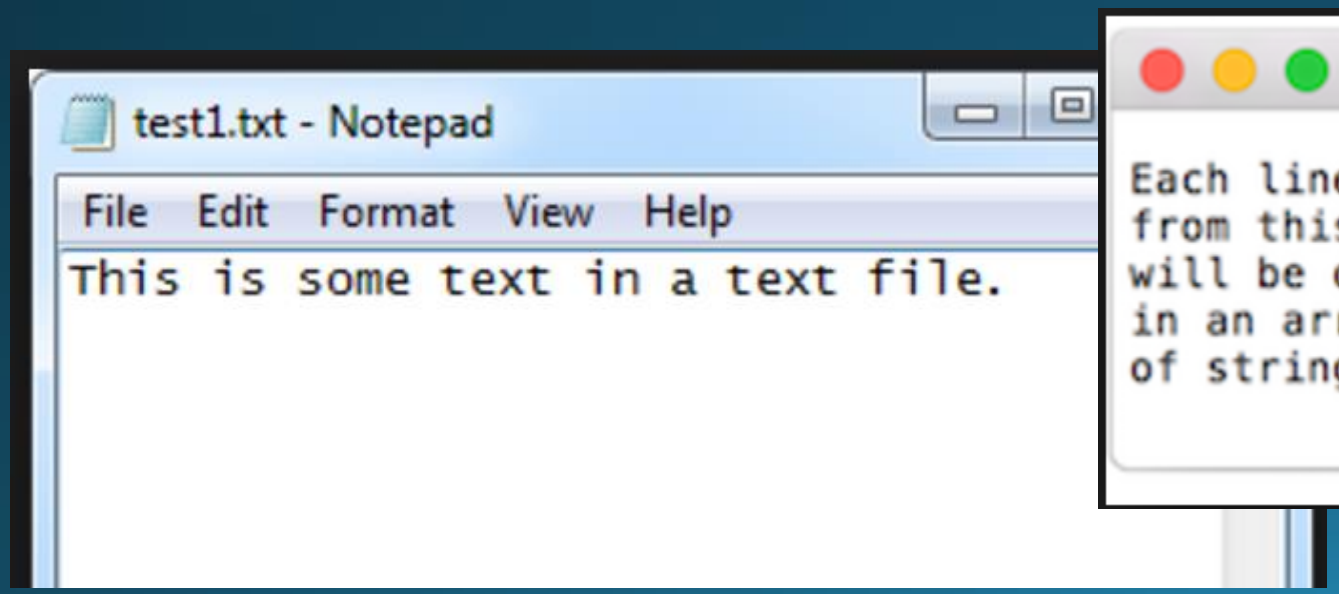
- Datenbanken und SQL Sprache
- Übungen

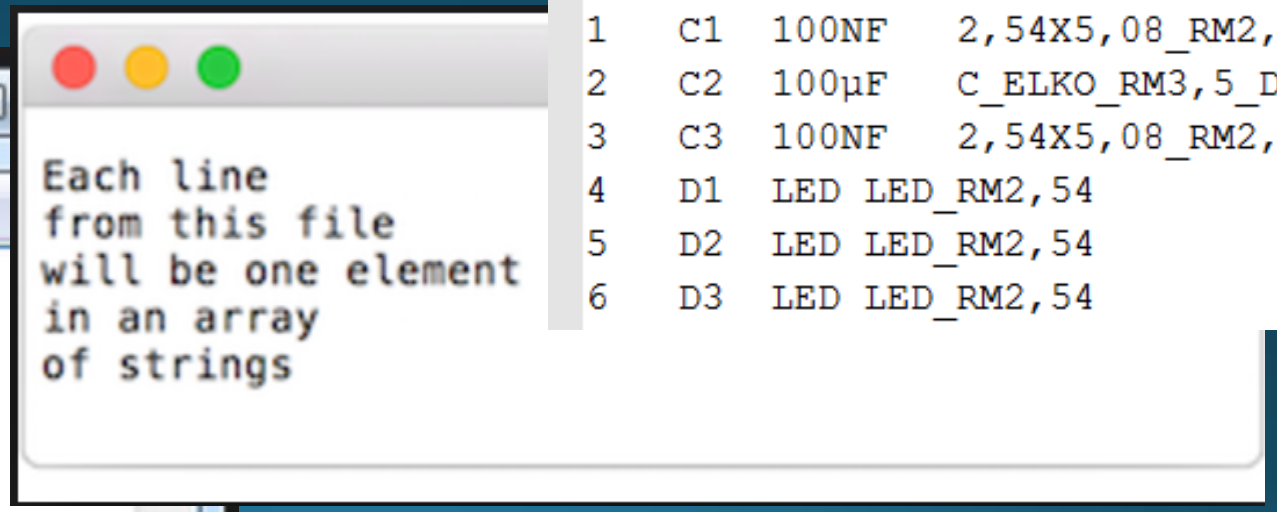
Textdateien als Datenspeicher

- Wenn Daten in einer Datei einfach hintereinander stehen, spricht man oft von einem «Flat File» -> es gibt keine Struktur darin

-> alles auf einer Zeile? Zeilenumbrüche?

-> alle Daten zuerst in Strings umwandeln






```
; Platine=<alle Bauteile>
; Autor=
;
Pos Name    Wert     Gehäuse
1   C1  100NF   2,54X5,08_RM2,54
2   C2  100µF   C_ELKO_RM3,5_DM8
3   C3  100NF   2,54X5,08_RM2,54
4   D1  LED LED_RM2,54
5   D2  LED LED_RM2,54
6   D3  LED LED_RM2,54
```

CSV Dateien (Comma-Separated Values)

- Bedeutung der Felder ist in der ersten Zeile
- 1 Eintrag pro Zeile, immer gleiche Struktur
- Trennzeichen zwischen den Feldern definiert
- Evtl. Felder in " " eingeschlossen

```
Name;Vorname;Wohnort;Geschlecht;Alter
Rossi;Angela;Rom;w;14
Habluetzel;Peter;Zuerich;m;31
Karlsson;Anna;Berlin;w;45
Dvorak;Milos;Prag;m;26
```

Übungen

Löse die Aufgaben auf dem Blatt:

Aufgabe 1: Palindrome

Aufgabe 2: CSV Dateien **a) bis c)**
