
DATENBANKEN UND INFORMATIONSSYSTEME

SOMMERSEMESTER 2012

PROBEKLAUSUR

1 Konzeptuelle Modellierung (12 Punkte)

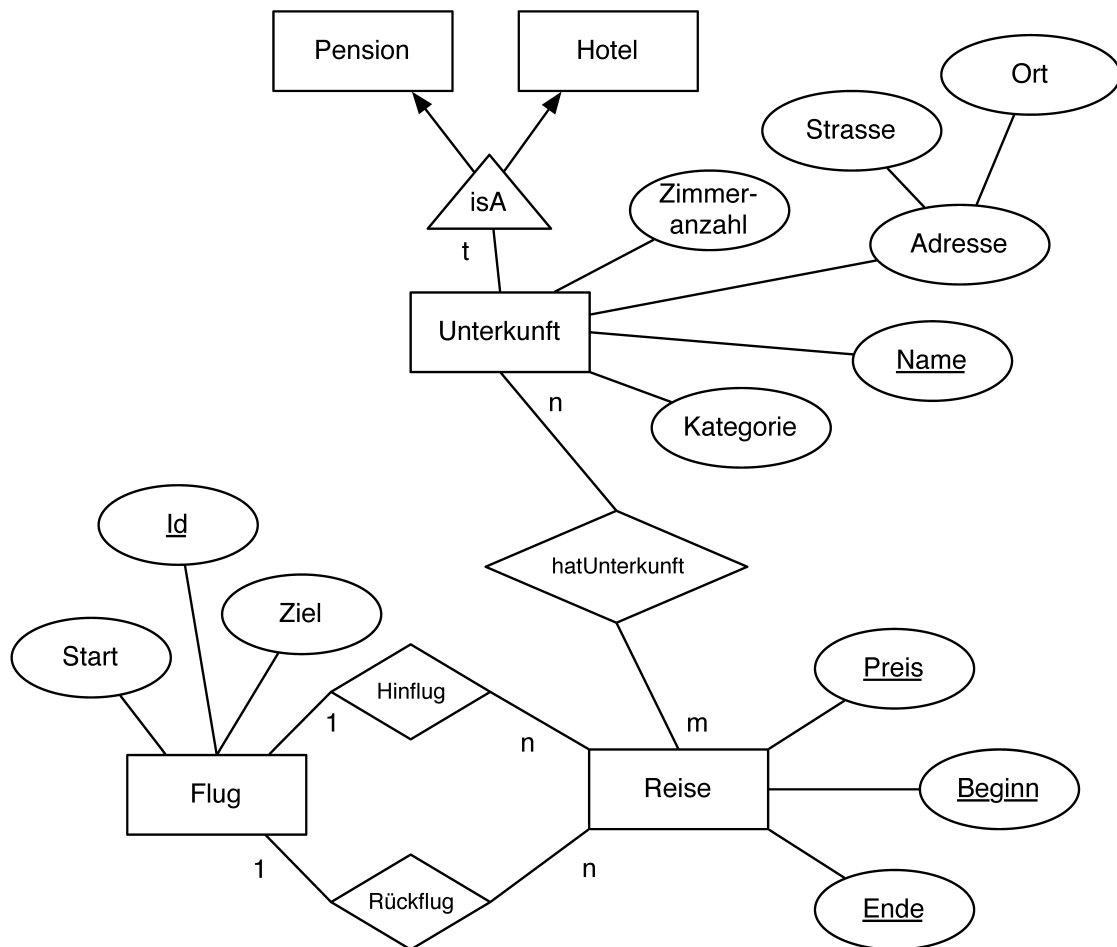
Die folgende Beschreibung skizziert ein Informationssystem zur Verwaltung von Musikern:

