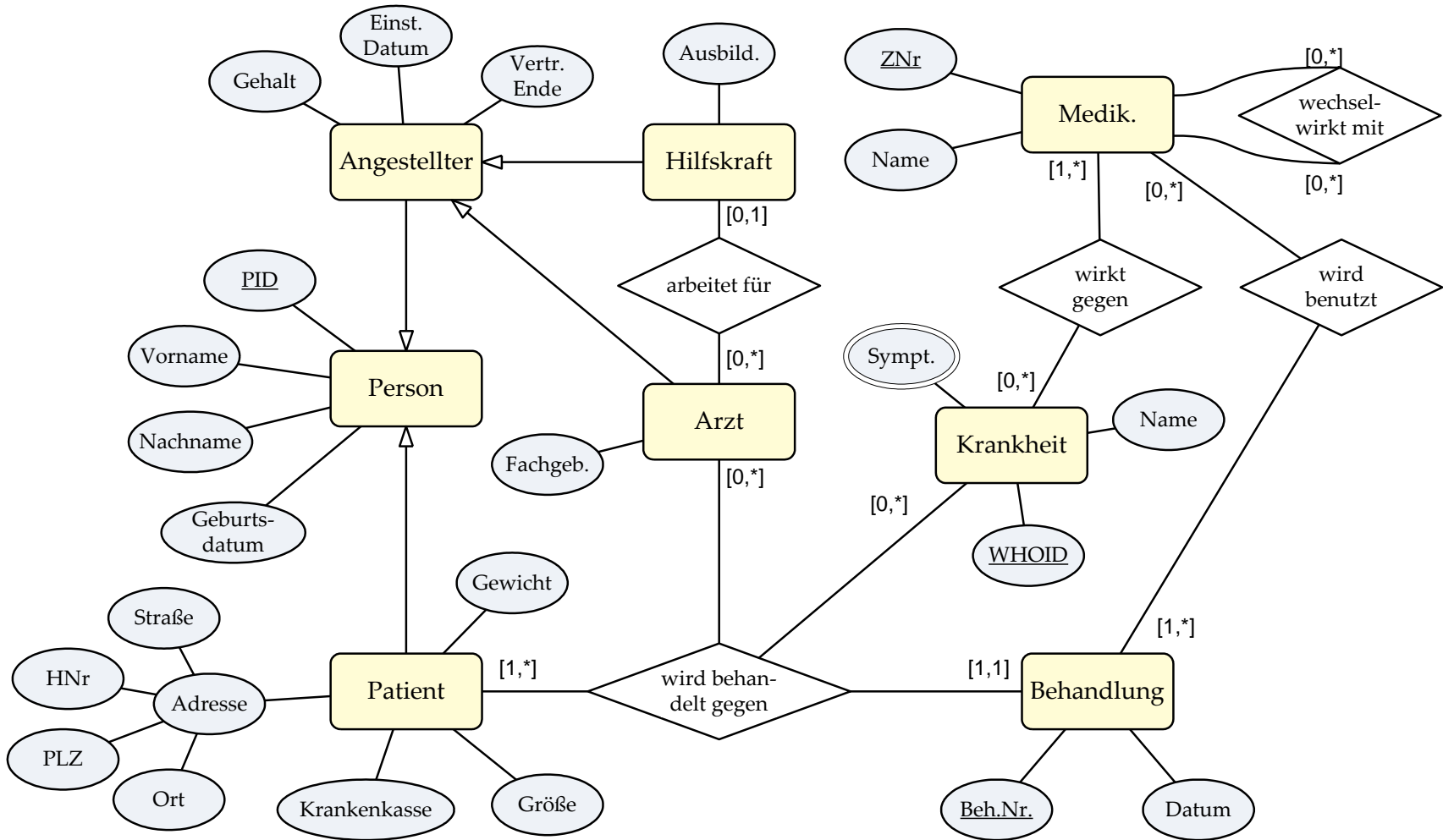


Aufgabe 1: Informationsmodellierung (1)

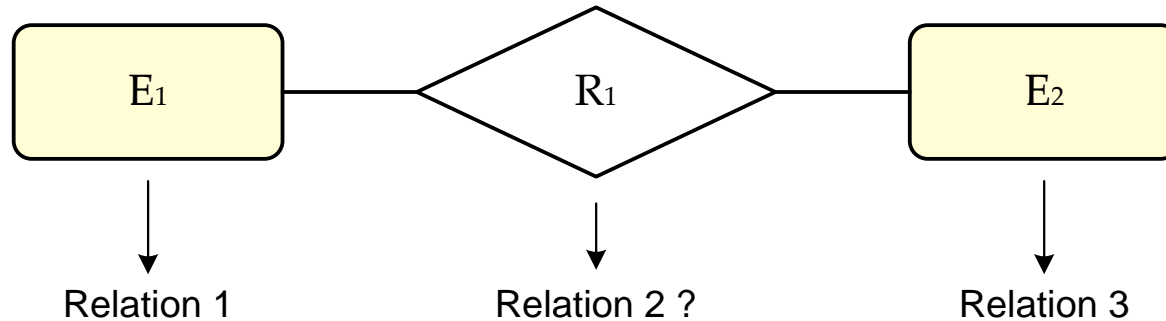
▪ Krankenhaus-Anwendung

- Objekte des Anwendungsbereichs
 - Behandlung: Beh.Nr., Datum
 - Krankheit: WHOID, Name, Symptome
 - Medikament: ZNr., Name
 - Personen: PID, Vorname, Nachname, Geb.Datum
 - Angestellter: Gehalt, Einstell.Datum, Vertragsende
 - Arzt: Fachgebiet
 - Hilfskraft: Ausbildung
 - Patient: Gewicht, Größe, Krankenkasse, Adresse (Straße, HNr , PLZ, Ort)
- Bedingungen und Beziehungen
 - Arzt und Hilfskraft sind Angestellte, Angestellte und Patienten sind Personen
 - Hilfskräfte können maximal einem Arzt zugeordnet sein, einem Arzt können wiederum mehrere Hilfskräfte zugeordnet werden
 - Ein Medikament kann gegen eine oder mehrere Krankheiten einsetzbar sein und für eine Krankheit gibt es beliebig viele Medikamente.
 - Ein Medikament kann Wechselwirkungen mit anderen Medikamenten haben.
 - Ein Medikament kann in beliebig vielen Behandlungen benutzt werden, aber an einer Behandlung muss mindestens ein Medikament genutzt werden.
 - Eine Behandlung erfolgt immer für einen Patienten aufgrund einer Krankheit durch einen Arzt. Ein Arzt kann dabei beliebig viele Behandlungen durchführen, für eine Krankheit kann es beliebig viele Behandlungen geben und ein Patient wird mindestens einmal behandelt.

Aufgabe 1: Informationsmodellierung (2)



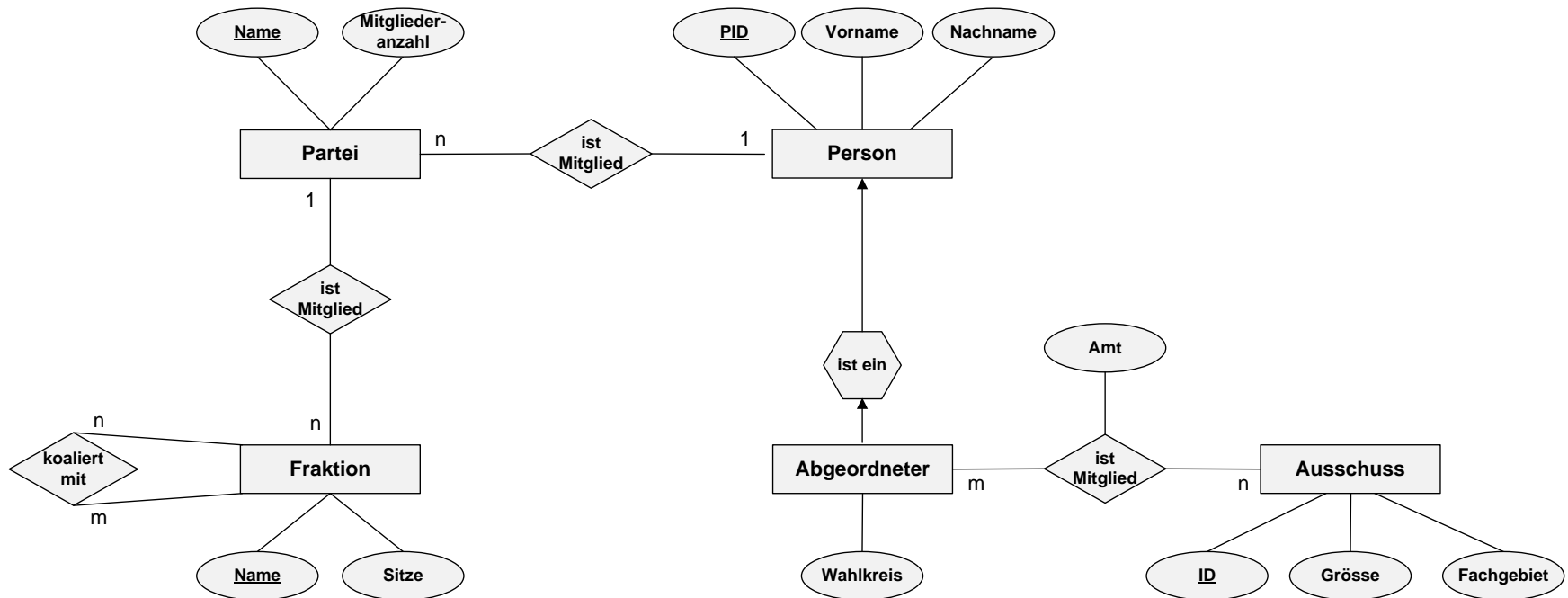
Aufgabe 2: Abbildung ER nach RM (1)



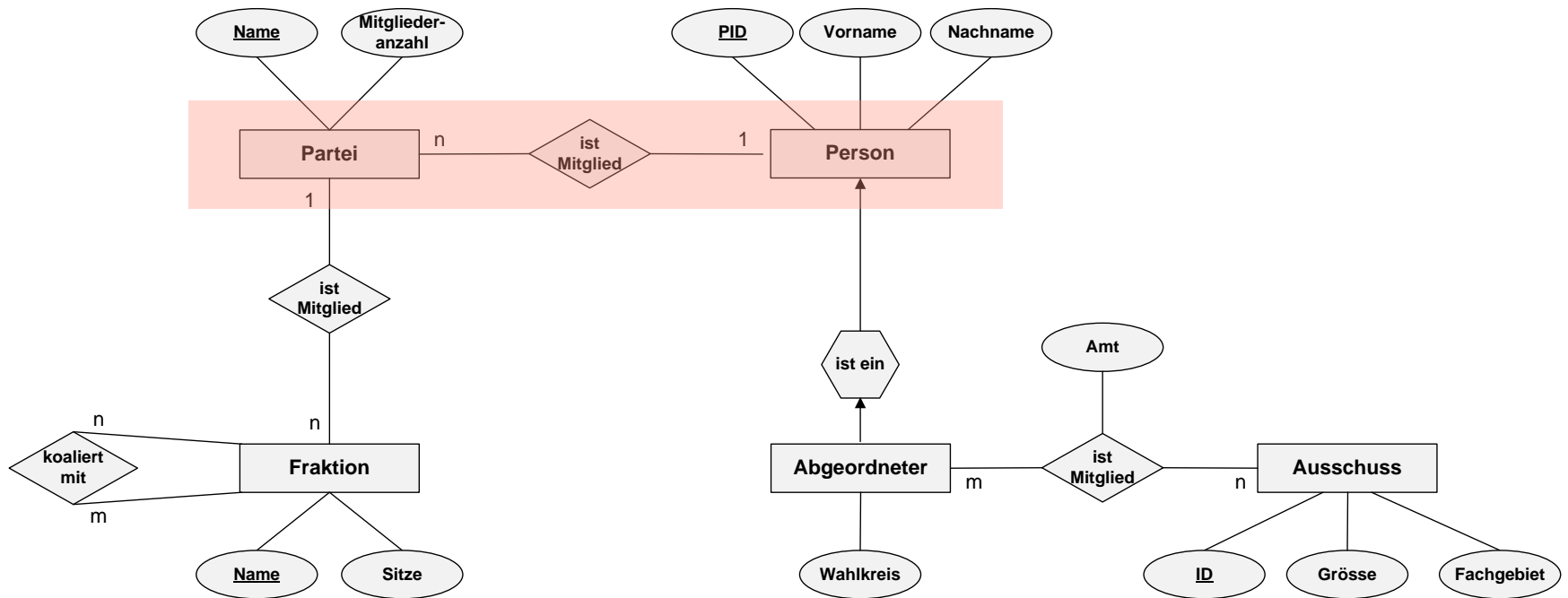
■ **Abbildungsregeln**

- 2 Entity-Mengen mit (n:m)-Verknüpfung:
→ Relationship als eigene Relation notwendig
- 1 Entity-Menge mit (n:m)-Verknüpfung:
→ Relationship als eigene Relation notwendig, Umbenennung erforderlich!
- 2 Entity-Mengen mit (1:n)-Verknüpfung:
→ ohne eigene Relation darstellbar
- 1 Entity-Menge mit (1:n)-Verknüpfung:
→ ohne eigene Relation darstellbar, Umbenennung erforderlich!

Aufgabe 2: Abbildung ER nach RM (2)

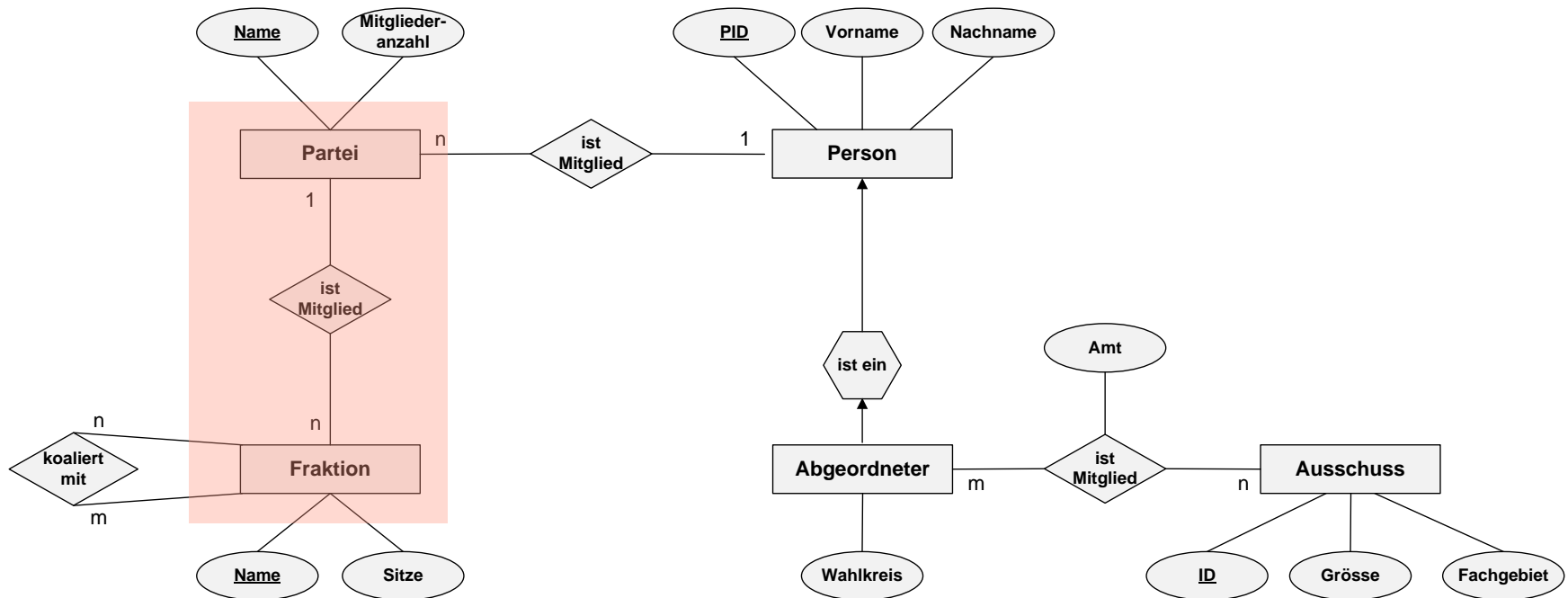


Aufgabe 2: Abbildung ER nach RM (3)



