

15.8.00

Datenbank- und Informationssysteme

Lösungsvorschläge zu Übungsblatt 2

Sommersemester 1999

Aufgabe 1

FAHRER (Pers-Nr, Name, Adresse, Geburtsdatum)

LKW (Kennzeichen, Baujahr, Typ)

TOUR (Tour-Nr, Entfernung, mehrtägig, Ruhezeiten, *Kennzeichen*, *Pers-Nr*)

KUNDE (Kunden-Nr, Name, Anschrift)

FRACHTAUFTRAG (Auftrag-Nr, Datum, Gewicht, *Kunden-Nr*, *von*, *nach*)

FRACHTGUT (Bezeichnung, Auftrag-Nr, Gefahrenklasse)

TEILSTRECKE (TS-Nr, Entfernung, Hinweis, *Tour-Nr*, *beginnt*, *endet*)

VERLADEORT (Ort-ID, Name/Firma, Straße, Hausnummer, PLZ, Ort)

ERLEDIGT (*Kennzeichen*, *Auftrag-Nr*, *TS-Nr*)

a) CREATE SCHEMA Speditionsverwaltung;

```
CREATE DOMAIN Pers-NrDomain AS CHAR(14)
CONSTRAINT Pers-NrDomainConstraint
CHECK ( VALUE LIKE '____-____-__');
```

```
CREATE DOMAIN GefahrenklasseDomain AS INTEGER
CONSTRAINT GefahrenklasseDomainConstraint
CHECK ( VALUE >= 1 AND VALUE <= 5);
```

```
CREATE DOMAIN KennzeichenDomain AS VARCHAR(9);
```

```
CREATE DOMAIN Tour-NrDomain AS INTEGER;
```

```
CREATE DOMAIN Kunden-NrDomain AS CHAR(9);
```

```
CREATE DOMAIN Auftrag-NrDomain AS INTEGER;
```

```
CREATE DOMAIN BezeichnungDomain AS VARCHAR(20);
```

```
CREATE DOMAIN TS-NrDomain AS INTEGER;
```

```
CREATE DOMAIN Orts-IDDomain AS INTEGER;
```

```
CREATE TABLE Fahrer
( Pers.-Nr      Pers-NrDomain PRIMARY KEY,
  Name         VARCHAR(30),
  Adresse      VARCHAR(50),
  Geburtsdatum DATE
);
```

```
CREATE TABLE LKW
( Kennzeichen  KennzeichenDomain PRIMARY KEY,
  Baujahr      INT,
  Typ          VARCHAR(30)
);
```

```
CREATE TABLE Tour
( Tour-Nr      Tour-NrDomain PRIMARY KEY,
  Entfernung   INT,
  mehrtägig    BIT,
  Ruhezeiten   INTERVAL HOUR TO MINUTE,
  Kennzeichen  KennzeichenDomain REFERENCES LKW(Kennzeichen),
  Pers-Nr      Pers-NrDomain REFERENCES Fahrer(Pers-Nr)
);
```

```
CREATE TABLE Kunde
( Kunden-Nr    Kunden-NrDomain PRIMARY KEY,
  Name         VARCHAR(30),
  Anschrift    VARCHAR(50)
);
```

```
CREATE TABLE Frachtauftrag
( Auftrag-Nr   Auftrag-NrDomain PRIMARY KEY,
  Datum        TIMESTAMP,
  Gewicht       INTEGER,
  Kunden-Nr    Kunden-NrDomain REFERENCES Kunde(Kunden-Nr),
  von          Orts-IDDomain REFERENCES Verladeort(Orts-ID),
  nach         Orts-IDDomain REFERENCES Verladeort(Orts-ID)
);
```

```

CREATE TABLE Frachtgut
( Bezeichnung    BezeichnungDomain,
  Auftrag-Nr     Auftrag-NrDomain
  REFERENCES Frachtauftrag(Auftrag-Nr),
  PRIMARY KEY (Bezeichnung, Auftrag-Nr),
  Gefahrenklasse GefahrenklasseDomain
);

CREATE TABLE Teilstrecke
( TS-Nr          TS-NrDomain PRIMARY KEY,
  Entfernung     INT,
  Hinweis        VARCHAR(50),
  Tour-Nr        Tour-NrDomain REFERENCES Tour(Tour-Nr),
  beginnt        Orts-IDDomain REFERENCES Verladeort(Orts-ID),
  endet          Orts-IDDomain REFERENCES Verladeort(Orts-ID)
);

CREATE TABLE Verladeort
( Orts-ID        Orts-IDDomain PRIMARY KEY,
  Name/Firma     VARCHAR(30),
  Straße         VARCHAR(30),
  Hausnummer     INT,
  PLZ            DEC(5),
  Ort            VARCHAR(15)
);

CREATE TABLE Erledigt
( Auftrag-Nr     Auftrag-NrDomain REFERENCES Frachtauftrag(Auftrag-Nr),
  TS-Nr          TS-NrDomain REFERENCES Teilstrecke(TS-Nr),
  PRIMARY KEY (Auftrag-Nr,TS-Nr),
  Kennzeichen    KennzeichenDomain REFERENCES LKW(Kennzeichen)
);