- Jeder Musiker soll mit seinem Namen, bestehend aus Vor- und Nachname, und der Anzahl an Berufsjahren gespeichert werden. Ein Musiker kann eindeutig über seine Künstler ID identifiziert werden. Ein Musiker ist entweder Schlagwerker, Bläser oder Streicher. In Zukunft kann es aber noch andere Musiker in dem Informationssystem geben.
- Ein Konzert wird von einem oder mehreren Orchestern gegeben, wobei jedes Orchester natürlich auch mehrere Konzerte geben kann. Ein Orchester besteht aus einem oder mehreren Musikern. Das Orchester kann eindeutig über seinen Namen identifiziert werden. Außerdem möchte man zu jedem Orchester den Träger (d.h., städtisch, privat etc.) speichern. Ein Musiker kann zu einem oder mehreren Orchestern gehören, muss dies aber nicht.
- Ein Konzert besteht aus mindestens zwei Musikstücken. Dabei wird ein Konzert durch das Datum, das Motto und den Ort eindeutig identifiziert. Ein Musikstück kann bei keinem, einem oder mehreren Konzerten von einem oder mehreren Musikern gespielt werden.
- Zu jedem Musikstück wird der Name und ein oder mehrere Tonarten gespeichert. Ein Musikstück wird über seinen Namen identifiziert. Da dies allein nicht ausreicht (z. B. "9. Symphonie"), wird es noch über seinen Komponisten identifiziert (z. B. "Wolfgang A. Mozart"). Der Komponist wird dabei über seinen Namen eindeutig identifiziert. Außerdem wird zu jedem Komponisten Geburtsdatum und Geburtsort gespeichert.

Entwerfen Sie ein Entity-Relationship-Diagramm, das die gegebenen Anforderungen umsetzt. Verwenden Sie zur Repräsentation der Kardinalitäten die Min-Max-Notation.

2 Relationales Datenbankschema (10 Punkte)

Geben Sie eine Umsetzung des folgenden konzeptuellen Schemas in ein relationales Datenbankschema an. Dokumentieren Sie auch die interrelationalen Abhängigkeiten. Verwenden Sie die in der Vorlesung vorgestellte Notation.



3 Relationale Algebra (6 (4) Punkte)

Das nachstehende Datenbankschema speichert die Daten eines *ProgShops*:

Kunde(Email, KName, Bezahlart)
Kauft(Email, PName, Datum)
Programm(PName, Beschreibung, Altersempfehlung, Preis)
HatKategorie(PName, Kat)

Die Relation *Kauft* speichert also die von einem Kunden erworbenen Programme und die Relation *HatKategorie* die Zuordnung von Programmen zu Kategorien.

Es gelten die folgenden interrelationalen Abhängigkeiten:

$\text{Kauft}(\text{Email}) \subseteq \text{Kunde}(\text{Email})$
 $\text{Kauft}(\text{PName}) \subseteq \text{Programm}(\text{PName})$
 $\text{HatKategorie}(\text{PName}) \subseteq \text{Programm}(\text{PName})$

Drücken Sie das folgende Informationsbedürfnis in **Relationaler Algebra** aus.

- Namen und Beschreibungen aller Programme, die von „Kreditkarten“-Kunden gekauft wurden.
Hinweis: „Kreditkarte“ ist eine *Bezahlart*. (4 Punkte)
- Bearbeiten Sie diese Aufgabe nicht, wenn Sie die 90-minütige Klausur schreiben!**
Die Namen aller Kunden, die noch kein Programm gekauft haben. (2 Punkte)

4 SQL (12 (7) Punkte)

Diese Aufgabe bezieht sich ebenfalls auf das Datenbankschema zum *ProgShop* aus Aufgabe 3.

Drücken Sie die nachstehenden Informationsbedürfnisse in **SQL** aus:

- Die Namen aller Programme, die eine Altersempfehlung kleiner als 6 Jahre haben und der Kategorie „Lernprogramme“ zugeordnet sind. (3 Punkte)
- Erstellen Sie eine Übersicht, die pro Kunde seinen Namen, die Zahl der erworbenen Programme und die Summe der Kosten hierfür enthält. (4 Punkte)
- Bearbeiten Sie diese Aufgabe nicht, wenn Sie die 90-minütige Klausur schreiben!**
Geben Sie zu jeder Kategorie Programmname und Preis des teuersten Programms aus und sortiere Sie die Ausgabe absteigend nach Preis. (5 Punkte)

5 Funktionale Abhängigkeiten & Normalformen (7 (6) Punkte)

- a) Geben Sie ein Relationenschema mit intrarelationalen Abhängigkeiten an, das in 3NF aber nicht in BCNF ist. (2 Punkte)

Betrachten Sie das Relationenschema:

$$S = (\{A, B, C, D, E, F\}, \{AB \rightarrow DE, C \rightarrow EF, F \rightarrow BC, D \rightarrow AE\})$$

