

Lösung Probeklausur



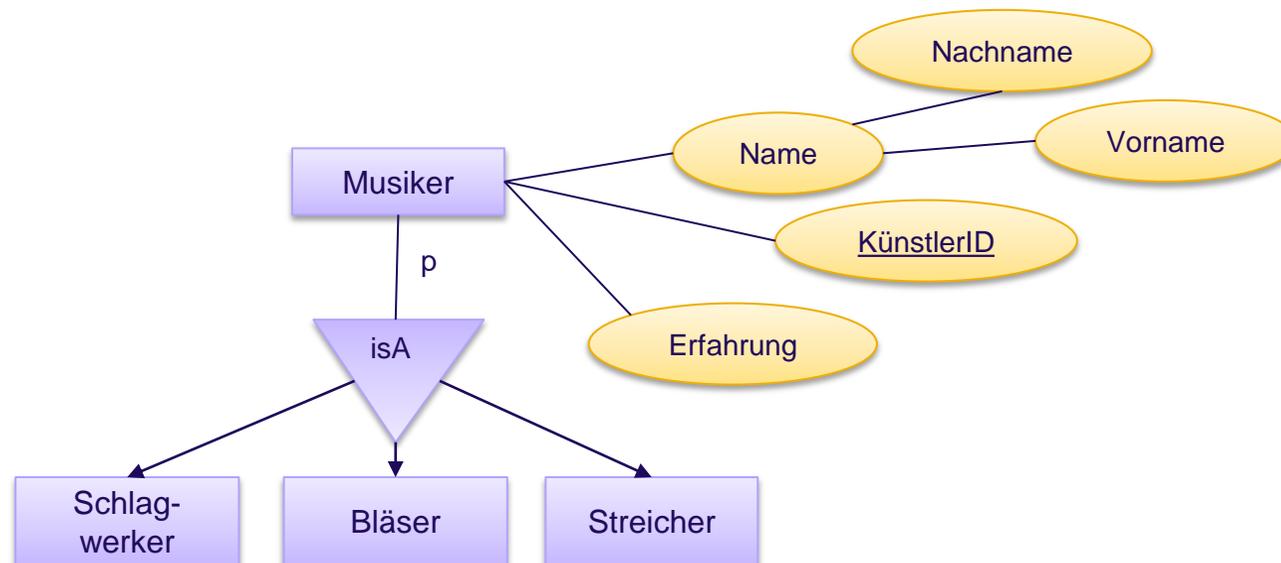
Sommersemester 2012

Aufgabe 1:

Konzeptuelle Modellierung

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.

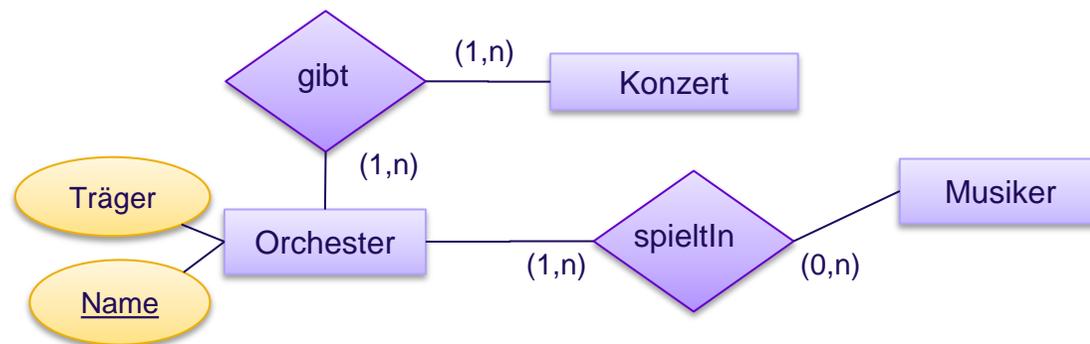
Ein Musiker ist entweder Schlagwerker, Bläser oder Streicher. In Zukunft kann es aber noch andere Musiker in dem Informationssystem geben.



Aufgabe 1:

Konzeptuelle Modellierung

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.

