Einzelprüfung „Softwaretechnologie / Datenbanksysteme (nicht vertieft)“

Einzelprüfungsnummer 46116 / 2014 / Frühjahr

Thema 2 / Teilaufgabe 2 / Aufgabe 3

(Mitfahrgelegenheiten)

Stichwörter: SQL, SQL mit Übungsdatenbank, GROUP BY, HAVING

„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;
```

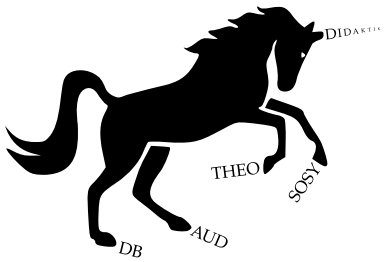
(b) Wie sieht die Ergebnisrelation zu folgenden Anfragen auf den Beispieldaten aus?

```
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;
```



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.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net.Der TEX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: `https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Examen/46116/2014/03/Thema-2/Teilaufgabe-2/Aufgabe-3.tex`