

Aufgabe 1

Sie möchten in einer Tabelle in einer Datenbank speichern, an welchem Datum Sie sich mit welcher Person an welchem Ort getroffen haben.

1. Geben Sie das Schema dieser Datenbank.
2. Geben Sie ein Beispiel für eine Instanz dieses Schemas.

Aufgabe 2

1. Erweitern Sie das Beispiel aus der letzten Aufgabe: Sie wollen in der selben Tabelle auch noch die Telefonnummer der Personen speichern.
2. Erklären Sie anhand dieses Beispiels Redundanz in einer Datenbank.
3. Erklären Sie, wie aus dieser Redundanz Inkonsistenz entstehen kann.

Aufgabe 3

Gegeben seien die Tabellen *instructor* und *department* als CSV-Dateien. Implementieren Sie folgende Abfragen einer Ihnen genehmen Programmiersprache:

1.

```
select name
  from instructor
 where ID = '22222'
```
2.

```
select building
  from   instructor, department
 where  instructor.dept_name = department.dept_name and
        name = 'Einstein'
```

Warum nutzt man eine deklarative Sprache wie SQL zur Datenbankabfrage?
Warum nicht eine prozedurale Programmiersprache wie Java oder Python?

Aufgabe 4

Gegeben sei ein Array in der globalen Variable **a** mit Kontoständen in nicht-flüchtigem Speicher. Es soll die Integritätsbedingung gelten, dass die Summe aller Werte des Arrays gleich bleibt. Nun soll das erste Konto geleert und auf das zweite Konto überwiesen werden. Dazu gibt es die Funktion:

```
void transact(int[] a)
{
    a[1] = a[1] + a[0];
    a[0] = 0;
}
```

Erklären Sie, wie bei einem Systemabsturz die Integritätsbedingung verletzt werden kann.

Erweitern Sie die Funktion auf geeignete Weise, um sicherzustellen, dass die Integritätsbedingung zu jedem Zeitpunkt gilt.