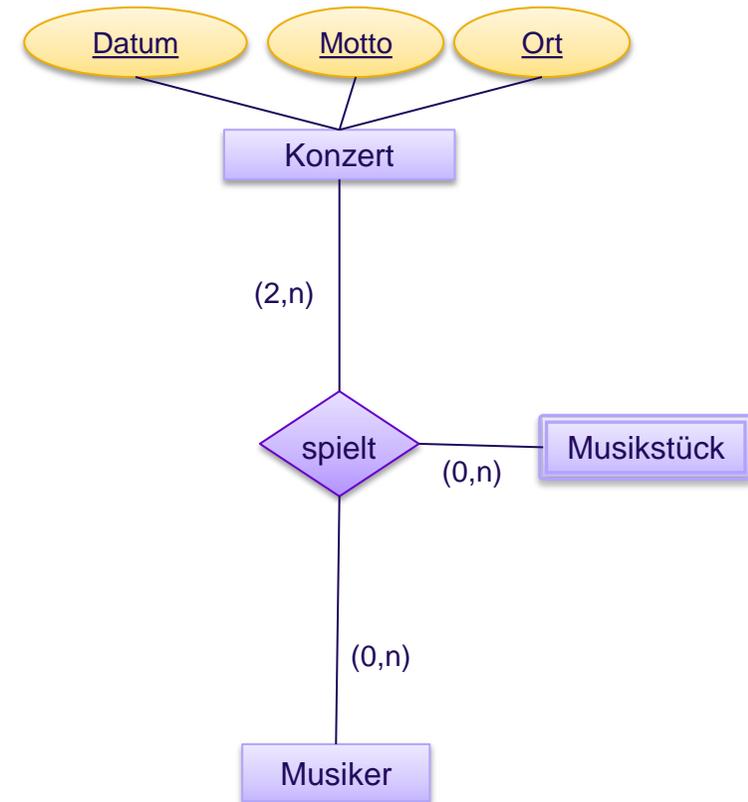


Aufgabe 1:

Konzeptuelle Modellierung

Ein Konzert besteht aus mindestens zwei **Musikstücken**. Dabei wird ein Konzert durch das Datum, das Motto und den Ort eindeutig identifiziert.

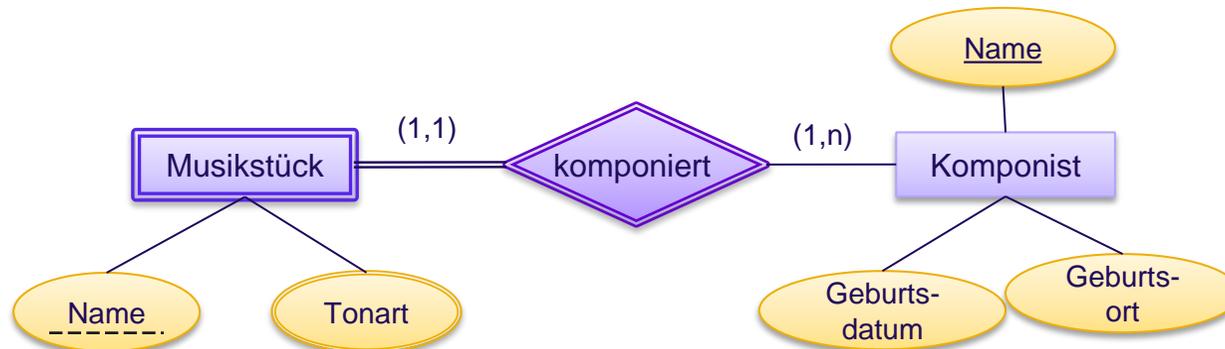
Ein Musikstück kann bei keinem, einem oder mehreren Konzerten von einem oder mehreren Musikern gespielt werden.



