



Modul
Grundlagen von Datenbanken
Wintersemester 2018/2019

Probeklausur
???.???.2019, ?? Uhr, ??

Musterlösung

Aufgabe	1	2	3	4	5	6	7	8	9	10	Gesamt
Punkte	18	12	12	14	12	6	8	11	15	12	120
Ergebnis											

Rechtsmittelbelehrung:

Gegen die Bewertung dieser Prüfungsleistung kann innerhalb eines Monats nach ihrer Bekanntgabe Widerspruch erhoben werden. In diesem Zeitraum kann die Bewertung der Klausur eingesehen werden. Der Widerspruch ist schriftlich oder zur Niederschrift beim Vorsitzenden des B.Sc./M.Sc.-Prüfungsausschusses einzulegen. Es wird darauf hingewiesen, dass ein erfolgloses Widerspruchsverfahren kostenpflichtig ist.

Bearbeitungshinweise:

- Die Bearbeitungsdauer der Klausur beträgt insgesamt 120 Minuten.
- Es werden maximal 120 Punkte vergeben. Dabei entsprechen die den einzelnen Aufgaben zugeordneten Punkte den jeweils vorgesehenen Bearbeitungszeiten (in Minuten). Zum Bestehen der Klausur sind 50% der erreichbaren Punkte hinreichend.
- Überprüfen Sie sofort nach Erhalt der Klausur die Vollständigkeit der Unterlagen (25 Seiten).
- Bitte lassen Sie die Klausur zusammengeheftet. Notieren Sie alle Lösungen direkt auf den Aufgabenblättern.
- Verwenden Sie kein eigenes Papier, sondern nur die zur Verfügung gestellten Blätter.
- Bitte schreiben Sie mit blauem oder schwarzem Schreiber – Bleistift und Rotstift sind nicht erlaubt.
- Schreiben Sie auf jedes Blatt Ihren Namen und Ihre Matrikelnummer.
- Es sind keine Hilfsmittel zugelassen.
- Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis auf den Tisch.
- Schalten Sie Ihr Mobiltelefon aus!

Viel Erfolg!

Matrikel-Nr:

Name:

Hiermit versichere ich, dass ich die Klausur selbstständig und unter ausschließlicher Nutzung der erlaubten Hilfsmittel bearbeitet und die Rechtsmittelbelehrung zur Kenntnis genommen habe.

Unterschrift: _____

**Blättern Sie erst weiter, wenn die
Bearbeitungszeit offiziell begonnen hat.**

Aufgabe 1: Informationsmodellierung mit dem Entity-Relationship-Modell

[18 P.]

