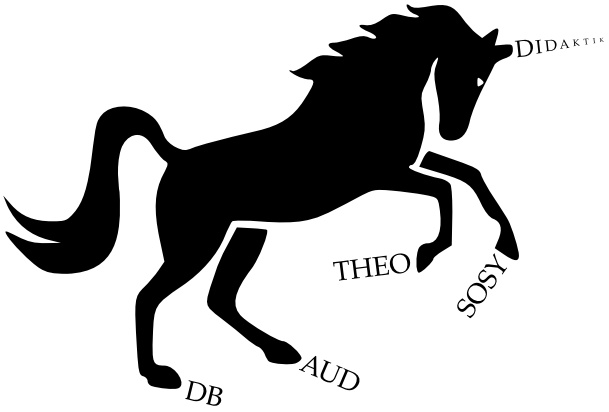
46116
Frühjahr
2014

Softwaretechnologie / Datenbanksysteme (nicht vertieft)

Aufgabenstellungen mit Lösungsvorschlägen



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Aufgabenübersicht

Thema Nr. 2 3

Teilaufgabe Nr. 1 3

Aufgabe 1 [Hanoi] 3

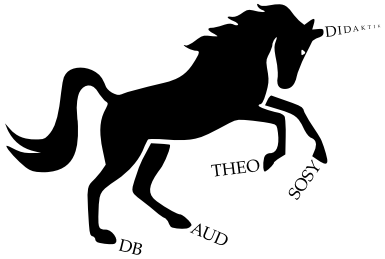
Teilaufgabe Nr. 2 4

Aufgabe 2: Relationale Algebra [Mitfahrgelegenheiten] 4

Aufgabe 2: Relationale Algebra 4

Aufgabe 2: Relationale Algebra 4

Aufgabe 3 [Mitfahrgelegenheiten] 7



Die Bschlangaul-Sammlung
Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

Thema Nr. 2

Teilaufgabe Nr. 1

Aufgabe 1 [Hanoi]

