

Einführung in die Informatik: Systeme und Anwendungen
SS 2014

Übungsblatt 10: Lösungsvorschlag

Aufgabe 10-1 Semaphore

In einer Fabrik werden Handys mit Akku hergestellt. Dazu gibt es zwei Erzeugerprozesse E_1 und E_2 und einen Verbraucherprozess V . E_1 stellt die Akkus her und legt sie in eine Zwischenablage. E_2 stellt die Handys her und legt sie in dieselbe Zwischenablage. Die Zwischenablage hat Platz für je einen Akku und ein Handy. Ist die Zwischenablage mit je einem Akku und einem Handy gefüllt, so kann V die beiden Teile aus der Zwischenablage nehmen und zusammenbauen.

Die Prozesse sind durch folgende unvollständige Prozessbeschreibungen gegeben:

```
/* dient dem wechselseitigen Ausschluss der Zugriffe auf die Zwischenablage */  
var mutex: binarysemaphore;  
init(mutex, 1);
```

```
/* zeigt an, ob der Platz für Akku bzw. Handy in der Zwischenablage frei ist */  
var Akku_genommen, Handy_genommen: semaphore;  
init(Akku_genommen, 1); init(Handy_genommen, 1);
```

```
/* zeigt an, ob Akku bzw. Handy in der Zwischenablage abgelegt wurde */  
var Akku_abgelegt, Handy_abgelegt: semaphore;  
init(Akku_abgelegt, 0); init(Handy_abgelegt, 0);
```

Prozess E_1

```
REPEAT {  
(1) stelle Akku her;  
(2) wait(Akku_genommen);  
(3) wait(mutex);  
(4) lege Akku in die Zwischenablage;  
(5) signal(mutex);  
(6) signal( Akku_abgelegt );  
}
```

Prozess E_2

```
REPEAT {  
(1) stelle Handy her;  
(2) wait(Handy_genommen);  
(3) wait(mutex);  
(4) lege Handy in die Zwischenablage;  
(5) signal(mutex);  
(6) signal( Handy_abgelegt );  
}
```

Prozess V

```
REPEAT {  
(1) wait(Akku_abgelegt);  
(2) wait(Handy_abgelegt);  
(3) wait(mutex);  
(4) nimm Akku und Handy aus der Zwischenablage;  
(5) signal(mutex);  
(6) signal( Akku_genommen );  
(7) signal( Handy_genommen );  
(8) baue Akku und Handy zusammen;  
}
```

- (a) Vervollständigen Sie die signal-Operationen der Prozesse E_1 , E_2 und V , so dass die drei Prozesse wie beschrieben parallel ablaufen können, ohne dass Deadlocks auftreten.
- (b) Ist die Reihenfolge der wait-Operationen (1) und (2) von Prozess V für den korrekten Ablauf von Belang? Begründen Sie Ihre Antwort kurz.

Lösungsvorschlag:

Nein, weil diese beiden wait-Operationen auch in der umgekehrten Reihenfolge sicherstellen, dass je ein Akku und Handy in der Zwischenablage liegen.

- (c) Ist die Reihenfolge der wait-Operationen (2) und (3) von Prozess V für den korrekten Ablauf von Belang? Begründen Sie Ihre Antwort kurz.

Lösungsvorschlag:

Ja, weil es zu einem Deadlock kommen kann, wenn V bereits *mutex* belegt hat und auf *Handy_abgelegt* wartet, während E_2 auf *mutex* wartet und das Handy nicht ablegen kann.

Aufgabe 10-2 *Anfragen*

Gegeben sei das folgende relationale Datenbank-Schema für belegte Lehrveranstaltungen im Sommersemester 2011 (die Primärschlüssel-Attribute sind jeweils unterstrichen):

Lehrveranstaltung (LNr, Titel, FNr)

Student (MatrNr, Name, Vorname, FNr, Fachsemester)

Studienfach (FNr, Bezeichnung, Regelstudienzeit)

Belegt (MatrNr, LNr)

Formulieren Sie folgende Anfragen in Relationaler Algebra:

- (a) Geben Sie Namen, Vornamen und Bezeichnung der Hauptfächer aller Studenten aus, die sich im 1. Fachsemester befinden.

