

# DBIS

Datenbanken  
und  
Informationssysteme

# Datenbanken und Informationssysteme

## Probeklausur



Sommersemester 2011



Prof. Dr. M. Jarke  
Lehrstuhl Informatik 5  
(Informationssysteme)  
RWTH Aachen

# Aufgabe 1: ER-Modell

- ◆ Die Pizzeria hat verschiedene Kundentypen. Sie unterscheidet ausschließlich zwischen Stammkunden, Onlinekunden sowie Telefonkunden. Kunden können natürlich in mehrere dieser Kategorien fallen (*z.B. ein Stammkunde, der mal telefonisch und mal online bestellt hat*). Kunden werden erst gespeichert, wenn sie mindestens einmal etwas bestellt haben.
- ◆ Jeder Kunde ist eindeutig über seine Kundennummer identifizierbar und hat einen Namen, sowie eine Adresse bestehend aus der Straße, PLZ und Ort. Bei Stammkunden wird zudem der Geburtstag gespeichert. Für jeden Onlinekunden wird eine E-Mail-Adresse gespeichert. Bei Telefonkunden wird die Telefonnummer (für eventuelle Rückfragen) gespeichert.

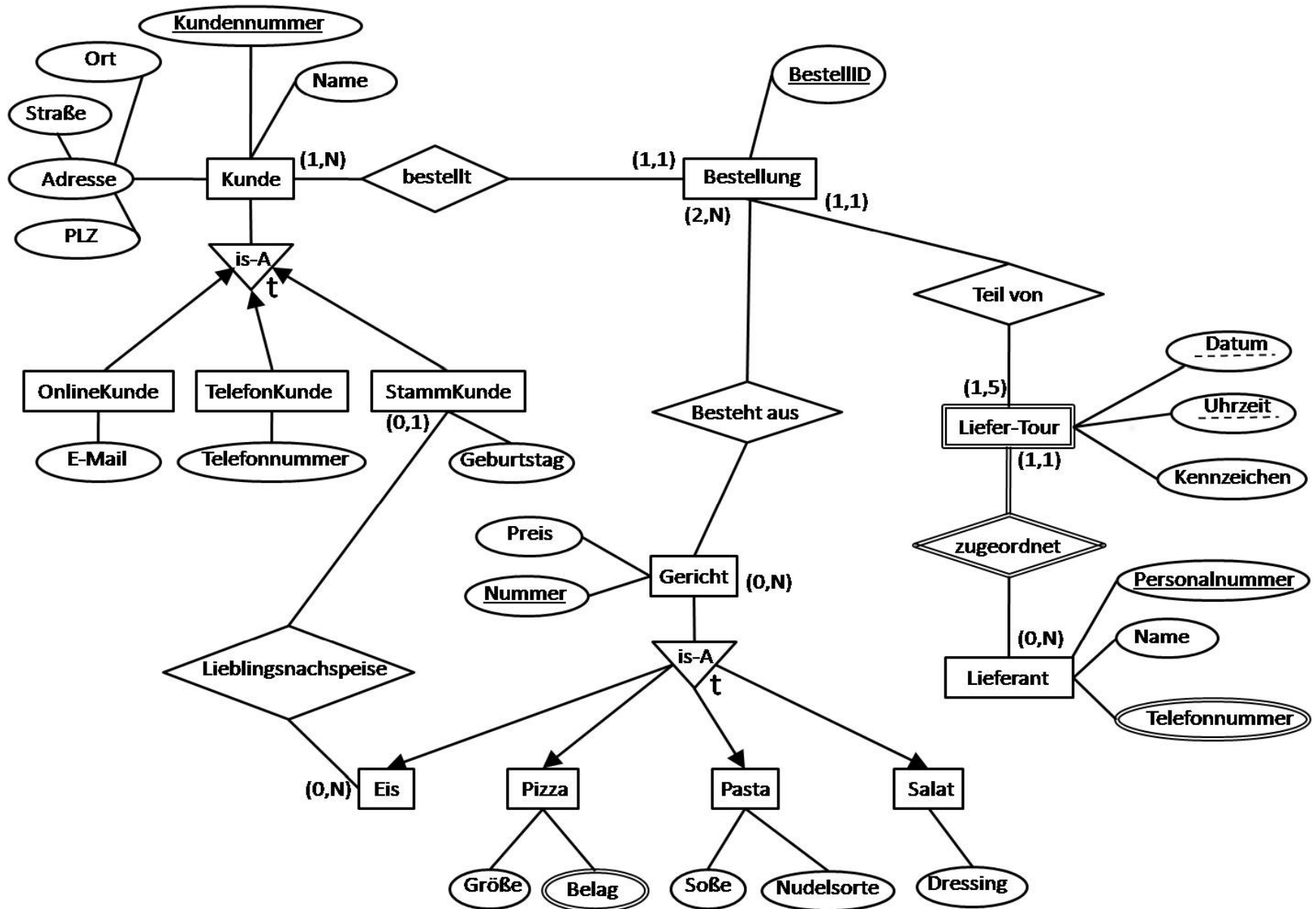
# Aufgabe 1: ER-Modell (2)