- b) Identifizieren Sie alle Schlüsselkandidaten von S . Zeigen Sie, dass es sich um Schlüsselkandidaten handelt und beweisen Sie, dass es keine anderen Schlüsselkandidaten gibt. (4 Punkte)
- c) **Bearbeiten Sie diese Aufgabe nicht, wenn Sie die 90-minütige Klausur schreiben!**
Ist die funktionale Abhängigkeit $F \rightarrow CE$ aus der obigen Menge funktionaler Abhängigkeiten der Relation S ableitbar? Geben Sie Ihren Lösungsweg an. (1 Punkt)

6 Synthese & Dekomposition (16 (10) Punkte)

Gegeben das folgende Relationenschema:

$$R = (\{A, B, C, D, E, F, G\}, \{A \rightarrow BC, BD \rightarrow CF, EG \rightarrow F, D \rightarrow GE\})$$

- a) Bestimmen Sie den Schlüsselkandidaten von R . (2 Punkte)
- b) Wenden Sie den Synthese-Algorithmus zur Überführung des Schemas in die dritte Normalform an. (8 Punkte)
- c) **Bearbeiten Sie diese Aufgabe nicht, wenn Sie die 90-minütige Klausur schreiben!**
Auf demselben Schema seien bereits zwei Schritte des Dekompositions-Algorithmus ausgeführt:
1. Eine Dekomposition entlang $A \rightarrow BC$ mit dem Ergebnis
 $R_1 = (\{A, B, C\}, \{A \rightarrow BC, \dots\})$ (ist in BCNF) und
 $R_2 = (\{A, D, E, F, G\}, \{EG \rightarrow F, D \rightarrow GE, \dots\})$ (nicht in BCNF!) und
 2. Eine Dekomposition entlang $D \rightarrow GE$ mit dem Ergebnis
 $R_{21} = (\{D, E, G\}, \{D \rightarrow GE, \dots\})$ und
 $R_{22} = (\{A, D, F\}, \{\dots\})$

Ist der Dekompositions-Algorithmus fertig?

Zeigen Sie also entweder, dass R_{21} und R_{22} in BCNF sind oder führen Sie den Algorithmus fort, bis dies für alle entstehenden Relationen der Fall ist. (6 Punkte)

7 Serialisierbarkeit & Nebenläufigkeit (14 (10) Punkte)

- a) Erläutern Sie kurz das Dirty-Read-Problem. Geben Sie einen Schedule an, in dem ein Dirty-Read-Problem auftritt. (2 Punkte)

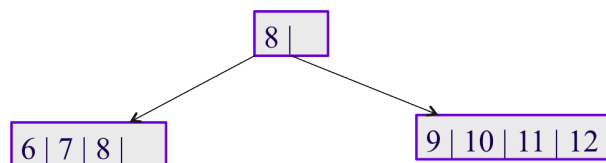
Gegeben sei der folgenden Schedule s :

$$s = r_1(x)r_2(x)w_3(y)w_2(z)r_1(z)c_3w_1(x)r_2(y)c_2r_1(y)w_1(z)c_1$$

- b) Bestimmen Sie, ob der Schedule konfliktserialisierbar ist. Geben Sie dazu die Konfliktmenge und den Konfliktgraphen an. Begründen Sie Ihre Entscheidung. (3 Punkte)
- c) Bestimmen Sie jeweils für die Klassen RC , ACA und ST , ob der Schedule s in dieser Klasse enthalten ist. Begründen Sie Ihre Entscheidung. (5 Punkte)
- d) **Bearbeiten Sie diese Aufgabe nicht, wenn Sie die 90-minütige Klausur schreiben!** Geben Sie an, wie die Ausgabe eines "S2PL" Schedulers, d. h. eines "strikten" 2PL Schedulers, wäre, wenn s seine Eingabe wäre. Beschreiben Sie kurz, den Unterschied zwischen einem "S2PL" und einem "2PL" Scheduler. (4 Punkte)