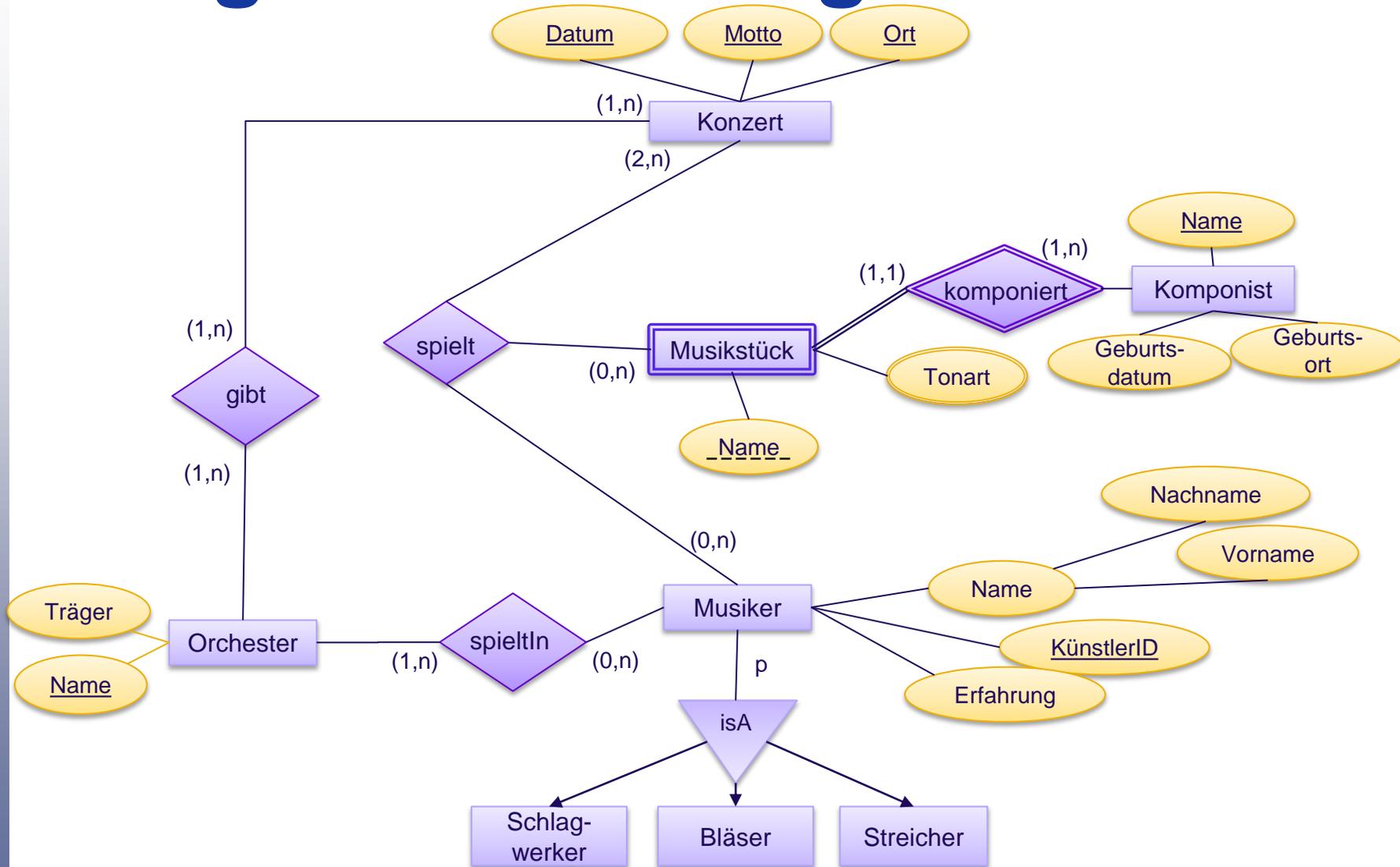
Aufgabe 1:

Konzeptuelle Modellierung

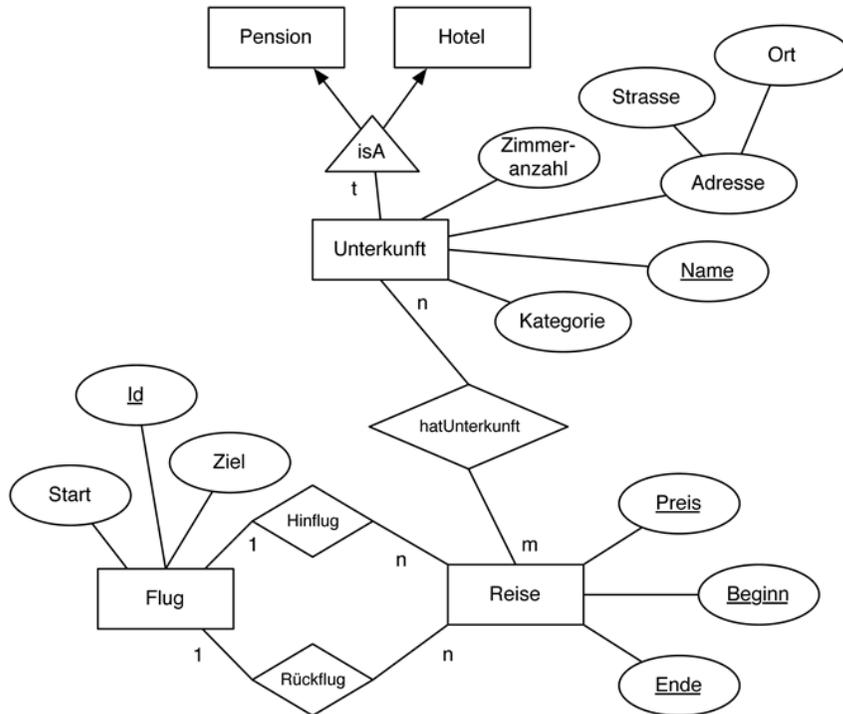
Zu jedem Musikstück wird der Name und ein oder mehrere Tonarten gespeichert. Ein Musikstück wird über seinen Namen identifiziert. Da dies allein nicht ausreicht (z. B. “9. Symphonie”), wird es noch über seinen **Komponisten** identifiziert (z. B. “Wolfgang A. Mozart”). Der Komponist wird dabei über seinen Namen eindeutig identifiziert. Außerdem wird zu jedem Komponisten Geburtsdatum und Geburtsort gespeichert.