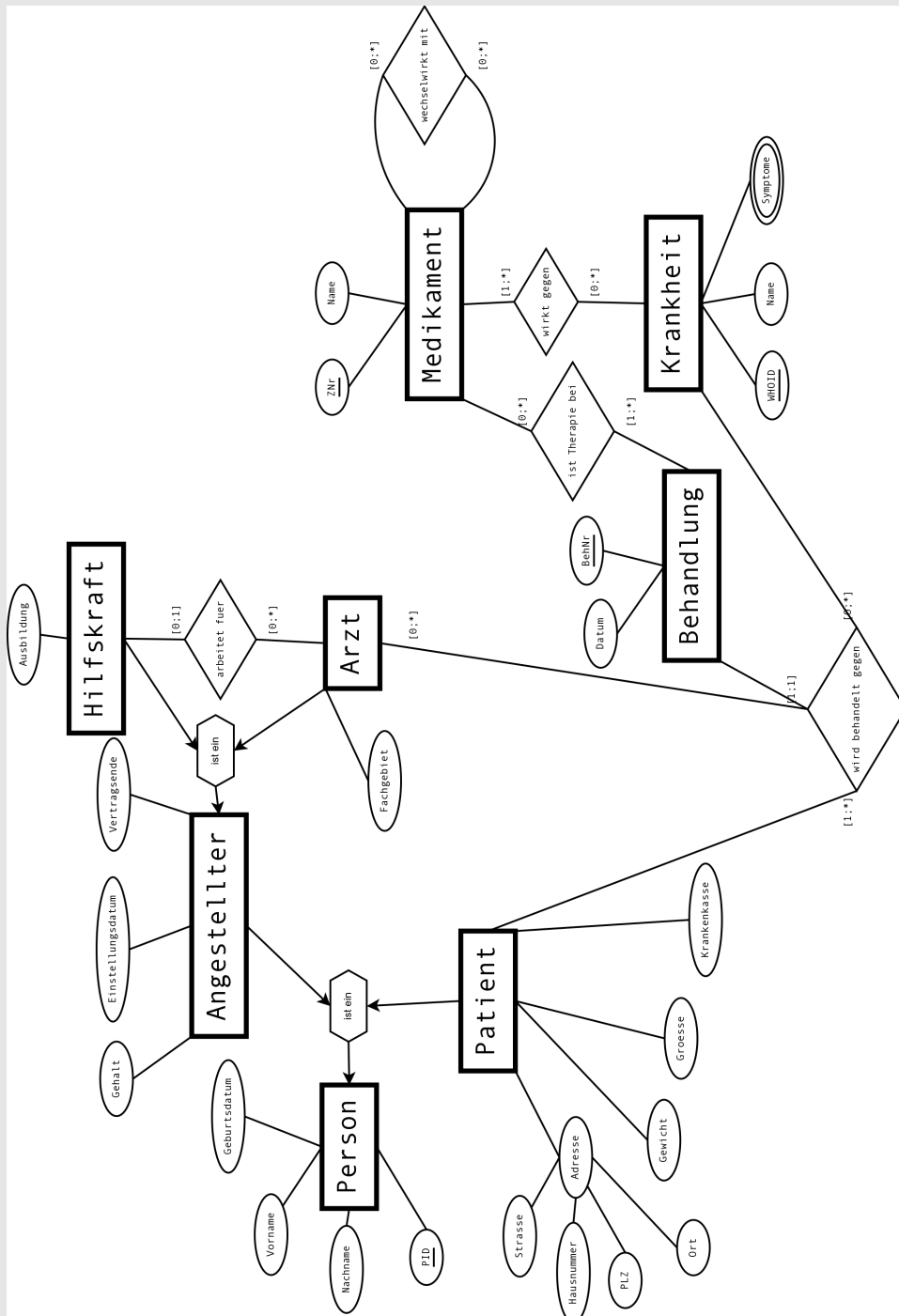
Erfassen Sie die im Folgenden beschriebenen Informationsstrukturen (Ausschnitt einer Anwendung zur Verwaltung der Patienten und deren Behandlungen in einem Ärztehaus) in einem ER-Diagramm. Beziehen Sie sich dabei genau auf die gegebene Beschreibung, ohne weiteres Wissen zu möglicherweise ähnlichen Anwendungsbereichen einfließen zu lassen. Markieren Sie in Ihrem Entwurf Primärschlüssel durch Unterstreichung und konkretisieren Sie die Abbildungstypen durch Kardinalitätsrestriktionen (Notation: [min,max]). Verwenden Sie unbedingt die aus der Vorlesung bekannte Notation. Andere Notationen werden nicht gewertet! Ausnahme: Die Zuordnung bei 1:n Abbildungstypen bleibt Ihnen überlassen, muss aber eindeutig als solche markiert werden (z.B. durch ein ausformuliertes Beispiel). Benutzen Sie möglichst wenige Entitäten (Ausnahme Vererbung).

Etwaige (Existenz-)Abhängigkeiten sollen nicht modelliert werden.

Es werden Angestellte und Patienten sowie deren Behandlungen betrachtet. Die Angestellten untergliedern sich in Hilfskräfte, welche eine Ausbildung haben, und Ärzte, welche ein Fachgebiet haben. Beiden Angestellten-Gruppen gemein ist, dass für sie ein Gehalt, ein Einstellungsdatum sowie ein Vertragsende festgelegt ist. Angestellte und Patienten haben personenspezifische Eigenschaften: eine eindeutige Identifikationsnummer (PID), einen Familiennamen, einen Vornamen und ein Geburtsdatum. Jede Hilfskraft kann (maximal) einem Arzt zugeordnet sein. Einem Arzt können wiederum mehrere Hilfskräfte zugeordnet werden. Für Patienten werden neben den bereits genannten personenspezifischen Eigenschaften noch die Größe, das Gewicht, die Adresse, welche sich aus einer Straße, einer Hausnummer, einer PLZ und einem Ort zusammensetzt, und der Name der Krankenkasse gespeichert.