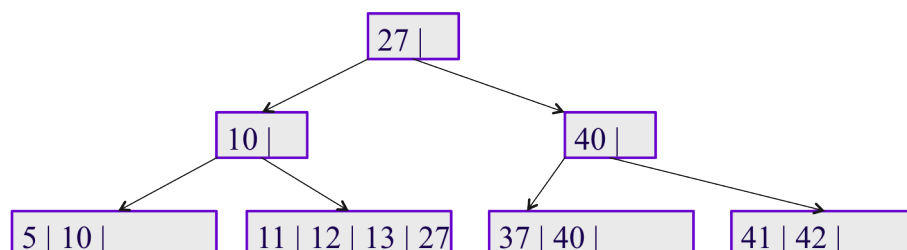
8 B⁺-Bäume (9 (4) Punkte)

- a) Gegeben sei der folgende B⁺-Baum mit $k = 1$ und $k^* = 2$:



Führen Sie die Operationen $insert(14)$ und anschließend $delete(14)$ darauf aus. Geben Sie die Zwischenschritte für jede Operation an und stellen Sie die Baumstruktur nach jeder Operation dar. (4 Punkte)

- b) **Bearbeiten Sie diese Aufgabe nicht, wenn Sie die 90-minütige Klausur schreiben!** Gegeben sei der folgende B⁺-Baum mit $k = 1$ und $k^* = 2$:



Führen Sie die Operationen $delete(40)$ und anschließend $delete(5)$ darauf aus. Geben Sie die Zwischenschritte für jede Operation an und stellen Sie die Baumstruktur nach jeder Operation dar. (5 Punkte)

9 XML (8 Punkte)

Betrachten Sie das folgende XML-Dokument als Beispiel eines XML-Dokuments gleicher Struktur, das eine Datenbank von Hotelketten, Hotels und deren Gästen darstellt:

```
<hotelliste>
  <hotelkette name="Tempelhotels">
    <hotel name="Devisen-Tempel">
      <ausstattung typ="Casino"/>
      <ausstattung typ="Sonnenbank"/>
      <gaeste>
        <gast name="Jürgen Silberpfennig">
          <zimmer>343</zimmer>
          <kundenstatus>silber</kundenstatus>
        </gast>
        <gast name="Matthias Nordmann">
          <zimmer>342</zimmer>
          <kundenstatus>platin</kundenstatus>
        </gast>
      </gaeste>
    </hotel>
    ...
  </hotelkette>
  <hotelkette name="Inselhotels">
    <hotel name="Wellness-Insel">
      <ausstattung typ="Massageliegen"/>
      <ausstattung typ="Whirlpool"/>
      <ausstattung typ="Dampfbad"/>
      <gaeste>
        <gast name="Chris Orange">
          <zimmer>333</zimmer>
          <kundenstatus>orange</kundenstatus>
        </gast>
      </gaeste>
    </hotel>
    <hotel name="Bier-Insel">
      <ausstattung typ="Bar"/>
      <ausstattung typ="Fassbad"/>
      <gaeste>
        <gast name="Alois Potton">
          <zimmer>523</zimmer>
          <kundenstatus>blau</kundenstatus>
        </gast>
        <gast name="Carl Heinrichs">
          <zimmer>423</zimmer>
          <kundenstatus>gold</kundenstatus>
        </gast>
      </gaeste>
    </hotel>
    ...
  </hotelkette>
  ...
</hotelliste>
```

Adressieren Sie durch Angabe entsprechender **XPath**-Ausdrücke die wie folgt beschriebenen XML-Knoten:

- a) Die Hotels der Hotelkette „Inselhotels“. (2 Punkte)
- b) Die Gäste mit Kundenstatus „gold“ der Hotels mit „Massageliegen“. (2 Punkte)

Drücken Sie das folgende Informationbedürfnis in **XQuery** aus.

- c) Alle Paare aus Hotelbewohnern, die übereinander wohnen. (4 Punkte)
Hinweis: Es wird davon ausgegangen, dass sich Hotelzimmer mit einer Differenz von 100 in der Zimmernummer übereinander befinden (z. B. liegt das Zimmer 312 über dem Zimmer 212).

Die Ausgabe soll dem folgenden Format entsprechen:

```
<obermieter>
  <wohntueber>
    <oben>Alois Potton</oben>
    <unten>Carl Heinrichs</unten>
  </wohntueber>
  <wohntueber>
    <oben>...</oben>
    <unten>...</unten>
  </wohntueber>
  ...
</obermieter>
```