



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. 05

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

Hausaufgabe 1

Diese Aufgabe ist die zweite 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 füllen. Sie werden ebenfalls lernen, wie Sie die Datenbank für bestimmte Anfragen optimieren können.

Benutzen Sie im Folgenden Ihr ER-Modell des Onlineshops aus dem vorangegangenen Blatt. Falls Sie nicht im Besitz einer zufriedenstellenden Lösung sind, können Sie auch die am Freitag Nachmittag veröffentlichte Beispiellösung als Ausgangspunkt verwenden.

- Übertragen Sie das ER-Modell in ein relationales Schema.
- Verfeinern Sie das relationale Schema soweit möglich durch Eliminierung von Relationen.
- Finden Sie für die Attribute sinnvolle und von Postgres unterstützte Datentypen. Beziehen Sie auch die Abhängigkeiten zwischen den einzelnen Relationen in Ihre Überlegungen ein. Eine Liste mit unterstützten Datentypen finden Sie hier: (<https://www.postgresql.org/docs/current/datatype.html>).

Hausaufgabe 2

Formulieren Sie folgende Anfragen auf dem bekannten Universitätsschema in SQL. Geben Sie alle Ergebnisse duplikatfrei aus.

- Finden Sie die *Studenten*, die Sokrates aus *Vorlesung(en)* kennen.
- Finden Sie die *Studenten*, die *Vorlesungen* hören, die auch Fichte hört.
- Finden Sie die *Assistenten* von *Professoren*, die den Studenten Carnap unterrichtet haben – z.B. als potentielle Betreuer seiner Bachelorarbeit.
- Geben Sie die Namen der *Professoren* an, die Theophrastos aus *Vorlesungen* kennt.
- Welche *Vorlesungen* werden von *Studenten* im Bachelorstudium (1. – 6. Semester) gehört? Geben Sie die Titel dieser *Vorlesungen* an.
- Bestimmen Sie für jede Vorlesung wie viele Studenten diese hören. Geben Sie auch Vorlesungen ohne Hörer aus. Sortieren Sie das Ergebnis absteigend nach Anzahl der Hörer.

Hausaufgabe 3

Bestimmen Sie für alle Studenten eine gewichtete Durchschnittsnote ihrer Prüfungen. Die Gewichtung der einzelnen Prüfungen erfolgt gemäß dem Vorlesungsumfang (SWS). Dies entspricht dem Verfahren der Durchschnittsnotenberechnung für Ihr Bachelor-Zeugnis.

HINWEIS: Wenn Sie das Ergebnis Ihrer Anfrage besser testen wollen, können Sie eine größere *prüfen*-Relationenausprägung benutzen. Fügen Sie dazu das folgende SQL-Statement vor ihrer Anfrage ein. Verwenden Sie dann `pruefenxl` statt `pruefen`:

```
with pruefenxl(matrn, vorlnr, persnr, note) as (
  select * from pruefen
  union all
  values
    (25403, 5049, 2126, 1),
    (26120, 5001, 2137, 1),
    (26120, 5043, 2126, 3),
    (26120, 5052, 2126, 4),
    (26120, 4630, 2137, 1)
)
```

Hausaufgabe 4

Folgender Ausdruck im Tupelkalkül gibt alle Studenten aus, die alle von ihnen gehörten Vorlesungen bestanden haben.

$$\{s \mid s \in \text{Studenten} \wedge \\ \forall h \in \text{ hoeren}(h.\text{MatrNr} = s.\text{MatrNr} \Rightarrow \\ \exists p \in \text{ pruefen}(p.\text{MatrNr} = s.\text{MatrNr} \wedge p.\text{VorlNr} = h.\text{VorlNr} \wedge p.\text{Note} \leq 4))\}$$

Übersetzen Sie diese Anfrage nun in SQL. Da SQL keine Allquantoren und Implikationen unterstützt, müssen Sie sie dazu zunächst umformen.

- Formen Sie den Ausdruck in einen Äquivalenten um, der keine Implikationen oder Allquantoren verwendet.
- Übersetzen Sie den so erlangten Ausdruck in SQL. Testen Sie ihn in der Webschnittstelle.

Zusatzaufgabe 5

Diese Aufgabe wird nicht in der Übung besprochen. Bei Installationsschwierigkeiten besuchen Sie bitte die Tutorsprechstunde am Mittwoch nach der Vorlesung.

In dieser Aufgabe richten Sie ein Datenbanksystem auf Ihrem eigenen Rechner ein, um später auch komplexe Anfragen an unseren Onlineshop aus den vorherigen Aufgaben auswerten zu können. Wir empfehlen hierfür Postgres, da es die gleiche Syntax wie HyPer benutzt, quellenoffen ist und vergleichsweise einfach installiert werden kann.

Die folgende Installationsanweisungen beziehen sich auf Ubuntu, wobei Postgres auch auf Windows und MacOS installiert werden kann. **Bei Installationsschwierigkeiten besuchen Sie bitte die Tutorsprechstunde am Mittwoch nach der Vorlesung.**

```
$ sudo apt install postgresql
```

Den Datenbankserver kann man nun mit folgenden Befehlen steuern:

```
$ sudo systemctl (start|stop|restart|reload|status) postgresql
```

Wir legen nun einen neuen Benutzer an, mit dem wir uns künftig in `psql` einloggen werden. Als Nutzernamen benutzen Sie am besten Ihren Username, mit dem Sie auch in Ubuntu eingeloggt sind.

```
sudo -u postgres createuser -P <nutzernamen>
```

Im nächsten Schritt wechseln wir wieder zum User `postgres` und können nun unsere erste Datenbank anlegen, die wir `universitaet` nennen. Außerdem müssen wir den neuen Benutzer berechtigen, diese Datenbank verwenden zu dürfen:

```
$ sudo -u postgres psql
```

```
postgres=# CREATE DATABASE universitaet;
```

```
postgres=# GRANT ALL ON DATABASE universitaet to <nutzernamen>;
```

Die `psql`-Umgebung kann mit `\q` wieder verlassen werden.

Wir importieren nun das aus der Vorlesung bekannte Unischema. Dazu verlassen wir `psql` und laden zunächst die folgende SQL Datei von der Vorlesungswebsite herunter:

```
$ wget https://db.in.tum.de/teaching/ws1920/grundlagen/uni_mysql.sql
```

Nun erstellen wir ein neues Schema in der Datenbank und importieren alle Tabellen und Daten, die in der Datei `uni_mysql.sql` angegeben sind.

```
$ psql universitaet < uni_mysql.sql
```

Ab sofort können wir uns über `psql` mit der Datenbank verbinden:

```
$ psql universitaet
```

Nun können wir in der Datenbank neue Tabellen erstellen, diese mit Daten befüllen und Anfragen ausführen.