d) ist Hausaufgabe

Relationale Datenbank (Wikipedia)

9.2 in Tigerjython: Datenbanken und SQL

«Eine relationale Datenbank dient zur elektronischen Datenverwaltung in Computersystemen und beruht auf einem tabellenbasierten relationalen Datenbankmodell.»

«Zum Abfragen und Manipulieren der Daten wird überwiegend die Datenbanksprache SQL (Structured Query Language) eingesetzt.»

Relation

Beispiel einer Relation „Buch“:

Buch-ID	Autor	Verlag	Verlagsjahr	Titel	Datum
1	Hans Vielschreiber	Musterverlag	2007	Wir lernen SQL	13.01.2007
2	J. Gutenberg	Gutenberg und Co.	1452	Drucken leicht gemacht	01.01.1452
3	G. I. Caesar	Handschriftverlag	−44	Mein Leben mit Asterix	16.03.−44
5	Galileo Galilei	Inquisition International	1640	Eppur si muove	1641
6	Charles Darwin	Vatikan-Verlag	1860	Adam und Eva	1862

Relationen sind verknüpft

Relation „Nutzer“

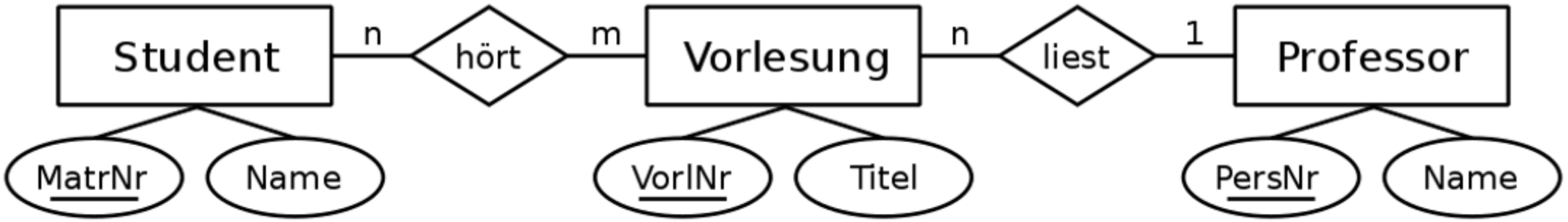
Nutzer-ID	Vorname	Nachname
10	Hans	Vielschreiber
11	Jens	Mittelleser
12	Erich	Wenigleser

Relation „Entliehen“

Nutzer-ID	Buch-ID
10	1
10	2
10	3
12	5
12	6

Relationen sind verknüpft

ER-Diagramm:



Relationen:

Student

MatrNr	Name
26120	Fichte
25403	Jonas
27103	Fauler

hört

MatrNr	VorlNr
25403	5001
26120	5001
26120	5045

Vorlesung

VorlNr	Titel	PersNr
5001	ET	15
5022	IT	12
5045	DB	12

Professor

PersNr	Name
12	Wirth
15	Tesla
20	Urlauber

SQL Sprache

- https://de.wikipedia.org/wiki/SQL

Vorlesung

VorlNr	Titel	PersNr
5001	ET	15
5022	IT	12
5045	DB	12

Abfrage mit Filter [Bearbeiten | Quelltext bearbeiten]

```
SELECT VorlNr, Titel
FROM Vorlesung
WHERE Titel = 'ET';
```

listet *VorlNr* und *Titel* aller derjenigen Zeilen der Tabelle *Vorlesung* auf, deren Titel 'ET' ist.

SQL Sprache

- https://de.wikipedia.org/wiki/SQL

Vorlesung

VorlNr	Titel	PersNr
5001	ET	15
5022	IT	12
5045	DB	12

Abfrage mit Filter [Bearbeiten | Quelltext bearbeiten]

```
SELECT VorlNr, Titel
FROM Vorlesung
WHERE Titel = 'ET';
```

listet *VorlNr* und *Titel* aller derjenigen Zeilen der Tabelle *Vorlesung* auf, deren Titel 'ET' ist.

Ergebnis:

VorlNr	Titel
5001	ET

Übungen - Personentabelle 9.2

PERSONENTABELLE MIT CREATE ERZEUGEN

Als exemplarisches Beispiel betrachtest du die Erfassung von Personendaten in einer Schule und die Verarbeitung von Sportleistungsdaten eines Schulsporttages. Dabei werden die erfassten Daten stark vereinfacht. Du verwendest hier als Datenbanksystem **SQLite**, das eine grosse Verbreitung auf vielen Rechnertypen hat, insbesondere auch auf mobilen Geräten.

Die Personendaten speicherst du in einer Tabelle mit dem Namen *person*. Für jede Person gibt es einen Zeileneintrag, auch **Datensatz** (**record**) genannt. Die Spalten (auch **Felder** genannt) enthalten die Eigenschaften (**Attribute**) der Datensätze.

Attribute (Felder)

Datensätze (records)

personid	familienname	vorname	wohnort	geschlecht	jahrgang
1	Meier	Peter	Gümligen	m	2001
2	Flückiger	Marc	Bern	m	2003
3	Huber	Anna	Bern	w	2003
4	Bauer	Paul	Muri	m	2000
5	Zwahlen	Noe	Ostermundigen	m	2002
6	Meier	Nina	Wohlen	w	2001

DB erzeugen und löschen

```
from sqlite3 import *

with connect("schule.db") as con:
    cursor = con.cursor()
    sql = """CREATE TABLE person
             (personid INTEGER PRIMARY KEY,
             familienname VARCHAR,
             vorname VARCHAR,
             wohnort VARCHAR,
             geschlecht VARCHAR,
             jahrgang INTEGER)"""
    cursor.execute(sql)
print "Done"
```

```
from sqlite3 import *

with connect("schule.db") as con:
    cursor = con.cursor()
    cursor.execute("DROP TABLE person")
print "Done"
```

Arbeiten mit Records

```
from sqlite3 import *

with connect("schule.db") as con:
    cursor = con.cursor()
    sql = """INSERT INTO person
             (familienname, vorname, wohnort, geschlecht, jahrgang)
             VALUES ('Huber', 'Anna', 'Bern', 'w',  2002)"""
    cursor.execute(sql)
print "Done"
```

Programmcode marki

Das Modul *prettytable* hilft dir, Tabellendaten als String zu formatieren und im Ausgabefenster auszuschreiben.

```
from sqlite3 import *
from prettytable import printTable

con = connect("schule.db")
with con:
    cursor = con.cursor()
    cursor.execute("SELECT familienname, vorname, wohnort FROM person")
    printTable(cursor)
```

Übung

Aufgabe: erstelle die schule.db und fülle die Daten wie in der Tabelle ein (Programme sind alle im Buch)

Löse die Aufgaben auf dem Blatt:

Aufgabe 3: a) und b)

Aufgabe 4: freiwillig

Hausaufgabe: 3c)

Ein SQL Datenbank System