



## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS19/20

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

<https://db.in.tum.de/teaching/ws1920/grundlagen/>

### Blatt Nr. 06

Tool zum Üben von SQL-Anfragen: <https://hyper-db.com/interface.html>.

### Hausaufgabe 1

*Diese Aufgabe ist die dritte in einer Reihe von Aufgaben, in denen Sie lernen werden, einen gegebenen Sachverhalt zu analysieren, geeignete Modelle zu entwerfen, diese in ein Datenbankschema zu überführen, das Schema in einer Datenbank aufzusetzen und mit Daten zu befüllen. Sie werden ebenfalls lernen, wie Sie die Datenbank für bestimmte Anfragen optimieren können.*

Gehen Sie in dieser Aufgabe von dem von Ihnen verfeinerten relationalen Schema des Onlineshops aus der letzten Aufgabe aus. Sollten Sie nicht im Besitz einer zufriedenstellenden Lösung sein, können Sie auch die am Freitag Nachmittag veröffentlichte Beispiellösung als Ausgangspunkt verwenden.

- a) Geben Sie DDL-Anweisungen an, die die für den Onlineshop notwendigen Relationen erstellen.
- b) Ergänzen Sie Ihre DDL-Anweisungen aus Teilaufgabe a) um die Angabe der Primär- und Fremdschlüsseln.

Zusätzlich zu den in Blatt 4 vorgestellten Anforderungen muss das Datenbankschema nun auch DSGVO-konform werden. Wir nehmen vereinfachend nur folgende Forderung an: *Kunden des Onlineshops müssen die Löschung all ihrer personenbezogenen Daten verlangen können.* Achten Sie außerdem darauf, dass die für die Buchhaltung notwendigen Informationen erhalten bleiben müssen (z.B. aus steuerrechtlichen Gründen).

- c) Erweitern Sie Ihre DDL-Anweisungen so, dass die oben genannten Anforderungen erfüllt werden. Das bedeutet also, dass Kunden gelöscht werden können, ihre Bestellungen aber *anonymisiert* im System gespeichert bleiben.
- d) Fügen Sie Ihrer Datenbank nun mittels DML-Anweisungen die minimale Anzahl an Tupeln hinzu, sodass es einen Kunden und eine ihm zugeordnete Bestellung gibt. Dabei muss sichergestellt sein, dass keine referentiellen Integritätsbedingungen verletzt werden. Löschen Sie dann den Kunden aus der Datenbank und überprüfen Sie, ob Teilaufgabe c) erfolgreich in der Datenbank umgesetzt ist.
- e) *Zusatzaufgabe:* Erstellen Sie ein SQL-Skript, welches alle Tabellen in einer neuen Datenbank anlegt. Sollte die Datenbank bereits Tabellen mit gleichem Namen enthalten, so sollen diese zunächst *restlos* gelöscht werden. Testen Sie das Skript in Ihrer aus dem letzten Blatt aufgesetzten (lokalen) Datenbank.

### Hausaufgabe 2

„Bekanntheitsgrad“: Formulieren Sie eine SQL-Anfrage, um den Bekanntheitsgrad von Studenten zu ermitteln. Gehen Sie dabei davon aus, dass Studenten sich aus gemeinsam besuchten Vorlesungen kennen. Sortieren Sie das Ergebnis absteigend nach Bekanntheitsgrad!

### Hausaufgabe 3

Gegeben sei die Relation `Fahrplan`, die strukturell dem folgenden Beispiel gleicht:

Von	Nach	Linie	Abfahrt	Ankunft
Garching, Forschungszentrum	Garching	U6	09:06	09:09
Garching	Garching-Hochbrück	U6	09:09	09:11
Garching-Hochbrück	Fröttmaning	U6	09:11	09:15
...	...			
Fröttmaning	Garching-Hochbrück	U6	09:00	09:04
Garching-Hochbrück	Garching	U6	09:04	09:06
Garching	Garching, Forschungszentrum	U6	09:06	09:09
...	...			
Garching, Forschungszentrum	Technische Universität	690	17:56	17:57

Formulieren Sie die folgenden Anfragen auf dieser Relation in SQL. Sie können dabei den Wert `CURRENT_TIME` für die aktuelle Zeit verwenden. Um einen Zeitwert zu einer Zahl umzuwandeln, verwenden Sie folgende Funktion:

```
EXTRACT(EPOCH FROM x - '00:00:00'::TIME)
```

Diese gibt die Anzahl der Sekunden zurück, die zwischen dem Zeitwert `x` und Mitternacht liegen. Mit dieser Funktion kann auch die Anzahl der Sekunden zwischen zwei beliebigen Zeitwerten berechnet werden.

Fallunterscheidungen können Sie mit dem `case`-Konstrukt ausdrücken:

```
select
  case
    when semester=1 then 'Ersti'
    when semester<=6 then 'Bachelor'
    else 'Master'
  end
from
  Studenten
```

- Geben Sie alle Einträge des Fahrplans aus, deren Abfahrtshaltestelle das Wort „Garching“ enthält.
- Geben Sie alle Einträge des Fahrplans mit dem zusätzlichen Attribut „Verkehrsmittel“ aus. Alle Linien, die mit „U“ anfangen, haben das Verkehrsmittel „U-Bahn“, alle die mit „S“ anfangen „S-Bahn“, und alle restlichen „Bus/Tram“.
- Finden Sie alle Abfahrten ab „Garching, Forschungszentrum“, die Sie heute noch erreichen können. Sortieren Sie das Ergebnis aufsteigend nach Abfahrtszeit.
- Finden Sie alle Fahrten zwischen zwei Haltestellen (nicht-transitiv), die mindestens drei und höchstens fünf Minuten dauern.

Bonus: Berücksichtigen Sie, dass Fahrten über Mitternacht möglich sind, z.B. Abfahrt um 23:58 Uhr und Ankunft um 00:02 Uhr.

Laden Sie zum Testen entweder die SQL-Datei von der Übungswebseite in Ihr lokal installiertes Datenbanksystem oder verwenden Sie die Webschnittstelle.

#### **Hausaufgabe 4**

Formulieren Sie die folgende Anfrage auf dem bekannten Unischema in SQL: Ermitteln Sie für jede Vorlesung, wie viele Studenten diese vorgezogen haben. Ein Student hat eine Vorlesung vorgezogen, wenn er in einem früheren Semester ist als der „Modus“ der Semester der Hörer dieser Vorlesung. Der Modus ist definiert als der Wert, der am häufigsten vorkommt – für diese Anfrage also das Semester, in dem die meisten Hörer dieser Vorlesung sind. Falls es mehrere Semester dieser Art gibt, soll nur das niedrigste zählen.

Beachten Sie, dass auch Vorlesungen ohne Hörer, sowie Vorlesungen deren Hörer alle im gleichen Semester sind, ausgegeben werden sollen.

Geben Sie für jede Vorlesung die Vorlesungsnummer, den Titel und die Anzahl der „Vorzieher“ aus.