

Einführung in Datenbanken

— Übungsblatt 5 (Schema-Entwurf, Logik) —

Ihre Lösungen laden Sie bitte in die Übungsplattform in StudIP hoch ([StudIP-Eintrag der Vorlesung], Reiter „Übungsplattform“).

Einsendeschluss ist Montag, der 18.11.2024, 18⁰⁰.

Hausaufgaben können einzeln oder in Zweier-Gruppen bearbeitet werden. Sie können die Gruppe für jede Aufgabe neu wählen.

Denken Sie daran, dass Sie bei jeder Aufgabe angeben müssen, ob Sie bereit sind, vorzurechnen:

- „VORRECHNEN:0“: Ich werde nicht zur Übung kommen.
- „VORRECHNEN:1“: Ich möchte diese Aufgabe nicht vorrechnen.
- „VORRECHNEN:2“: Ich möchte diese Aufgabe nur ungern vorrechnen.
- „VORRECHNEN:3“: Ich kann vorrechnen, lasse aber gern anderen den Vortritt.
- „VORRECHNEN:4“: Ich kann problemlos vorrechnen.
- „VORRECHNEN:5“: Ich möchte gerne, dass meine Abgabe besprochen wird.

Falls Sie als Gruppe abgeben, muss jedes Gruppenmitglied einzeln die Bereitschaft zum Vorrechnen erklären (VORRECHNEN1:N ist der Wert für den Studierenden, der die Aufgabe in die Übungsplattform hochgeladen hat, und VORRECHNEN2:M der Wert für den anderen Studierenden).

Vergessen Sie nicht, eventuell verwendete Quellen wie ChatGPT (oder auch die Lösung eines anderen Studierenden) anzugeben (mit der in Kapitel 0, Folien 22 bis 27 beschriebenen Codierung). Ohne Quellenangabe werden „zu ähnliche Lösungen“ als Plagiat behandelt. Mit Quellenangabe werden sie normal korrigiert. In der Klausur müssen Sie aber ähnliche Aufgaben ohne Hilfe lösen! Sie sind erwachsene Menschen und können sich ja denken, dass Sie nur durch eigene Beschäftigung mit den Aufgaben lernen.

Wir würden uns freuen, wenn Sie die für die jeweilige Aufgabe verwendete Zeit in Minuten in der Form „ZEIT:N“ angeben würden (bei Gruppenarbeit „ZEIT1:N“ und „ZEIT2:M“). Diese Angabe ist freiwillig.

Hinweis: SQL-Anfragen mit Syntaxfehlern werden automatisch mit 0 Punkten bewertet! Testen Sie also Ihre Anfragen (z.B. im Adminer).

Aufgabe 1 (6 Punkte)

Wählen Sie einen Fachvortrag vom Industrietag Informationstechnologie IT², der am 12.11.2024 von 14 bis 18 Uhr stattfindet. Die Vorträge sind von 14 bis 16 Uhr in Raum 3.07. Modellieren Sie eine im gewählten Vortrag beschriebene Anwendung, einen relevanten Teil, oder eine Beispiel-Anwendung aus dem Vortrag als relationales Datenbankschema mit etwa 2 bis 3 Tabellen.

- Geben Sie die Tabellen mit Primär- und Fremdschlüsseln in Kurznotation wie auf Folie 6-37 an.
- Geben Sie weiterhin für alle Tabellen Tupel an, die verdeutlichen, was der Inhalt der einzelnen Spalten ist. Der gesamte Datenbankzustand soll etwa 8–12 Tupel umfassen.
- Beschreiben Sie knapp und verständlich, was die Attribute der Tabellen bedeuten sollen (soweit es aus den Spaltennamen und den Beispieldaten nicht schon völlig klar ist).
- Beschreiben Sie weiterhin mindestens einen Anwendungsprozess, der Daten in die Tabellen einfügt, liest oder aktualisiert.

Sie können die Lösung als `.txt` oder als PDF-Datei abgeben.