- ◆ Kunden geben Bestellungen auf, wobei jede Bestellung von genau einem Kunden stammt. Zu jeder Bestellung wird eine eindeutige BestellID gespeichert.
- ◆ Eine Bestellung besteht aus mindestens zwei Gerichten.
- ◆ Die Pizzeria unterscheidet zwischen genau vier Arten von Gerichten: Pizza, Pasta, Salat und Eis. Alle Gerichte haben einen Preis und eine eindeutige Nummer. Zu der Pizza wird die Größe gespeichert und die Beläge (in der Regel mehrere!). Zu der Pasta werden die Nudelsorte, sowie die Soße und bei Salaten das Dressing abgespeichert.
- ◆ Jedem Stammkunden kann genau ein Eis als Lieblingsnachspeise zugeordnet werden.

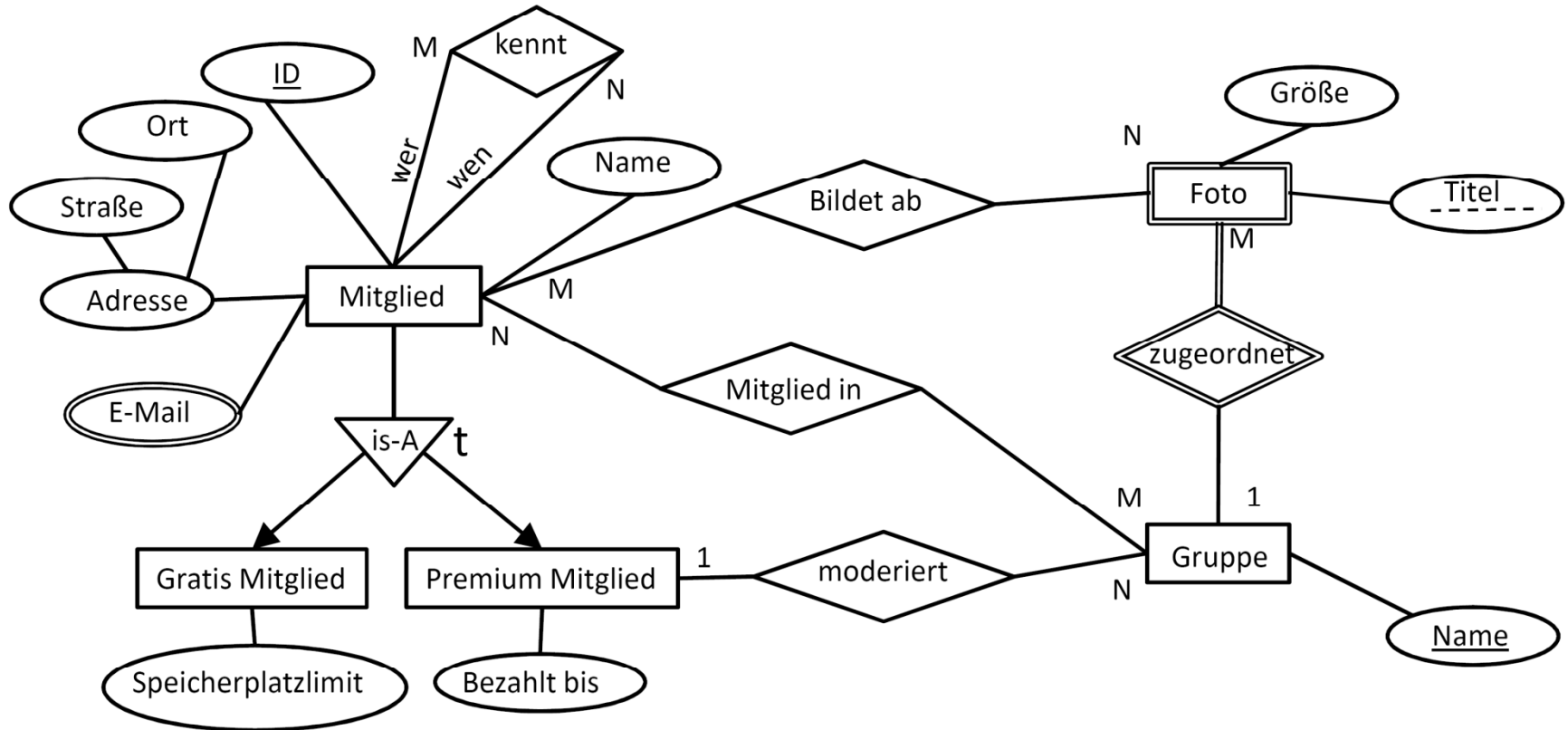
# Aufgabe 1: ER-Modell (3)