Eine Krankheit hat einen eindeutigen Identifikationscode (WHOID), einen Namen und eine Reihe von Symptomen. Ein Medikament hat eine eindeutige Zulassungsnummer (ZNr) und einen Namen. Ein Medikament kann gegen eine oder mehrere Krankheiten einsetzbar sein. Zwar kann es für jede Krankheit mehrere geeignete Medikamente geben, es ist aber auch möglich, dass eine Krankheit nicht durch Medikamente behandelbar ist. Zu beachten ist, dass ein Medikament Wechselwirkungen mit anderen Medikamenten haben kann, aber nicht haben muss.

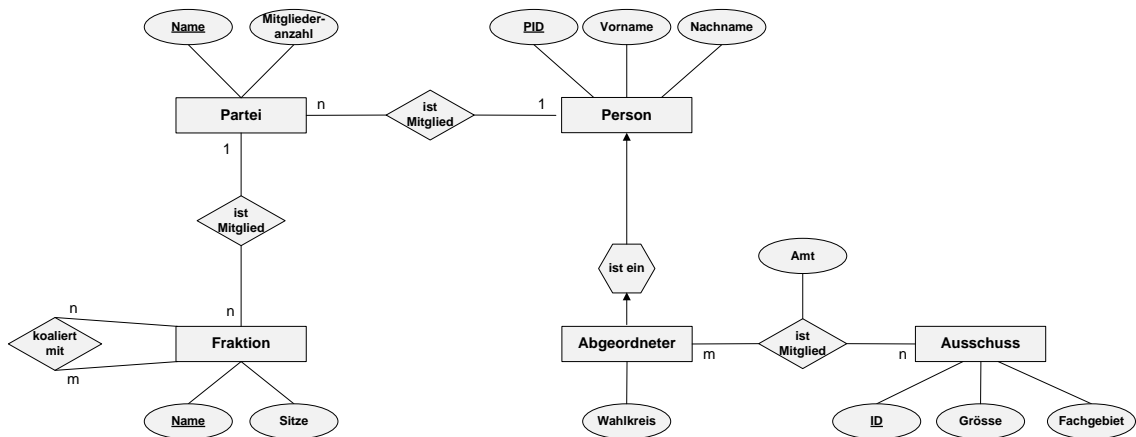
Eine Behandlung ist durch eine eindeutige Behandlungsnummer (BehNr) gekennzeichnet. Zudem besitzt sie ein Datum an dem sie stattfand. Eine Behandlung betrifft immer genau einen Patienten, einen Arzt und eine Krankheit. Ein Patient kann mehrmals (aber mindestens einmal) behandelt werden, ein Arzt kann mehrmals behandeln und eine Krankheit kann ebenfalls mehrmals behandelt werden. In einer Behandlung können beliebig viele (aber mindestens ein) Medikament angewendet werden. Ein Medikament kann in beliebig vielen Behandlungen Anwendung finden.



Aufgabe 2: Abbildung eines ER-Diagramms auf das relationale Datenmodell

[12 P.]

Gegeben sei folgendes ER-Diagramm:



Entwickeln Sie aus dem dargestellten ER-Diagramm ein entsprechendes relationales Datenbankschema anhand der in der Vorlesung erläuterten Abbildungsregeln. Stellen Sie sicher, dass Ihr Datenbankschema die minimale Anzahl von Relationen aufweist. Stellen Sie das resultierende DB-Schema dar, in dem Sie die notwendigen Relationenschemata in der Form

$$\text{Relationenname}(\text{Attributname}_1, \text{Attributname}_2, \dots, \text{Attributname}_n)$$

anführen und dabei jeweils den Primärschlüssel unterstreichen. Gegebenenfalls enthaltene Fremdschlüssel sind zu 'unterstricheln' und durch die aus den Übungen bekannte Pfeilnotation zu spezifizieren:

$$\underline{\text{Attr}_i} \rightarrow \underline{\text{Rel}_b}.\underline{\text{Attr}_j}$$

Hinweis: Die dargestellten (1:n)-Beziehungen haben folgende Semantik:

- (i) Eine Person kann nur in einer Partei Mitglied sein; eine Partei kann mehrere Mitglieder haben.
- (ii) Eine Partei ist Teil höchstens einer Fraktion; einer Fraktion können mehrere Parteien angehören.