Gegeben sei folgende Methode zur Berechnung der Anzahl der notwendigen Züge beim Spiel *„Die Türme von Hanoi“*:

```java
int hanoi(int nr, char from, char to) {
  char free = (char) ('A' + 'B' + 'C' - from - to);
  if (nr > 0) {
    int moves = 1;
    moves += hanoi(nr - 1, from, free);
    System.out.println("Move piece nr. " + nr + " from " + from + " to " + to);
    moves += hanoi(nr - 1, free, to);
    return moves;
  } else {
    return 0;
  }
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46116/jahr_2014/fruehjahr/Hanoi.java`

(a) Beweisen Sie formal mittels vollständiger Induktion, dass zum Umlegen von k Scheiben (z. B. vom Turm A zum Turm C) insgesamt $2^k - 1$ Schritte notwendig sind, also dass für $k \geq 0$ folgender Zusammenhang gilt:

$$\text{hanoi}(k, \text{'A'}, \text{'C'}) = 2^k - 1$$

Lösungsvorschlag

Zu zeigen:

$$\text{hanoi}(k, \text{'A'}, \text{'C'}) = 2^k - 1$$

Induktionsanfang

— Beweise, dass $A(1)$ eine wahre Aussage ist. ———————————————

$k = 0$

$$\text{hanoi}(0, \text{'A'}, \text{'C'}) = 0$$

$$2^0 - 1 = 1 - 1 = 0$$

Induktionsvoraussetzung

— Die Aussage $A(k)$ ist wahr für ein beliebiges $k \in \mathbb{N}$. ——————————

$$\text{hanoi}(k, \text{'A'}, \text{'C'}) = 2^k - 1$$

Induktionsschritt

— Beweise, dass wenn $A(n = k)$ wahr ist, auch $A(n = k + 1)$ wahr sein muss. ——

$$\text{hanoi}(k, \text{'A'}, \text{'C'}) = 1 + \text{hanoi}(k - 1, \text{'A'}, \text{'B'}) + \text{hanoi}(k - 1, \text{'B'}, \text{'C'})$$

$k \to k + 1$

$$\begin{aligned}
\text{hanoi}(k+1, \text{'A'}, \text{'C'}) &= 1 + \text{hanoi}((k+1) - 1, \text{'A'}, \text{'B'}) + \\
&\quad\ \text{hanoi}((k+1) - 1, \text{'B'}, \text{'C'}) \\
&= 1 + \text{hanoi}(k, \text{'A'}, \text{'B'}) + \\
&\quad\ \text{hanoi}(k, \text{'B'}, \text{'C'}) && k + 1 - 1 = k \\
&= 1 + 2^k - 1 + 2^k - 1 && \text{Formeln eingesetzt} \\
&= 2^k + 2^k - 1 && 1 - 1 - 1 = -1 \\
&= 2 \cdot 2^k - 1 && 2^k + 2^k = 2 \cdot 2^k \\
&= 2^{k+1} - 1 && 2 \cdot 2^k = 2^{k+1}
\end{aligned}$$

(b) Geben Sie eine geeignete Terminierungsfunktion an und begründen Sie kurz Ihre Wahl!

Lösungsvorschlag

Betrachte die Argumentenfolge $k, k-1, k-2, \ldots, 0$. Die Terminierungsfunktion ist offenbar $T(k) = k$. $T(k)$ ist bei jedem Rekursionsschritt auf der Folge der Argumente streng monoton fallend. Bei der impliziten Annahme k ist ganzzahlig und $k \geq 0$ ist $T(k)$ nach unten durch 0 beschränkt.

Teilaufgabe Nr. 2

Aufgabe 2: Relationale Algebra [Mitfahrgelegenheiten]

Aufgabe 2: Relationale Algebra

Gegeben sei das folgende relationale Schema mitsamt Beispieldaten für eine Datenbank von Mitfahrgelegenheiten. Die Primärschlüssel-Attribute sind jeweils unterstrichen, Fremdschlüs-

sel sind überstrichen.

„Kunde":

KID	Name	Vorname	Stadt
K1	Meier	Stefan	S3
K2	Müller	Peta	S3
K3	Schmidt	Christine	S2
K4	Schulz	Michael	S4

„Stadt"

SID	SName	Bundesland
S1	Berlin	Berlin
S2	Nürn	Bayern
S3	Köln	Nordrhein-Wesffalen
S4	Stuttgart	Baden-Württemberg