- ◆ Eine Liefer-Tour besteht aus maximal 5 Bestellungen. Zu jeder Liefer-Tour wird das Datum und die Uhrzeit gespeichert, zudem wird zu jeder Liefer-Tour das Kennzeichen des zugehörigen Fahrzeugs gespeichert.
- ◆ Jeder Liefer-Tour ist genau ein Lieferant zugeordnet. Die Liefer-Tour ist eindeutig anhand des Lieferanten und dem Datum und der Uhrzeit zu identifizieren.
- ◆ Lieferanten selbst werden durch ihre Personalnummer identifiziert, zudem werden der Name und (ggf. mehrere) Telefonnummern gespeichert. Auch neue Lieferanten, die noch nichts ausgeliefert haben, werden direkt in die Datenbank aufgenommen.

## Aufgabe 1: Lösung



## Aufgabe 2: Relationale DBen



# Aufgabe 2: Lösung Transformation

- ◆  $D = (R, \Sigma_R)$
- ◆  $R = \{\text{Mitglied, GratisMitglied, PremiumMitglied, Kennt, Email, MitgliedIn, Gruppe, Foto, BildetAb}\}$
- ◆  $\text{Mitglied} = (\{\text{Ort, Straße, } \underline{\text{ID}}, \text{Name}\})$
- ◆  $\text{GratisMitglied} = (\{\underline{\text{ID}}, \text{Speicherplatzlimit}\})$
- ◆  $\text{PremiumMitglied} = (\{\underline{\text{ID}}, \text{BezahltBis}\})$
- ◆  $\text{Kennt} = (\{\underline{\text{IDwer}}, \underline{\text{IDwen}}\})$
- ◆  $\text{Email} = (\{\underline{\text{Email}}, \text{ID}\})$
- ◆  $\text{MitgliedIn} = (\{\underline{\text{ID}}, \underline{\text{GruppenName}}\})$
- ◆  $\text{Gruppe} = (\{\underline{\text{Name}}, \text{ModeratorID}\})$
- ◆  $\text{Foto} = (\{\text{Größe}, \underline{\text{Titel}}, \underline{\text{GruppenName}}\})$
- ◆  $\text{BildetAb} = (\{\underline{\text{ID}}, \underline{\text{FotoTitel}}, \underline{\text{GruppenName}}\})$

# Aufgabe 2: Lösung

## Interrelationale Abhängigkeiten

- ◆  $\text{Kennt}[\text{IDwer}] \subseteq \text{Mitglied}[\text{ID}]$
- ◆  $\text{Kennt}[\text{IDwen}] \subseteq \text{Mitglied}[\text{ID}]$
- ◆  $\text{Email}[\text{ID}] \subseteq \text{Mitglied}[\text{ID}]$
- ◆  $\text{MitgliedIn}[\text{ID}] \subseteq \text{Mitglied}[\text{ID}]$
- ◆  $\text{BildetAb}[\text{ID}] \subseteq \text{Mitglied}[\text{ID}]$
- ◆  $\text{BildetAb}[\text{FotoTitel}, \text{GruppenName}] \subseteq \text{Foto}[\text{Titel}, \text{GruppenName}]$
- ◆  $\text{MitgliedIn}[\text{GruppenName}] \subseteq \text{Gruppe}[\text{Name}]$
- ◆  $\text{Gruppe}[\text{ModeratorID}] \subseteq \text{PremiumMitglied}[\text{ID}]$
- ◆  $\text{Foto}[\text{GruppenName}] \subseteq \text{Gruppe}[\text{Name}]$
- ◆  $\text{PremiumMitglied}[\text{ID}] \cap \text{GratisMitglied}[\text{ID}] = \emptyset$
- ◆  $\text{PremiumMitglied}[\text{ID}] \cup \text{GratisMitglied}[\text{ID}] = \text{Mitglied}[\text{ID}]$



## Aufgabe 2: SQL-Anfrage b)

- ◆ *Die Namen aller "Premium Mitglieder", die auf keinem Foto gemeinsam mit dem Mitglied "Hans Werner" abgebildet sind.*

# Aufgabe 2: Lösung SQL-Anfrage b)

```
◆ SELECT m.name
FROM Mitglied m, PremiumMitglied pm
WHERE
    pm.ID = m.ID AND
    m.ID NOT IN
    ( SELECT b.ID
      FROM BildetAb b, BildetAb b2, Mitglied m2
      WHERE
          m2.Name = "Hans Werner" AND
          b2.ID = m2.ID AND
          b.FotoTitel = b2.FotoTitel AND
          b.GruppenName = b2.GruppenName
    )
```

