

Prof. Dr. V. Linnemann  
Christoph Reinke  
Universität zu Lübeck  
Institut für Informationssysteme

Lübeck, den 29. Juli 2010

# **Datenbanken**

## Sommersemester 2010

### **Probeklausur**

#### **Hinweise:**

- Es ist sinnvoll, die Aufgaben zum Test zunächst in der in der Klausur vorgegebenen Zeit zu lösen (90 Minuten).
- Anschließend sollten Sie die Aufgaben gründlicher bearbeiten und Ihre Lösungen angeben.
- Die angegebenen Punkte dienen nur Ihrer Orientierung und werden nicht für das Erlangen der Zugangsvoraussetzungen zur Klausur berücksichtigt.

## Aufgabe 1: ER-Modell und relationales Modell

Eine Internetauktionsseite kann mit folgenden Entitäten modelliert werden:

- Auktion(laufende sowie abgeschlossene),
- Produkt und
- Person.

- a) Modellieren Sie eine Internetauktionsseite basierend auf den angegebenen Entitäten in Form eines ER-Diagrammes. Ordnen Sie dabei allen Entitäten sinnvolle Attribute und Beziehungen zu. (5 Punkte)
- b) Geben Sie im ER-Diagramm für alle Beziehungen sinnvolle Kardinalitäten an. (5 Punkte)
- c) Setzen Sie das ER-Diagramm in ein relationales Schema um. Verwenden Sie dabei möglichst wenige Relationen. Kennzeichnen Sie Primär- und Fremdschlüssel. (10 Punkte)

## Aufgabe 2: Anfragesprachen

Die Datenbank eines Kaufhauses wird mit folgenden SQL-Befehlen definiert:

```
CREATE TABLE Abteilung (  
    Name          CHAR(20) NOT NULL,  
    Chefnummer   SMALLINT,  
    PRIMARY KEY(Name));  
  
CREATE TABLE Angestellter(  
    Nummer        SMALLINT NOT NULL,  
    Name          CHAR(30) NOT NULL,  
    Adresse       CHAR(50) NOT NULL,  
    Abteilungsname CHAR(20) NOT NULL,  
    Chefnummer    SMALLINT NOT NULL,  
    PRIMARY KEY(Nummer) ,  
    FOREIGN KEY(Abteilungsname) REFERENCES Abteilung ,  
    FOREIGN KEY(Chefnummer) REFERENCES Angestellter);
```

```

ALTER TABLE Abteilung
  ADD CONSTRAINT FK_CHEFNR
  FOREIGN KEY (Chefnummer) REFERENCES Angestellter;

CREATE TABLE Artikel(
  Artikelnr    SMALLINT NOT NULL,
  Bezeichnung  CHAR(50) NOT NULL,
  Verk_Preis   DECIMAL(10,2) NOT NULL,
  PRIMARY KEY(Artikelnr));

CREATE TABLE Lieferant(
  Name        CHAR(20) NOT NULL,
  Adresse     CHAR(50) NOT NULL,
  PRIMARY KEY(Name));

CREATE TABLE Bietet_An(
  Abteilungsname CHAR(20) NOT NULL,
  Artikelnr      SMALLINT NOT NULL,
  PRIMARY KEY(Abteilungsname, Artikelnr),
  FOREIGN KEY(Abteilungsname) REFERENCES Abteilung,
  FOREIGN KEY(Artikelnr) REFERENCES Artikel);

CREATE TABLE Liefert(
  Lieferantename CHAR(20) NOT NULL,
  Artikelnr      SMALLINT NOT NULL,
  Produktnr     SMALLINT NOT NULL,
  Eink_Preis    DECIMAL(10,2) NOT NULL,
  PRIMARY KEY(Lieferantename, Artikelnr),
  FOREIGN KEY(Lieferantename) REFERENCES Lieferant,
  FOREIGN KEY(Artikelnr) REFERENCES Artikel);

```

Die Tabellen **Abteilung**, **Angestellter**, **Artikel** und **Lieferant** sind selbsterklärend. Die Relation **Bietet\_An** speichert die Beziehung "Abteilung bietet Artikel an". Die Tabelle **Liefert** speichert die Beziehung "Lieferant liefert Artikel", wobei Produktnr eine vom Lieferanten vergebene Nummer ist.