S5	München	Bayer

„Angebot":

KID	Start	Ziel	Datum	Plätze
K4	S4	S5	08.07.2011	3
K4	S5	S4	10.07.2011	3
K1	S1	S5	08.07.2011	3
K3	S2	S3	15.07.2011	1
K4	S4	S1	15.07.2011	3
K1	S5	S5	09.07.2011	2

„Anfrage":

KID	Start	Ziel	Datum
K2	S4	S5	08.07.2011
K2	S5	S4	10.07.2011
K3	S2	S3	08.07.2011
K3	S3	S2	10.07.2011
K2	S4	S5	05.07.2011
K2	S5	S4	17.07.2011

(a) Formulieren Sie die folgenden Anfragen auf das gegebene Schema in relationaler Algebra:

- Finden Sie die Namen aller Städte in Bayern!

Lösungsvorschlag

$$\pi_{\text{SName}}(\sigma_{\text{Bundesland}=\text{Bayern}}(\text{Stadt}))$$

- Finden Sie die SIDs aller Städte, für die weder als Start noch als Ziel eine Anfrage vorliegt!

Lösungsvorschlag

$$\pi_{\text{SID}}(\text{Stadt}) - \pi_{\text{Start}}(\text{Anfage}) - \pi_{\text{Ziel}}(\text{Anfrage})$$

- Finden Sie alle IDs von Kunden, welche eine Fahrt in ihrer Heimatstadt starten.

Lösungsvorschlag

$$\pi_{\text{KID}}(\text{Kunde} \bowtie_{\text{Kunde.KID}=\text{Anfrage.KID} \wedge \text{Kunde.Stadt}=\text{Anfrage.Stadt}} \text{Anfrage}) \wedge \pi_{\text{KID}}(\text{Kunde} \bowtie_{\text{Kunde.KID}=\text{Angebot.KID} \wedge \text{Kunde.Stadt}=\text{Angebot.Stadt}} \text{Angebot})$$

- Geben Sie das Datum aller angebotenen Fahrten von München nach Stuttgart aus!

Lösungsvorschlag

$$\pi_{\text{Datum}}(\\ (\text{Angebot} \bowtie_{\text{Start=SID} \wedge \text{SName='München'}} \text{Stadt}) \\ \bowtie_{\text{Ziel=SID} \wedge \text{SName='Stuttgart'}} \\ \text{Stadt} \\ )$$

Variante 2:

Lösungsvorschlag

$$\pi_{\text{Datum}}(\\ \sigma_{\text{Sname='München'} \wedge \text{Zname='Stuttgart'}}(\\ \rho_{\text{Zname} \leftarrow \text{Sname,SID1} \leftarrow \text{SID}}(\text{Stadt}) \\ \bowtie_{\text{Ziel=SID1}} \\ \text{Angebot} \\ \bowtie_{\text{Start=SID}} \\ \text{Stadt}))$$

(b) Geben Sie das Ergebnis (bezüglich der Beispieldaten) der folgenden Ausdrücke der relationalen Algebra als Tabellen an:

- $\pi_{\text{KID}}(\text{Angebot}) \bowtie \text{Kunde}$

Lösungsvorschlag

Zeile mit der Petra Müller fällt weg.

KID	Name	Vorname	Stadt
K1	Meier	Stefan	S3
K3	Schmidt	Christine	S2
K4	Schulz	Michael	S4

- $\pi_{(\text{KID,Stadt})}(\text{Kunde}) \bowtie_{\text{Kunde.Stadt=Angebot.Ziel}} \pi_{\text{Plaetze}}(\text{Angebot})$

Lösungsvorschlag

KID	Stadt	Plätze
K1	S3	1
K2	S3	1
K4	S4	1
K4	S4	2

Aufgabe 3 [Mitfahrgelegenheiten]

„Kunde":

KID	Name	Vorname	Stadt
K1	Meier	Stefan	S3
K2	Müller	Peta	S3
K3	Schmidt	Christine	S2
K4	Schulz	Michael	S4

„Stadt"

SID	SName	Bundesland
S1	Berlin	Berlin
S2	Nürnberg	Bayern
S3	Köln	Nordrhein-Wesffalen
S4	Stuttgart	Baden-Württemberg
S5	München	Bayern

„Angebot":

KID	Start	Ziel	Datum	Plätze
K4	S4	S5	08.07.2011	3
K4	S5	S4	10.07.2011	3
K1	S1	S5	08.07.2011	3
K3	S2	S3	15.07.2011	1
K4	S4	S1	15.07.2011	3
K1	S5	S5	09.07.2011	2

„Anfrage":

KID	Start	Ziel	Datum
K2	S4	S5	08.07.2011
K2	S5	S4	10.07.2011
K3	S2	S3	08.07.2011
K3	S3	S2	10.07.2011
K2	S4	S5	05.07.2011
K2	S5	S4	17.07.2011