Aufgabe 1: Lösung ER-Modell



Aufgabe 2: ER nach RDM



◆ Unterkunft(Name, Str, Ort, Zimmer, Kat, Typ)

◆ Reise(Preis, Beginn, Ende, HinID, RückID)

◆ Flug(ID, Start, Ziel)

◆ HatUnterkunft(Name, Preis, Beginn, Ende)

◆ HatUnterkunft[Name] ⊆ Unterkunft[Name]

◆ HatUnterkunft[Preis, Beginn, Ende] ⊆ Reise[Preis, Beginn, Ende]

◆ Reise[HinID] ⊆ Flug[ID]

◆ Reise[RückID] ⊆ Flug[ID]

Aufgabe 2: Varianten

- ◆ ISA: 1 Relation für Unterkunft, Pension und Hotel
- ◆ ISA: 1 Relation für Hotel und Pension
 - Für die ersten beiden Fälle sind auch weitere interrel. Abhängigkeiten erforderlich
- ◆ Falsch: 1 Relation mit zwei Attributen „isHotel“, „isPension“

Aufgabe 3+4: „ProgShop“

Das nachstehende Datenbankschema speichert die Daten eines *ProgShops*:

Kunde(*Email*, *KName*, *Bezahlart*)

Kauft(*Email*, *PName*, *Datum*)

Programm(*PName*, *Beschreibung*, *Altersempfehlung*, *Preis*)

HatKategorie(*PName*, *Kat*)

Es gelten die folgenden interrelationalen Abhängigkeiten:

$Kauft(Email) \subseteq Kunde(Email)$

$Kauft(PName) \subseteq Programm(PName)$

$HatKategorie(PName) \subseteq Programm(PName)$

Aufgabe 3: Lösung

Kunde(Email, KName, Bezahlart)

Kauft(Email, PName, Datum)

Programm(PName, Beschreibung,
Altersempfehlung, Preis)

HatKategorie(PName, Kat)

a) Relationale Algebra: Namen und Beschreibungen aller Programme, die von „Kreditkarten“-Kunden gekauft wurden.

Hinweis: „Kreditkarte“ ist eine *Bezahlart*.

$$\pi_{PName, Beschreibung} (Programm \bowtie (Kauft \bowtie (\sigma_{Bezahlart="Kreditkarte"} Kunde)))$$

b) Relationale Algebra: Die Namen aller Kunden, die noch kein Programm gekauft haben.

$$\pi_{KName} (Kunde - (Kunde \bowtie Kauft))$$

Aufgabe 4: Lösung

a) Die Namen aller Programme, die eine Altersempfehlung kleiner als 6 Jahre haben und der Kategorie „Lernprogramme“ zugeordnet sind.

```
SELECT Programm.PName
FROM Programm, HatKategorie
WHERE Programm.PName=HatKategorie.PName
        AND Kat='Lernprogramme'
        AND Altersempfehlung<6;
```

b) Erstellen Sie eine Übersicht, die pro Kunde seinen Namen, die Zahl der erworbenen Programme und die Summe der Kosten hierfür enthält.

```
SELECT Kunde.Email, KName, COUNT (*), SUM(Preis)
AS Kosten
FROM Programm, Kauft, Kunde
WHERE Programm.PName=Kauft.PName
        AND Kauft.Email=Kunde.Email
GROUP BY Kunde.Email, Kunde.KName
```

Aufgabe 4: Lösung (Forts.)

- c) Geben Sie zu jeder Kategorie Programmname und Preis für das (die) teuerste(n) Programm(e) aus. Sortieren Sie absteigend nach Preis.

```
SELECT Kat, K.PName, Preis
FROM HatKategorie K, Programm P
WHERE K.PName = P.PName
        AND P.Preis=
        (SELECT MAX(P2.Preis)
         FROM Programm P2, HatKategorie K2
         WHERE P2.PName=K2.PName
         AND K2.Kat = K.Kat)
ORDER BY Preis DESC
```

Aufgabe 5: Funktionale Abhängigkeiten & Normalisierung

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

$$R=(ABC, \{AB \rightarrow C, C \rightarrow B\})$$

- *Schlüssel: AB und AC*
- *3NF \rightarrow alle NSA sind direkt von allen Schlüsseln abhängig*
- *Hier: keine NSA, daher in 3NF*
- *BCNF \rightarrow alle Attribute sind direkt von allen Schlüsseln abhängig*

Aufgabe 5: Funktionale Abhängigkeiten & Normalisierung

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 . [..]