Hinweis: Die dargestellte (m:n)-Beziehung 'istMitglied' zwischen den Entity-Typen Abgeordneter und Ausschuss hat folgende Semantik:

- (iv) Ein Abgeordneter kann in einem Ausschuss maximal ein Amt belegen.

Matrikel-Nr:

Name:

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: Relationenalgebra und SQL

[12 P.]

Gegeben seien die folgenden Relationenschemata:

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)

Benutzen Sie zur Lösung der folgenden Aufgaben ausschließlich die in der Vorlesung vorgestellten Operatoren der Relationenalgebra!

[3 P.]

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

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

[3 P.]

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

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

oder

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

[3 P.]

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

$$\pi_{Name}((Person \bowtie_{PNr=Bieter} Gebot \bowtie_{Artikel=ANr} Artikel) \bowtie_{Werber=PNr \wedge Verkäufer=PNr} (\pi_{PNr}(Person)))$$

```

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


Matrikel-Nr:

Name:

- d) Der kleinste in der Datenbank gespeicherte Startpreis eines Artikels beträgt 10 € und der größte 159 €. Bestimmen Sie die Selektivität der Selektion $\sigma_{Startpreis \geq 46}$ unter der Annahme einer Gleichverteilung: [3 P.]

$$sel(Startpreis \geq 46) = \frac{159-45}{159-9} = \frac{114}{150} = \frac{38}{50} = 0.76$$

In der Literatur wird vereinfacht häufig auch folgende Berechnung verwendet:

$$sel(Startpreis \geq 46) = \frac{\maxSel - \minSel}{\maxDB - \minDB} = \frac{159-46}{159-10} = \frac{113}{149} = 0.758$$

Aufgabe 4: SQL

[14 P.]

Gegeben seien die Relationenschemata aus Aufgabe 3:

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)

Formulieren Sie für die in den nachfolgenden Teilaufgaben angeführten, natürlichsprachlich formulierten Anfragen entsprechende SQL-Anweisungen. **Verwenden Sie den in der Vorlesung verwendeten SQL-Standard.** Das SQL-Schlüsselwort JOIN darf dabei nicht verwendet werden.

[2 P.]

- a) 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;
```

[3 P.]

- b) 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);
```

Matrikel-Nr:

Name:

- c) Die Namen und EMail aller Personen, die noch nie geboten haben. [3 P.]

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

oder

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

- d) Die ANr, die Bezeichnung und den Startpreis jedes Artikels, dessen Verkäufer von einer Person mit dem Namen 'Bond' erworben wurde. [3 P.]

```
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';
```

- e) Die PNr und das durchschnittliche Alter der von ihm erworbenen Personen für jeden Werber, der maximal 10 Personen erworben hat. [3 P.]

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

Aufgabe 5: Änderungsoperationen auf Sichten

[12 P.]

Ausgehend von der Basisrelation

Musiker (PNr, Name, Alter, Genre, Vermögen, Status)

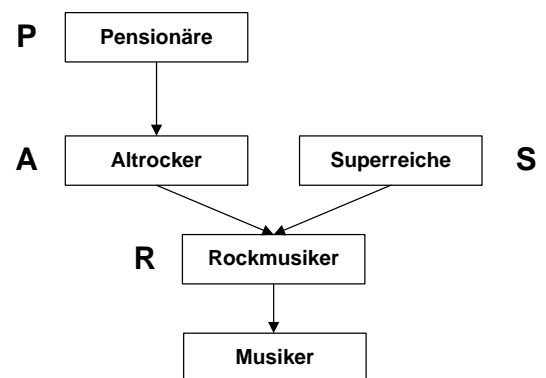
seien folgende Sichten definiert:

```
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;
```



Da diese Sichten Änderungsoperationen erlauben, **könnten** sie alle mit dem Zusatz `WITH CASCADED CHECK OPTION` definiert werden. In der Tabelle auf Seite 11 unten sind verschiedene Konfigurationen der Sichten mit bzw. ohne Check-Option dargestellt. Ein Spaltenwert 'CASC' bedeutet dabei, dass die Option `WITH CASCADED CHECK OPTION` für die zugehörige Sicht gesetzt ist; ein Spaltenwert '-' bedeutet, dass die zugehörige Sicht ohne `CHECK OPTION` definiert wurde.

Betrachten Sie folgende Änderungsoperationen:

- ```
UPDATE Rockmusiker
SET Alter = 49
WHERE Alter = 51
AND Status = 'aktiv'
AND Vermögen < 50M;
```
- ```
INSERT INTO Superreiche
VALUES (12, 'Madonna', 50, 'Pop', 500M, 'aktiv');
```
- ```
UPDATE Rockmusiker
SET Status = 'aktiv'
WHERE Name = 'Beatles'
AND Alter > 60
AND Vermögen > 200M;
```

Matrikel-Nr:

Name:

---

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

Markieren Sie im unteren Teil der untenstehenden Tabelle, welche dieser Änderungsoperationen von einem Datenbanksystem in den verschiedenen Konfigurationen erlaubt bzw. zurückgewiesen werden. Die Lösungen zur ersten Änderungsoperation sind als Hilfestellung bereits angegeben. Für die Fälle, in denen die Änderung bzw. das Einfügen zulässig ist, geben Sie in der Tabelle außerdem an, in welchen Sichten **auf jeden Fall alle** geänderten/eingefügten Tupel nach Abschluss der Operation sichtbar werden. Verwenden Sie dabei die folgende Notation:

nein                    Operation wird für die Konfiguration zurückgewiesen  
ja/Sicht-Liste        Operation ist für die Konfiguration zulässig.  
                          Alle geänderten/eingefügten Tupel sind anschließend in  
                          den in der Sicht-Liste aufgeführten Sichten sichtbar.

Zur Vereinfachung sind anstelle der kompletten Sichtennamen die folgenden Abkürzungen zu verwenden:

Rockmusiker    R  
Altrockler      A  
Pensionäre      P  
Superreiche     S

|             |          |          |          |          |
|-------------|----------|----------|----------|----------|
| Rockmusiker | -        | -        | CASC     | -        |
| Altrockler  | -        | CASC     | -        | -        |
| Pensionäre  | -        | -        | -        | CASC     |