$$\pi_{Name, Vorname, Bezeichnung}((\sigma_{Fachsemester=1}(Student) \bowtie Studienfach))$$

- (b) Geben Sie die Namen aller Studenten zusammen mit den Titeln ihrer belegten Vorlesungen aus.

$$\pi_{Name, Titel}(Student \bowtie Belegt \bowtie Lehrveranstaltung)$$

- (c) Bestimmen Sie die Namen und Vornamen aller Studenten, die nur Lehrveranstaltungen aus ihrem Hauptfach belegen.

$$\pi_{Name, Vorname}(\sigma_{Student.FNr=Lehrveranstaltung.FNr}(Student \bowtie Belegt \bowtie Lehrveranstaltung)) - \pi_{Name, Vorname}(\sigma_{Student.FNr <> Lehrveranstaltung.FNr}(Student \bowtie Belegt \bowtie Lehrveranstaltung))$$

Formulieren Sie folgende Anfragen in SQL:

- (e) Bestimmen Sie die durchschnittliche Fachsemesterzahl der Studenten, die die Vorlesung "Systeme und Anwendungen" belegen. Geben Sie zusätzlich den Titel der Vorlesung aus.

```
select Titel, avg(Fachsemester) from Lehrveranstaltung natural join Belegt
natural join Student where Titel = 'Systeme und Anwendungen' group by Titel;
```

- (f) Geben Sie Nummer und Titel aller Lehrveranstaltungen aus, welche nicht aus dem Studienfach "Informatik" kommen.

```
select LNr, Titel, from Lehrveranstaltung natural join Studienfach where
Bezeichnung not in ('Informatik');
```

```
select LNr, Titel from Lehrveranstaltung where FNr not in (select FNR from
Studienfach where Bezeichnung = 'Informatik');
```

- (g) Bestimmen Sie die Titel aller Lehrveranstaltungen, die von mindestens 100 Studenten belegt werden. Geben Sie Anzahl der Studenten für jede Lehrveranstaltung mit aus.

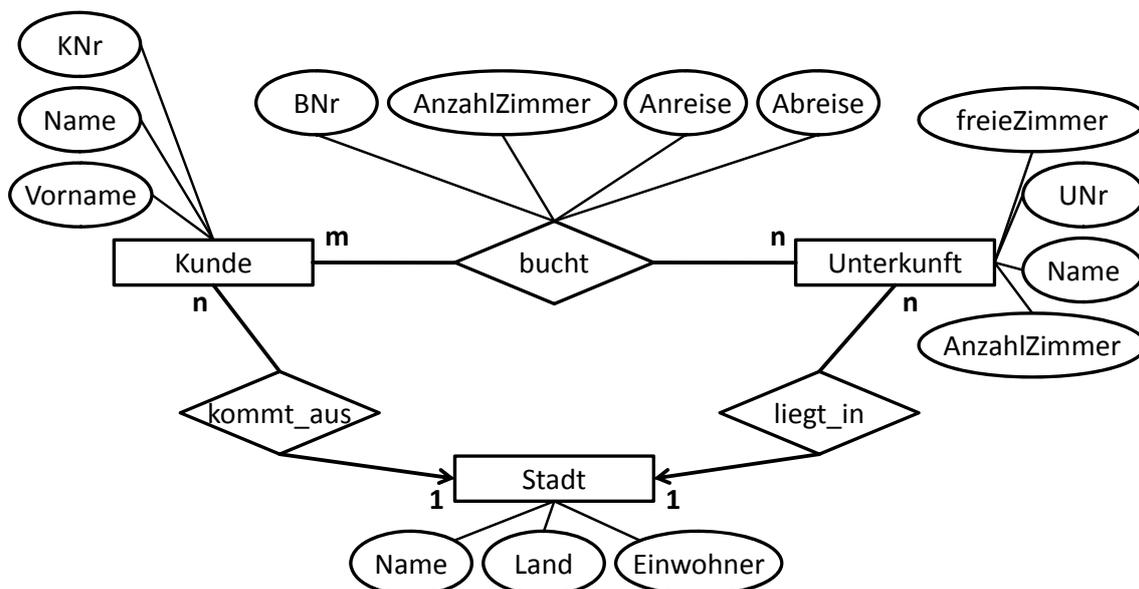
```
select Titel, count(*) from Lehrveranstaltung natural join Belegt group
by Titel having count(*) >= 100;
```