# Aufgabe 2: RA-Anfrage c)

- ◆ Alle Namen von Mitgliedern, die den Moderator mindestens einer der Gruppen kennen, in der sie selbst Mitglied sind.

$$\pi_{\text{Name}} ($$

$$\text{Mitglied} \bowtie$$

$$(\rho_{\text{ID}} \leftarrow \text{IDwer}, \text{ModeratorID} \leftarrow \text{IDwen}(\text{Kennt})) \bowtie$$

$$(\rho_{\text{GruppenName}} \leftarrow \text{Name}(\text{Gruppe})) \bowtie$$

$$\text{MitgliedIn})$$

## Aufgabe 2: SQL-Anfrage d)

- ◆ Zu jeder Gruppe soll die Anzahl der Mitglieder und der Name des Moderators ausgegeben werden. Ausgeschlossen werden sollen die Gruppen, deren Moderator höchstens bis zum 1.9.2010 gezahlt hat. Es sollen nur Gruppen beachtet werden, die mindestens 50 Mitglieder und ein Gratis-Mitglied haben.

# Aufgabe 2: Lösung SQL-Anfrage d)

```
SELECT count(*) as AnzMitglieder, m.name
FROM Mitglied m, PremiumMitglied pm, Gruppe g,
    MitgliedIn mi
WHERE pm.ID = m.ID AND mi.GruppenName = g.Name
    AND g.ModeratorID = pm.ID
    AND pm.BezahltBis > "1.9.2010"
    AND g.name IN (
        SELECT mi2.GruppenName
        FROM GratisMitglied gm, MitgliedIn mi2
        WHERE gm.ID = mi2.ID)
GROUP BY g.name, m.name
HAVING AnzMitglieder >= 50
```

## Aufgabe 2: RDK-Anfrage e)

- ◆ Alle Namen von "Gratis-Mitgliedern", die in den Gruppen, in denen sie Mitglieder sind, keine anderen Gruppenmitglieder kennen. Dies schließt auch die "Gratis-Mitglieder" ein, die in keiner Gruppe Mitglied sind.

# Aufgabe 2: Lösung

## RDK-Anfrage e)

$$\{ (\text{name}) \mid (\exists \text{ort})(\exists \text{strasse})(\exists \text{id})(\exists \text{speicher})$$