| Superreiche | -        | -        | CASC     | -        |
| a)          | ja/R     | ja/R     | ja/R     | ja/R     |
| b)          | ja/-     | ja/-     | nein     | ja/-     |
| c)          | ja/R,A,S | ja/R,A,S | ja/R,A,S | ja/R,A,S |
| d)          | ja/R,S   | nein     | ja/R,S   | nein     |

## Aufgabe 6: Normalformenlehre

[6 P.]

Gegeben ist die Relation R mit den Attributen N, O, D, T, Z und S, sowie der Menge F an funktionalen Abhängigkeiten

$$F = \{FA_1, FA_2, FA_3, FA_4, FA_5\}.$$

Die Wertebereiche der Attribute sind alle atomar.

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

$$FA_1 = N \rightarrow S$$

$$FA_2 = T \rightarrow O, Z$$

$$FA_3 = N, D \rightarrow T$$

$$FA_4 = S, D \rightarrow N$$

$$FA_5 = Z \rightarrow O$$

[2 P.]

i) Bestimmen Sie die Schlüsselkandidaten von R bezüglich F.

Schlüsselkandidat 1: N, D  
Schlüsselkandidat 2: S, D

[1 P.]

ii) Bestimmen Sie die Nicht-Primärattribute (Nicht-Schlüsselattribute) von R bezüglich F.

Nicht-Primärattribut 1: T  
Nicht-Primärattribut 2: Z  
Nicht-Primärattribut 3: O

[3 P.]

iii) Nehmen Sie an, dass einer der in Aufgabenteil i) ermittelnden Schlüsselkandidaten als Primärschlüssel verwendet wird. In welchen Normalformen befindet sich das Relationenschema R bezüglich F? Begründen Sie Ihre Antwort, indem Sie darlegen, warum sich das Relationenschema in genau diesen Normalformen befindet und warum die anderen Normalformen nicht vorliegen.

(Anmerkung: Betrachten Sie dabei lediglich die 1., 2. und 3. Normalform.)

Matrikel-Nr:

Name:

---

Das Relationenschema befindet sich in der 2. Normalform aber nicht in der 3. Normalform, da:

- Attributwerte sind atomar  $\Rightarrow$  1. NF
- Keines der Nicht-Primärattribute T, Z oder O hängt partiell von einem der Schlüsselkandidaten ab  $\Rightarrow$  2. NF
- Die Nicht-Primärattribute O und Z hängen über T transitiv von dem Schlüsselkandidaten N,D ab  $\Rightarrow \neg$  3. NF

## Aufgabe 7: Transaktionen

[8 P.]

[6 P.]

- a) Gegeben sind die drei Objekte  $x$ ,  $y$  und  $z$ , welche von den Transaktionen  $T_1$ ,  $T_2$  und  $T_3$  gelesen bzw. geschrieben werden. Dabei bezeichnet  $w_i(x)$  den Schreibzugriff der Transaktion  $T_i$  auf das Objekt  $x$  und  $r_i(x)$  den Lesezugriff der Transaktion  $T_i$  auf  $x$ .