- *Attributhüllen der einzelnen Attribute:*

$$A^+: A \rightarrow A$$

$$B^+: B \rightarrow B$$

$$C^+: C \rightarrow CEF \rightarrow BCEF$$

$$D^+: D \rightarrow ADE$$

$$E^+: E \rightarrow E$$

$$F^+: F \rightarrow BCF \rightarrow BCEF$$

- *Nur aus C erhält man F und umgekehrt, daher muss eines der beiden Teil des Schlüssels sein.*
- *Ähnlich für A und D . Allerdings setzt das im Falle von A auch noch das Vorhandensein von C oder F voraus.*
- *B und E sind aus diesen Vorgaben immer ableitbar.*
- *Damit ergeben sich die folgende Schlüssel:*
 $AC, CD, AF, \text{ und } DF$.
- *Wg. der obigen Begründung kann es keine weiteren Schlüssel geben und Schlüssel sind minimal*

Aufgabe 5: Funktionale Abhängigkeiten & Normalisierung

Betrachten Sie das Relationenschema:

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

- c) Ist die funktionale Abhängigkeit $F \rightarrow CE$ aus der obigen Menge funktionaler Abhängigkeiten der Relation S ableitbar?

Kann mit Verwendung der RAP-Regeln und Menge der FDs gezeigt werden:

$$F_{\text{(Ref.)}} \rightarrow F \rightarrow_{\text{(mit FD 3, Akk.)}} FBC \rightarrow_{\text{(mit FD 2, Akk.)}} FBCE \rightarrow_{\text{(Projektion)}} CE \checkmark$$

Aufgabe 6: Normalisierung

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 .

Prüfe AD , da weder A noch D auf rechter Seite vorkommen:

$$AD \xrightarrow{FD_1} ABCD \xrightarrow{FD_2} ABCDF \xrightarrow{FD_4} ABCDEFG$$

Also ist AD ein Schlüssel

(minimal, da weder A und D ausreichen, und alle Attribute ableitbar mit AD)

Aufgabe 6: Normalisierung

Gegeben das folgende Relationenschema:

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

- b) Wenden Sie den Synthese-Algorithmus zur Überführung des Schemas in die dritte Normalform an.

rechts-minimal:

$$A \rightarrow B, A \rightarrow C, BD \rightarrow C, BD \rightarrow F, EG \rightarrow F, D \rightarrow G, D \rightarrow E$$

links-minimal:

Prüfe $BD \rightarrow C$: B^+ : $B \rightarrow B$ D^+ : $D \rightarrow D \rightarrow DEG \rightarrow DEFG$ ✘

Prüfe $BD \rightarrow F$: JA (s.o.) ✓ Ersetze durch $D \rightarrow F$

Prüfe $EG \rightarrow F$: E^+ : $E \rightarrow E$ G^+ : $G \rightarrow G$ ✘

Aufgabe 6 (Forts.)

$$F' = \{A \rightarrow B, A \rightarrow C, BD \rightarrow C, D \rightarrow F, EG \rightarrow F, D \rightarrow G, D \rightarrow E\}$$

Redundant?

$A \rightarrow B?$: A^+ : $A \rightarrow AC$ ✗

$A \rightarrow C?$: A^+ : $A \rightarrow AB$ ✗

$BD \rightarrow C?$: $(BD)^+$: $BD \rightarrow BDF \rightarrow BDEF \rightarrow BDEFG$ ✗

$D \rightarrow F?$: D^+ : $D \rightarrow DE \rightarrow DEG \rightarrow DEFG$ ✓ *Streiche $D \rightarrow F$*

$EG \rightarrow F?$: $(EG)^+$: $EG \rightarrow EG$ ✗

$D \rightarrow G?$: D^+ : $D \rightarrow DE$ ✗

$D \rightarrow E?$: D^+ : $D \rightarrow DG$ ✗

Aufgabe 6: Normalisierung

Synthese:

F' : $A \rightarrow B, A \rightarrow C, BD \rightarrow C, EG \rightarrow F, D \rightarrow G, D \rightarrow E$

$R_1 = (\{\underline{A}, B, C\}, \{A \rightarrow BC\})$

$R_2 = (\{\underline{B}, C, \underline{D}\}, \{BD \rightarrow C\})$

$R_3 = (\{\underline{E}, F, \underline{G}\}, \{EG \rightarrow F\})$

$R_4 = (\{\underline{D}, E, G\}, \{D \rightarrow EG\})$

Schlüssel AD in keiner Relation enthalten, daher

$R_5 = (\{\underline{A}, \underline{D}\}, \emptyset)$

Aufgabe 6: Normalisierung

c) Dekomposition:

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

2. Eine Dekomposition entlang $D \rightarrow GE$ mit dem Ergebnis

$$R_{21} = (\{D, E, G\}, \{D \rightarrow GE, \dots\}) \text{ und}$$

$$R_{22} = (\{A, D, F\}, \{\dots\})$$

R_{21} ist in BCNF, denn E und G hängen nicht voneinander ab, so dass es keine transitiven Abhängigkeiten geben kann.