$$(\text{Mitglied}(\text{ort}, \text{strasse}, \text{id}, \text{name}) \wedge$$

$$\text{GratisMitglied}(\text{id}, \text{speicher}) \wedge$$

$$\neg(\exists \text{gname})(\exists \text{id2})$$

$$\text{MitgliedIn}(\text{id}, \text{gname}) \wedge$$

$$\text{MitgliedIn}(\text{id2}, \text{gname}) \wedge$$

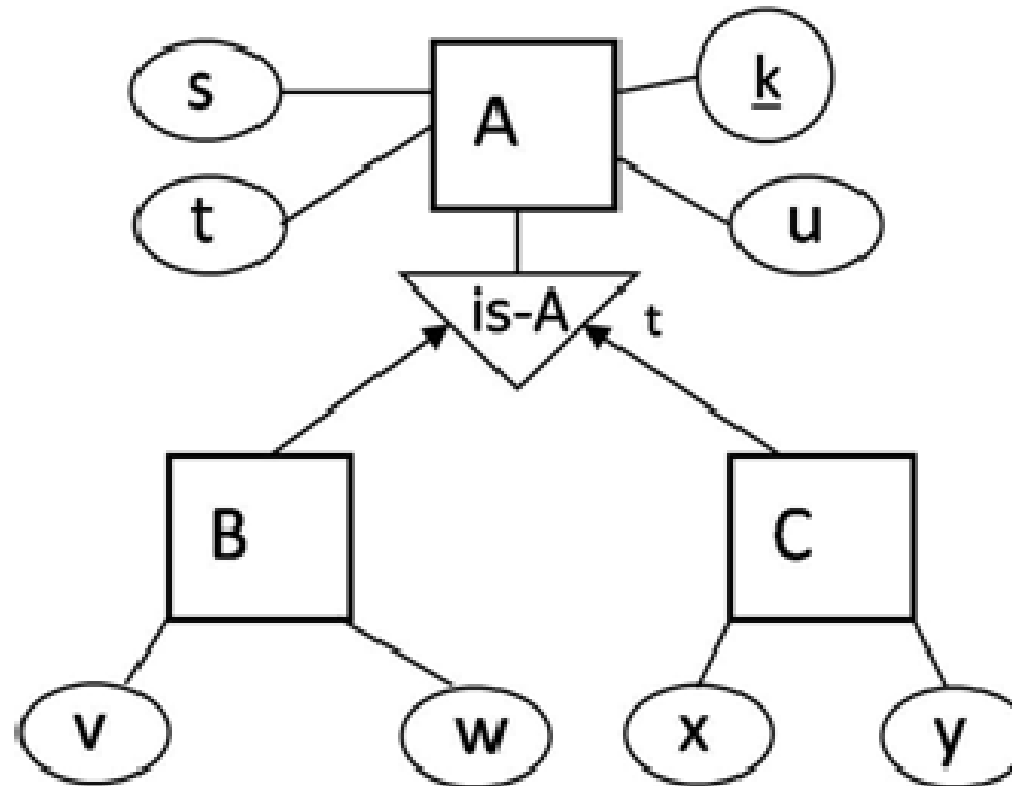
$$\text{Kennt}(\text{id}, \text{id2})$$

$$)$$

$$)$$

$$\}$$

# Aufgabe 3: Transformation IsA



- ◆ Welche Transformationen sind möglich?
- ◆ Vor- und Nachteile?



# Aufgabe 3: Lösung

- ◆ **Eine Relation mit einem Typ-Attribut geht nicht, da IsA überlappend ist**
- ◆ **Es bleibt:**
  1. Eine Relation für alle drei Typen mit booleschen Attributen.
  2. Je eine Relation für B und C (Tabellen nur für Subtypen).
  3. Je eine Relation für A, B und C

# Aufgabe 3: Lösung

## Vor- und Nachteile

### ◆ Nachteile

1. Viele Nullwerte (bei Objekten die nur B bzw. nur C sind)
2. S,t,u müssen in B und C modelliert werden, für ein Objekt, das sowohl B als auch C ist, werden diese Attribute zweimal gespeichert.
3. Bei vielen Anfragen ist ein Join nötig.

### ◆ Vorteile

1. Keine Joins nötig bei Anfragen.
2. Keine Nullwerte und keine Joins wenn man nur B bzw. C Objekte abfragt. Erst kombinierte B-C Typen brauchen einen Join.
3. Keine Nullwerte, keine Redundanz.

# Aufgabe 4: Entscheidungsfragen

- a) Für die Definition von SQL Schemata verwendet man SQL-DDL.
- b) In einem ER-Diagramm zeigen bei einer disjunkten isA-Beziehung die Pfeile auf die erbenden Entitätstypen.
- c) Wenn  $X \rightarrow A$  und  $Y \rightarrow B$  funktionale Abhängigkeiten für eine Relation  $r \in \text{Rel}(Z)$  sind (mit  $X, Y \subseteq Z$ ,  $A, B \in Z$ ), so gilt auch  $XY \rightarrow AB$ .

# Aufgabe 4:

## Entscheidungsfragen (2)

- d) Zwei Operationen  $p \in op(t)$  und  $q \in op(t')$  zweier Transaktionen  $t$  und  $t'$  ( $t \neq t'$ ) stehen genau dann in einem Schedule  $s$  in Konflikt, wenn sie auf demselben Objekt  $x$  arbeiten und sowohl  $p$  als auch  $q$  Schreiboperationen sind.
- e) In einem B+-Baum mit  $k = 2$  und  $k^* = 2$  haben innere Knoten mindestens 3, aber höchstens 5 Söhne.
- f) Für einen nicht leeren B-Baum der Ordnung  $k$  gilt: Ein innerer Knoten mit  $b$  Schlüsseln hat genau  $b$  Söhne.
- g) Sei  $A$  eine Attributmenge und  $a \in Rel(A)$ , dann gilt:  
 $a \bowtie b = \Pi_A(a \bowtie b)$ .

## Aufgabe 4: Lösung

- a) Richtig.
- b) Richtig.
- c) Richtig.
- d) Falsch.
- e) Richtig.
- f) Falsch.
- g) Falsch.

# Aufgabe 5:

## FAs und Normalformen

- ◆  $A \rightarrow BE, AE \rightarrow BD, G \rightarrow CD, CD \rightarrow BEG, CG \rightarrow B$
- ◆ Schlüsselkandidaten?

- ◆ Welche Normalform hat R?

# Aufgabe 5: FDs und Normalformen

## ◆ Synthese-Algorithmus

# Aufgabe 5: Lösung

## ◆ Schlüsselkandidaten

- a) A ist in keiner FD rechts, daher ist A Teil jedes Schlüssels.
- b) G kann nur bestimmt werden, wenn C bekannt ist und umgekehrt. Daher enthält jeder Schlüssel entweder C oder G.
- c) AG ist Superschlüssel, denn  
 $AG \rightarrow ABEG \rightarrow ABDEG \rightarrow ABCDEG$   
AC ist Superschlüssel, denn  
 $AC \rightarrow ABCE \rightarrow ABCED \rightarrow ABCEDG$
- d) AG ist minimal  
 $A \rightarrow ABE \rightarrow ABDE$  keine weitere FD anwendbar  
 $G \rightarrow CDG \rightarrow BCDEG$  keine weitere FD anwendbar
- e) AC ist minimal  
 $C \rightarrow C$  keine weitere FD anwendbar  
K1 = AG, K2 = AC sind Schlüsselkandidaten
- f) Aus a) - e) folgt, dass es keine anderen Schlüsselkandidaten geben kann.

