

Einführung in Datenbanken

— Übungsblatt 6 (Logik) —

Ihre Lösungen laden Sie bitte in die Übungsplattform in StudIP hoch ([StudIP-Eintrag der Vorlesung], Reiter „LTI-Tool“, dann auf „Anwendung starten“).

Einsendeschluss ist Montag, der 27.11.2023, 18⁰⁰.

Hausaufgaben müssen einzeln bearbeitet werden. „Zu ähnliche Lösungen“ führen automatisch zu 0 Punkten für alle Beteiligten. Das gilt auch dann, wenn Sie nicht direkt abgeschrieben haben, sondern nur zufällig die gleiche Quelle benutzt haben (ohne diese zu nennen).

Wir haben aber vereinbart, dass bei korrekter Nennung der Quelle (als SQL-Kommentar im vorgeschriebenen Format `--QUELLE: ...`) selbst identische Abgaben akzeptiert werden (ohne Punktabzug).

Vergessen Sie auch nicht, anzugeben, ob Sie die Aufgabe gerne vorrechnen möchten (`--Vorrechnen:5`) oder eher nicht. Bei der Angabe `--Vorrechnen:1` werden Sie nicht drangenommen, mit `--Vorrechnen:0` erklären Sie, dass Sie nicht zur Übung kommen werden. Sie müssen bei mindestens 20% der Aufgaben einen Wert ≥ 2 angeben.

Aufgabe 1 (4 Punkte)

Dieses Logikrätsel stammt aus 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 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: „Raum 3 ist leer“.
- Raum 2: „In Raum 1 ist ein Tiger“.
- Raum 3: „Dieser Raum ist leer“.

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.
- Die Inschrift des Schildes am leeren Raum kann wahr oder falsch sein.

Wir codieren die Möglichkeiten als Zeilen einer Tabelle:

MOEGLICH		
Raum1	Raum2	Raum3
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):

- $\text{Raum1} = 'D' \rightarrow \text{Raum3} = ' '$
- $\text{Raum1} = 'T' \rightarrow \neg \text{Raum3} = ' '$
- $\text{Raum2} = 'D' \rightarrow \text{Raum1} = 'T'$
- $\text{Raum2} = 'T' \rightarrow \neg \text{Raum1} = 'T'$
- $\text{Raum3} = 'D' \rightarrow \text{Raum3} = ' '$
- $\text{Raum3} = 'T' \rightarrow \neg \text{Raum3} = ' '$

In SQL gibt es leider kein „ $\rightarrow$ “ (wenn-dann), aber im Skript steht, wie man es durch NOT und OR ersetzen kann (vielleicht wissen Sie das auch schon aus anderen Vorlesungen, diese Äquivalenz wird öfters gebraucht). In den obigen Formeln steht $\neg$ für die Negation NOT (wenn ein Tiger im Raum ist, ist die Inschrift des Schildes falsch).

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(Raum1, Raum2, Raum3) 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/db23/homework/h6-dt.sql>]

Fügen Sie die **WHERE**-Bedingung an und geben Sie die komplette ausführbare SQL-Anfrage ab.

Aufgabe 2 (2 Punkte)

Die folgenden Aufgaben beziehen sich auf eine Datenbank mit Informationen über US-Präsidenten. Im Adminer ist dies das Schema **president_public**. Die Tabellen haben folgenden Aufbau:

- **state**(state_name, admin_entered, year_entered)
- **president**(pres_name, birth_year, years_serv, death_age, party, state_born→state)
- **pres_hobby**(pres_name→president, hobby)
- **administration**(admin_nr, pres_name→president, year_inaugurated)
- **admin_pr_vp**((admin_nr, pres_name)→administration, vice_pres_name)
- **pres_marriage**(pres_name→president, spouse_name, pr_age, sp_age, nr_children, mar_year)
- **election**(election_year, candidate, votes, winner_loser_indic)

Sie finden diese Datenbank auch im Adminer:

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

Das Passwort steht Sie in StudIP-Eintrag der Vorlesung, Reiter „Informationen“.