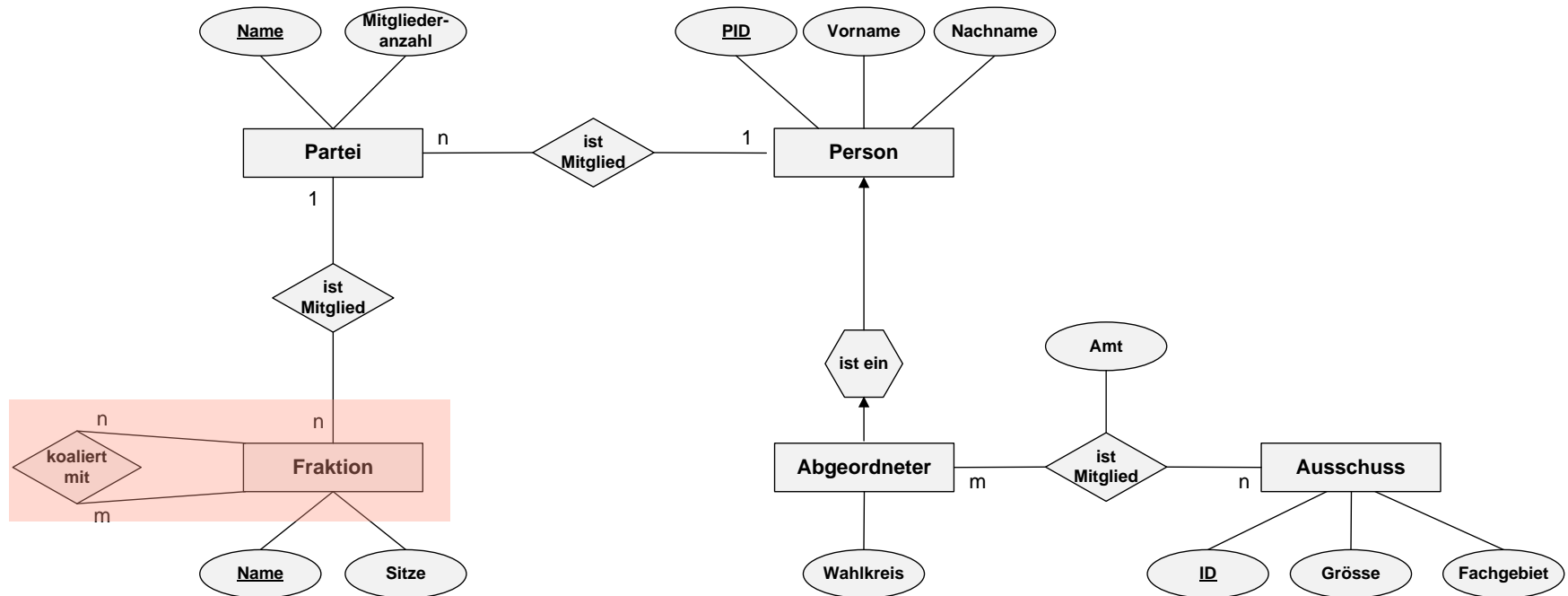
Person(PID, Vorname, Nachname, Partei → Partei.Name)
Partei(Name, Mitgliederzahl)

Aufgabe 2: Abbildung ER nach RM (4)



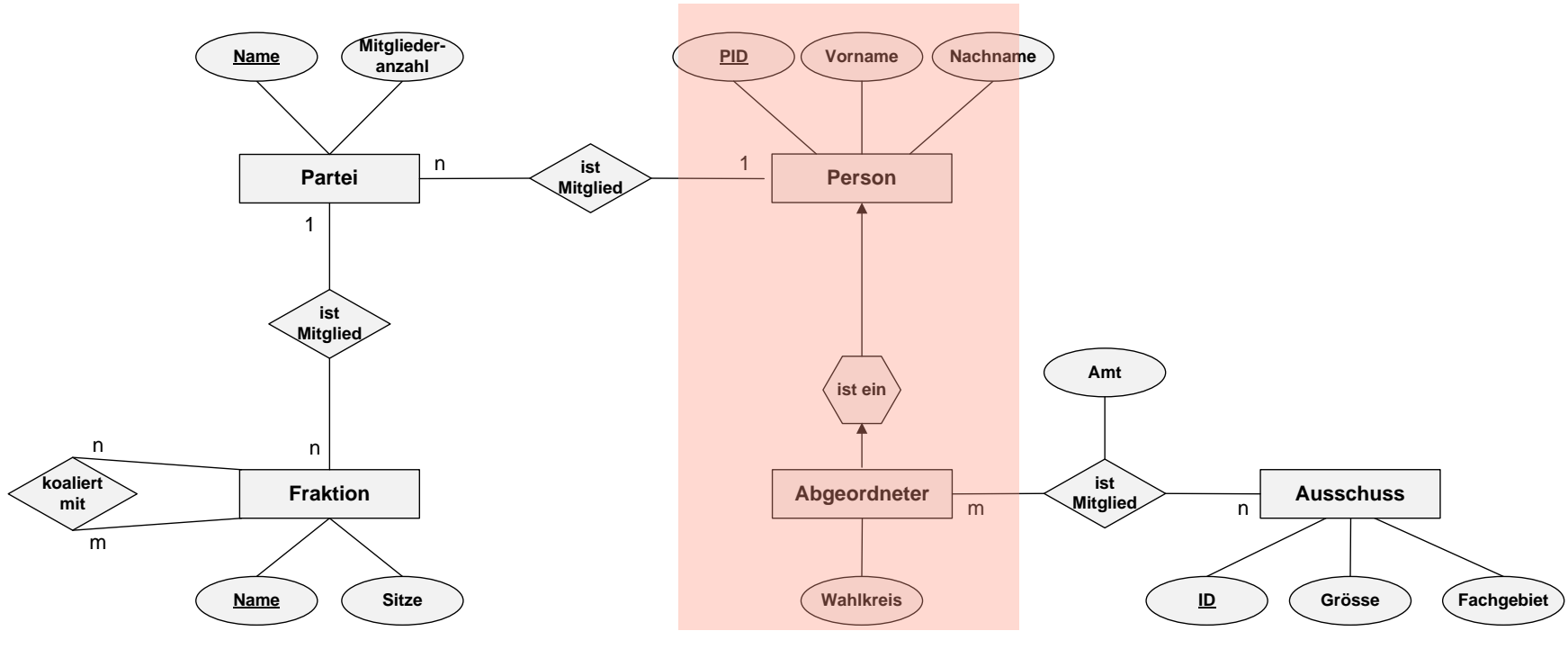
Partei(Name, Mitgliederzahl, Fraktion → Fraktion.Name)
Fraktion(Name, Sitze)

Aufgabe 2: Abbildung ER nach RM (5)



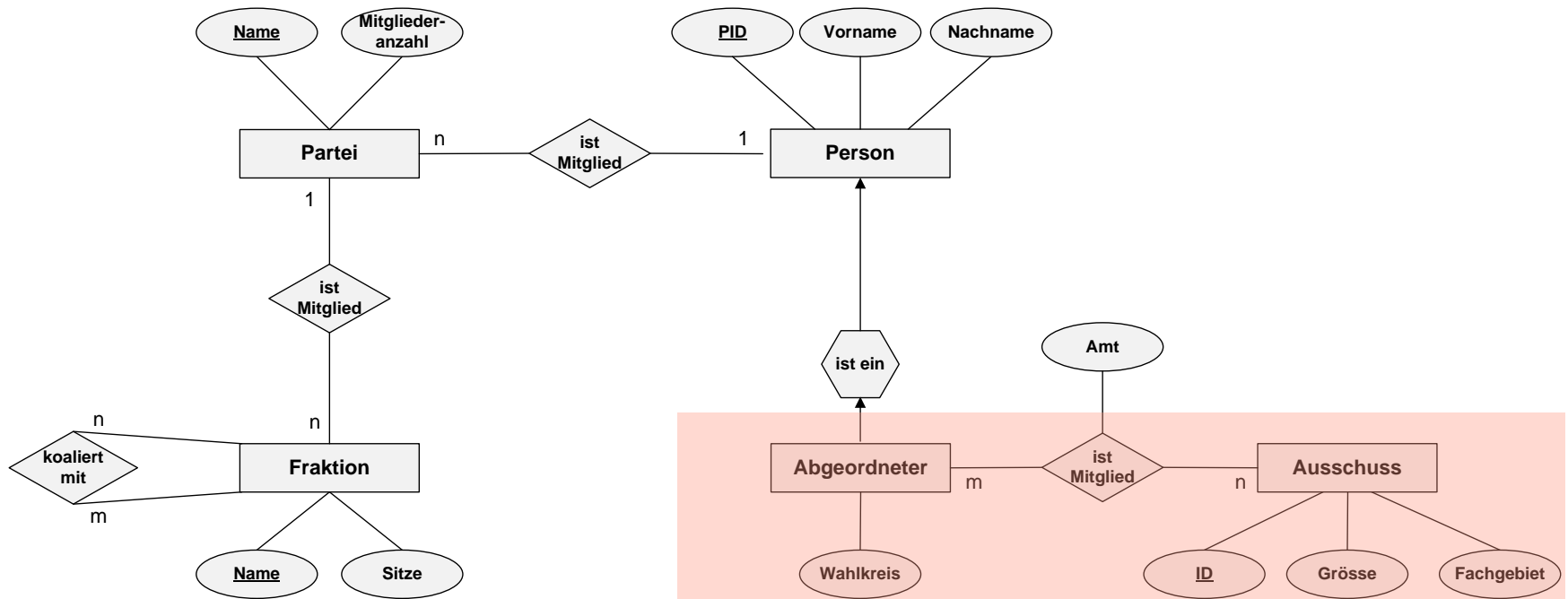
Fraktion(Name, Sitze)
koaliert_mit(Partner1 → Fraktion.Name, Partner2 → Fraktion.Name)

Aufgabe 2: Abbildung ER nach RM (6)



Person(PID, Vorname, Nachname)
Abgeordneter(PID → Person.PID, Wahlkreis)

Aufgabe 2: Abbildung ER nach RM (7)



Abgeordneter(PID → Person.PID, Wahlkreis)

Ausschuss(ID, Grösse, Fachgebiet)

ist_Mitglied(Abgeordneter → Abgeordneter.PID, Ausschuss → Ausschuss.ID, Amt)

Aufgabe 2: Abbildung ER nach RM (8)

▪ Resultierendes relationales Datenbankschema:

Person(PID, Vorname, Nachname, Partei → Partei.Name)

Abgeordneter(PID → Person.PID, Wahlkreis)

Partei(Name, Mitgliederzahl, Fraktion → Fraktion.Name)

Fraktion(Name, Sitze)

Ausschuss(ID, Groesse, Fachgebiet)

ist_Mitglied(Abgeordneter → Abgeordneter.PID, Ausschuss → Ausschuss.ID, Amt)

koaliert_mit(Partner1 → Fraktion.Name, Partner2 → Fraktion.Name)

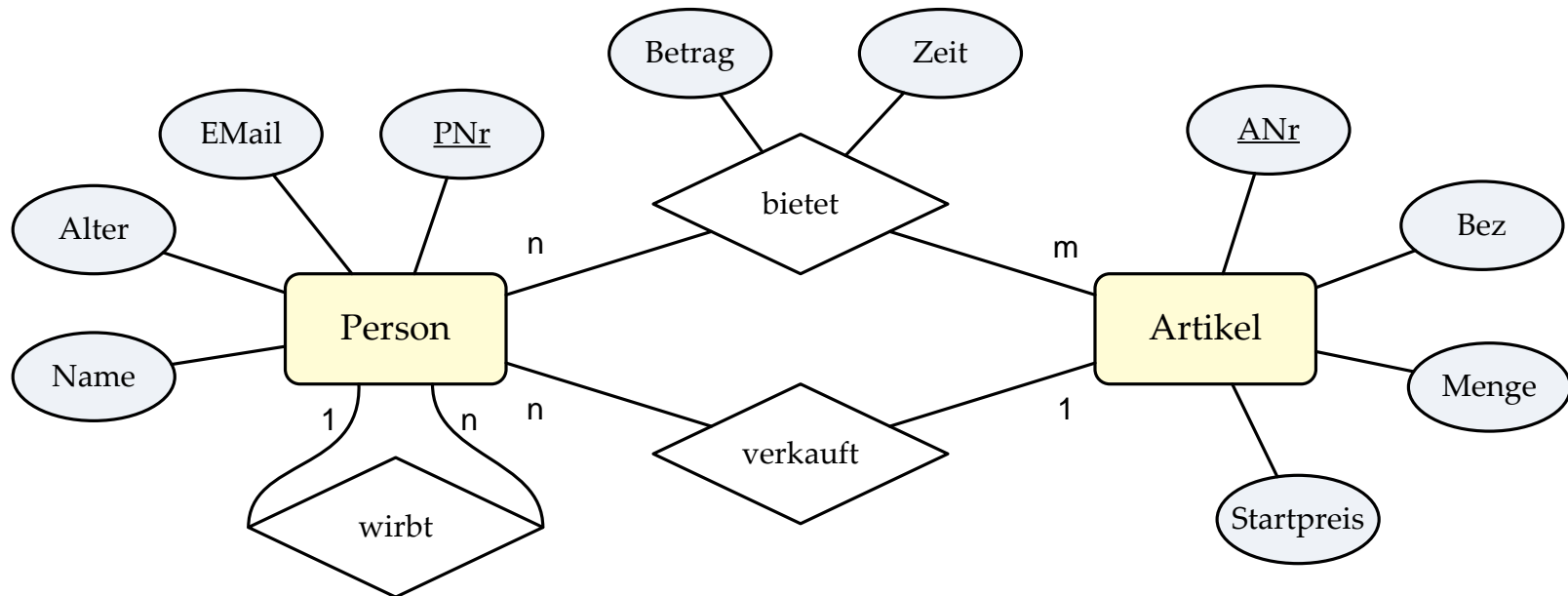
Aufgabe 3: Relationale Algebra

■ Schema

Person(PNr, Name, EMail, Alter, Werber \rightarrow Person.PNr)

Artikel(ANr, Bezeichnung, Menge, Startpreis, Verkäufer \rightarrow Person.PNr)

Gebot(Bieter \rightarrow Person.PNr, Artikel \rightarrow Artikel.ANr, Zeit, Betrag)



Aufgabe 3: Relationale Algebra

a) Anfrage in der Relationalen Algebra

Geben Sie einen Relationenalgebra-Ausdruck an, der alle Artikel (ANr, Bezeichnung), auf welche eine Person mit dem Namen 'Müller' geboten hat, ausgibt.

Person(PNr, Name, EMail, Alter, Werber → Person.PNr)

Artikel(ANr, Bezeichnung, Menge, Startpreis, Verkäufer → Person.PNr)

Gebot(Bieter → Person.PNr, Artikel → Artikel.ANr, Zeit, Betrag)

Aufgabe 3: Relationale Algebra

a) Anfrage in der Relationalen Algebra

Geben Sie einen Relationalenalgebra-Ausdruck an, der alle **Artikel (ANr, Bezeichnung)**, auf welche eine **Person** mit dem **Namen 'Müller'** **geboten** hat, ausgibt.

Person(PNr, Name, EMail, Alter, Werber → Person.PNr)

Artikel(ANr, **Bezeichnung**, Menge, Startpreis, Verkäufer → Person.PNr)

Gebot(Bieter → Person.PNr, Artikel → Artikel.ANr, Zeit, Betrag)

Aufgabe 3: Relationale Algebra

a) Anfrage in der Relationalen Algebra

Geben Sie einen Relationalenalgebra-Ausdruck an, der alle **Artikel (ANr, Bezeichnung)**, auf welche eine **Person** mit dem Namen 'Müller' **geboten** hat, ausgibt.