```

```

b) CREATE VIEW fahrer36
AS (SELECT Pers-Nr, Name
    FROM Fahrer
    WHERE Geburtsdatum > '12.12.1936');

```

Aufgabe 2

```

a) SELECT *
    FROM LKW
    ORDER BY Baujahr ASC;

b) SELECT Kennzeichen
    FROM LKW
    WHERE Kennzeichen LIKE 'KA%';

```

```

c) SELECT Name, Anschrift
    FROM Kunde K, Frachtauftrag FA
    WHERE K.Kunden-Nr = FA.Kunden-Nr
    AND FA.Datum BETWEEN '15.1.1996 - 0.00' AND '23.3.1996 - 23.59';

d) SELECT K1.Kunden-Nr
    FROM Kunde K1, Kunde K2, Frachtauftrag FA1, Frachtauftrag FA2
    WHERE K1.Kunden-Nr <> K2.Kunden-Nr
    AND FA1.Kunden-Nr = K1.Kunden-Nr
    AND FA2.Kunden-Nr = K2.Kunden-Nr
    AND FA1.Datum = FA2.Datum;

e) SELECT Kunde.*, Fahrer.*      (oder SELECT *)
    FROM Fahrer, Kunde
    WHERE Fahrer.Name = Kunde.Name
    AND Fahrer.Geburtsdatum > '1.1.1924';

f) SELECT MAX(Entfernung)
    FROM Teilstrecke
    GROUP BY Tour-Nr;

g) SELECT DISTINCT Name
    FROM Fahrer, Tour, Teilstrecke
    WHERE Fahrer.Pers-Nr = Tour.Pers-Nr
    AND Tour.Tour-Nr = Teilstrecke.Tour-Nr
    AND Teilstrecke.Entfernung > 250;

h) SELECT AVG(Gewicht)
    FROM Frachtauftrag
    GROUP BY Kunden-Nr;

i) SELECT Name, COUNT(Tour-Nr.), SUM(Entfernung)
    FROM Fahrer, Tour
    WHERE Fahrer.Pers-Nr = Tour.Pers-Nr
    GROUP BY Fahrer.Pers-Nr, Name;

j) SELECT Kunden-Nr, SUM(Gewicht)
    FROM Frachtauftrag
    GROUP BY Kunden-Nr
    HAVING SUM(Gewicht) >= 300;

```

Aufgabe 3

```

a) SELECT Kunden-Nr
    FROM Kunde
    WHERE Name = 'DHS AG'; (Ergebnis : 004582)

    DELETE FROM Frachtauftrag
    WHERE Kunden-Nr = '004582' (Kundennummer der DHS AG);

b) SELECT Pers.-Nr
    FROM Fahrer
    WHERE Name = 'Karl Schmidt'; (Ergebnis : 789126)