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.  $T_1T_2T_3$ ) oder einen auftretenden Konfliktzyklus angeben.

$$\text{i) } S_1 = r_1(y) \ r_2(z) \ w_3(y) \ w_2(x) \ r_3(y) \ r_1(y) \ r_3(x)$$

Hier ergibt sich ein Konfliktzyklus:  
 Mit  $r_1(y) \rightarrow w_3(y)$  folgt  $T_1$  vor  $T_3$   
 Mit  $w_3(y) \rightarrow r_1(y)$  folgt  $T_3$  vor  $T_1$   
 Mit  $w_2(x) \rightarrow r_3(x)$  folgt  $T_2$  vor  $T_3$

$$\text{ii) } S_2 = r_1(z) \ r_3(x) \ r_2(x) \ r_1(y) \ w_2(z) \ r_3(y) \ w_3(y)$$

mögliche seriellen Abfolgen lauten:  $T_1T_2T_3$  und  $T_1T_3T_2$

$$\text{iii) } S_3 = w_2(y) \ r_3(z) \ r_1(x) \ r_1(y) \ w_3(x) \ w_3(z)$$

mögliche serielle Abfolge lautet:  $T_2T_1T_3$

- b) **Anomalien bei unkontrolliertem Mehrbenutzerbetrieb**

[2 P.]

Betrachten Sie die zwei Transaktionen  $T_1$  und  $T_2$ , deren Operationen ohne Mehrbenutzerkontrolle in der nachfolgend angegebenen Reihenfolge ausgeführt werden. Geben Sie eine Operation von  $T_2$  zum Zeitschritt 3 an (bei Platzmangel bitte unter, **nicht** in die Tabelle schreiben), die für  $T_1$  ein non-repeatable Read verursacht, und begründen Sie kurz die Entstehung dieser Anomalie im vorliegenden Beispiel.



Matrikel-Nr:

Name:

---

| Zeitschritt | T <sub>1</sub>                                                         | T <sub>2</sub>                                                              |
|-------------|------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| 1           | SELECT SUM(Gehalt)<br>FROM Personal<br>WHERE Abteilung = 'Treadstone'; |                                                                             |
| 2           |                                                                        | SELECT AVG(Alter)<br>FROM Personal;                                         |
| 3           |                                                                        | UPDATE Personal<br>SET Gehalt = Gehalt*2<br>WHERE Abteilung = 'Treadstone'; |
| 4           | SELECT SUM(Gehalt)<br>FROM Personal<br>WHERE Abteilung = 'Treadstone'; |                                                                             |
| 5           | COMMIT;                                                                |                                                                             |