Person(PNr, Name, EMail, Alter, Werber → Person.PNr)

Artikel(ANr, Bezeichnung, Menge, Startpreis, Verkäufer → Person.PNr)

Gebot(Bieter → Person.PNr, Artikel → Artikel.ANr, Zeit, Betrag)

$\pi_{\text{ANr, Bezeichnung}} (\sigma_{\text{Name='Müller'}} (\text{Person}) \bowtie_{\text{PNr=Bieter}} \text{Gebot} \bowtie_{\text{Artikel=ANr}} \text{Artikel})$

Aufgabe 3: Relationale Algebra

b) Anfrage in der Relationalen Algebra

Geben Sie einen Relationalenalgebra-Ausdruck an, der alle Personen (PNr), welche noch keinen Artikel mit der Bezeichnung 'Buch' verkauft haben, ausgibt.

Person(PNr, Name, EMail, Alter, Werber → Person.PNr)

Artikel(ANr, Bezeichnung, Menge, Startpreis, Verkäufer → Person.PNr)

Gebot(Bieter → Person.PNr, Artikel → Artikel.ANr, Zeit, Betrag)

Aufgabe 3: Relationale Algebra

b) Anfrage in der Relationalen Algebra

Geben Sie einen Relationalenalgebra-Ausdruck an, der alle Personen (PNr), welche noch keinen Artikel mit der Bezeichnung 'Buch' verkauft haben, ausgibt.

Person(PNr, Name, EMail, Alter, Werber → Person.PNr)

Artikel(ANr, Bezeichnung, Menge, Startpreis, Verkäufer → Person.PNr)

Gebot(Bieter → Person.PNr, Artikel → Artikel.ANr, Zeit, Betrag)

Aufgabe 3: Relationale Algebra

b) Anfrage in der Relationalen Algebra

Geben Sie einen Relationalenalgebra-Ausdruck an, der alle Personen (PNr), welche noch keinen Artikel mit der Bezeichnung 'Buch' verkauft haben, ausgibt.

Person(PNr, Name, EMail, Alter, Werber → Person.PNr)

Artikel(ANr, Bezeichnung, Menge, Startpreis, Verkäufer → Person.PNr)

Gebot(Bieter → Person.PNr, Artikel → Artikel.ANr, Zeit, Betrag)

$$\pi_{\text{PNr}}(\text{Person}) - \pi_{\text{Verkäufer}}(\sigma_{\text{Bezeichnung}='Buch'}(\text{Artikel}))$$

Aufgabe 3: Relationale Algebra

c) Anfrage in der Relationalen Algebra

Geben Sie eine SQL-Anweisung an, welche die gleiche Ergebnismenge liefert wie der folgende Relationenalgebra-Ausdruck:

$$\pi_{\text{Name}} \left(\left(\text{Person} \underset{\text{PNr=Bieter}}{\bowtie} \text{Gebot} \underset{\text{Artikel=ANr}}{\bowtie} \text{Artikel} \right) \underset{\substack{\text{Werber=PNr} \\ \wedge \text{Verkäufer=PNr}}}{\bowtie} \left(\pi_{\text{PNr}}(\text{Person}) \right) \right)$$

Aufgabe 3: Relationale Algebra

c) Anfrage in der Relationalen Algebra

Geben Sie eine SQL-Anweisung an, welche die gleiche Ergebnismenge liefert wie der folgende Relationenalgebra-Ausdruck:

$$\pi_{\text{Name}} \left(\left(\text{Person} \underset{\text{PNr}=\text{Bieter}}{\bowtie} \text{Gebot} \underset{\text{Artikel}=\text{ANr}}{\bowtie} \text{Artikel} \right) \underset{\substack{\text{Werber}=\text{PNr} \\ \wedge \text{Verkäufer}=\text{PNr}}}{\bowtie} \left(\pi_{\text{PNr}} (\text{Person}) \right) \right)$$

```
SELECT DISTINCT p1.Name
FROM Person p1, Gebot g, Artikel a, Person p2
WHERE p1.PNr=g.Bieter AND g.Artikel=a.ANr
AND p2.PNr=a.Verkäufer AND p2.PNr=p1.Werber;
```

Aufgabe 3: Relationale Algebra

c) Anfrage in der Relationalen Algebra

Geben Sie eine SQL-Anweisung an, welche die gleiche Ergebnismenge liefert wie der folgende Relationenalgebra-Ausdruck:

$$\pi_{\text{Name}} \left(\left(\text{Person} \bowtie_{\text{PNr}=\text{Bieter}} \text{Gebot} \bowtie_{\text{Artikel}=\text{ANr}} \text{Artikel} \right) \bowtie_{\substack{\text{Werber}=\text{PNr} \\ \wedge \text{Verk\u00e4ufer}=\text{PNr}}} \left(\pi_{\text{PNr}} (\text{Person}) \right) \right)$$

```
SELECT DISTINCT p1.Name
FROM Person p1, Gebot g, Artikel a, Person p2
WHERE p1.PNr=g.Bieter AND g.Artikel=a.ANr
AND p2.PNr=a.Verk\u00e4ufer AND p2.PNr=p1.Werber;
```

Aufgabe 3: Relationale Algebra

d) Anfrage in der Relationalen Algebra

Der kleinste in der Datenbank gespeicherte Startpreis eines Artikels beträgt 10 € und der größte 159 €. Bestimmen Sie die Selektivität der Selektion $\sigma_{\text{Startpreis} \geq 46}$ unter der Annahme einer Gleichverteilung.

Bei Gleichverteilung:

SF = Anteil an Elementen, die die Bedingung erfüllen

SF = (Anzahl an Preisen größer gleich 46) / (Anzahl an Preisen)

SF = $114/150 = 0.76$

Aufgabe 3: Relationale Algebra

d) Anfrage in der Relationalen Algebra

Der kleinste in der Datenbank gespeicherte Startpreis eines Artikels beträgt 10 € und der größte 159 €. Bestimmen Sie die Selektivität der Selektion $\sigma_{\text{Startpreis} \geq 46}$ unter der Annahme einer Gleichverteilung.

Bei Gleichverteilung:

SF = Anteil an Elementen, die die Bedingung erfüllen

SF = (Anzahl an Preisen größer gleich 46) / (Anzahl an Preisen)

SF = 114/150 = 0.76

Aufgabe 3: Relationale Algebra

d) Anfrage in der Relationalen Algebra

Der kleinste in der Datenbank gespeicherte Startpreis eines Artikels beträgt 10 € und der größte 159 €. Bestimmen Sie die Selektivität der Selektion $\sigma_{\text{Startpreis} \geq 46}$ unter der Annahme einer Gleichverteilung.

Bei Gleichverteilung:

SF = Anteil an Elementen, die die Bedingung erfüllen

SF = (Anzahl an Preisen größer gleich 46) / (Anzahl an Preisen)

SF = 114/150 = 0.76

Häufig wird vereinfacht berechnet:

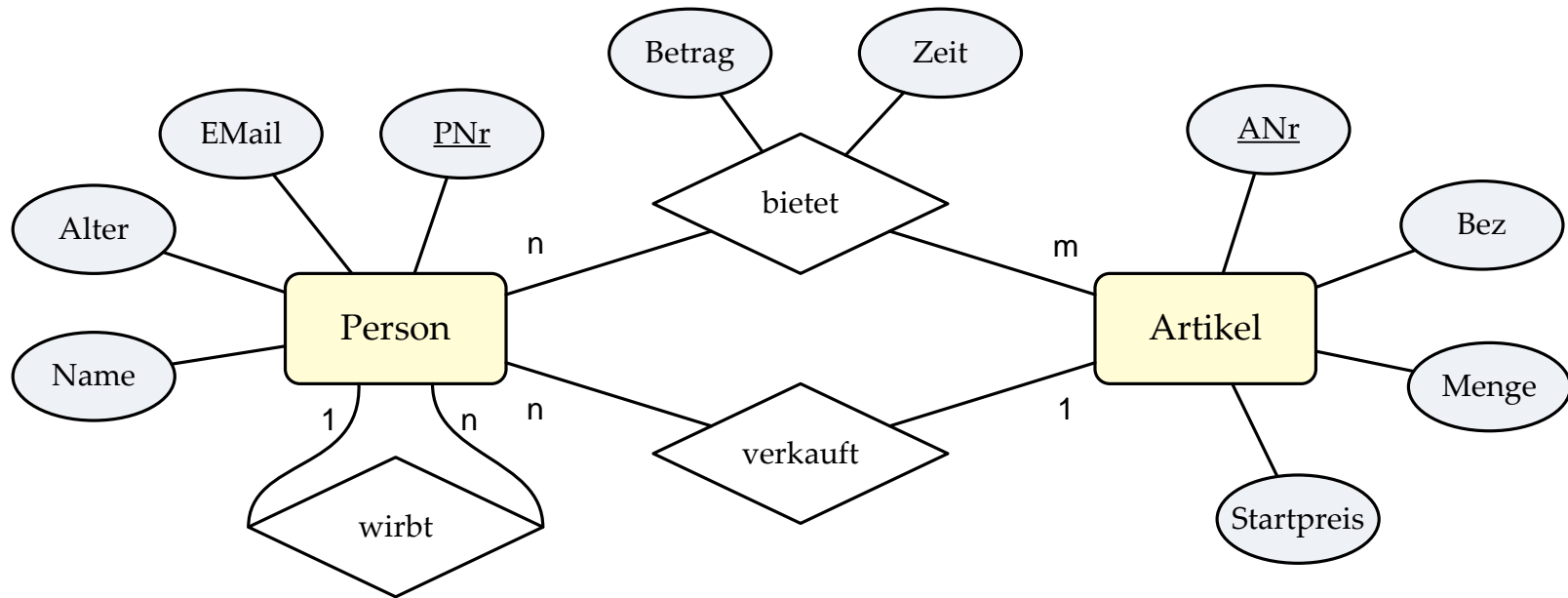
SF = (maxSel-minSel) / (max-min)

= (159-46) / (159-10) = 113/149 = 0.758

Aufgabe 4: SQL (1)

■ Schema

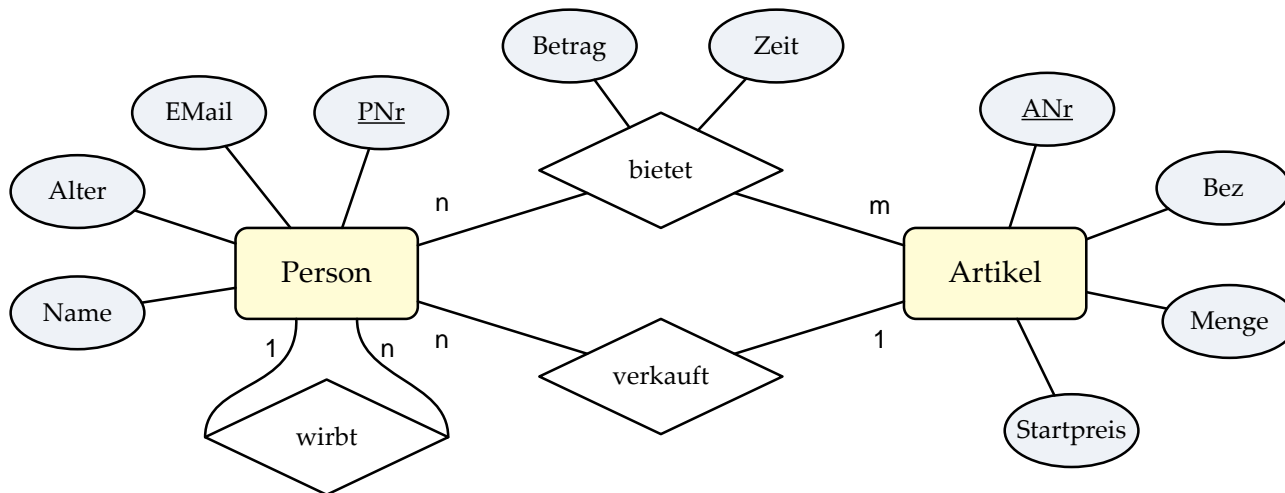
Person(PNr, Name, EMail, Alter, Werber → Person.PNr)
Artikel(ANr, Bezeichnung, Menge, Startpreis, Verkäufer → Person.PNr)
Gebot(Bieter → Person.PNr, Artikel → Artikel.ANr, Zeit, Betrag)



Aufgabe 4: SQL (2)

- Die PNr, die EMail und alle gebotenen Beträge der Personen namens 'Harry' nach PNr absteigend sortiert.

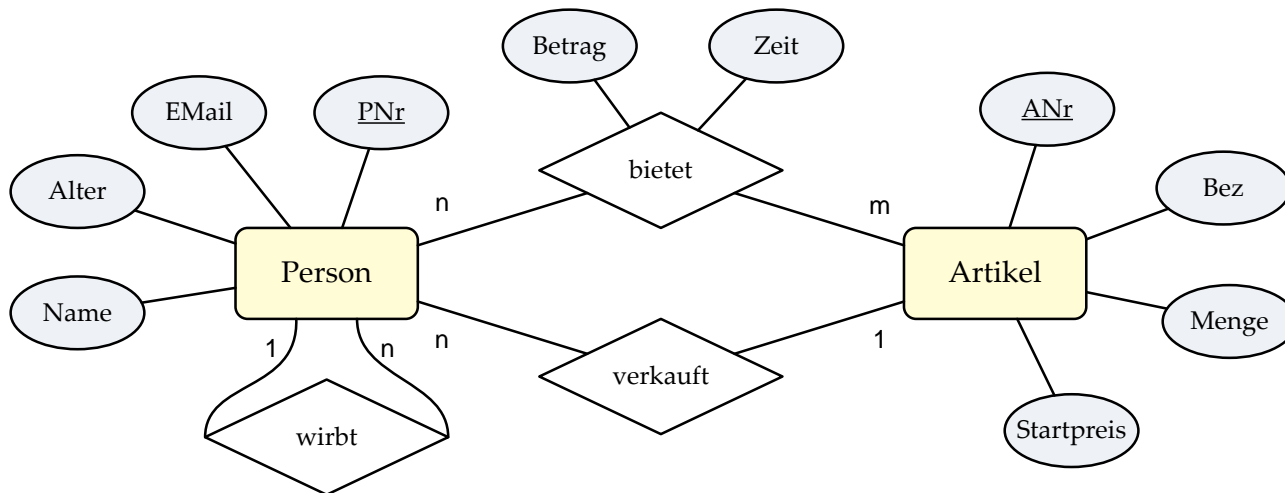
```
SELECT p.PNr, p.EMail, g.Betrag
FROM Person p, Gebot g
WHERE g.Bieter = p.PNr
AND p.Name = 'Harry'
ORDER BY p.PNr DESC
```



Aufgabe 4: SQL (3)

- Die PNr, der Name sowie der in allen Geboten abgegebene Gesamtbetrag jeder Person, die älter als 30 Jahre alt ist

```
SELECT p.PNr, p.Name, SUM(g.Betrag) as Gesamtbetrag
FROM Person p, Gebot g
WHERE p.Alter > 30
AND g.Bieter = p.PNr
GROUP BY p.PNr, p.Name;
```