- ## ◆ R ist in 1NF, 2NF z.B. wegen $A \rightarrow BE$ verletzt.



# Aufgabe 5: Lösung Synthese

- ◆ Mache F rechtsminimal:

$$G_1 = \{A \rightarrow B, A \rightarrow E, AE \rightarrow B, AE \rightarrow D, G \rightarrow C, G \rightarrow D, \\ CD \rightarrow B, CD \rightarrow E, CD \rightarrow G, CG \rightarrow B\}$$

- ◆ Mache F linksminimal:

$$G_2 = \{A \rightarrow B, A \rightarrow E, \mathbf{A \rightarrow B}, \mathbf{A \rightarrow D}, G \rightarrow C, G \rightarrow D, \\ CD \rightarrow B, CD \rightarrow E, CD \rightarrow G, \mathbf{G \rightarrow B}\}$$

- ◆ Entferne redundante FDs:

$$G_3 = \{A \rightarrow B, A \rightarrow E, A \rightarrow D, G \rightarrow C, G \rightarrow D, CD \rightarrow E, \\ CD \rightarrow G, G \rightarrow B\}$$

*Alternativ hätte statt **CD** → **B** auch **G** → **B** wegfallen können*

# Aufgabe 5: Lösung Synthese

- ◆ Zerlegung mittels Synthese

$$R_1 = (\{A, B, D, E\}, \{A \rightarrow B, A \rightarrow D, A \rightarrow E\})$$

$$R_2 = (\{B, C, D, G\}, \{G \rightarrow C, G \rightarrow D, G \rightarrow B, CD \rightarrow G\})$$

$$R_3 = (\{C, D, E, G\}, \{G \rightarrow C, G \rightarrow D, CD \rightarrow E, CD \rightarrow G\})$$

Keine der Attributmengen ist Schlüssel für U, daher zusätzliche Schlüsselrelation:

$$R_4 = (\{A, C\}, \{\})$$

# Aufgabe 6: Transaktionen

- ◆  $s_1 = w_1(x) r_1(y) r_2(y) w_2(y) c_2 r_3(y) w_3(x) w_1(y) c_3 r_1(x) c_1$
- ◆  $s_2 = r_1(y) w_1(y) r_2(y) w_1(x) r_1(x) c_1 w_2(y) w_3(x) c_2 r_3(y) c_3$
- ◆ Sind  $s_1$  und  $s_2$  konfliktäquivalent?
- ◆ Sind  $s_1$  und  $s_2$  konfliktserialisierbar?
- ◆ Sind  $s_1$  und  $s_2$  in RC, ACA und ST?

# Aufgabe 6:

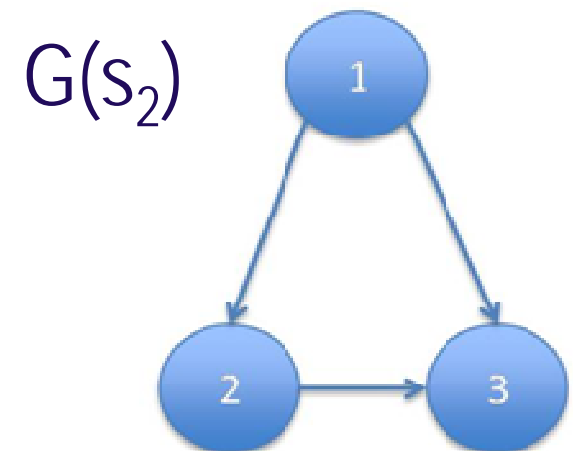
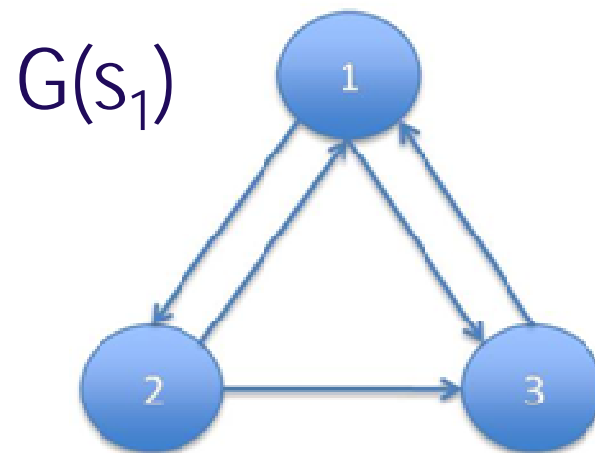
## a) Konfliktäquivalent