R_{22} ist **nicht** in BCNF, denn wg. $D \rightarrow GE$ und $EG \rightarrow F$ gilt $D \rightarrow F \in F^+$ insbesondere $D \rightarrow F \in F_{22}^+$ (Alternativ: $A \rightarrow F \in F^+$)

Daher Dekomposition entlang dieser Abhängigkeit

$$R_{221} = (\{D, F\}, \{D \rightarrow F, \dots\}) \text{ und}$$

$$R_{222} = (\{A, D\}, \{\dots\})$$

Diese sind beide offensichtlich in BCNF.

Aufgabe 7: Serialisierbarkeit

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

Beim Dirty-Read-Problem wird durch eine TA t ein Wert gelesen, der von einer TA t' geschrieben wurde, die noch nicht abgeschlossen ist.

Wenn t' abbricht, dann hat t einen ungültigen Wert gelesen.

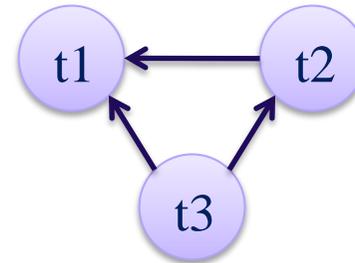
Beispiel: $r_1(x)w_1(x)r_2(x)a_1\dots$

Aufgabe 7: Lösung

$$s = r_1(x) r_2(x) w_3(y) w_2(z) r_1(z) c_3 w_1(x) r_2(y) c_2 r_1(y) w_1(z) c_1$$

$$b) \text{ conf}(s) = \{(r_2(x), w_1(x)), (w_3(y), r_2(y)), (w_3(y), r_1(y)), (w_2(z), r_1(z)), (w_2(z), w_1(z))\}$$

$s \in \text{CSR}$, da Konfliktgraph
azyklisch



c)

- t_2 liest y von t_3
- t_1 liest y von t_3
- t_1 liest z von t_2 .
- $c_3 <_s c_2$, $c_3 <_s c_1$ und $c_2 <_s c_1 \rightarrow s \in \text{RC}$
- $c_3 <_s r_2(y)$ und $c_3 <_s r_1(y)$, aber $c_2 \not<_s r_1(z) \rightarrow s \notin \text{ACA}$
- $s \notin \text{ST}$, da $s \notin \text{ACA}$ und $\text{ST} \subset \text{ACA}$

Aufgabe 7: Lösung

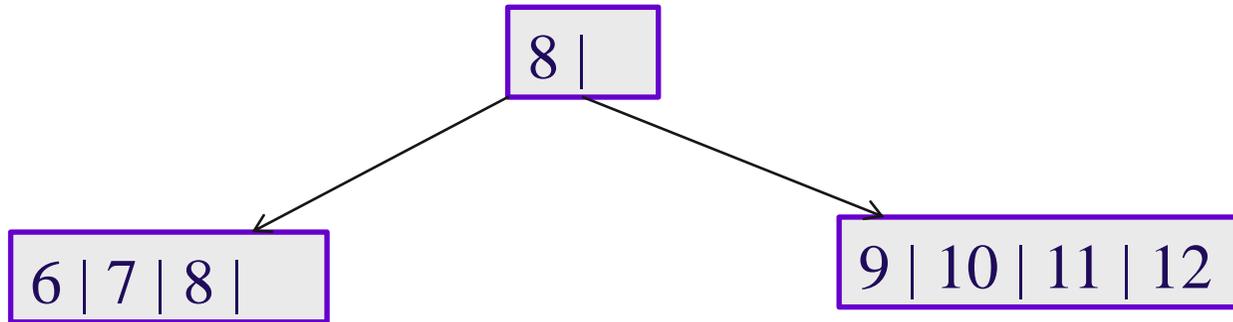
$$s = r_1(x) r_2(x) w_3(y) w_2(z) r_1(z) c_3 w_1(x) r_2(y) c_2 r_1(y) w_1(z) c_1$$