```
CREATE TABLE Stadt (
  SID VARCHAR(100) NOT NULL PRIMARY KEY,
  SName VARCHAR(100) NOT NULL,
  Bundesland VARCHAR(100) NOT NULL
);

CREATE TABLE Anfrage (
  KID VARCHAR(100) NOT NULL,
  Start VARCHAR(100) DEFAULT NULL REFERENCES Stadt (SID),
  Ziel VARCHAR(100) DEFAULT NULL REFERENCES Stadt (SID),
  Datum date NOT NULL,
  PRIMARY KEY (KID, Start, Ziel, Datum)
);

CREATE TABLE Angebot (
  KID VARCHAR(100) NOT NULL,
  Start VARCHAR(100) DEFAULT NULL REFERENCES Stadt (SID),
  Ziel VARCHAR(100) DEFAULT NULL REFERENCES Stadt (SID),
  Datum date NOT NULL,
```

```
  Plätze integer DEFAULT NULL,
  PRIMARY KEY (Datum, KID)
);

CREATE TABLE Kunde (
  KID VARCHAR(100) NOT NULL PRIMARY KEY,
  Name VARCHAR(100) DEFAULT NULL,
  Vorname VARCHAR(100) DEFAULT NULL,
  Stadt VARCHAR(100) DEFAULT NULL REFERENCES Stadt (SID)
);

INSERT INTO Stadt (SID, SName, Bundesland) VALUES
  ('S1', 'Berlin', 'Berlin'),
  ('S2', 'Nürnberg', 'Bayern'),
  ('S3', 'Köln', 'NRW'),
  ('S4', 'Stuttgart', 'BW'),
  ('S5', 'München', 'Bayern');

INSERT INTO Anfrage (KID, Start, Ziel, Datum) VALUES
  ('K2', 'S4', 'S5', '2011-07-05'),
  ('K2', 'S4', 'S5', '2011-07-08'),
  ('K3', 'S2', 'S3', '2011-07-08'),
  ('K2', 'S5', 'S4', '2011-07-10'),
  ('K3', 'S3', 'S2', '2011-07-10'),
  ('K2', 'S5', 'S4', '2011-07-17');

INSERT INTO Kunde (KID, Name, Vorname, Stadt) VALUES
  ('K1', 'Meier', 'Stefan', 'S3'),
  ('K2', 'Müller', 'Petra', 'S3'),
  ('K3', 'Schmidt', 'Christine', 'S2'),
  ('K4', 'Schulz', 'Michael', 'S4');

INSERT INTO Angebot (KID, Start, Ziel, Datum, Plätze) VALUES
  ('K1', 'S1', 'S5', '2011-07-08', 3),
  ('K4', 'S4', 'S5', '2011-07-08', 3),
  ('K1', 'S5', 'S4', '2011-07-09', 2),
  ('K4', 'S5', 'S4', '2011-07-10', 3),
  ('K3', 'S2', 'S3', '2011-07-15', 1),
  ('K4', 'S4', 'S1', '2011-07-15', 3);
```

(a) Formulieren Sie die folgenden Anfragen in SQL:

(i) Geben Sie alle Attribute aller Anfragen aus, für die passende Angebote existieren! Ein Angebot ist passend zu einer Anfrage, wenn Start, Ziel und Datum identisch sind!

Lösungsvorschlag

```
SELECT Anfrage.KID, Anfrage.Start, Anfrage.Ziel, Anfrage.Datum
FROM Anfrage, Angebot
WHERE
  Anfrage.Start = Angebot.Start AND
  Anfrage.Ziel = Angebot.Ziel AND
  Anfrage.Datum = Angebot.Datum;
```

(ii) Finden Sie Nachnamen und Vornamen aller Kunden, für die kein Angebot existiert!

Lösungsvorschlag

```
SELECT k.Name, k.Vorname
FROM Kunde k
WHERE NOT EXISTS ( SELECT * FROM Angebot a WHERE a.KID = k.KID )
```

oder:

```
SELECT k.Name, k.Vorname
FROM Kunde k
WHERE k.KID NOT IN ( SELECT KID FROM Angebot );
```

(iii) Geben Sie das Datum aller angebotenen Fahrten von München nach Stuttgart aus und sortieren Sie das Ergebnis aufsteigend!

```
SELECT Datum
FROM Angebot, Stadt
WHERE
  (SID = Start OR
  SID = Ziel)
  AND
  (SName = 'München' OR SName = 'Stuttgart')
```

(iv) Geben Sie für jeden Startort einer Anfrage den Namen der Stadt und die Anzahl der Anfragen aus.

```
SELECT SName, COUNT(*)
FROM Anfrage, Stadt
WHERE SID = Start
GROUP BY SID;
```

(b) Wie sieht die Ergebnisrelation zu folgenden Anfragen auf den Beispieldaten aus?

```
SELECT *
FROM
Stadt
WHERE
NOT EXISTS ( SELECT *
FROM Anfrage
WHERE Start = SID OR Ziel = SID ) ;
```

Lösungsvorschlag

S1 Berlin Berlin

```
SELECT KID, SUM (Plätze)
FROM Angebot
WHERE Plätze > 2
GROUP BY KID
HAVING SUM (Plätze) > 4;
```