Aufgabe 10-3 E/R-Modellierung

Für das Buchungssystem eines Reisebüros sollen in einer relationalen Datenbank die angebotenen Unterkünfte und die Kunden des Reisebüros sowie Infos über die Städte der Unterkünfte verwaltet werden. Folgende Eigenschaften bilden die Grundlagen des Systems:

- Eine Stadt wird durch ihren Namen und das Land gekennzeichnet, in dem sie liegt. Außerdem hat jede Stadt eine Einwohnerzahl. In einer Stadt können mehrere Unterkünfte liegen.
- Unterkünfte haben eine eindeutige Nummer, einen Namen und eine Anzahl an Zimmern, wobei zusätzlich die Anzahl der freien Zimmer angegeben sein soll. Eine Unterkunft liegt in einer bestimmten Stadt und kann gleichzeitig mehrere Kunden aufnehmen.
- Ein Kunde besitzt eine eindeutige Kundennummer, einen Namen und einen Vornamen und kommt aus einer bestimmten Stadt. Aus einer Stadt können mehrere Kunden kommen.
- Kunden können für einen beliebigen Zeitraum ein oder mehrere Zimmer einer beliebigen Unterkunft buchen. Eine Buchung ist durch eine Buchungsnummer gekennzeichnet. Der gewünschte Zeitraum soll dabei durch die Tage der An- und Abreise eingetragen werden können.

- (a) Entwerfen Sie zu diesem Zweck ein E/R-Modell. Markieren Sie die Funktionalität jeder Beziehung. Schlüsselkandidaten müssen hier nicht markiert werden.



- (b) Setzen Sie das vollständige E/R-Diagramm in ein entsprechendes relationales Datenbankschema um. Identifizieren Sie für jede Relation einen Primärschlüssel und unterstreichen Sie diesen. Achten Sie auf eine geeignete Modellierung der Relationships. Sie müssen hier keine SQL-DDL-Befehle angeben.

Relationenname	Attribute (Schlüsselattribute unterstrichen)
Kunde	<u>KNr</u> , Name, Vorname, StadtName, StadtLand
Unterkunft	<u>UNr</u> , Name, AnzahlZimmer, freieZimmer, StadtName, StadtLand
Stadt	<u>Name</u> , <u>Land</u> , Einwohner
Buchung	<u>BNr</u> , KNr, UNr, AnzahlZimmer, Anreise, Abreise

Aufgabe 10-4 *Normalisierung*

Gegeben sei das Relationenschema $R(A, B, C, D, E, F)$, sowie die Menge der zugehörigen nicht-trivialen funktionalen Abhängigkeiten:

$$\{A, B \rightarrow C \quad D \rightarrow E \quad F \rightarrow A, B, D, E \quad B \rightarrow C, D\}$$

- (a) Begründen Sie, dass F Schlüsselkandidat ist und es keine weiteren Schlüsselkandidaten gibt.

Lösungsvorschlag:

A, B, D, E und F sind direkt von F abhängig.
Wegen $B \rightarrow CD$ bzw. $AB \rightarrow C$ gilt auch: $F \rightarrow C$ (*Transitivität*).

F ist also eindeutig.

Da F nur aus einem Element besteht ist es auch minimal und damit Schlüsselkandidat.

Auf der anderen Seite ist F von keinem anderen Attribut abhängig und somit der einzige Schlüsselkandidat.

- (b) In welcher *höchsten* Normalform befindet sich das Relationenschema R ?
Begründen Sie Ihre Antwort.

Lösungsvorschlag:

R befindet sich in 2. Normalform.
Einzigster Schlüsselkandidat F besteht aus nur einem Attribut.

R erfüllt aber nicht die 3. Normalform.
Für die erste funktionale Abhängigkeit gilt:
 AB enthält keinen Schlüsselkandidaten und C ist nicht prim.