```

```
SELECT Pers.-Nr
FROM Fahrer
WHERE Name = 'Peter Bauer'; (Ergebnis : 123456)
```

```
UPDATE tour
SET Pers-Nr= '789126' (Personalnummer von Karl Schmidt)
WHERE Pers-Nr = '123456'; (Personalnummer von Peter Bauer)
```

- c) INSERT INTO LKW VALUES ('M-B-123', 1996, 'Mercedes Benz');
- d) Es muß eine eindeutige Abbildung existieren zwischen der Basistabelle und der VIEW.
Die Eindeutigkeit der Abbildung von der Basistabelle auf die VIEW ist durch die VIEW-Definition gegeben. Wichtig ist aber auch die umgekehrte Richtung.

UPDATE ist nur möglich, wenn folgende Bedingungen erfüllt sind:

- SELECT darf keine GROUP-BY-Klauseln enthalten (HAVING auch nicht!)
- SELECT darf keine Mengenoperatoren enthalten
- SELECT darf kein DISTINCT enthalten
- VIEW darf sich nur über eine Tabelle erstrecken, die FROM Klausel darf sich nur auf eine Tabelle beziehen
- WHERE darf kein CORRELATED SUBQUERY enthalten
- SELECT darf keine Aggregationsfunktion enthalten

Aufgabe 4

a)

```
i) SELECT PNAME
FROM projekt
WHERE P# IN
      ( SELECT P#
        FROM lieferung
        WHERE L# = 'L1');
```

oder

```
SELECT PNAME
FROM projekt p, lieferung l
WHERE p.P# = l.P#
AND l.L# = 'L1';
```

oder

```
SELECT PNAME
FROM projekt NATURAL JOIN lieferung
WHERE lieferung.L# = ,L1';
```

```
ii) SELECT DISTINCT L#
FROM lieferung
WHERE T# IN
      (SELECT T#
       FROM lieferung
       WHERE L# IN
         (SELECT L#
          FROM lieferung
          WHERE T# IN
            (SELECT T#
             FROM teil
             WHERE FARBE = 'ROT'))));
```

oder

```
SELECT DISTINCT L#
FROM ((teil NATURAL JOIN lieferung L1)
      NATURAL JOIN lieferung L2 USING L#)
      NATURAL JOIN lieferung L3 USING T#
WHERE teil.FARBE = 'ROT';
```

```
iii) SELECT L#
FROM lieferung L1
WHERE T# = 'T1'
AND MENGE >
      (SELECT AVG(MENGE)
       FROM lieferung L2
       WHERE T# = 'T1'
       AND L1.P# = L2.P#);
```

```
iv) SELECT lieferant.*, teil.*
FROM lieferant, teil
WHERE lieferant.STADT = teil.STADT
AND lieferant.LNAME NOT LIKE 'J%';
```

```
v) SELECT DISTINCT P#
FROM lieferung Li1
WHERE NOT EXISTS
      (SELECT *
       FROM lieferung Li2
       WHERE Li1.P# = Li2.P#
       AND Li2.L# <> 'L1');
```

```
vi) SELECT LNAME
   FROM lieferant
   WHERE STADT = „LONDON“
   UNION
   SELECT LNAME
   FROM lieferant
   WHERE L# IN
     (SELECT L#
      FROM lieferung
      WHERE P# IN
        (SELECT P#
         FROM projekt
         WHERE STADT = „LONDON“));
```

oder

```
SELECT LNAME
FROM lieferant
WHERE STADT = „LONDON“
UNION
SELECT LNAME
FROM (projekt NATURAL JOIN lieferung)
     NATURAL JOIN lieferant USING L#
WHERE projekt.STADT = 'LONDON';
```

```
b) SELECT *
   FROM projekt NATURAL LEFT OUTER JOIN ext-berater;
```

```
c) SELECT T#, TNAME, FARBE
   FROM teil UNION JOIN teil-usa
   WHERE GEWICHT < '100';
```