d) S2PL: behält alle Schreibsperrern bis Commit

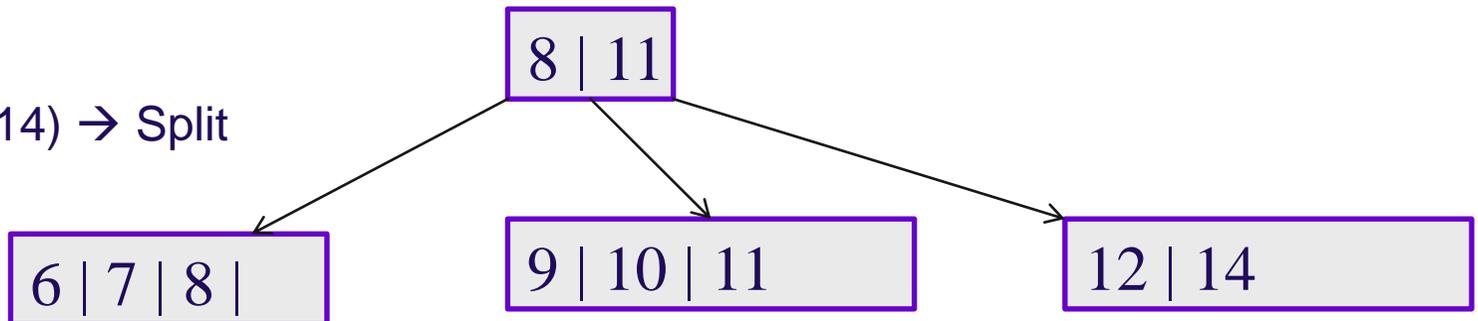
$$S_{S2PL} = \{rl_1(x) r_1(x) rl_2(x) r_2(x) \\ wl_3(y) w_3(y) wl_2(z) \\ w_2(z) c_3 wu_3(y) rl_2(y) \\ r_2(y) ru_2(y) ru_2(x) c_2 \\ wu_2(z) rl_1(z) r_1(z) wl_1(x) \\ w_1(x) rl_1(y) r_1(y) \\ wl_1(z) w_1(z) ru_1(x) ru_1(y) \\ ru_1(z) c_1 wu_1(x) wu_1(z)\}$$

Aufgabe 8 a)