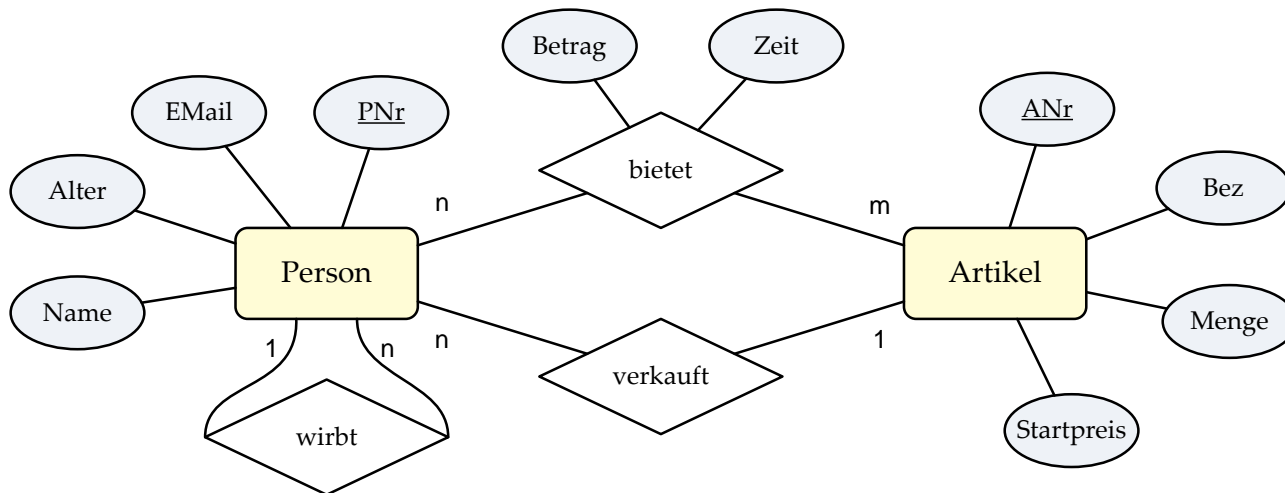


Aufgabe 4: SQL (4)

- Die Namen und EMail aller Personen, die noch nie geboten haben.

```
SELECT DISTINCT p.Name, p.EMail
FROM Person p
WHERE p.PNr NOT IN(
    SELECT g.Bieter
    FROM Gebot g)
```

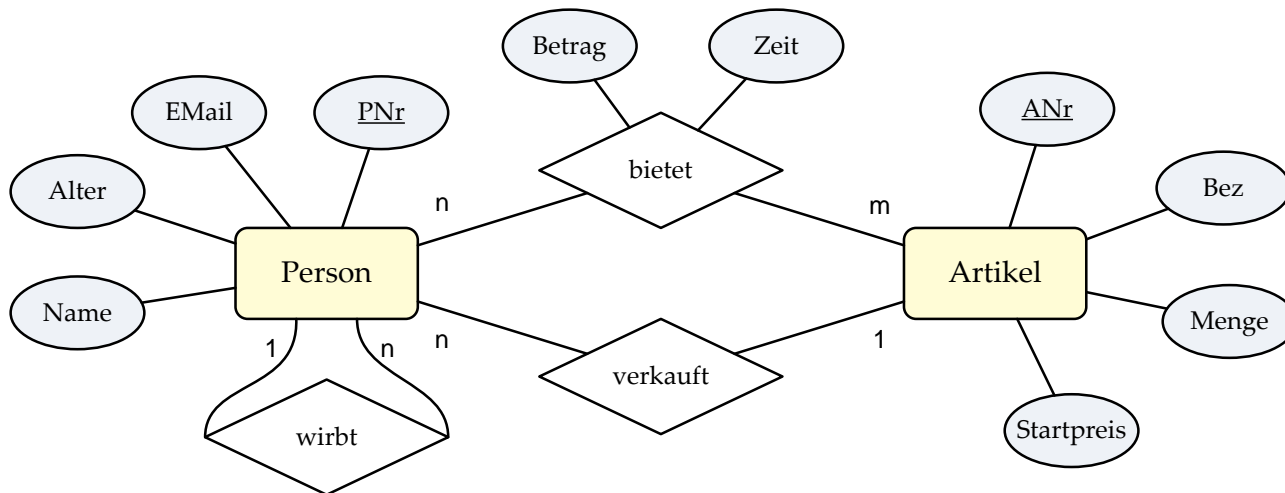
```
SELECT DISTINCT p.Name, p.EMail
FROM Person p
WHERE NOT EXISTS( SELECT *
    FROM Gebot g
    WHERE g.Bieter=p.PNr)
```



Aufgabe 4: SQL (5)

- Die ANr, die Bezeichnung und den Startpreis jedes Artikels, dessen Verkäufer von einer Person mit dem Namen 'Bond' geworben wurde.

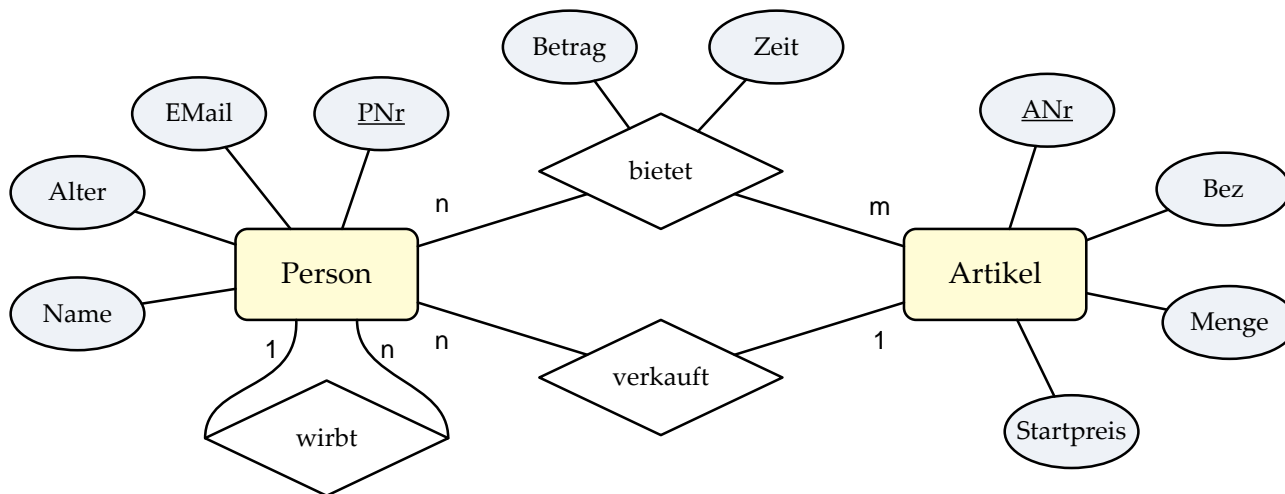
```
SELECT a.ANr, a.Bezeichnung, a.Startpreis
FROM Person p1, Person p2, Artikel a
WHERE a.Verkäufer = p1.PNr
AND p1.Werber = p2.PNr
AND p2.Name = 'Bond'
```



Aufgabe 4: SQL (6)

- Die PNr und das durchschnittliche Alter der von ihm geworbenen Personen für jeden Werber, der maximal 10 Personen geworben hat.

```
SELECT p.Werber, AVG(p.Alter) as Durchschnittsalter
FROM Person p
GROUP BY p.Werber
HAVING(COUNT(*)<=10)
```



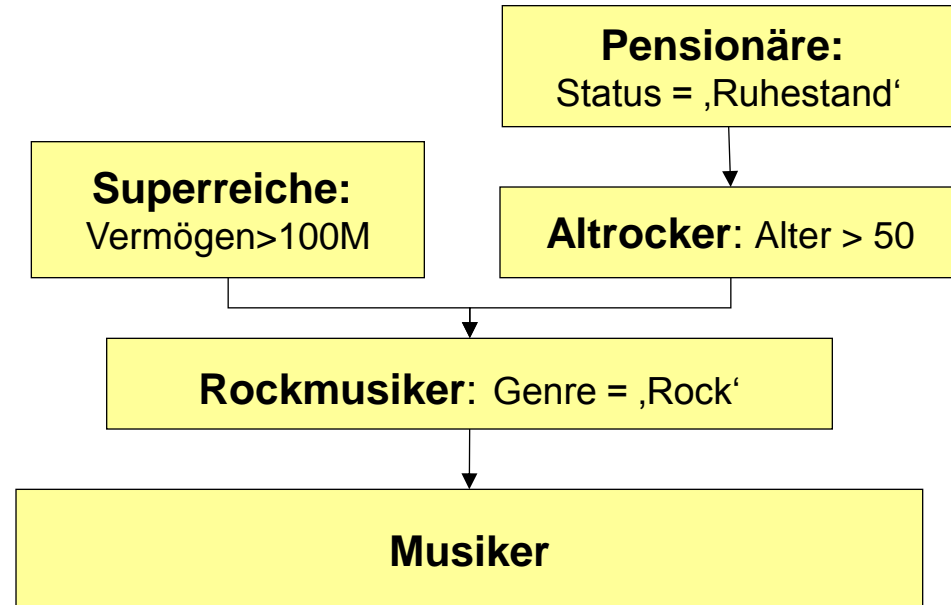
Aufgabe 5: Änderungen auf Sichten

```
CREATE VIEW Rockmusiker  
AS SELECT * FROM Musiker  
WHERE Genre = 'Rock'
```

```
CREATE VIEW Altrocker  
AS SELECT * FROM Rockmusiker  
WHERE Alter > 50;
```

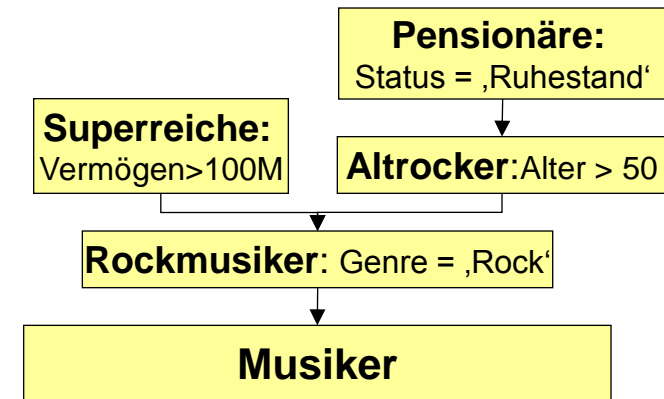
```
CREATE VIEW Pensionäre  
AS SELECT * FROM Altrocker  
WHERE Status = 'Ruhestand';
```

```
CREATE VIEW Superreiche  
AS SELECT * FROM Rockmusiker  
WHERE Vermögen > 100M;
```



Aufgabe 5: Änderungen auf Sichten

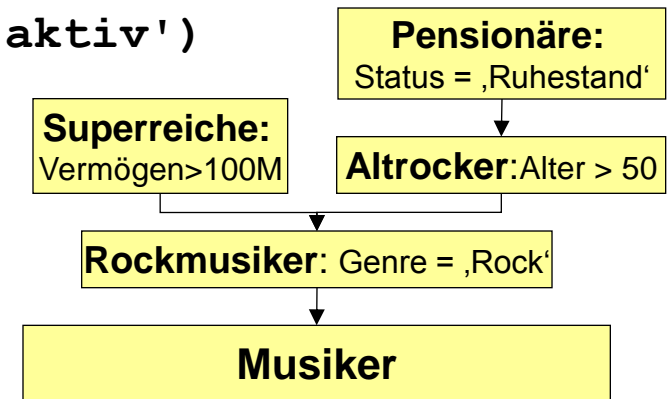
```
UPDATE Rockmusiker SET Alter = 49
WHERE Alter = 51
AND Status = 'aktiv'
AND Vermögen < 50M
```



	Konf. 1	Konf. 2	Konf. 3	Konf. 4
Rockmusiker	-	-	CASC	-
Altrockler	-	CASC	-	-
Pensionäre	-	-	-	CASC
Superreiche	-	-	CASC	-
a)	Ja / R	Ja / R	Ja / R	Ja / R

Aufgabe 5: Änderungen auf Sichten

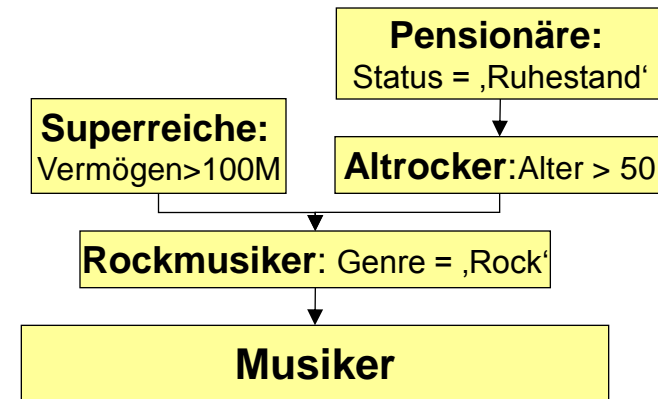
```
INSERT INTO Superreiche
VALUES (12, 'Madonna', 50, 'Pop', 500M, 'aktiv')
```



	Konf. 1	Konf. 2	Konf. 3	Konf. 4
Rockmusiker	-	-	CASC	-
Altrockler	-	CASC	-	-
Pensionäre	-	-	-	CASC
Superreiche	-	-	CASC	-
b)	Ja / -	Ja / -	Nein	Ja / -

Aufgabe 5: Änderungen auf Sichten

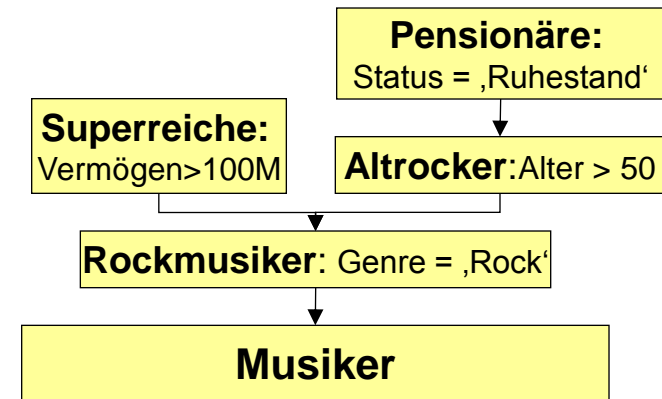
```
UPDATE Rockmusiker SET Status = 'aktiv',
WHERE Name = 'Beatles'
AND Alter > 60
AND Vermögen > 200M
```



	Konf. 1	Konf. 2	Konf. 3	Konf. 4
Rockmusiker	-	-	CASC	-
Altrockler	-	CASC	-	-
Pensionäre	-	-	-	CASC
Superreiche	-	-	CASC	-
c)	Ja / R,A,S	Ja / R,A,S	Ja / R,A,S	Ja / R,A,S

Aufgabe 5: Änderungen auf Sichten

UPDATE Pensionäre SET **Alter = 48**
WHERE Vermögen > 200M



	Konf. 1	Konf. 2	Konf. 3	Konf. 4
Rockmusiker	-	-	CASC	-
Altrocker	-	CASC	-	-
Pensionäre	-	-	-	CASC
Superreiche	-	-	CASC	-
d)	Ja / R,S	Nein	Ja / R,S	Nein

Aufgabe 6: Normalformen

R(N,O,D,T,Z,S)

Fkt. Abhängigkeiten

FA1: $N \rightarrow S$

FA2: $T \rightarrow O, Z$

FA3: $N, D \rightarrow T$

FA4: $S, D \rightarrow N$

FA5: $Z \rightarrow O$

i) Schlüsselkandidaten: (Attributkombinationen die eindeutig und minimal sind)

SK1: N,D

SK2: S,D

Aufgabe 6: Normalformen

R(N,O,D,T,Z,S)

Fkt. Abhängigkeiten

FA1: N → S

FA2: T → O, Z

FA3: N,D → T

FA4: S,D → N

FA5: Z → O

i) Schlüsselkandidaten: (Attributkombinationen die eindeutig und minimal sind)

SK1: N,D

SK2: S,D

Aufgabe 6: Normalformen

R(N,O,D,T,Z,S)

Fkt. Abhängigkeiten

FA1: $N \rightarrow S$

FA2: $T \rightarrow O, Z$

FA3: $N, D \rightarrow T$

FA4: $S, D \rightarrow N$

FA5: $Z \rightarrow O$

i) Schlüsselkandidaten: (Attributkombinationen die eindeutig und minimal sind)

SK1: N,D

SK2: S,D

Aufgabe 6: Normalformen

R(N,O,D,T,Z,S)