Aufgabe 5

```
a) SELECT f-art, d-typ
   FROM fisch, dose
   WHERE f-länge <= d-länge;
```

```
b) SELECT f-art
   FROM fisch, dose
   WHERE f-länge = d-länge
   AND d-fassungsvermögen >=
     (SELECT MIN(b-anzahl)
      FROM bestellung
      WHERE f-art = b-fischart);
```

```
c) SELECT t-name
   FROM testesser
   WHERE t-stücklohn > 50
   EXCEPT
     (SELECT t-name
      FROM testesser
      WHERE t-name IN
        (SELECT b-fischart
         FROM bestellung b
         WHERE t-wohntort = b.zielort));
```

Aufgabe 6

```
a) SELECT Tierart
   FROM TIERE
   WHERE ZooName = 'Wilhelm'
   AND Tierart LIKE 'L%'
```

```
b) SELECT Tierklasse, SUM (Anzahl)
   FROM TIERE
   WHERE Zooname = 'karlsruhe'
   GROUP BY Tierklasse
```

```
c) SELECT Nachname
   FROM WAERTER
   WHERE Alter IN
     (SELECT MAX (ALTER)
      FROM ADRESSE A, WAERTER W
      WHERE W.ZName = A.ZooName
      AND A.Stadt = 'Muenchen')
```

```
d) SELECT DienstNr
   FROM WAERTER W
   WHERE Alter >
     (SELECT SUM (Tierart)
      FROM TIERE T
      WHERE W.ZName = T.ZooName
      AND W.Tierklasse = T.Tierklasse)
```

```
e) UPDATE TIERE
   SET Anzahl = Anzahl + 1
   WHERE Tierart = 'Elefant'
   AND ZooName = 'Karlsruher Zoo'
```

```
f) UPDATE TIERE
   SET Anzahl = Anzahl - 1
   WHERE Tierart = 'Elefant'
   AND ZooName = 'Karlsruher Zoo'
```

```
g) DELETE FROM WAERTER
   WHERE DienstNr = 50101
   AND ZName = 'Karlsruher Zoo'
```

h) INSERT INTO WAERTER (ZName, DienstNr, Nachname, Alter, Tierklasse)
VALUES ('Karlsruher Zoo', 67023, Müller, 39, 'Dickhaeuter')

Aufgabe 7

a)

- SELECT K.KNr, Typ
FROM KUNDE K, ANGEBOT A
WHERE K.KNr = A.KNr
AND A.WNr IN
(SELECT WNr
FROM WOHNUNG
WHERE STName = 'Sued')
- SELECT K.KNr, Typ
FROM KUNDE K, ANGEBOT A, WOHNUNG W
WHERE K.KNr = A.KNr
AND A.WNr = W.WNr
AND W.STName = 'Sued'

b)

- SELECT W.NR, W.STName
FROM WOHNUNG W, ANGEBOT A
WHERE W.WNr = A.WNr
AND A.KNr IN
(SELECT KNR
FROM KUNDE
WHERE STName = 'Waldstadt')
- SELECT W.NR, W.STName
FROM WOHNUNG W, ANGEBOT A, KUNDE K
WHERE W.WNr = A.WNr
AND A.KNr = K.KNr
AND K.STName = 'Waldstadt'

c)

- SELECT WNR
FROM WOHNUNG
WHERE STName = 'Neureut'
AND Miete > 2000

EXCEPT

SELECT W.WNR,
FROM WOHNUNG W, ANGEBOT A
WHERE W.WNr = A.WNr
AND A.KNr IN
(SELECT KNR
FROM KUNDE
WHERE STName = 'Waldstadt')
- d) SELECT DISTINCT KNr
FROM ANGEBOT
- e) SELECT KNr, COUNT (*)
FROM ANGEBOT
GROUP BY KNr
- f) SELECT KNr, COUNT (*)
FROM ANGEBOT
GROUP BY KNr
HAVING COUNT(*) >2