Hinweis: Für den Besuch des IT-Tags müssen Sie sich frühzeitig über die Web-Seite anmelden. Sonst kann kein Namensschild für Sie vorbereitet werden und Sie müssen lange beim Einlass stehen, weil alle Daten handschriftlich erfasst werden. Die Web-Seite des 39. IT-Tags ist:

[<https://www.uni-halle.de/uzi/veranstaltungen/39it/>].

Bei Problemen sprechen Sie bitte Alexander Hinneburg an.

Aufgabe 2 (4 Punkte)

Dieses Logikrätsel ist inspiriert von dem Buch „Dame oder Tiger“ von Raymond Smullyan, der ein bekannter Logiker war. Sie können Auszüge aus dem titelgebenden Abschnitt des Buches auf dieser Webseite finden:

[<https://emath.de/Referate/Smullyan.pdf>]

Ich habe die Geschichte um das Rätsel sowie auch das Rätsel selbst leicht verändert.

Ein König lässt eine junge Dame einsperren, weil sie für mehr Umweltschutz demonstriert hat. Ihr Verlobter möchte sie gern befreien. Der König ist ein Fan von Logik-Rätseln, und stellt den Verlobten vor die Wahl zwischen drei Türen:

- In einem der drei Räume ist seine Herzens-Dame. Wenn er diese Tür öffnet, darf er mit ihr gehen.
- In einem anderen Raum ist ein hungriger Tiger. Wenn er diese Tür öffnet, wird er höchstwahrscheinlich aufgefressen.
- Der dritte Raum ist leer. Er überlebt dann zwar, muss aber ohne seine Liebste gehen.

An den drei Räumen gibt es Schilder mit folgenden Inschriften:

- Raum 1: „Dieser Raum ist leer“.
- Raum 2: „Der Tiger ist in Raum 1“.
- Raum 3: „Die Dame ist in Raum 1“.

Um die Sache etwas komplizierter zu machen, gelten folgende Regeln:

- Wenn die Dame in dem Raum ist, ist die Inschrift des Schildes wahr.
- Die Inschrift des Schildes an dem Raum, wo der Tiger ist, ist falsch.
- Beim leeren Raum kann die Inschrift des Schildes wahr oder falsch sein.

Wir codieren die Möglichkeiten als Zeilen einer Tabelle:

MOEGLICH		
R1	R2	R3
D	T	
D		T
T	D	
	D	T
T		D
	T	D

Aus den Inschriften und den Zusatzbedingungen ergeben sich folgende logischen Formeln (jeweils zwei Wenn-Dann-Bedingungen pro Schild):

- $R1 = 'D' \rightarrow R1 = ' '$
- $R1 = 'T' \rightarrow \neg R1 = ' '$
- $R2 = 'D' \rightarrow R1 = 'T'$
- $R2 = 'T' \rightarrow \neg R1 = 'T'$
- $R3 = 'D' \rightarrow R1 = 'D'$
- $R3 = 'T' \rightarrow \neg R1 = 'D'$

In SQL gibt es leider kein „ \rightarrow “ (wenn-dann), aber $A \rightarrow B$ ist bekanntlich äquivalent zu $\neg A \vee B$, und kann damit in SQL mit **NOT** und **OR** ausgedrückt werden. Wenn ein Tiger im jeweiligen Raum ist, ist die Inschrift des Schildes falsch und wird deswegen mit \neg (**NOT** in SQL) negiert.

Ihre Aufgabe ist nun, die obigen sechs logischen Formeln in eine **WHERE**-Bedingung in SQL zu überführen (alle sechs Bedingungen müssen gelten, also mit **AND** verknüpft werden). Sie dürfen, wenn Sie möchten, **NOT** entfernen, indem Sie den jeweiligen Vergleich invertieren. Ansonsten bleiben Sie bitte möglichst nahe an den gegebenen Bedingungen (in der Geschichte wäre es ja fatal, wenn irgendein Fehler geschehen würde).

