



## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS18/19

Moritz Sichert, Lukas Vogel (gdb@in.tum.de)

<https://db.in.tum.de/teaching/ws1819/grundlagen/>

### Blatt Nr. 01

#### Hausaufgabe 1

Als moderner Netzbürger besitzen Sie einen Twitter-Account mit 150 Followern. Nach der Vorlesung schreiben Sie folgenden Tweet:

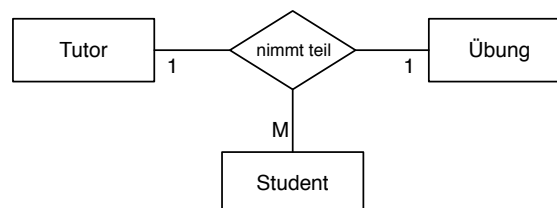


- a) Welche Daten werden durch das Absenden dieses Tweets generiert? An welchen Stellen im System von Twitter könnten diese gespeichert werden? Denken Sie nicht nur an den Text des Tweets sondern auch an potentielle Metadaten.

Welche zusätzlichen Daten werden generiert, wenn Ihr Sitznachbar auf das Herz ("Like") klickt?

- b) Seit dem 25.5.2018 gilt die Datenschutz-Grundverordnung (DSGVO), wegen der u.A. personenbezogene Daten auf Anfrage gelöscht werden müssen. Sie sind bei Twitter zuständig, diese Anfragen zu bearbeiten. Überlegen Sie sich, welche Daten bei so einer Anfrage gelöscht werden müssen und welche Probleme dabei auftreten können.

#### Hausaufgabe 2



Ignorieren Sie die Funktionalitätsangaben und beantworten Sie:

- a) Wie viele partielle Funktionen der Form  $A \times B \rightarrow C$  können in einer ternären Beziehung auftreten (Ignorieren Sie beim Zählen die Reihenfolge auf der linken Seite der Beziehung).
- b) Nennen Sie alle möglichen partiellen Beziehungen in der hier gezeigten Beziehung „nimmt teil“.

- c) Nennen Sie für jede Funktion in Prosa, welche Einschränkung diese darstellt, falls sie gilt.

Unter Berücksichtigung der Funktionalitätsangaben:

- d) Welche partiellen Funktionen gelten hier?

### **Gruppenaufgabe 1 (nicht Zuhause vorbereiten)**

Sie designen eine Webanwendung zur Univerwaltung. Früh entschließen Sie sich zum Einsatz eines Datenbanksystems als Backend für Ihre Daten. Ihr Kollege ist skeptisch und würde die Datenverwaltung lieber selbst implementieren. Überzeugen Sie ihn von Ihrem Entschluss. Finden Sie stichhaltige Antworten auf die folgenden von Ihrem Kollegen in den Raum gestellten Äußerungen:

- a) Die Installation und Wartung eines Datenbanksystems ist aufwendig, die Erstellung eines eigenen Datenformats ist straight-forward und flexibler.
- b) Mehrbenutzersynchronisation wird in diesem Fall nicht benötigt.
- c) Es ist unsinnig, das jeder Entwickler zunächst eine eigene Anfragesprache (SQL) lernen muss, nur um Daten aus der Datenbank zu extrahieren.
- d) Redundanz ist hilfreich, wieso sollte man auf sie verzichten?

### **Gruppenaufgabe 2 (nicht Zuhause vorbereiten)**

Finden Sie ein Beispiel für ein Problem (bzw. eine Inkonsistenz), die auftreten kann, wenn unkontrolliert parallel auf Daten zugegriffen wird. Ein traditionelles Beispiel hierfür ist eine gegenseitige Bank-Überweisung zwischen zwei Konten A und B. Wenn A einen Betrag  $x$  zu B überweist und B einen Betrag  $x'$  zu A, sollte immer gelten  $Kontostand(A) + Kontostand(B)$  ist konstant, da sonst Geld verschwunden ist. Konstruieren Sie einen Ablauf zweier gegenseitiger Überweisungen, bei dem die Eigenschaft, dass die Kontostandssumme konstant sein soll nach dem Abschluss der zwei Überweisungen verletzt ist.