Formulieren Sie folgende Anfragen jeweils in SQL und relationaler Algebra. Formulieren Sie die Anfragen außerdem entweder im Tupel- oder Domänenkalkül. Insgesamt sind also 9 Anfragen zu formulieren.

- a) Welche Lieferanten liefern keine Artikel? Geben Sie die Namen der Lieferanten aus. (6 Punkte)

- b) Wie lauten die Artikel, die in mehr als einer Abteilung angeboten werden? Geben Sie die Artikelbezeichnung und den Namen der Abteilung aus. (9 Punkte)
- c) Welches ist der/sind die Artikel mit dem höchsten Verkaufspreis? Auszugeben sind Artikelbezeichnung und Verkaufspreis. (9 Punkte)

### Aufgabe 3: Funktionale Abhängigkeiten

Ein erster Entwurf eines Krankenhaus-Informationssystems (KIS) führte zur Identifizierung einer Menge  $F = \{KT \rightarrow S, T \rightarrow K, KP \rightarrow T\}$  von funktionalen Abhängigkeiten für die Attribute Patientenummer (P), Therapienummer (T), Krankheit (K) und Symptome (S).

- a) Bestimmen Sie sämtliche Schlüssel für die Relation  $R(P, T, K, S)$  bzgl. der funktionalen Abhängigkeiten  $F$ . (2 Punkte)
- b) Ist die Menge  $F$  minimal? Begründung! (4 Punkte)
- c) Finden Sie eine zu  $F$  äquivalente und minimale Menge von funktionalen Abhängigkeiten  $G$ . Beweisen Sie anschließend, dass  $G$  wirklich äquivalent zu  $F$  und minimal ist. (4 Punkte)

### Aufgabe 4: Verbundtreue Dekomposition

Nach einer genaueren Überarbeitung der funktionalen Abhängigkeiten für das KIS wurde die Menge  $F$  aus der letzten Aufgabe verworfen und an deren Stelle die Menge  $F_{KIS} = \{T \rightarrow K, PT \rightarrow Z, P \rightarrow N, K \rightarrow S\}$  mit den zusätzlichen Attributen Patientename (N) und Behandlungszeitraum (Z) aufgestellt.

- a) Zeigen Sie, dass  $PK$  kein Schlüssel von  $F_{KIS}$  ist. (2 Punkte)
- b) Überprüfen Sie mit dem Algorithmus der Vorlesung, ob die Dekomposition mit  $R_1(P, N)$ ,  $R_2(T, K, S)$  und  $R_3(P, Z, T)$  verbundtreu ist. (6 Punkte)

### Aufgabe 5: Dekomposition

Für die Relation  $R(F, D, J, A, V, N, M)$  über Flugdaten mit F(Flugnummer), D(Datum), J(Flugzeugnummer), A(Fluggesellschaft), V(Von), N(Nach) und M(Meilen) des letzten Zettels sei folgende verbundtreue Dekomposition gegeben  $R_1(F, A, N)$ ,  $R_2(F, D, J)$  und  $R_3(F, V, M)$ . Darüberhinaus ist  $F$  die Menge von funktionalen Abhängigkeiten mit  $F = \{FD \rightarrow JVNMA, AFD \rightarrow J, VN \rightarrow M, F \rightarrow AVNM, F \rightarrow A, VND \rightarrow M\}$ , die gelten soll.

- Ist diese Dekomposition abhängigkeiterhaltend? (Herleitung) (9 Punkte)

### Aufgabe 6: Normalformen

Grundlage dieser Aufgabe ist eine Datenbank mit den folgenden Attributen: B (Broker), O (Büro eines Brokers), I (Investor), S (Aktie), Q (Anzahl Aktien, die ein Investor besitzt), D (Dividende für eine Aktie). Die folgenden funktionalen Abhängigkeiten sollen gelten:  $S \rightarrow D$ ,  $I \rightarrow B$ ,  $IS \rightarrow Q$  und  $B \rightarrow O$ .

Finden Sie eine verbundtreue Dekomposition des Schemas  $R(B,O,S,Q,I,D)$  in Boyce-Codd Normalform mittels des in der Vorlesung angegebenen Algorithmus. (9 Punkte)