- ◆  $op(s_1) = op(s_2)$
- ◆  $conf(s_1) = \{$   
 $(w_1(x), w_3(x)),$   
 $(r_1(y), w_2(y)),$   
 $(r_2(y), w_1(y)),$   
 $(w_2(y), r_3(y)),$   
 $(w_2(y), w_1(y)),$   
 $(r_3(y), w_1(y)),$   
 $(w_3(x), r_1(x))\}$
- ◆  $conf(s_2) = \{$   
 $(r_1(y), w_2(y)),$   
 $(w_1(y), r_2(y)),$   
 $(w_1(y), w_2(y)),$   
 $(w_1(y), r_3(y)),$   
 $(w_1(x), w_3(x)),$   
 $(r_1(x), w_3(x)),$   
 $(w_2(y), r_3(y)) \}$
- ◆  $conf(s_1) \neq conf(s_2) \rightarrow s_1$  und  $s_2$  sind nicht konfliktäquivalent!

## Aufgabe 6:

## b) Konfliktserialisierbar

- ◆  $s_1$  ist nicht in CSR, da z.B. Zykel zwischen  $t_1$  und  $t_2$ :  
 $(r_1(y), w_2(y)), (r_2(y), w_1(y))$  sind in  $\text{conf}(s_1)$
- ◆  $s_2$  ist in CSR, da Konfliktgraph keinen Zykel enthält.

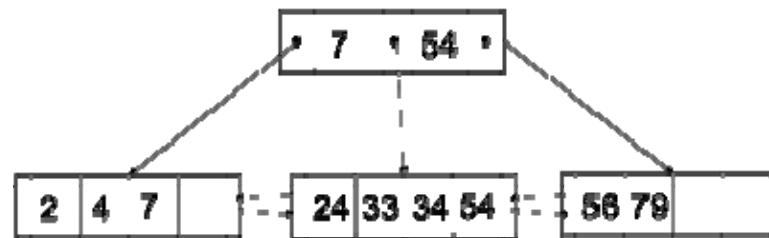


# Aufgabe 6:

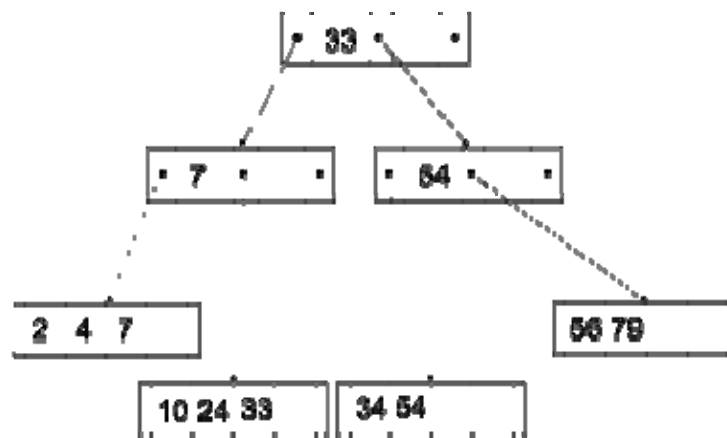
## c) RC, ACA, ST

- ◆ RC: Keine Transaktion wird freigegeben, bis alle anderen Transaktionen, von denen sie gelesen hat, auch freigegeben wurden.
  - $s_1, s_2$  sind in RC.
- ◆ ACA: Eine Transaktion darf nur Werte von bereits erfolgreich abgeschlossenen Transaktionen lesen.
  - $s_1$  ist in ACA,  $s_2$  ist nicht in ACA.
- ◆ ST: Ein Schedule ist strikt, falls ein Objekt  $x$  weder gelesen noch überschrieben wird, bis die Transaktion, die  $x$  zum letzten Mal geschrieben hat, abgeschlossen ist.
  - Weder  $s_1$  noch  $s_2$  sind in ST.