Im folgenden sollen Sie jeweils zwei Anfragen vergleichen, u.a. hinsichtlich Äquivalenz. Beachten Sie, dass Sie alle möglichen Datenbank-Zustände in Betracht ziehen müssen, nicht nur den aktuellen Zustand. Es ist daher nur begrenzt hilfreich, die Anfragen auszuprobieren. Sie können voraussetzen, dass Schlüssel, Fremdschlüssel und die NOT NULL-Bedingungen aus dem Schema erfüllt sind (und auch die Datentypen so sind, wie im Adminer angegeben).

Anfrage 1:

```
SELECT pres_name
FROM    president
WHERE   (state_born = 'Arizona' OR state_born = 'Nevada')
AND     party = 'Democratic'
```

Anfrage 2:

```
SELECT pres_name
FROM    president
WHERE   party = 'Democratic' AND state_born = 'Arizona'
OR      party = 'Democratic' AND state_born = 'Nevada'
```

Welche der folgenden Aussagen ist korrekt? Wenn mehrere korrekt sein sollten, wählen Sie die erste korrekte Aussage.

- A. Die beiden Anfragen liefern immer die gleiche Antwort (äquivalent)
- B. Die beiden Anfragen liefern bis auf Duplikate die gleiche Antwort
- C. Das Ergebnis von Anfrage 1 ist immer leer (inkonsistent)
- D. Das Ergebnis von Anfrage 2 ist immer leer (inkonsistent)
- E. Anfrage 1 liefert immer eine Obermenge ($\supseteq$) von Anfrage 2
- F. Anfrage 1 liefert immer eine Teilmenge ($\subseteq$) von Anfrage 2
- G. Keine der Aussagen trifft zu

Bei „B.“ ist gemeint, dass die Tupelmengen (nach Duplikat-Eliminierung) identisch sind. Auch „E.“ und „F.“ beziehen sich auf die Tupel-Mengen nach Eliminierung eventueller Duplikate (d.h. gemeint ist die übliche Mengen-Inklusion und nicht die Multimengen-Inklusion — blenden Sie ggf. Duplikate in Ihren Überlegungen aus).

Geben Sie den Buchstaben der ersten korrekten Aussage ab sowie eine kurze Begründung (maximal 3 Zeilen, die entscheidenden Stichworte reichen). Bitte laden Sie eine `.txt`-Datei mit Ihrer Antwort hoch (kein PDF, kein Word).

Aufgabe 3 (2 Punkte)

Wie bei Aufgabe 2 vergleichen Sie bitte die folgenden beiden Anfragen.

Anfrage 1:

```
SELECT pres_name
FROM   president
WHERE  NOT (years_serv > 1 OR party <> 'Democratic')
```

Anfrage 2:

```
SELECT pres_name
FROM   president
WHERE  NOT years_serv > 1 AND party = 'Democratic'
```

Aufgabe 4 (2 Punkte)

Wie bei Aufgabe 2 vergleichen Sie bitte die folgenden beiden Anfragen.

Anfrage 1:

```
SELECT pres_name
FROM   president
WHERE  state_born = 'Arizona'
```

Anfrage 2:

```
SELECT p.pres_name
FROM   president p, pres_marriage m
WHERE  p.pres_name = m.pres_name
AND    'Arizona' = p.state_born
```

Hinweis: Der Name der Tupelvariable wird nicht Bestandteil des Namens der Ergebnisspalte (d.h. durch die Verwendung einer Tupelvariable in einem **SELECT**-Ausdruck wird die Äquivalenz nicht gestört).

Aufgabe 5 (3 Punkte)

Welche Präsidenten haben als Hobby „Running“ oder „Walking“? Geben Sie den Namen des Präsidenten, das Hobby sowie das Geburts- und Todesjahr des Präsidenten aus. Das Todesjahr müssen Sie approximieren als Summe von Geburtsjahr und dem erreichten Alter (**death_age**). Die Spalte soll in der Ausgabe „Todesjahr (ca.)“ heissen. Offensichtlich muss Ihre Anfrage die Tabellen **pres_hobby** und **president** miteinander verknüpfen.

Das Ergebnis sollte so aussehen:

pres_name	hobby	birth_year	Todesjahr (ca.)
Adams J Q	Walking	1767	1847
Lincoln A	Walking	1809	1865
McKinley W	Walking	1843	1901
Wilson W	Walking	1856	1923
Truman H S	Walking	1884	1972
Clinton W J	Running	1946	NULL