

**Lösungsvorschläge
zur Nachklausur
„1671 Datenbanken I“**

8.9.2012

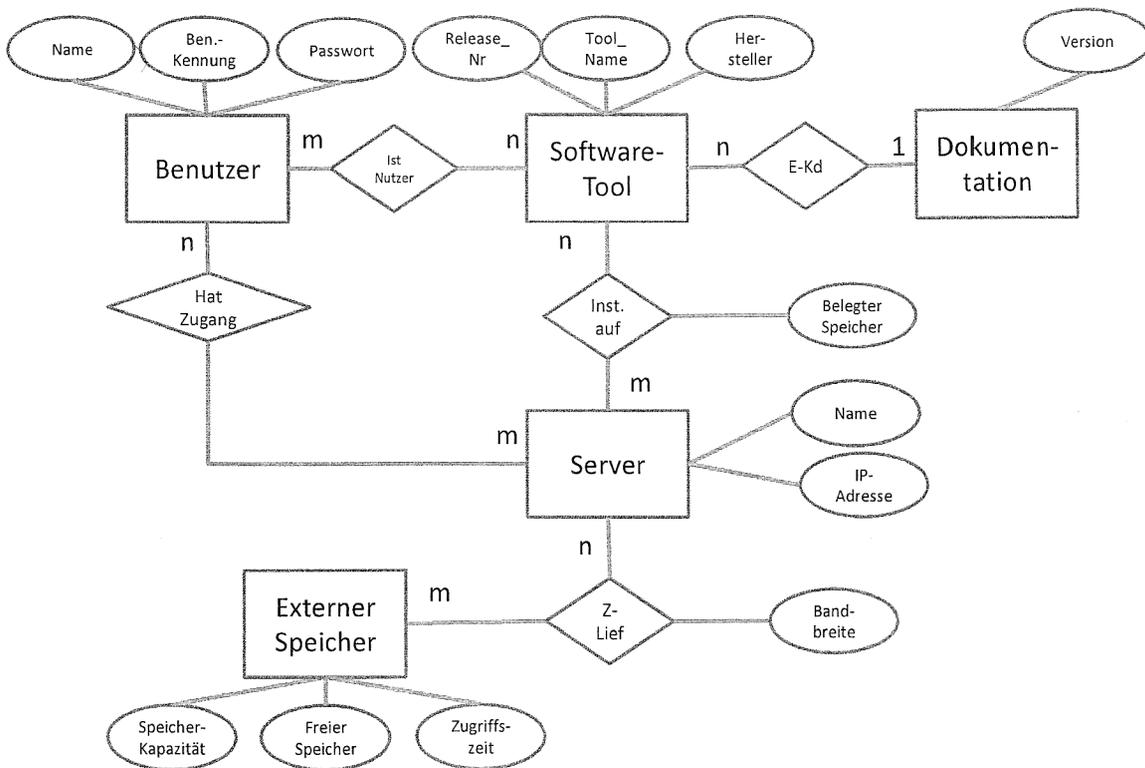
Lösung Aufgabe 1: Multiple Choice**(6 Punkte)**

Welche der folgenden Aussagen sind korrekt, welche sind nicht korrekt? Kreuzen Sie die Spalte Wahr (W) oder Falsch (F) an.

W	F	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Wegen der logischen Datenunabhängigkeit brauchen Anwendungsprogramme nicht geändert werden, wenn eine 1:n Beziehung in eine n:m Beziehung umgewandelt wird.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Eine Tupelvariable wird in SQL in der FROM-Klausel definiert.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Ein OUTER JOIN wird dazu benutzt, die Elemente der einen Relation, die über den Join kein Gegenstück in der anderen Relation finden, trotzdem in das Ergebnis des Joins zu überführen.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Sei $R(A_1, \dots, A_n)$ ein Relationenschema und X und Y Teilmengen von $\{A_1, \dots, A_n\}$. Dann ist Y funktional abhängig von X , geschrieben $X \rightarrow Y$, wenn es keine Relation vom Typ R geben kann, in der zwei Tupel denselben Wert für Y , aber verschiedene Werte für X haben.
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ein Attribut A ist voll funktional abhängig vom Schlüssel S der Relation R , wenn es keine funktionale Abhängigkeit $B \rightarrow A$ gibt mit $S \subset B$.
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Bei der algebraischen Optimierung von Queries wird versucht, Selektionen möglichst nah zu den Blättern des Operatorbaumes zu verschieben.