Fkt. Abhängigkeiten

FA1: $N \rightarrow S$

FA2: $T \rightarrow O, Z$

FA3: $N, D \rightarrow T$

FA4: $S, D \rightarrow N$

FA5: $Z \rightarrow O$

i) Schlüsselkandidaten: (Attributkombinationen die eindeutig und minimal sind)

SK1: N,D

SK2: S,D

ii) Nicht-Primärattribute: (Attribute die in keinem Schlüsselkandidaten vorkommen)

NP1: O

NP2: T

NP3: Z

Aufgabe 6: Normalformen

R(N,O,D,T,Z,S)

Fkt. Abhängigkeiten

FA1: $N \rightarrow S$

FA2: $T \rightarrow O, Z$

FA3: $N, D \rightarrow T$

FA4: $S, D \rightarrow N$

FA5: $Z \rightarrow O$

iii) Normalformen:

- 1NF: JA, da ALLE Attribute atomar sind (siehe Aufgabenstellung)

Aufgabe 6: Normalformen

R(N,O,D,T,Z,S)

Fkt. Abhängigkeiten

FA1: N → S

FA2: T → O, Z

FA3: N,D → T

FA4: S,D → N

FA5: Z → O

iii) Normalformen:

- 1NF: JA, da ALLE Attribute atomar sind (siehe Aufgabenstellung)
- 2NF: JA, da kein Nicht-Primärattribut partiell von einem der Schlüsselkandidaten abhängt (!! S ist kein Nicht-Primärattribut)

Aufgabe 6: Normalformen

R(N,O,D,T,Z,S)

Fkt. Abhängigkeiten

FA1: $N \rightarrow S$

FA2: $T \rightarrow O, Z$

FA3: $N, D \rightarrow T$

FA4: $S, D \rightarrow N$

FA5: $Z \rightarrow O$

iii) Normalformen:

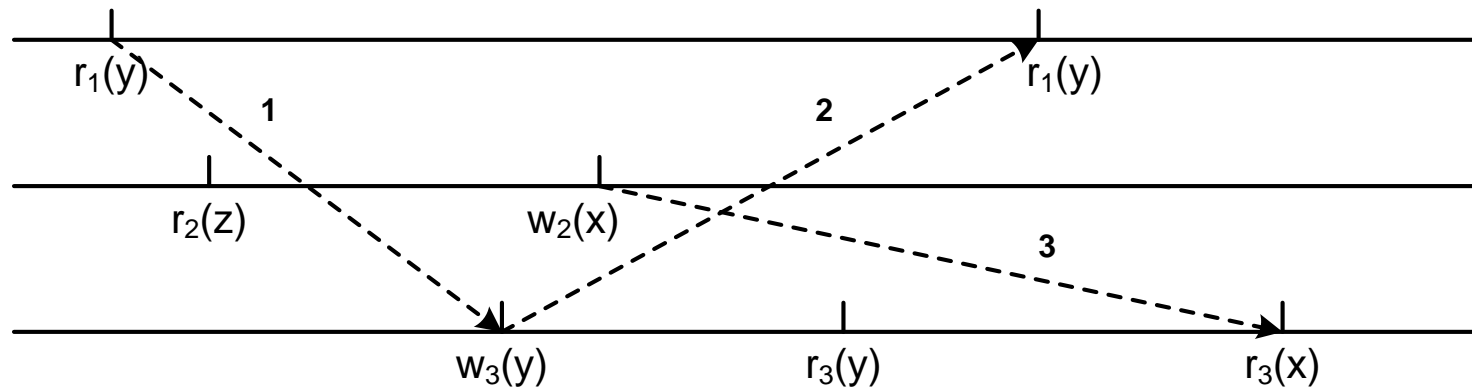
- 1NF: JA, da ALLE Attribute atomar sind (siehe Aufgabenstellung)
- 2NF: JA, da kein Nicht-Primärattribut partiell von einem der Schlüsselkandidaten abhängt (!! S ist kein Nicht-Primärattribut)
- 3NF: NEIN, da O und Z über T transitiv von dem Schlüsselkandidaten N,D abhängen

Aufgabe 7: Transaktionen (1)

- **A) Bestimmen Sie bei den folgenden drei Schedules S_1 , S_2 und S_3 jeweils,**
 - ob diese serialisierbar sind oder nicht.
 - Begründen Sie Ihre Antwort, indem Sie entweder alle äquivalenten, seriellen Transaktionsreihenfolgen (z.B. $T_1 T_2 T_3$) oder einen auftretenden Konfliktzyklus angeben.
- **Erinnerung:**
 - Die parallele Ausführung einer Menge von Transaktionen ist **serialisierbar**, wenn es eine **serielle Ausführung** derselben TA-Menge gibt, die den **gleichen DB-Zustand** und **die gleichen Ausgabewerte** wie die ursprüngliche Ausführung erzielt

Aufgabe 7: Transaktionen (2)

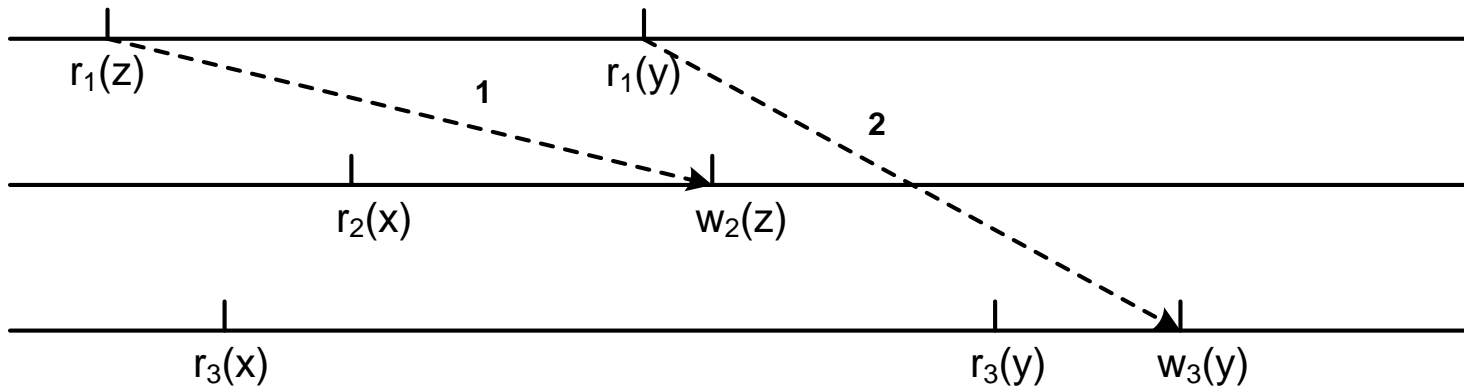
- **A) i) $S_1 = r_1(y) r_2(z) w_3(y) w_2(x) r_3(y) r_1(y) r_3(x)$**



- Aus 1 ergibt sich: T_1 vor T_3
- Aus 2 ergibt sich: T_3 vor T_1
- Aus 3 ergibt sich: T_2 vor T_3
- $\rightarrow S_1$ ist nicht serialisierbar!

Aufgabe 7: Transaktionen (3)

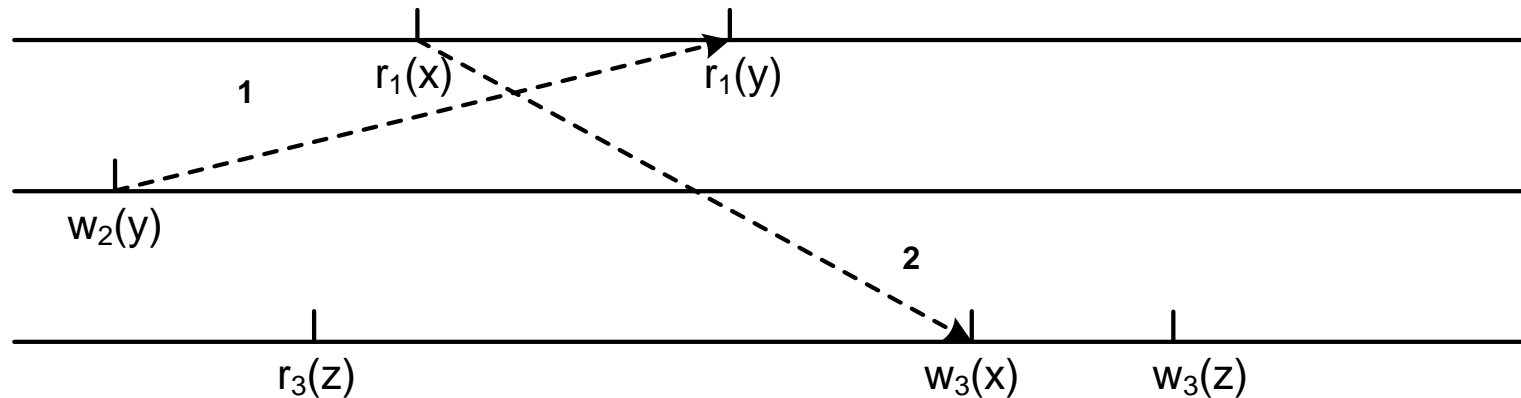
- **A) ii) $S_2 = r_1(z) r_3(x) r_2(x) r_1(y) w_2(z) r_3(y) w_3(y)$**



- Aus 1 ergibt sich: T_1 vor T_2
- Aus 2 ergibt sich: T_1 vor T_3
- $\rightarrow S_2$ ist serialisierbar: $T_1T_2T_3$ oder $T_1T_3T_2$

Aufgabe 7: Transaktionen (4)

- **A) iii) $S_3 = w_2(y) r_3(z) r_1(x) r_1(y) w_3(x) w_3(z)$**



- Aus 1 ergibt sich: T_2 vor T_1
- Aus 2 ergibt sich: T_1 vor T_3
- $\rightarrow S_3$ ist serialisierbar: $T_2 T_1 T_3$

Aufgabe 7: Transaktionen (5)

- **B) Geben Sie eine Operation von T₂ zum Zeitschritt 3 an die für T₁ ein non-repeatable Read verursacht**

Zeitschritt	T1	T2
1	SELECT SUM(Gehalt) FROM Personal WHERE Abteilung = 'Treadstone';	
2		SELECT AVG(Alter) FROM Personal;
3		
4	SELECT SUM(Gehalt) FROM Personal WHERE Abteilung = 'Treadstone';	
5	COMMIT;	
6		COMMIT;

- Erinnerung: Voraussetzung für non-repeatable Read ist eine Abhängigkeit der Form $r_1(X) \rightarrow w_2(X)$

Aufgabe 7: Transaktionen (6)

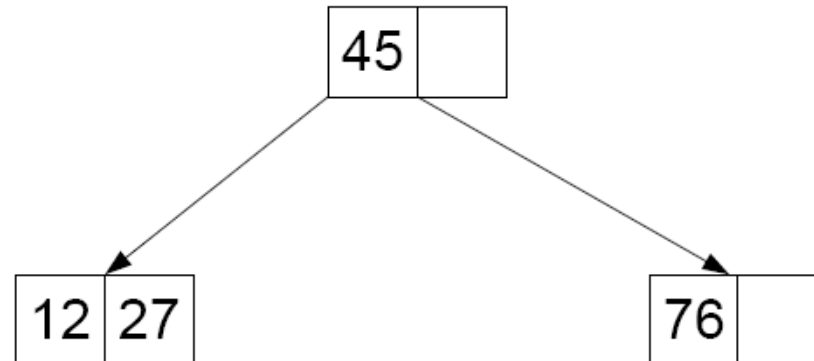
■ B) Lösung: Update oder Delete eines gelesenen Datensatzes

Zeitschritt	T1	T2
1	SELECT SUM(Gehalt) FROM Personal WHERE Abteilung = 'Treadstone';	
2		SELECT AVG(Alter) FROM Personal;
3		UPDATE Personal SET Gehalt = Gehalt*2 WHERE Abteilung = 'Treadstone';
4	SELECT SUM(Gehalt) FROM Personal WHERE Abteilung = 'Treadstone';	
5	COMMIT;	
6		COMMIT;

- Warum kein Insert
 - Insert → Phantomproblem

Aufgabe 8: Einfügen und Löschen in B-Bäumen

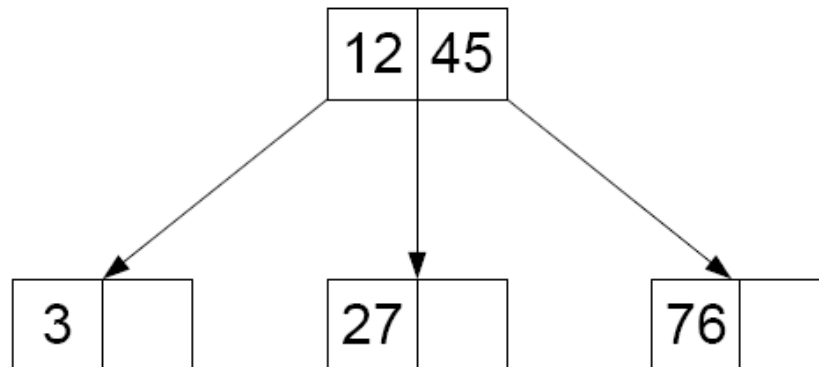
a) **Einfügen:** Gegeben ein Baum der Klasse $\tau(1,2)$



Aufgabe 8: Einfügen und Löschen in B-Bäumen

a) **Einfügen:** Gegeben ein Baum der Klasse $\tau(1,2)$

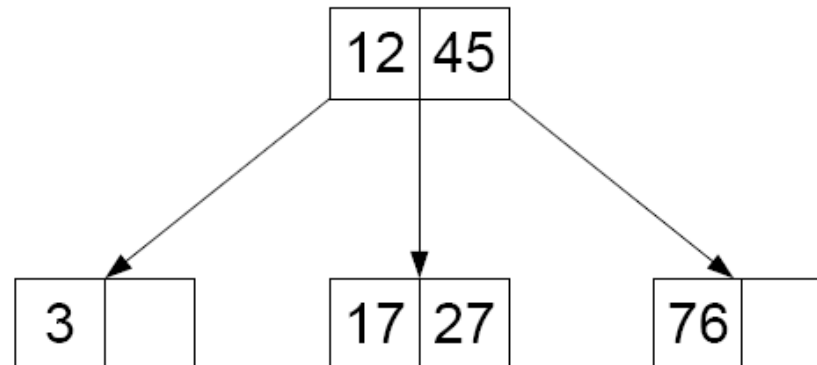
Schlüsselwert 3 (Splitten)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

a) **Einfügen:** Gegeben ein Baum der Klasse $T(1,2)$

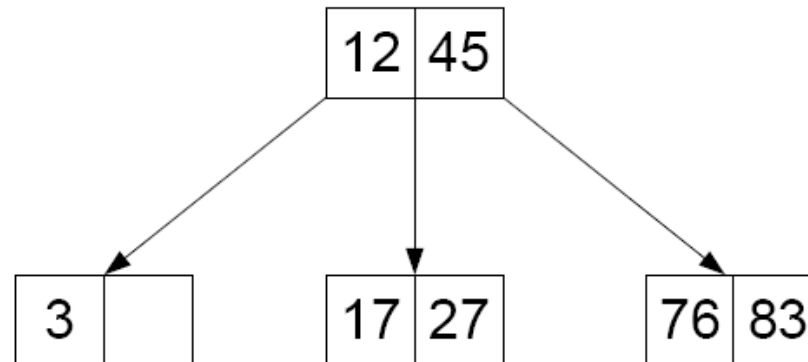
Schlüsselwert 17 (einfaches Einfügen)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