Aufgabe 10-5 *Relationale Algebra*

Gegeben seien die folgenden Relationen R und S :

R

A	B	C	D
1	2	3	1
1	2	3	4
2	1	4	2
2	1	4	3
2	1	4	4
4	3	4	3

S

C	D	E
2	4	3
4	3	2
3	1	4

Geben Sie die Ergebnisrelationen folgender Ausdrücke der relationalen Algebra als Tabellen an.

(a) $\pi_{C,D}(R) \cap \pi_{C,D}(S)$

C	D
3	1
4	3

(b) $R \bowtie S$

A	B	C	D	E
1	2	3	1	4
2	1	4	3	2
4	3	4	3	2

(c) $\pi_{A,B,C}(R) \bowtie S$

A	B	C	D	E
1	2	3	1	4
2	1	4	3	2
4	3	4	3	2

(d) $\pi_{A,B}(R) \bowtie_{B < C} S$

A	B	C	D	E
1	2	4	3	2
1	2	3	1	4
2	1	2	4	3
2	1	4	3	2
2	1	3	1	4
4	3	4	3	2

(e) $R \div \pi_D(\sigma_{D \geq 3}(S))$

A	B	C
2	1	4

Aufgabe 10-6 SQL

Eine Agentur zur Vermittlung von Zeitschriftenabonnements hält ihre Daten über die laufenden Abonnements in einer relationalen Datenbank nach folgendem Schema (Schlüsselattribute sind unterstrichen):

Abonnent (Kundennr, Name, Adresse)

bezieht (Kundennr, Titel)

Zeitschrift (Titel, Preis, Verlag)

Die Relation *bezieht* ordnet den Kundennummern von Abonnenten die abonnierten Zeitschriftentitel zu.

Formulieren Sie die folgenden Anfragen bezüglich dieses Datenbankschemas in SQL.

- (a) Finden Sie für jedes Abonnement den Namen des Abonnenten sowie Titel, Preis und Verlag der Zeitschrift. Das Ergebnis soll in alphabetischer Reihenfolge nach dem Verlag sortiert werden. Bei gleichem Verlag soll absteigend nach dem Preis sortiert werden.

Lösungsvorschlag:

```
SELECT name, titel, preis, verlag
FROM abonnent NATURAL JOIN bezieht NATURAL JOIN zeitschrift
ORDER BY verlag ASC, preis DESC;
```

```
SELECT name, z.titel, preis, verlag
FROM abonnent a, bezieht b, zeitschrift z
WHERE a.kundenr = b.kundenr AND b.titel = z.titel
ORDER BY 4, 3 DESC;
```

- (b) Finden Sie die Titel derjenigen Zeitschriften, die von keinem Abonnenten bezogen werden.

Lösungsvorschlag:

```
SELECT titel
FROM zeitschrift
EXCEPT
SELECT titel
FROM bezieht;
```

```
SELECT titel
FROM zeitschrift z
WHERE NOT EXISTS
  (SELECT *
   FROM bezieht b
   WHERE z.titel = b.titel);
```

- (c) Finden Sie die Titel derjenigen Zeitschriften, die einen überdurchschnittlichen Preis haben.

Lösungsvorschlag:

```
SELECT titel
FROM zeitschrift
WHERE preis > (SELECT avg(preis) FROM zeitschrift);
```

- (d) Finden Sie die Kundennummern derjenigen Abonnenten, die genau zwei Zeitschriften beziehen.

Lösungsvorschlag:

```
SELECT kundennr
FROM bezieht
GROUP BY kundennr
HAVING count(*) = 2;
```

- (e) Finden Sie die Kundennummern der Abonnenten, für die gilt, dass alle Zeitschriften, die sie beziehen, einen Preis von mindestens 5 Euro haben.

Lösungsvorschlag:

```
SELECT kundennr
FROM bezieht NATURAL JOIN zeitschrift
GROUP BY kundennr
HAVING min(preis) >= 5;

SELECT kundennr
FROM bezieht b
WHERE NOT EXISTS
  (SELECT *
   FROM zeitschrift z
   WHERE b.titel = z.titel AND preis < 5);
```