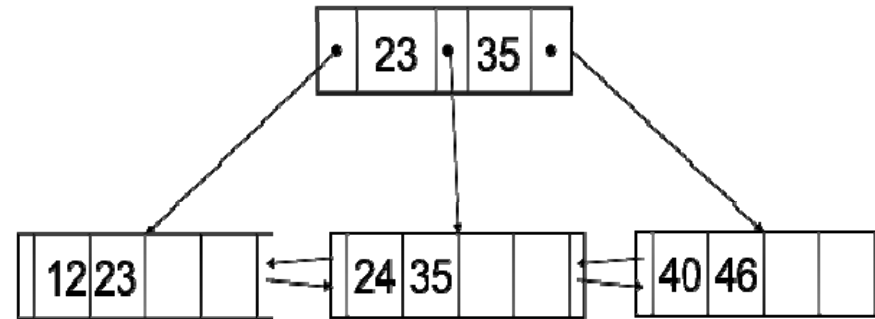
## Aufgabe 7: B<sup>+</sup>-Bäume



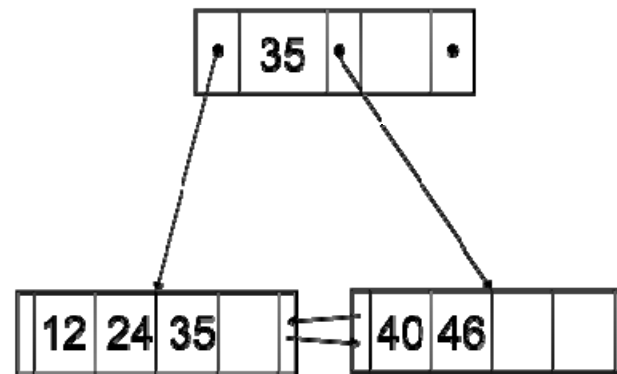
- ◆ 10 einfügen → Überlauf in Blatt → Überlauf in Wurzel



## Aufgabe 7: B<sup>+</sup>-Bäume



- ◆ Löschen der 23 → Verschmelzen





# Aufgabe 8: XML

```
<Auftragsdatenbank>  
  <Kunde>  
    <Name>Gabis Imbiss</Name>  
    <ID KundenID="4327" />  
    <Adresse>  
      <Strasse>Hamsterweg 7</Strasse>  
      <PLZ>91282</PLZ>  
      <Ort>Hunger</Ort>  
    </Adresse>  
    <Zahlungsmoral Bewertung="gut" />  
  </Kunde>
```

...

# Aufgabe 8: XML

```
<Auftrag Art="Printwerbung">  
  <Kosten PreisInEuro="1000" Anzahlung="500"/>  
  <Kunde>9458</Kunde>  
  <Mitarbeiter>Marietta Weyand</Mitarbeiter>  
  <Mitarbeiter>Lisa Paulus</Mitarbeiter>  
  <Optionen Farbe="SchwarzWeiss"  
    Schmuckfarbe="keine"/>  
</Auftrag>  
...  
</Auftragsdatenbank>
```

# Aufgabe 8:

## a) Xpath-Anfragen

1. Alle Printwerbungen, deren Kosten unter 1000 Euro liegen.
2. Alle IDs von Kunden, die vom Mitarbeiter "Anton Wetzel" betreut werden.
3. Alle Aufträge für Printwerbungen, bei denen weder Farbe (somit Farboption "SchwarzWeiss") noch Schmuckfarben verwendet werden.

# Aufgabe 8: Lösung

## a) Xpath-Anfragen

1. `//Auftrag[@Art="Printwerbung" and Kosten/@PreisInEuro < 1000]`
2. `Auftragsdatenbank/Auftrag[Mitarbeiter/text()='Anton Wetzel']/Kunde/text()`
3. `Auftragsdatenbank/Auftrag[@Art='Printwerbung']/Option[@Farbe="SchwarzWeiss" and Schmuckfarbe="keine"]/..`

# Aufgabe 8:

## b) XQuery

- ◆ Formulieren Sie eine XQuery-Anfrage, die alle Projektkosten und Anzahlungen von Projekten aller Kunden auflistet, die eine gute Zahlungsmoral haben. Das Ergebnis soll in folgender Form angegeben werden:

```
<Projektliste>
  <Kunde>
    <Firma Name="Gabis Imbiss" />
    <Projekt>
      <Radio-Werbung>
        <Kosten PreisInEuro="3000"
          Anzahlung="500" />
      </Radio-Werbung>
    </Projekt>
    ...<!-- evtl. weitere Projekte der Firma-->
  </Kunde>
</Projektliste>
```

# Aufgabe 8: b) XQuery

```
<Projektliste>
  { for $z in
    doc(Auftragsdb.xml)//Kunde[Zahlungsmoral[@Bewertung="gut"]]
    return
      <Kunde>
        <Firma Name="{ $z/Name/text() }"/>
          {
            for $p in
              doc(Auftragsdb.xml)//Auftrag[Kunde/text()=$z/ID/@KundenID
                and @Art="Printwerbung"]
            return
              <Projekt>
                <Printwerbung>
                  { $p/Kosten }
                </Printwerbung>
              </Projekt>
          }
  }
```

# Aufgabe 8: b) XQuery

```
{
  for $r in
    doc(Auftragsdb.xml)//Auftrag[Kunde/text()=$z/ID/@KundenID
      and @Art="Radio-Werbung"]
  return
    <Projekt>
      <Radio-Werbung>
        {$r/Kosten}
      </Radio-Werbung>
    </Projekt>
}
</Kunde>
}
```

</Projektliste>