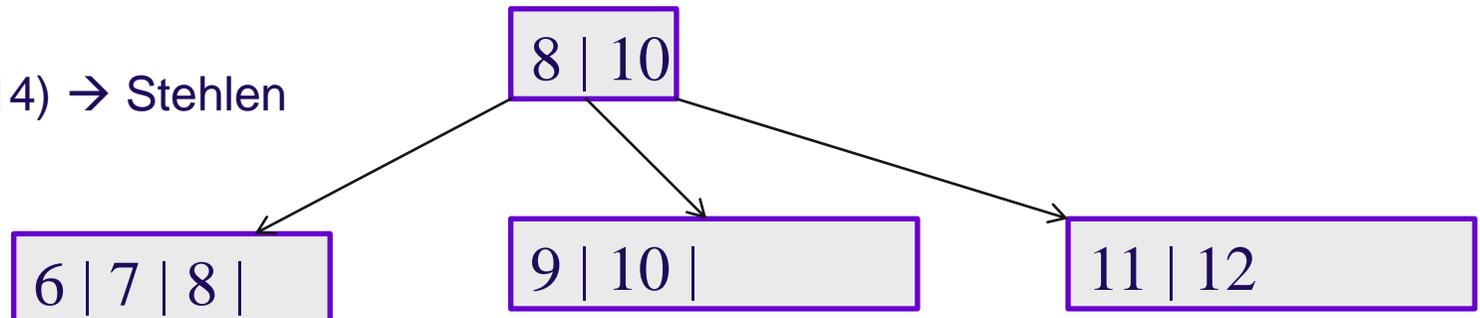
$k=1, k^*=2$



Insert(14) → Split

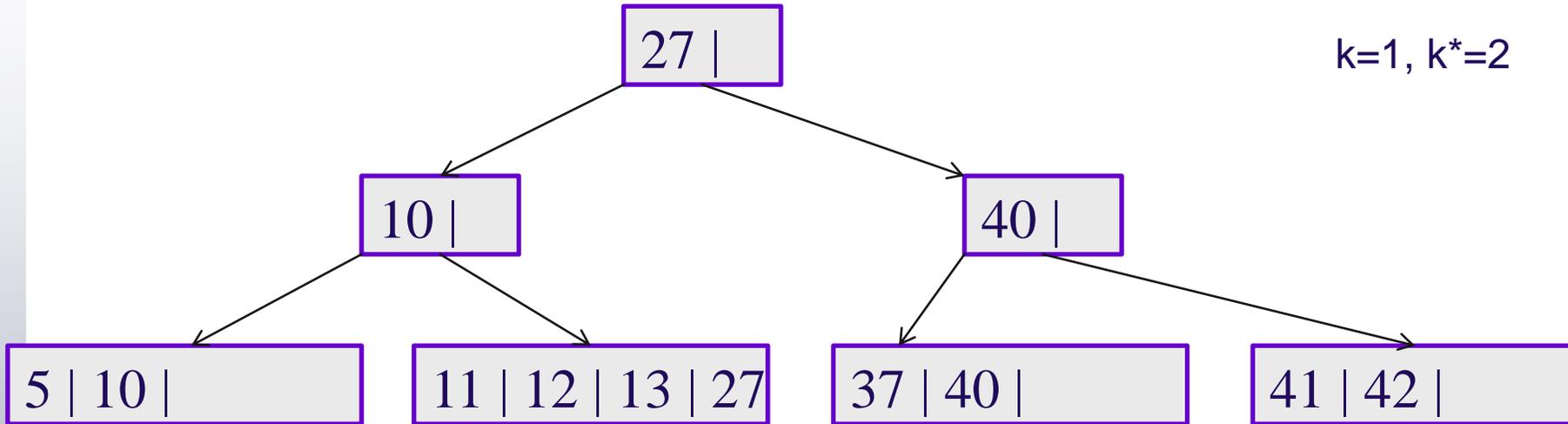


Delete(14) → Stehlen

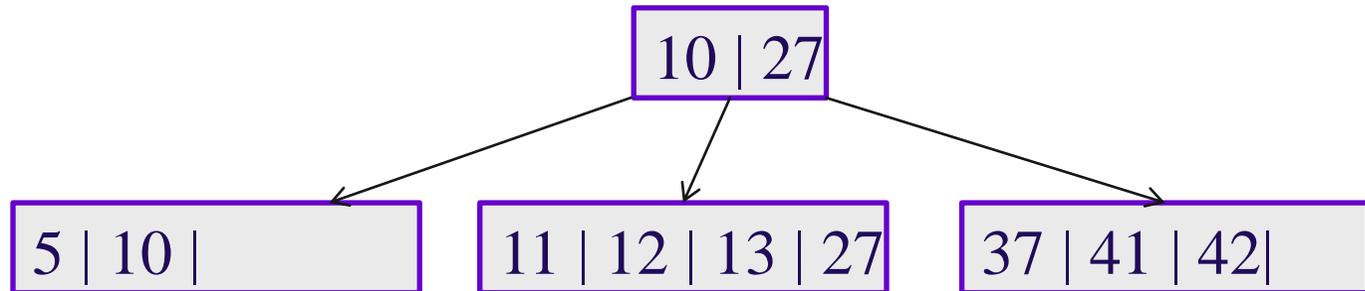


Aufgabe 8 b)

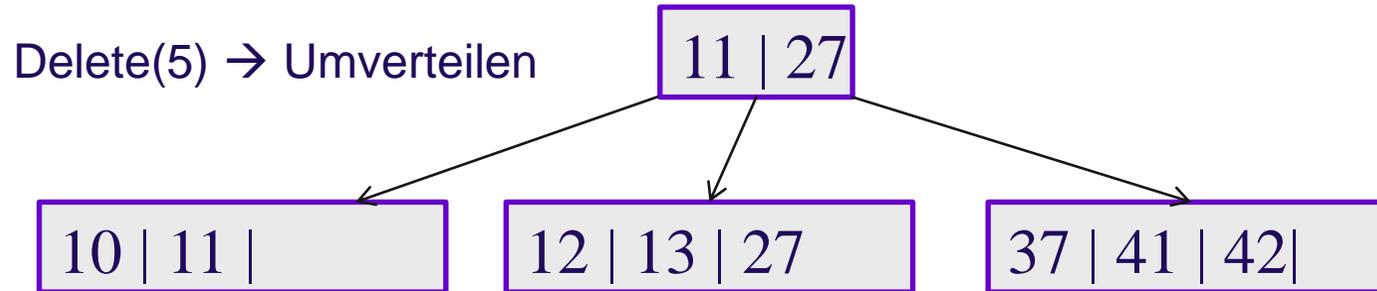
$k=1, k^*=2$



Delete(40) → Merge der beiden rechten Blätter,
daher Löschen der 40 im inneren B-Baum notwendig,
deshalb Verschmelzen von 10 und 27



Aufgabe 8 b) (Forts.)



12 könnte auch in linkes Blatt, falls man anders aufteilt

Aufgabe 9: Lösung a), b)

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

a) Die Hotels der Hotelkette „Inselhotels“.

```
//hotelkette[@name=„Inselhotels“]/hotel
```

b) Die Gäste mit Kundenstatus „gold“ der Hotels mit „Massageliegen“.

```
//hotel[ausstattung/@typ=„Massageliegen“]  
/gaeste/gast[kundenstatus=„gold“]
```

Aufgabe 9: Lösung c)

Drücken Sie das folgende Informationsbedürfnis in **XQuery** aus.
c) Alle Paare aus Hotelbewohnern, die übereinander wohnen.

```
<obermieter>
  {for $g in doc(„hotel.xml“)//gast
   for $g2 in $g/../../gast
   where $g2/zimmer=$g/zimmer+100
   return
     <wohntueber>
       <oben>{$g2/@name/string()}</oben>
       <unten>{$g/@name/string()}</unten>
     </wohntueber>
  }
</obermieter>
```

Viel Erfolg!!!

