

# Pflichtmodul Informationssysteme (SS 2020)

Prof. Dr. Jens Teubner

Leitung der Übungen: Thomas Lindemann, Christoph Stahl

## Übungsblatt Nr. 9

Ausgabe: 17.06.2020

Abgabe: 24.06.2020 – 16:00 Uhr

### Aufgabe 1 (Natürlichsprachliche Anfragen → SQL-Anfragen)

1. Gebt für jede der folgenden Anfragen eine entsprechende SQL-Anfrage an, die *nicht* auf Unteranfragen zurückgreift. Entscheidet zudem, ob die Anfrage Monoton ist.
  - (a) Wie lauten die Namen, Geburtsjahre und Geburtsstaaten aller Präsidenten?
  - (b) Welche Präsidenten waren nicht verheiratet? Es sollen nur die Namen dieser Präsidenten ausgegeben werden.
  - (c) Zu welchen Parteien gehören die Präsidenten, die kinderlose Ehen hatten? Es soll jeweils der Name des Präsidenten und seine Partei ausgegeben werden.
2. Gebt zur Präsidentendatenbank SQL-Anfragen an, die die folgenden Fragen mit Hilfe von Aggregatfunktionen beantworten. Entscheidet zudem, ob die Anfrage monoton ist.
  - (a) Welcher Präsident hat mit wie vielen Vizepräsidenten zusammengearbeitet? Es soll jeweils der Name des Präsidenten und die Anzahl der Vizepräsidenten ausgegeben werden.
  - (b) Wie viele Kinder stammen aus Ehen von Präsidenten, die vor 1900 geschlossen wurden?  
Es soll jeweils der Name des Präsidenten und die Gesamtzahl seiner Kinder ausgegeben werden. Dabei sollen nur Präsidenten berücksichtigt werden, die insgesamt mindestens fünf Kinder hatten, und die Ausgabe soll absteigend nach der Gesamtzahl der Kinder sortiert werden.  
*Hinweis: Beachtet dabei, dass ein Präsident Kinder aus mehreren Ehen haben kann!*
  - (c) Bestimmt für jeden Bundesstaat, in dem mindestens ein Präsidenten geboren wurde, sowohl das minimale, das maximale als auch das durchschnittliche Todesalter aller dort geborenen Präsidenten.
  - (d) Bestimmt für jeden Bundesstaat, wie viele Präsidenten in diesem Bundesstaat geboren wurden. Dabei sollen in der Ausgabe auch die Bundesstaaten berücksichtigt werden, aus denen bislang noch kein Präsident stammt.  
Die Ausgabe zu dieser Anfrage soll absteigend nach der Anzahl der Präsidenten sortiert werden. Wenn es innerhalb dieser Ordnung Bundesstaaten gibt, aus denen gleich viele Präsidenten stammen, sollen diese Bundesstaaten in aufsteigender lexikographischer Reihenfolge sortiert ausgegeben werden.

3. Gebt zur Präsidentendatenbank SQL-Anfragen an, die die folgenden Fragen mit durch EXISTS oder IN eingebundene Unteranfragen beantworten.
  - (a) Über welche Präsidenten sind *keine* Hobbys bekannt? Es sollen für jeden dieser Präsidenten sein Name und seine Partei ausgegeben werden.
  - (b) Welche Präsidenten wurden nie zum Präsidenten gewählt?
  - (c) Wann kam zum ersten Mal ein Präsident an die Macht, der in im Bundesstaat *New York* geboren wurde? Es soll der Name dieses Präsidenten und das Jahr seiner Amtseinführung ausgegeben werden.

## Aufgabe 2 (Transformation von SQL-Anfragen)

Betrachtet die folgende SQL-Anfrage an die Präsidenten-Datenbank:

```
SELECT P.PRES_NAME
FROM PRESIDENT P
WHERE NOT EXISTS (
    SELECT M.PRES_NAME
    FROM PRES_MARRIAGE M
    WHERE P.PRES_NAME = M.PRES_NAME AND
    M.NR_CHILDREN > 0
);
```

1. Beschreibt umgangssprachlich, welche Ausgabe diese SQL-Anfrage erzeugt.  
Ist diese Anfrage korreliert?
2. Formuliert diese Anfrage semantisch äquivalent mit Hilfe einer per NOT IN eingebundenen Unteranfrage.  
Ist eure Anfrage korreliert?
3. Betrachtet nun die folgende Anfrage:

```
SELECT DISTINCT P.PRES_NAME
FROM PRESIDENT P, PRES_MARRIAGE M
WHERE P.PRES_NAME = M.PRES_NAME AND
    M.NR_CHILDREN = 0;
```

Ist diese Anfrage semantisch äquivalent zu der ursprünglichen Anfrage? Begründet eure Antwort!

## Aufgabe 3 (CRUD)

Unter *CRUD* versteht man die Operationen *Create*, *Read*, *Update* und *Delete*, die benutzt werden um Daten in einer Datenbank abzurufen und zu verändern. Welche SQL Operationen entsprechen den einzelnen CRUD Operationen?

**Aufgabe 4 (Operationen auf Datenbanken)**

Gebt SQL Statements an, mit denen die folgenden Datenbankzustände in einander überführt werden können. Views werden als graue Tabellen dargestellt. Nutzt möglichst kurze und möglichst wenige SQL Statements.

1.

Von:

S		
A	B	C
1	x	a
1	x	c

Nach:

S			R	
A	B	C	A	C
1	x	a	1	a
1	x	c	1	c
2	y	d	2	d

2.

Von:

S		
A	B	C
1	x	a
1	x	c

Nach:

S			R	
A	B	C	A	C
1	x	a	1	a
1	x	c	1	c
2	y	d		

3.

