

Databases Auftrag 1

Aufgabe 1

```
create table kontakte(id integer, name text, datum text, ort text);
```

ID	NAME	DATUM	ORT
1	Karl	2021-02-12 13:55:00	Bahnhof Bern
2	Lydia	2021-02-12 16:23:31	Lydia's Zuhause
3	Karl	2021-02-23 19:52:03	Tramdepot

Aufgabe 2

```
create table kontakte(id integer, name text, datum text, ort text, telefonnr text);
```

Die Telefonnummer wird immer zum gleichen Namen gehören, dieser Wert ist also redundant.

Eine Inkonsistenz kann entstehen, wenn nun Karls Telefonnummer ändert, man aber vergisst, diese Telefonnummer in jeder Zeile, in der Karl vorkommt, zu aktualisieren.

Aufgabe 3

Code: siehe Projekt im Zip

Wird eine deklarative Sprache wie SQL verwendet, ist der Datenbank selber überlassen, wie die Daten selektiert werden. Dies ermöglicht der Datenbank, ihre Performance selber zu optimieren. Bei der Anwendung einer prozeduralen Sprache hingegen gibt es für den Entwickler mehr zu tun, die Abfragen sind fehleranfälliger, da sie selbst geschrieben werden müssen und es gibt mehr Code. Eine deklarative Sprache ist ausserdem kürzer und lesbarer.

Aufgabe 4

Sollte ein Problem auftreten, kann es passieren, dass der Betrag nur beim ersten Konto hinzuaddiert wird, beim zweiten Konto aber nie abgezogen wird. Somit würde die Integritätsbedingung verletzt.

Der Return-Typ der transact-Funktion sollte von «void» auf «int[]» geändert werden. Das Array, das in die Funktion übergeben wird, ist eine Kopie des ursprünglichen Arrays. Dieses Array wird dann modifiziert. Schlägt dieser Prozess fehl, ist das originale Array unberührt. Ist der Prozess erfolgreich verlaufen, wird die modifizierte Kopie des Arrays zurückgegeben und überschreibt dann das originale Array. So passiert eine Änderung «ganz oder gar nicht».

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