a) **Einfügen:** Gegeben ein Baum der Klasse $T(1,2)$

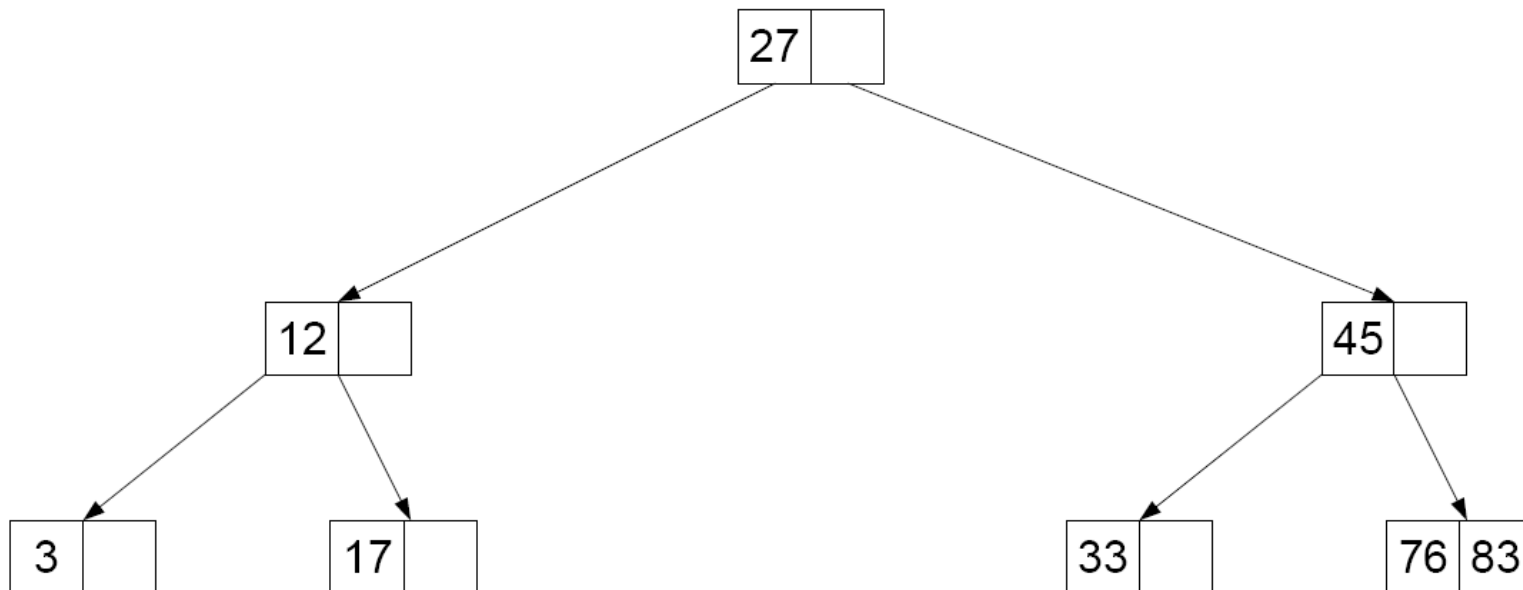
Schlüsselwert 83 (einfaches Einfügen)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

a) **Einfügen:** Gegeben ein Baum der Klasse $\tau(1,2)$

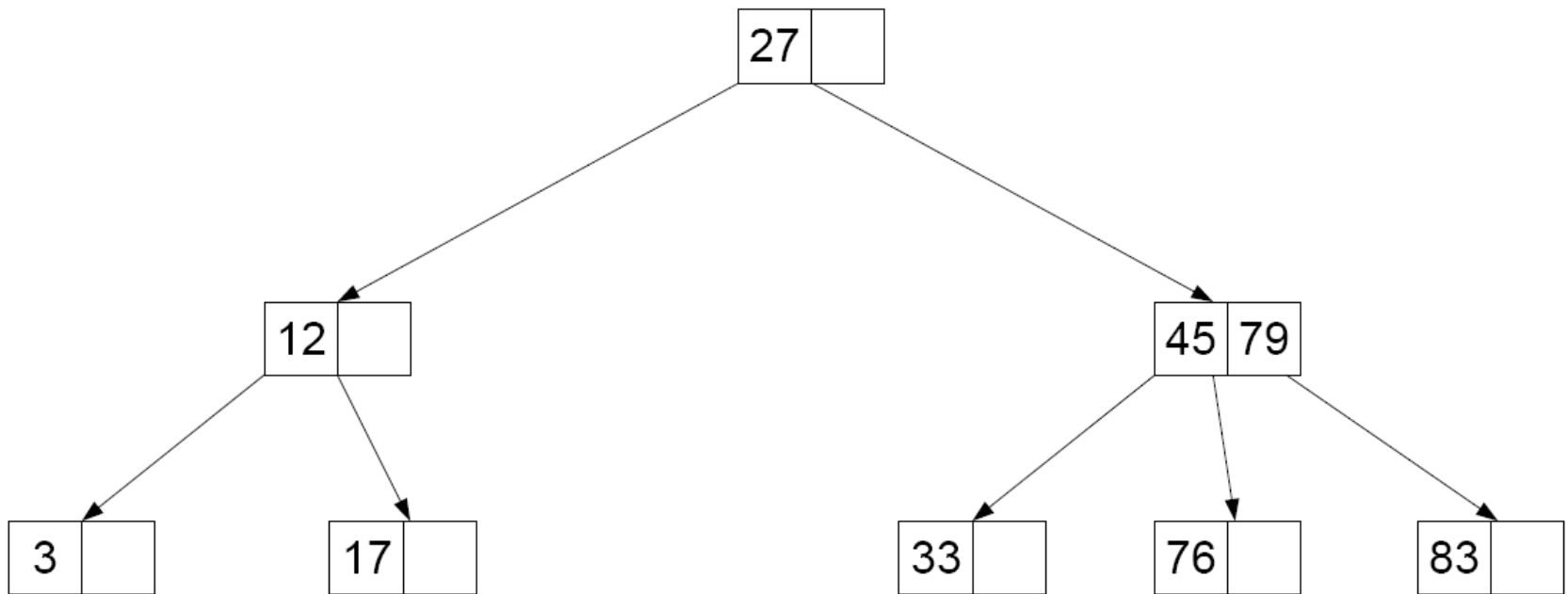
Schlüsselwert 33 (Splitten)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

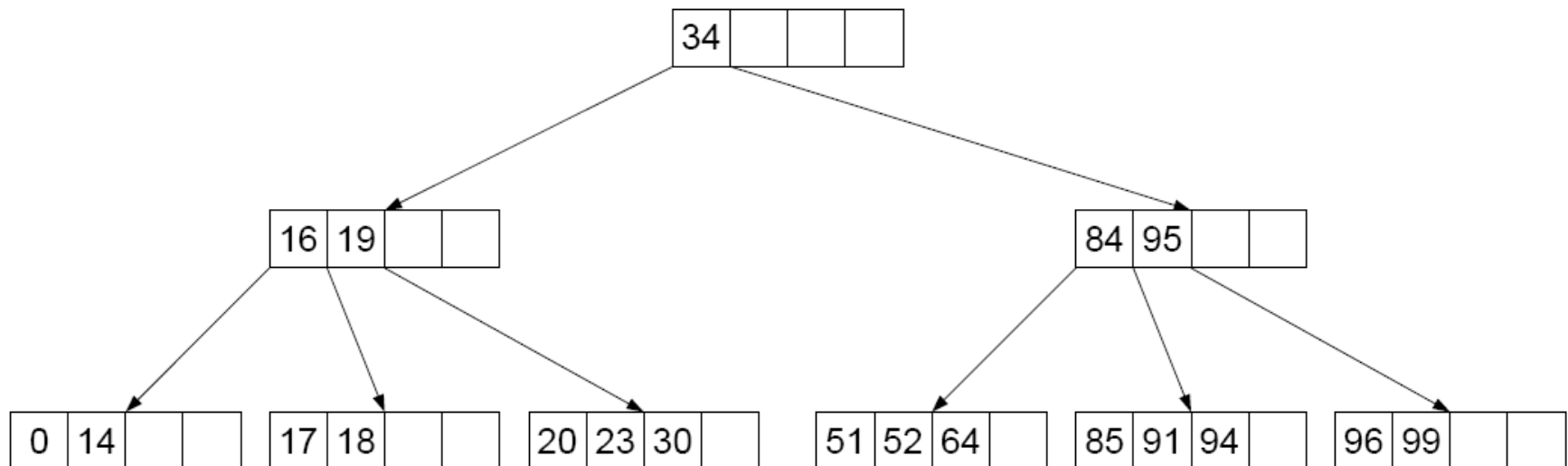
a) **Einfügen:** Gegeben ein Baum der Klasse $\tau(1,2)$

Schlüsselwert 79 (Splitten)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

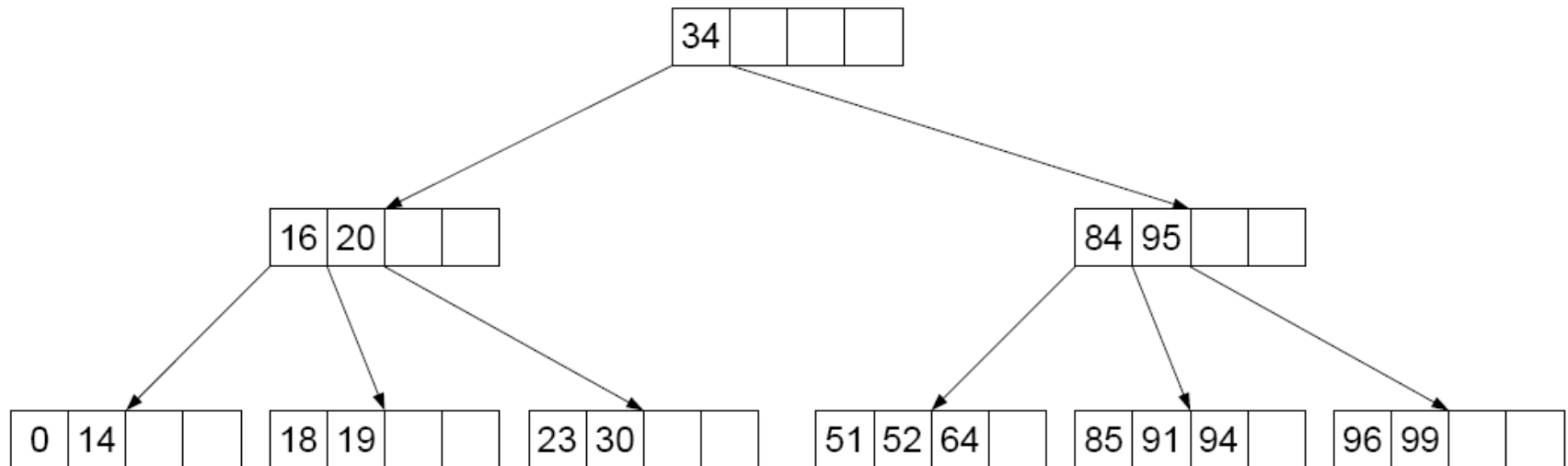
b) **Löschen:** Gegeben ein Baum der Klasse $\tau(2,3)$



Aufgabe 8: Einfügen und Löschen in B-Bäumen

b) **Löschen:** Gegeben ein Baum der Klasse $\tau(2,3)$

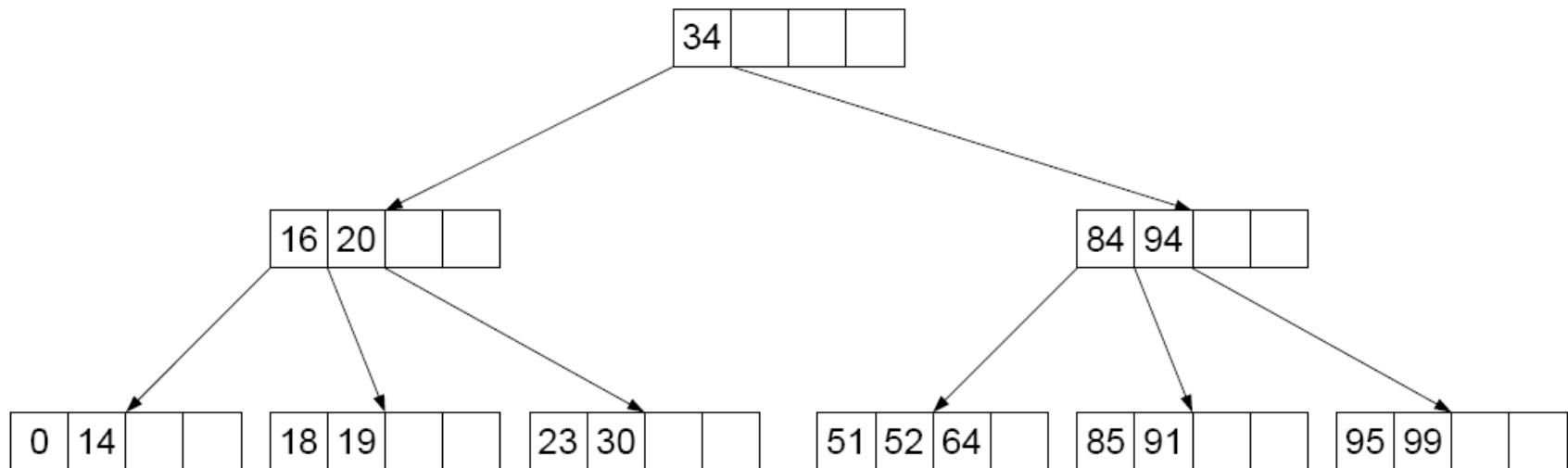
Schlüsselwert 17 (Ausgleichen)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

b) **Löschen:** Gegeben ein Baum der Klasse $\tau(2,3)$

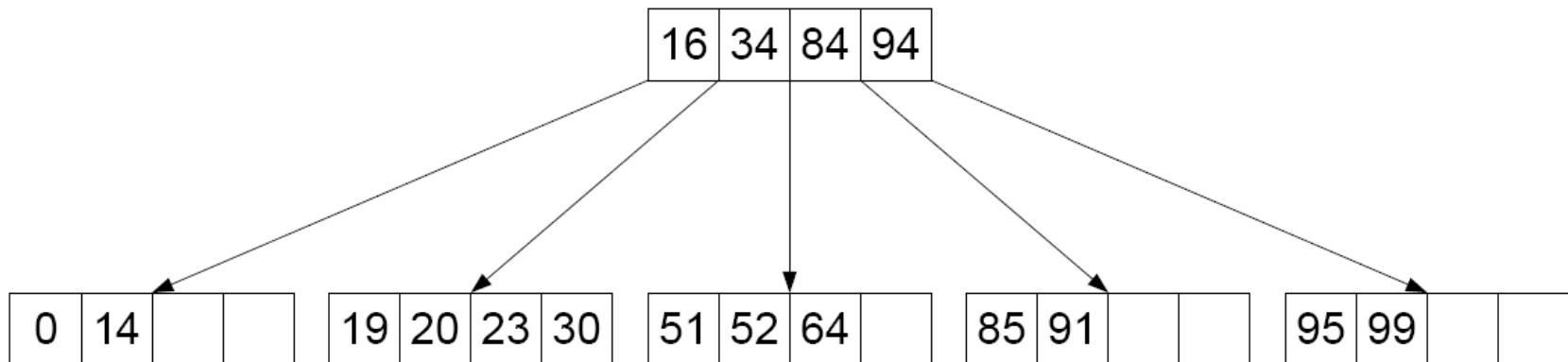
Schlüsselwert 96 (Ausgleichen)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

b) **Löschen:** Gegeben ein Baum der Klasse $\tau(2,3)$

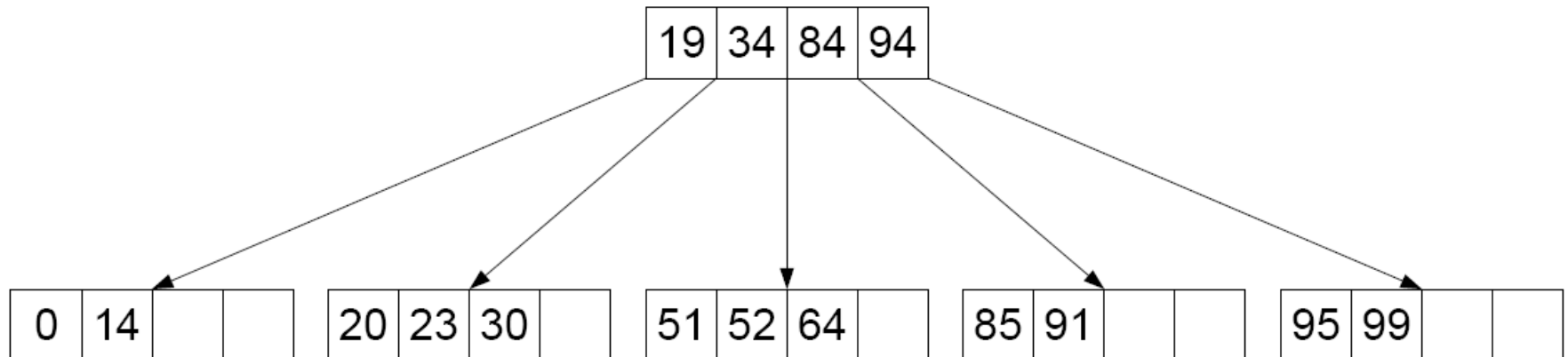
Schlüsselwert 18 (Mischen)