Leider gibt es die Tabelle **MOEGLICH** nicht in unserer Adminer-Installation, aber man kann auf folgende Weise eine lokale Tabelle nur für die eine Anfrage definieren:

```
WITH MOEGLICH(R1, R2, R3) AS
  (VALUES
    ('D', 'T', ' '),
    ('D', ' ', 'T'),
    ('T', 'D', ' '),
    (' ', 'D', 'T'),
    ('T', ' ', 'D'),
    (' ', 'T', 'D'))
SELECT *
FROM MOEGLICH
```

Sie können den SQL-Code herunterladen unter

[https://users.informatik.uni-halle.de/~brass/db24/homework/h5_dt.sql]

Fügen Sie die **WHERE**-Bedingung an und geben Sie die komplette ausführbare SQL-Anfrage ab. Natürlich sollen Sie wie immer Ihre Anfrage ausprobieren. Loggen Sie sich über das Adminer-Webinterface bei der PostgreSQL-Datenbank für diese Übungen ein:

[https://dbs.informatik.uni-halle.de/edb?pgsql=db&username=student_gast&db=postgres&ns=komponist_public]

Da Sie keine Tabelle der Datenbank benötigen, ist das Schema egal.

Aufgabe 3 (4 Punkte)

Diese Aufgabe bezieht sich auf die schon bekannte Datenbank mit Informationen über Musik-CDs (klassische Musik). Sie finden Sie im Adminer, Schema „komponist_public“. Die Webadresse ist:

```
https://dbs.informatik.uni-halle.de/edb?pgsql=db&username=student_gast&
db=postgres&ns=komponist_public
```

Sie benötigen die Tabellen

- `KOMPONIST(KNR, NAME, VORNAME°, GEBOREN°, GESTORBEN°)`
- `STUECK(SNR, KNR°→KOMPONIST, TITEL, TONART°, OPUS°)`

Sie wollen wissen, welche Stücke in der Tonart „F-dur“ von Johann Sebastian Bach und Georg Friedrich Händel in der Datenbank sind. Geben Sie Name und Vorname des Komponisten sowie den Titel des Stückes aus. Nennen Sie die Spalten der Ausgabe bitte „Nachname“, „Vorname“ und „Titel“ (auch in genau dieser Groß-/Kleinschreibung). Tipp: Sie brauchen über beiden Tabellen jeweils eine Tupelvariable. Das erwartete Ergebnis ist:

Nachname	Vorname	Titel
Händel	Georg Friedrich	Wassermusik, Suite in F-dur
Händel	Georg Friedrich	Concerto grosso op.3 Nr.4
Händel	Georg Friedrich	Concerto grosso op.6 Nr.2
Händel	Georg Friedrich	Concerto grosso op.6 Nr.9
Bach	Johann Sebastian	Brandenburgisches Konzert Nr.1
Bach	Johann Sebastian	Brandenburgisches Konzert Nr.2

Hinweise:

- Sie dürfen nur die Informationen im Aufgabentext verwenden (und ein wenig Allgemeinbildung). Sie dürfen z.B. nicht die Komponistennummern der beiden Komponisten erst in der Beispieldatenbank nachschlagen, und dann in die Anfrage einsetzen. Ihre Anfrage muss auch mit anderen Testzuständen funktionieren, nicht nur mit dem gegebenen Beispiel-Zustand.
- Bitte verwenden Sie möglichst nur die in der Vorlesung bereits vorgestellten Konstrukte von SQL.
- Anfragen mit Syntaxfehlern werden mit 0 Punkten bewertet.
- Wenn Sie Ihre Anfrage mit einem Semikolon abschließen und danach noch Kommentare schreiben, führt das zumindest im Adminer zu einer Fehlermeldung (nicht in der Kommandoschnittstelle `psql`). Für den Adminer beginnt nach dem Semikolon eine neue Anfrage. Offenbar kann er mit der Antwort von PostgreSQL auf leere Anfragen (die nur aus einem Kommentar bestehen) nicht umgehen. SQL-Anfragen müssen nicht mit einem Semikolon abgeschlossen werden. Sie können so auch nach der eigentlichen Anfrage noch Kommentare schreiben.