Lösung Aufgabe 2: Datenmodellierung

(15 Punkte)



Lösung Aufgabe 3: Datenmodellierung**(15 Punkte)**

Grundüberlegungen:

- Die Entities Tierart, Tier, Pfleger und Gehege werden jeweils auf eine Relation abgebildet.
- Die 1:n Beziehungen werden durch Attribute dargestellt, also
 - „Tier-Tierart“ wird durch ein Attribut in „Tier“ modelliert, dass als Fremdschlüssel den Schlüssel „Name“ der Relation „Tierart“ referenziert.
 - „Tier-Gehege“ wird durch ein Attribut in „Tier“ dargestellt, dass als Fremdschlüssel den Schlüssel „G_Nr“ der Relation „Gehege“ referenziert.
- Die n:m Beziehung wird durch eine eigene Relationen „Pflege“ dargestellt. Ihre einzigen Attribute sind die Referenzen auf die zu verbindenden Objekte.

Daraus ergeben sich die folgenden Tabellendefinitionen:

```

1. CREATE TABLE Tierart
      ( Name CHAR(100),
        Verbreitungsgebiet CHAR (100),
        PRIMARY KEY (Name))

2. CREATE TABLE Tier
      ( T_Nr INTEGER,
        Name CHAR(100),
        Geb-Datum DATE,
        Zoo-Eintritts-Datum DATE,
        Tierart CHAR(100),
        Gehege INTEGER,
        PRIMARY KEY (T_Nr),
        FOREIGN KEY (Tierart) REFERENCES
          Tierart (Name)),
        FOREIGN KEY (Gehege) REFERENCES
          Gehege (G_Nr))

3. CREATE TABLE Gehege
      ( G_Nr INTEGER,
        Name CHAR(100),
        Koordinaten CHAR (100),
        PRIMARY KEY (G_Nr))

4. CREATE TABLE Pfleger
      ( P_Nr INTEGER,
        Name CHAR(100),
        Adresse CHAR (500),
        PRIMARY KEY (P_Nr))

5. CREATE TABLE Pflege
      ( P_Nr INTEGER,
        T_Nr INTEGER,
        PRIMARY KEY (P_Nr, T_Nr),
        FOREIGN KEY (P_Nr) REFERENCES
          Pfleger (P_Nr),
        FOREIGN KEY (T_Nr) REFERENCES
          Tier (T_Nr))

```

Lösung Aufgabe 4: Queries**(25 Punkte)****a) 4 Punkte**

```

SELECT Verkaeufer.v_id,
       Verkaeufer.v_name,
       Verkaeufer.v_vorname
FROM   Verkaeufer
WHERE  Verkaeufer.v_werber IS NULL

```

b) 4 Punkte

```

SELECT Produkt.p_name AS Produkt,
       SUM(Umsatz.anzahl) AS Stückzahl
FROM   Produkt NATURAL JOIN Umsatz
WHERE  Umsatz.datum between 01.01.2011 and 31.12.2011
GROUP BY produkt.p_name
HAVING SUM (Umsatz.anzahl) > 9

```

c) 6 Punkte

Zunächst die Query, die nur die Vertreter als Werber anzeigt, die tatsächlich bereits andere Vertreter geworben haben:

```

SELECT Werber.v_id           AS Werber_id,
       Werber.v_name        AS Werber_Name,
       Werber.v_vorname     AS Werber_Vorname,
       Geworbener.v_id      AS Geworbener_id,
       Geworbener.v_name    AS Geworbener_Name,
       Geworbener.v_vorname AS Geworbener_Vorname
FROM   Verkaeufer AS Geworbener
       JOIN Verkaeufer AS Werber
           ON Geworbener.v_werber = Werber.v_id

```

Dann die Query, die auch die Vertreter als Werber anzeigt, die noch niemanden geworben haben:

```

SELECT Werber.v_id           AS Werber_id,
       Werber.v_name        AS Werber_Name,
       Werber.v_vorname     AS Werber_Vorname,
       Geworbener.v_id      AS Geworbener_id,
       Geworbener.v_name    AS Geworbener_Name,
       Geworbener.v_vorname AS Geworbener_Vorname
FROM   Verkaeufer AS Geworbener
       RIGHT OUTER JOIN Verkaeufer AS Werber
           ON Geworbener.v_werber = Werber.v_id

```

In der ersten Query wird der normale Join benutzt, wodurch die Tupel nicht in das Ergebnis übernommen werden, die keinen Wert enthalten oder kein Gegenstück in der anderen Relation haben. In der zweiten Query wird ein outer Join benutzt. Dieser bewirkt, dass alle Tupel der Relation Verkäufer in der Rolle Werber in das Join-Ergebnis übernommen werden, auch wenn sie keinen Wert enthalten oder kein