Aufgabe 8: Einfügen und Löschen in B-Bäumen

b) **Löschen:** Gegeben ein Baum der Klasse $\tau(2,3)$

Schlüsselwert 16 (Ausgleichen)

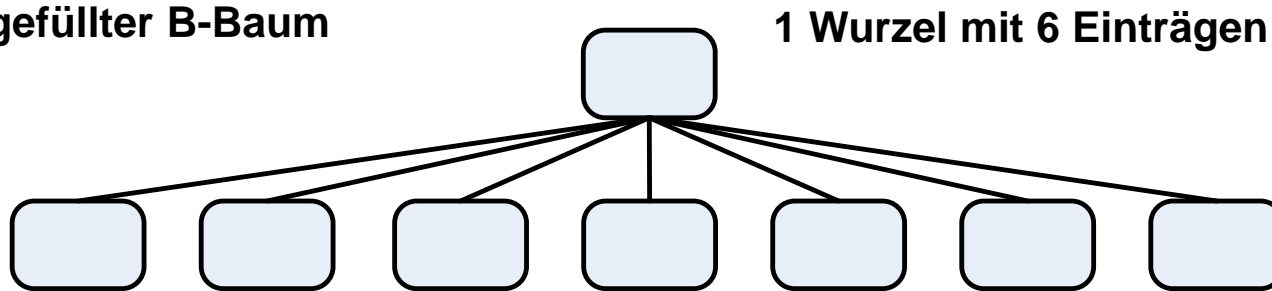


Aufgabe 9: Berechnungen in B- und B*-Bäumen

a) Gegeben seien ein maximal gefüllter B-Baum der Klasse $\tau(3,2)$ und ein minimal gefüllter B*-Baum der Klasse $\tau(2, 3, 3)$.

(1) Wie groß ist die Differenz der Anzahl an Datensätzen der jeweiligen Bäume?

Maximal gefüllter B-Baum



1 Wurzel mit 6 Einträgen

**7 Blattknoten
mit je 6 Einträgen**

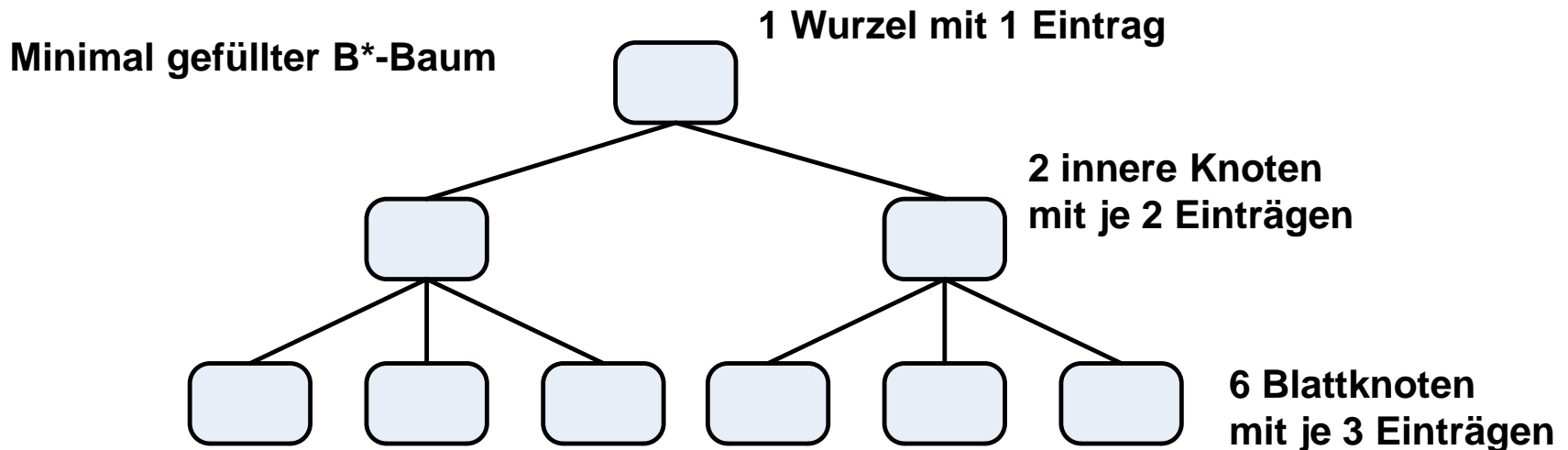
Datensätze befinden sich in **ALLEN** Knoten

$\Rightarrow 8 * 6 = 48$ Datensätze

Aufgabe 9: Berechnungen in B- und B*-Bäumen

a) Gegeben seien ein maximal gefüllter B-Baum der Klasse $\tau(3,2)$ und ein minimal gefüllter B*-Baum der Klasse $\tau(2, 3, 3)$.

(1) Wie groß ist die Differenz der Anzahl an Datensätzen der jeweiligen Bäume?



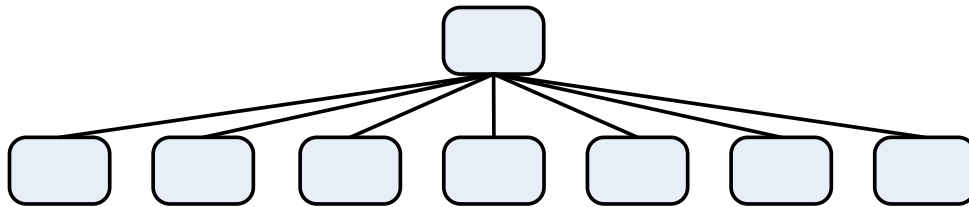
Datensätze befinden sich in **NUR** in den Blattknoten
 $\Rightarrow 6 * 3 = 18$ Datensätze

Aufgabe 9: Berechnungen in B- und B*-Bäumen

a) Gegeben seien ein maximal gefüllter B-Baum der Klasse $\tau(3,2)$ und ein minimal gefüllter B*-Baum der Klasse $\tau(2, 3, 3)$.

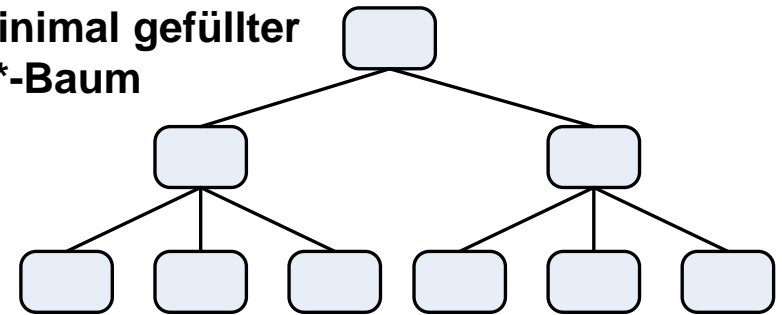
(1) Wie groß ist die Differenz der Anzahl an Datensätzen der jeweiligen Bäume?

**Maximal gefüllter
B-Baum**



48 Datensätze

**Minimal gefüllter
B*-Baum**



18 Datensätze

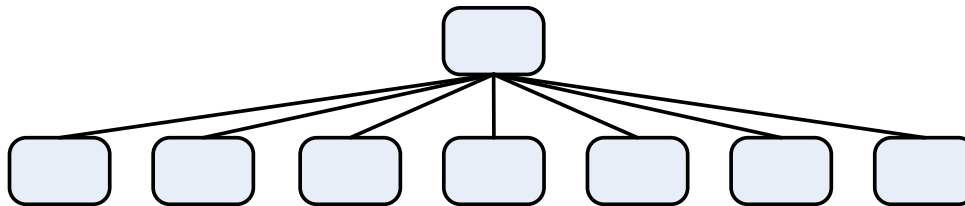
Der Unterschied beträgt **30 Datensätze**

Aufgabe 9: Berechnungen in B- und B*-Bäumen

a) Gegeben seien ein maximal gefüllter B-Baum der Klasse $\tau(3,2)$ und ein minimal gefüllter B*-Baum der Klasse $\tau(2, 3, 3)$.

(2) Wenn man sequentiell alle Datensätze aus einem der beiden Bäume lesen möchte, wieviele verschiedene Knoten (Seiten) muss man in dem jeweiligen Baum lesen?

**Maximal gefüllter
B-Baum**

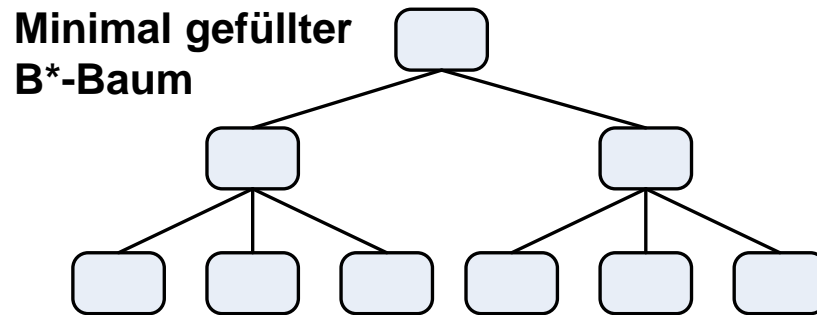


**Man muss ALLE Knoten lesen
⇒ 8 Knoten**

Aufgabe 9: Berechnungen in B- und B*-Bäumen

a) Gegeben seien ein maximal gefüllter B-Baum der Klasse $\tau(3,2)$ und ein minimal gefüllter B*-Baum der Klasse $\tau(2, 3, 3)$.

(2) Wenn man sequentiell alle Datensätze aus einem der beiden Bäume lesen möchte, wieviele verschiedene Knoten (Seiten) muss man in dem jeweiligen Baum lesen?



Man muss **ALLE** Blattknoten lesen.
Dafür muss man einmal den Baum der Höhe nach durchlaufen

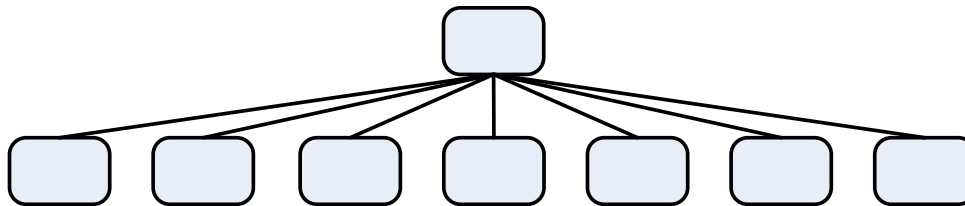
⇒ **6 Blätter + 2 innere Knoten = 8 Knoten**

Aufgabe 9: Berechnungen in B- und B*-Bäumen

a) Gegeben seien ein maximal gefüllter B-Baum der Klasse $\tau(3,2)$ und ein minimal gefüllter B*-Baum der Klasse $\tau(2, 3, 3)$.

(3) Wenn man wahlfrei einen Datensatz aus einem der beiden Bäume lesen möchte, wieviel Knoten (Seiten) muss man im jeweiligen Baum maximal lesen?

**Maximal gefüllter
B-Baum**

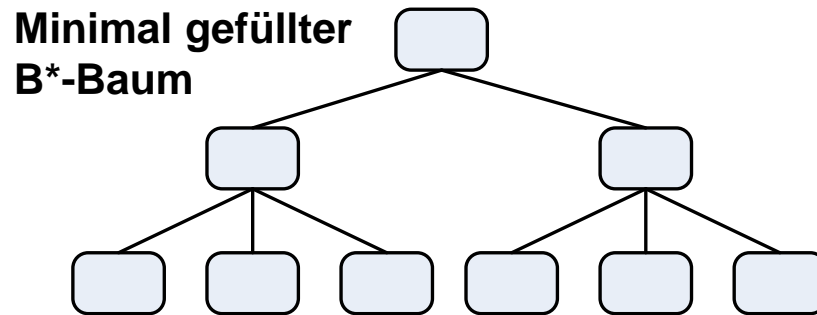


Man muss den Baum im schlechtesten Fall von Wurzel zu Blatt durchlaufen
⇒ **2 Knoten**

Aufgabe 9: Berechnungen in B- und B*-Bäumen

a) Gegeben seien ein maximal gefüllter B-Baum der Klasse $\tau(3,2)$ und ein minimal gefüllter B*-Baum der Klasse $\tau(2, 3, 3)$.

(3) Wenn man wahlfrei einen Datensatz aus einem der beiden Bäume lesen möchte, wieviel Knoten (Seiten) muss man im jeweiligen Baum maximal lesen?



Man muss den Baum immer von Wurzel zu Blatt durchlaufen
⇒ **3 Knoten**

Aufgabe 9: Berechnungen in B- und B*-Bäumen

b) Gegeben sei ein **B*-Baum** der Klasse (k, k^*, h) . Innerhalb des zugehörigen (fiktiven) DBS wird vereinfachend eine Seitengröße von $800B$ ($L=800B$) angenommen. Die weiteren Kenngrößen sind dabei wie folgt festgelegt:

$$\begin{array}{ll} I_M = 4B & I_D = 160B \\ I_K = 5B & I_P = 6B \end{array}$$

Die inneren Baum- und Blattknoten entsprechen jeweils genau einer Seite. Wie groß sind demnach k und k^* ?