Von:

S			R	
A	B	C	A	C
1	x	a	1	a
1	x	c	1	c
2	y	d		

Nach:

S		
A	B	C
2	y	q

4.

Von:

S			R		
A	B	C	A	B	C
1	x	a	1	x	a
2	y	d	1	v	c

Nach:

S			R			T		
A	B	C	A	B	C	A	B	C
1	x	a	1	x	a	1	x	a
2	y	d	1	v	c	2	y	d
						1	x	a
						1	v	c

5. Die folgende Tabelle *Lagerbuch* speichert alle Eingänge (AN) und Ausgänge (AB) von Getränken in einem Lokal. Die Tabelle *Karte* speichert die Verkaufspreise der Getränke. Gebt für beide Schritte jeweils einen Ausdruck an:

**Von:**

Lagerbuch				Karte	
DATUM	WARE	ART	ANZAHL	WARE	PREIS
11.05.2019	Wasser	AN	100	Wasser	1
11.05.2019	Bier	AN	100	Wein	5
11.05.2019	Wein	AN	100	Bier	2.5
11.05.2019	Schorle	AN	100	Schorle	1.5
11.05.2019	Saft	AN	100	Saft	1
12.05.2019	Bier	AB	50		
14.05.2019	Schorle	AB	20		
14.05.2019	Wasser	AB	70		
15.05.2019	Bier	AN	20		
15.05.2019	Wein	AB	10		
15.05.2019	Saft	AB	100		

**Über:**

Lagerbuch				Karte	
DATUM	WARE	ART	ANZAHL	WARE	PREIS
11.05.2019	Wasser	AN	100	Wasser	1
11.05.2019	Bier	AN	100	Wein	5
11.05.2019	Wein	AN	100	Bier	2.5
11.05.2019	Schorle	AN	100	Schorle	1.5
11.05.2019	Saft	AN	100	Saft	1
12.05.2019	Bier	AB	50		
14.05.2019	Schorle	AB	20		
14.05.2019	Wasser	AB	70		
15.05.2019	Bier	AN	20		
15.05.2019	Wein	AB	10		
15.05.2019	Saft	AB	100		

  

Umsatz			
WARE	VERKAUFT	PREIS	UMSATZ
Bier	50	2.5	125.0
Saft	100	1	100.0
Wasser	70	1	70.0
Wein	10	5	50.0
Schorle	20	1.5	30.0

**Nach:**

Lagerbuch				Karte	
DATUM	WARE	ART	ANZAHL	WARE	PREIS
11.05.2019	Wasser	AN	100	Wasser	1
11.05.2019	Bier	AN	100	Wein	5
11.05.2019	Wein	AN	100	Bier	2.5
11.05.2019	Schorle	AN	100	Schorle	1.5
11.05.2019	Saft	AN	100	Saft	1
12.05.2019	Bier	AB	20		
14.05.2019	Schorle	AB	20		
14.05.2019	Wasser	AB	70		
14.05.2019	Bier	AB	30		
15.05.2019	Bier	AN	20		
15.05.2019	Wein	AB	10		
15.05.2019	Saft	AB	100		
16.05.2019	Wein	AB	20		

  

Umsatz			
WARE	VERKAUFT	PREIS	UMSATZ
Wein	30	5	150.0
Bier	50	2.5	125.0
Saft	100	1	100.0
Wasser	70	1	70.0
Schorle	20	1.5	30.0

Auszug aus dem Schema der Präsidenten-Datenbank:

SQL> DESCRIBE PRESIDENT

Name	Null?	Type
PRES_NAME	NOT NULL	VARCHAR2(17)
BIRTH_YEAR	NOT NULL	NUMBER(4)
YEARS_SERV	NOT NULL	NUMBER(2)
DEATH_AGE		NUMBER(3)
PARTY	NOT NULL	VARCHAR2(12)
STATE_BORN	NOT NULL	VARCHAR2(17)

SQL> DESCRIBE ADMIN\_PR\_VP

Name	Null?	Type
ADMIN_NR	NOT NULL	NUMBER(3)
PRES_NAME	NOT NULL	VARCHAR2(17)
VICE_PRES_NAME	NOT NULL	VARCHAR2(17)

SQL> DESCRIBE ADMINISTRATION

Name	Null?	Type
ADMIN_NR	NOT NULL	NUMBER(3)
PRES_NAME	NOT NULL	VARCHAR2(17)
YEAR_INAUGURATED	NOT NULL	NUMBER(4)

SQL> DESCRIBE PRES\_MARRIAGE

Name	Null?	Type
PRES_NAME	NOT NULL	VARCHAR2(17)
SPOUSE_NAME	NOT NULL	VARCHAR2(17)
PR_AGE	NOT NULL	NUMBER(3)
SP_AGE	NOT NULL	NUMBER(3)
NR_CHILDREN	NOT NULL	NUMBER(2)
MAR_YEAR	NOT NULL	NUMBER(4)

SQL> DESCRIBE PRES\_HOBBY

Name	Null?	Type
PRES_NAME	NOT NULL	VARCHAR2(17)
HOBBY	NOT NULL	VARCHAR2(18)

```
SQL> DESCRIBE ELECTION
```

Name	Null?	Type
ELECTION_YEAR	NOT NULL	NUMBER(4)
CANDIDATE	NOT NULL	VARCHAR2(17)
VOTES		NUMBER(3)
WINNER_LOSER_INDIC	NOT NULL	CHAR(1)

```
SQL> DESCRIBE STATE
```

Name	Null?	Type
STATE_NAME	NOT NULL	VARCHAR2(17)
ADMIN_ENTERED		NUMBER(3)
YEAR_ENTERED	NOT NULL	NUMBER(4)