Gegenstück in der anderen Relation haben. Somit werden also Verkäufer angezeigt, auch wenn sie noch keinen anderen Verkäufer geworben haben.

d) 6 Punkte

```

SELECT Verkaeufer.v_id      AS ID,
       Verkaeufer.v_name   AS Name,
       Verkaeufer.v_vorname AS Vorname
FROM   Verkaeufer
WHERE  Verkaeufer.v_id =
      (SELECT Verkaeufer.v_werber
       FROM   Verkaeufer
       WHERE  Verkaeufer.v_id =
            (SELECT Umsatz.v_id
             FROM   Umsatz
             WHERE  Umsatz.anzahl = 7
             AND    Umsatz.datum = '12.04.2011'
             AND    Umsatz.p_id IN
                  (SELECT Produkt.p_id
                   FROM   Produkt
                   WHERE  Produkt.p_name = 'SAT-Receiver'
                  )
            )
       )
      )

```

e) 5 Punkte

```

SELECT Verkaeufer.v_id      AS ID,
       Verkaeufer.v_name   AS Name,
       SUM (umsatz.anzahl * produkt.p_preis)
FROM   Verkäufer
       NATURAL JOIN Umsatz
       NATURAL JOIN Produkt
WHERE  p_name = 'HD-SAT Receiver XY'
AND    Umsatz.Datum between 2011-01-01 and 2011-12-31
GROUP BY Verkaeufer.v_id

```

Lösung Aufgabe 5: Fkt. Abhängigkeiten, Normalformen (15 Punkte)

a)

Den Schlüssel der Relation Versicherter bilden die beiden Attribute Name und M_Nr.

Schlüssel der Relation Medikament ist das Attribut M_Nr.

b)

Die Relation Medikament ist in der 2. Normalform, da sie keinen zusammengesetzten Schlüssel hat und da damit jedes Attribut voll funktional vom Schlüssel abhängig ist. Sie ist aber nicht in der 3. Normalform, da es eine funktionale Abhängigkeit gibt, die nicht von einem Schlüssel ausgeht (Hersteller → Hersteller_Zulassungs_Nr).

Die Relation Versicherter ist nicht in 2. Normalform, da vom Schlüsselattribut Name funktionale Abhängigkeiten ausgehen, die aber nicht vom ganzen Schlüssel ausgehen (z.B. Name → Adresse).

c)

Um Medikament in die 3NF zu überführen, muss die transitive funktionale Abhängigkeit über das Nicht-Schlüssel-Attribut Hersteller eliminiert werden. Dies geschieht durch Auslagerung der Attribute in eine eigene Tabelle. Damit erhalten wir die folgenden Relationen:

Medikament1:

M_Nr, M_Name, Wirkstoff, Hersteller, M_Zulassungs_Nr

Produzent:

Hersteller, Hersteller_Zulassungs_Nr

Man beachte, dass Hersteller in der Relation Medikament1 verbleiben muss, sonst wäre es nicht möglich, die ursprüngliche Relation durch einen Join wieder herzustellen.

Die Relation Versicherter müssen wir zunächst in die 2NF bringen, indem wir die nicht voll funktionalen Abhängigkeiten eliminieren. Dazu lagern wir die vom ganzen Schlüssel abhängigen Attribute mit dem ganzen Schlüssel aus in die Relation Medikamentenkonsum:

Medikamentenkonsum:

M_Nr, Name, verschriebene_Menge

Versicherter1:

Name, Adresse, Geburtsdatum, KK_Nr, KK_Name, KK-Adresse,
Arzt_Zulassungs_Nr, Hausarzt_Name, Arzt_Adresse

Versicherter1 hat nun Name als Schlüssel und ist nun auch in 2NF, ebenso wie Medikamentenkonsum. Medikamentenkonsum ist auch in 3NF.

Um Versicherter1 in 3NF zu überführen, müssen wir wiederum die transitiven Abhängigkeiten eliminieren. Dies führt zu den folgenden Relationen:

Versicherter2:

Name, Adresse, Geburtsdatum, Krankenkasse, Hausarzt_Name

KK:

KK_Nr, KK_Name, KK_Adresse

Hausarzt:

Hausarzt_Name, Arzt_Zulassungs_Nr, Arzt_Adresse

Lösung Aufgabe 6: Normalformen

(6 Punkte)

Eine Relation mit nur einem Schlüsselattribut ist immer in 2. Normalform.

Da es nur ein Schlüsselattribut gibt, ist jedes Attribut voll funktional vom Schlüssel → Relation in 2NF.

Da es aber nicht ausgeschlossen ist, dass es weitere Abhängigkeiten zwischen Nicht-Schlüsselattributen gibt, ist sie nicht unbedingt in 3NF.