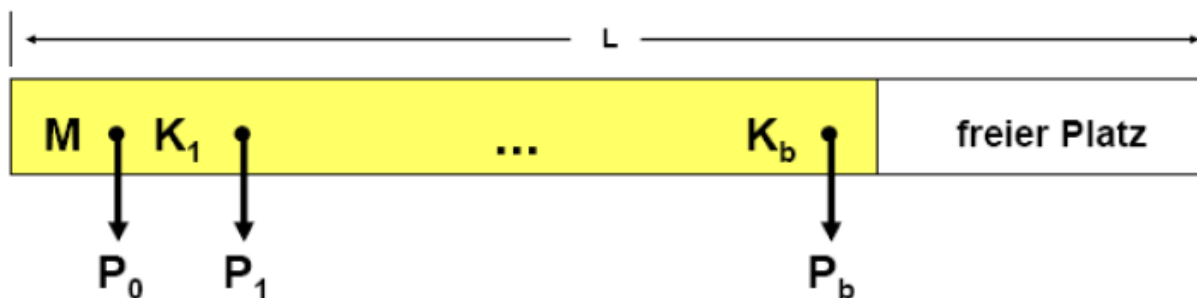
Aufgabe 9: Berechnungen in B- und B*-Bäumen

b) Gegeben sei ein **B*-Baum** der Klasse (k, k^*, h) . Innerhalb des zugehörigen (fiktiven) DBS wird vereinfachend eine Seitengröße von $800B$ ($L=800B$) angenommen. Die weiteren Kenngrößen sind dabei wie folgt festgelegt:

$$\begin{array}{ll} I_M = 4B & I_D = 160B \\ I_K = 5B & I_P = 6B \end{array}$$

Die inneren Baum- und Blattknoten entsprechen jeweils genau einer Seite. Wie groß sind demnach k und k^* ?

innerer Knoten

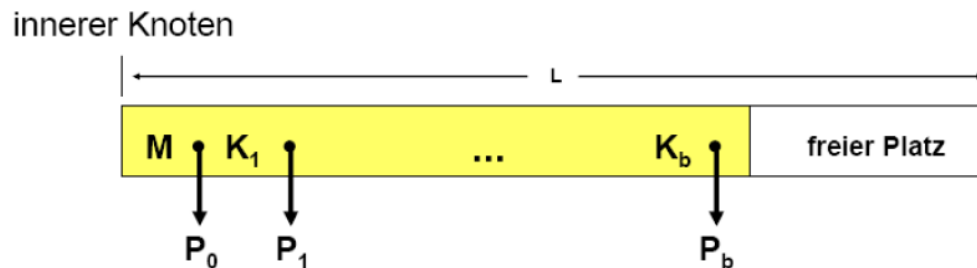


Aufgabe 9: Berechnungen in B- und B*-Bäumen

b) Gegeben sei ein **B*-Baum** der Klasse (k, k^*, h) . Innerhalb des zugehörigen (fiktiven) DBS wird vereinfachend eine Seitengröße von $800B$ ($L=800B$) angenommen. Die weiteren Kenngrößen sind dabei wie folgt festgelegt:

$$\begin{array}{ll} l_M = 4B & l_D = 160B \\ l_K = 5B & l_P = 6B \end{array}$$

Die inneren Baum- und Blattknoten entsprechen jeweils genau einer Seite. Wie groß sind demnach k und k^* ?



$$k = \left\lfloor \frac{L - l_M - l_P}{2 \cdot (l_K + l_P)} \right\rfloor = \left\lfloor \frac{800B - 4B - 6B}{2 \cdot (5B + 6B)} \right\rfloor = \left\lfloor \frac{790B}{22B} \right\rfloor \simeq [35, 9] = 35$$

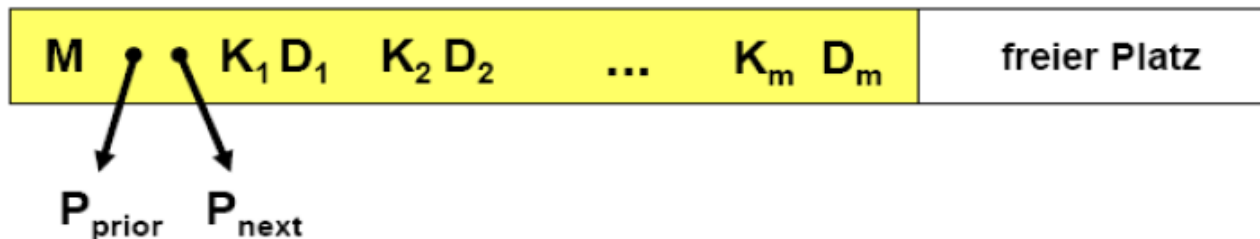
Aufgabe 9: Berechnungen in B- und B*-Bäumen

b) Gegeben sei ein **B*-Baum** der Klasse (k, k^*, h) . Innerhalb des zugehörigen (fiktiven) DBS wird vereinfachend eine Seitengröße von $800B$ ($L=800B$) angenommen. Die weiteren Kenngrößen sind dabei wie folgt festgelegt:

$$\begin{array}{ll} I_M = 4B & I_D = 160B \\ I_K = 5B & I_P = 6B \end{array}$$

Die inneren Baum- und Blattknoten entsprechen jeweils genau einer Seite. Wie groß sind demnach k und k^* ?

Blattknoten



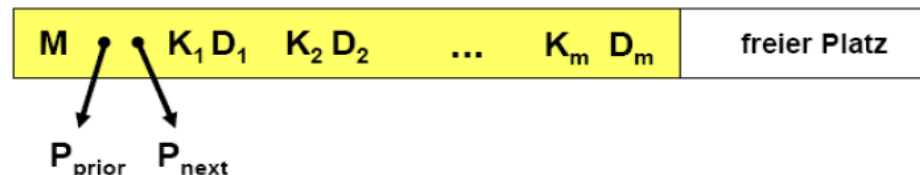
Aufgabe 9: Berechnungen in B- und B*-Bäumen

b) Gegeben sei ein **B*-Baum** der Klasse (k, k^*, h) . Innerhalb des zugehörigen (fiktiven) DBS wird vereinfachend eine Seitengröße von $800B$ ($L=800B$) angenommen. Die weiteren Kenngrößen sind dabei wie folgt festgelegt:

$$\begin{array}{ll} l_M = 4B & l_D = 160B \\ l_K = 5B & l_P = 6B \end{array}$$

Die inneren Baum- und Blattknoten entsprechen jeweils genau einer Seite. Wie groß sind demnach k und k^* ?

Blattknoten



$$k^* = \left\lfloor \frac{L - l_M - 2 \cdot l_P}{2 \cdot (l_K + l_D)} \right\rfloor = \left\lfloor \frac{800B - 4B - 12B}{2 \cdot (5B + 160B)} \right\rfloor = \left\lfloor \frac{784B}{330B} \right\rfloor \simeq [2, 376] = 2$$

Zusatzinfo: Doppelter Überlauf (Splitfaktor $m=2$)

- Split von 2 auf 3 Knoten wenn zwei Nachbarknoten voll sind
- Verschiedene Implementationen/Variationen denkbar
- Variation der Klausur:
 1. Lesen des rechten Nachbarn
 2. Entweder Ausgleichen oder gemeinsamer Split

Ausnahme: Der ganz rechte Knoten einer Ebene hat keinen rechten Nachbarn => Verwendung des linken Nachbarns
- Bei dreifachem Überlauf (Splitfaktor $m=3$) werden immer beide Nachbarknoten angeschaut. Reihenfolge erst links, dann rechts.
 - Implementationsspezifisch: Behandlung der Knoten mit nur einem Nachbarn (kommt in der Klausur nicht vor)

Aufgabe 10: Referenzielle Aktionen (1)

- 1. Welche Anforderung erfüllt ein (bzgl. der referenziellen Aktionen) sicheres Schema?**

Aufgabe 10: Referenzielle Aktionen (1)

1. Welche Anforderung erfüllt ein (bzgl. der referenziellen Aktionen) sicheres Schema?

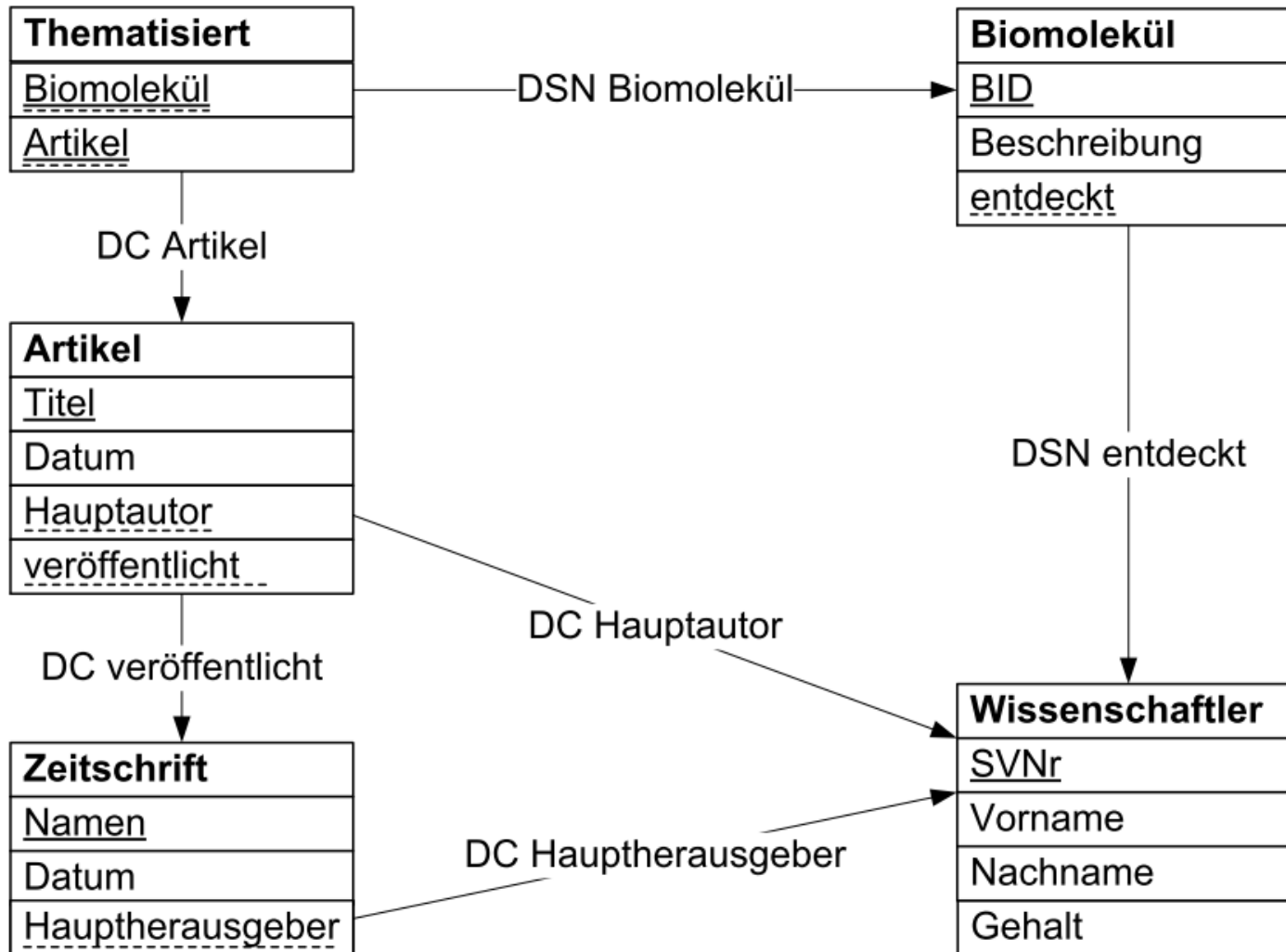
→ Bei einem sicheren Schema ist das Ergebnis einer Änderungsoperation unabhängig von der Reihenfolge, in der die referenziellen Aktionen ausgeführt werden; es treten also **keine reihenfolgeabhängigen Ergebnisse** auf.

Aufgabe 10: Referenzielle Aktionen (2)

2. Ist das vorliegende Schema sicher?

→ Ja.

Aufgabe 10: Referenzielle Aktionen (2)



Aufgabe 10: Referenzielle Aktionen (3)

3. Geben Sie eine Folge von SQL-DDL-Anweisungen an, welche die Schemaänderung realisiert und dabei die Löschung der Relation selbst vermeidet.

```
CREATE TABLE Artikel (  
    Titel          VARCHAR(50) PRIMARY KEY,  
    Datum          DATE          NOT NULL,  
    Hauptautor     INTEGER       NOT NULL,  
    veröffentlicht VARCHAR(50) NOT NULL,  
    CONSTRAINT fkHauptautor FOREIGN KEY (Hauptautor)  
        REFERENCES Wissenschaftler(SVNr) ON DELETE CASCADE ,  
    CONSTRAINT fkVeröffentlicht FOREIGN KEY (veröffentlicht)  
        REFERENCES Zeitschrift(Name) ON DELETE CASCADE  
);
```

Aufgabe 10: Referenzielle Aktionen (3)

3. Geben Sie eine Folge von SQL-DDL-Anweisungen an, welche die Schemaänderung realisiert und dabei die Löschung der Relation selbst vermeidet.



```
ALTER TABLE Artikel DROP CONSTRAINT fkVeröffentlicht;  
ALTER TABLE Artikel ADD CONSTRAINT fkVeröffentlicht  
    FOREIGN KEY (veröffentlicht)  
    REFERENCES Zeitschrift(Name) ON DELETE RESTRICT;
```

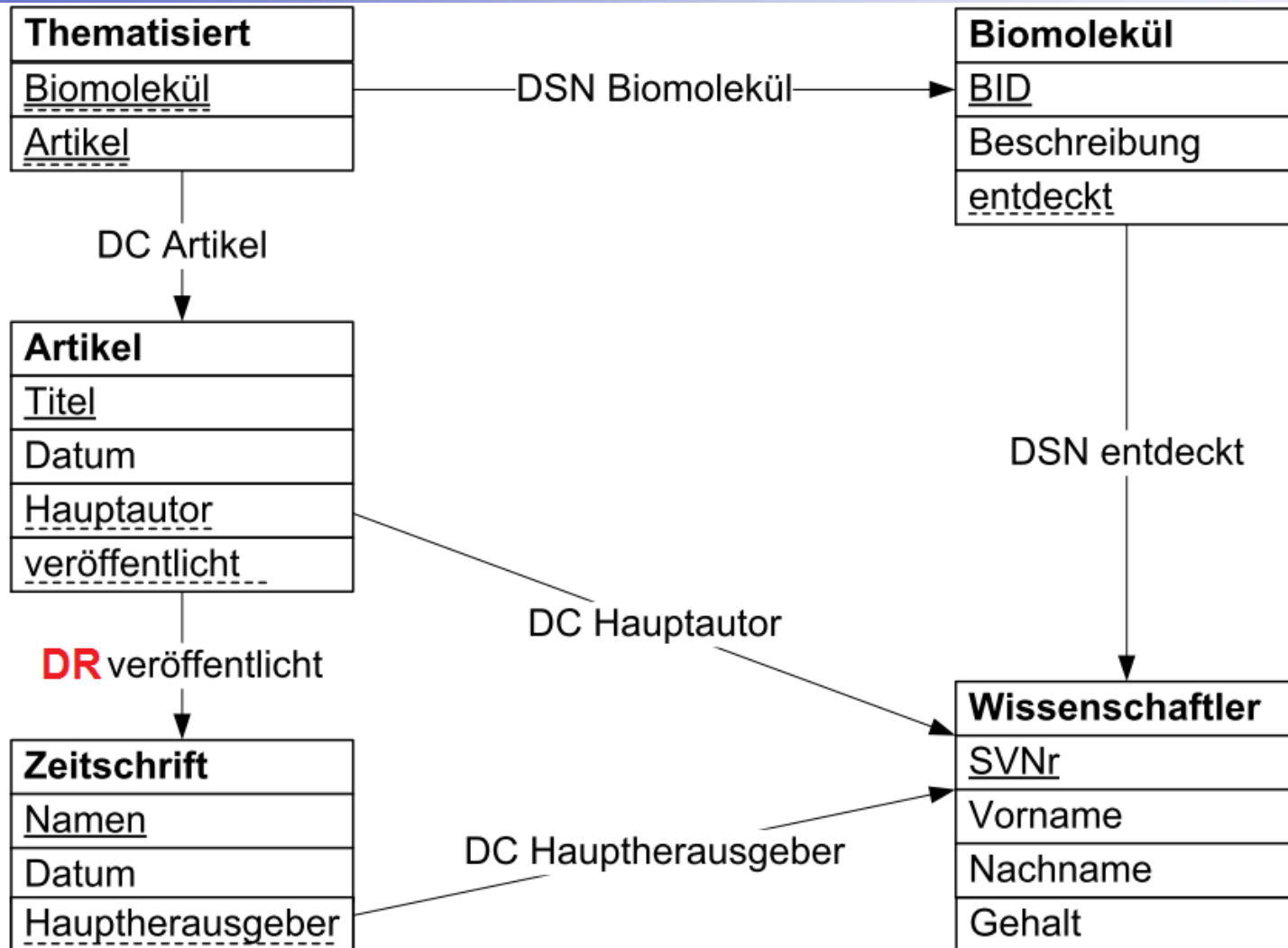
Aufgabe 10: Referenzielle Aktionen (4)

4. Ist das geänderte Schema sicher?

→ Nein

Reihenfolgeabhängige Ergebnisse treten auf, wenn der zu löschende Wissenschaftler Hauptherausgeber einer Zeitschrift ist, die **ausschließlich** Artikel veröffentlicht hat bei denen dieser Wissenschaftler Hauptautor ist.

Aufgabe 10: Referenzielle Aktionen (4)



WE WANT YOU!



GDB-Übungsgruppenleiter für das WS 2019/2020 gesucht!

[email an panse@informatik.uni-hamburg.de](mailto:panse@informatik.uni-hamburg.de)