| 6           |                                                                        | COMMIT;                                                                     |

T<sub>2</sub> ändert in Zeitschritt 3 Tupel, die von T<sub>1</sub> in Zeitschritt 1 gelesen wurden. Ein erneutes Lesen dieser Tupel, wie es in Zeitschritt 4 geschieht, liefert ein anderes Ergebnis als das erste Lesen. Weitere Lösungen sind möglich, zum Beispiel eine Löschoption auf der Relation 'Personal', welche Tupel mit dem Wert 'Treadstone' im Attribut 'Abteilung' betrifft.

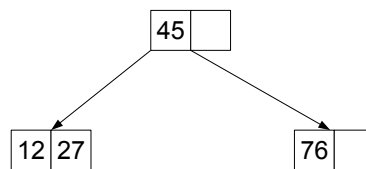
## Aufgabe 8: B-Bäume

[11 P.]

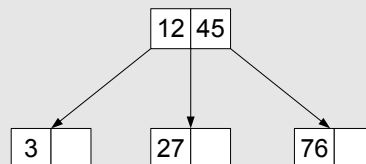
Anmerkung: Die B-Bäume sind im Folgenden stets gemäß der vereinfachten Darstellungsart aus der Vorlesung abgebildet.

[5 P.]

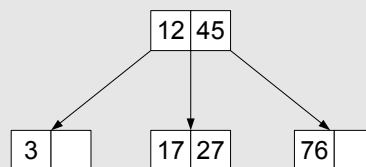
- a) Nehmen Sie den (Standard-) Split-Faktor 1 an und fügen Sie in den unten abgebildeten **B-Baum** der Klasse  $\tau(1, h)$  die Datensätze mit den Schlüsselwerten **3, 17, 83, 33 und 79** in dieser Reihenfolge ein. Zeichnen Sie den Baum nach jedem Split-Vorgang und erläutern Sie kurz die durchgeführten Maßnahmen (Splitten, einfaches Einfügen).



Schlüsselwert 3 (Splitten)



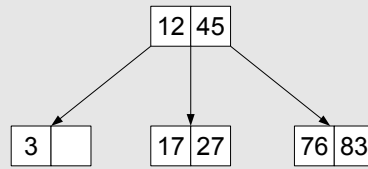
Schlüsselwert 17 (einfaches Einfügen)



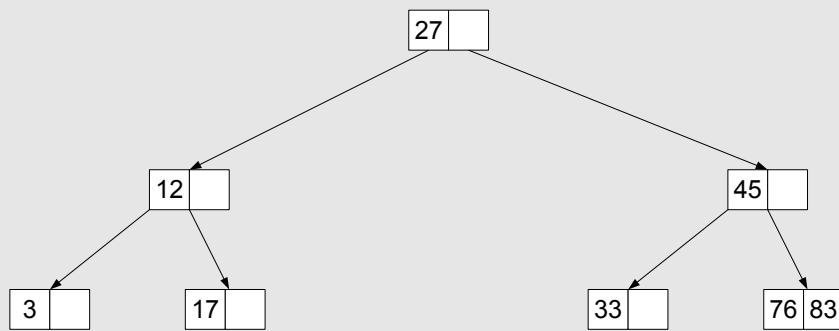
Schlüsselwert 83 (einfaches Einfügen)

Matrikel-Nr:

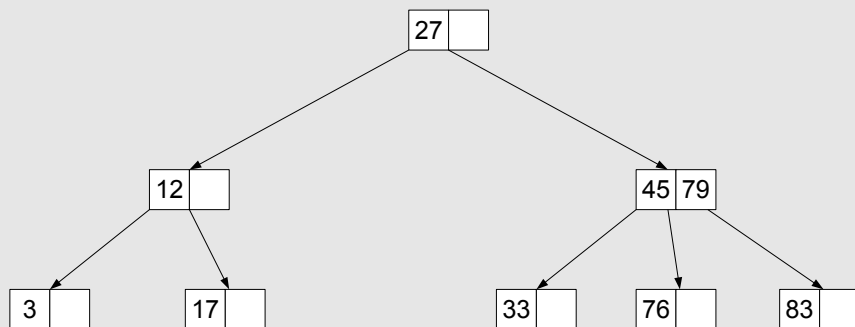
Name:



Schlüsselwert 33 (splitten)

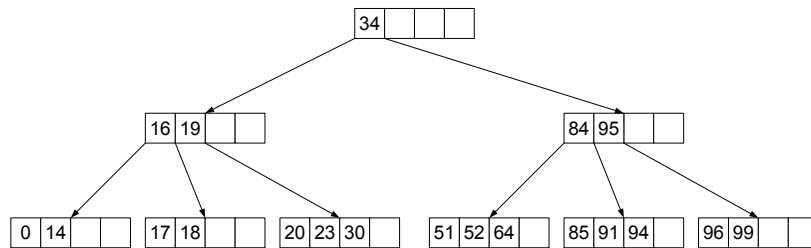


Schlüsselwert 79 (splitten)

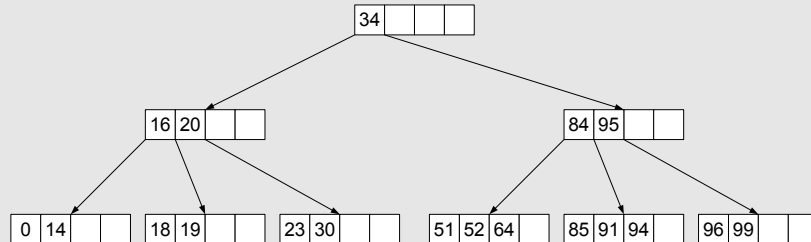


[6 P.]

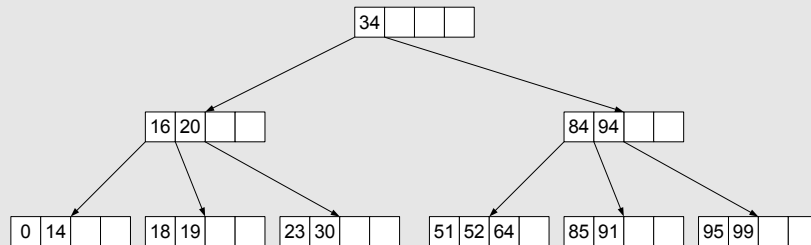
- b) Löschen Sie aus dem unten abgebildeten **B-Baum** der Klasse  $\tau(2, h)$  die Datensätze mit den Schlüsselwerten **17, 96, 18 und 16** (in dieser Reihenfolge). Zeichnen Sie den Baum nach jedem Löschvorgang und geben Sie jeweils kurz an, welche konkrete Maßnahme Sie durchgeführt haben (Mischen, Ausgleichen, einfaches Löschen). Für Ausgleichs- und Mischoperationen sollen nur direkt benachbarte Geschwisterknoten (bevorzugt der rechte) herangezogen werden.



Schlüsselwert 17 (Ausgleichen)



Schlüsselwert 96 (Ausgleichen)

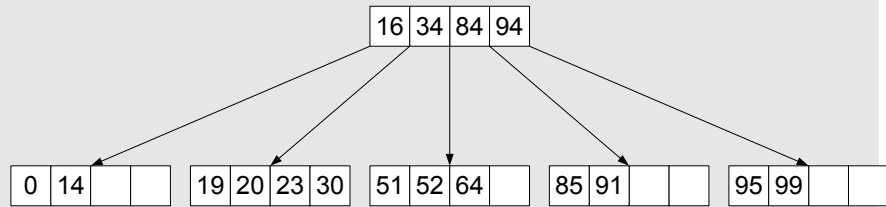


Schlüsselwert 18 (Mischen)

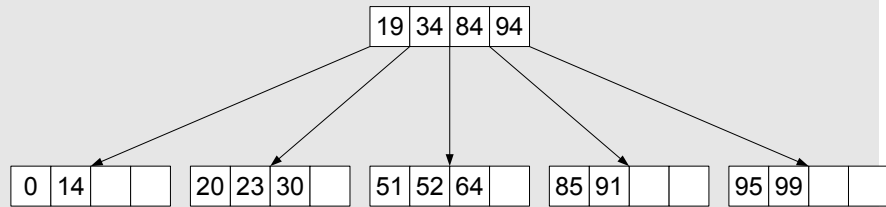
Matrikel-Nr:

Name:

---



Schlüsselwert 16 (Ausgleichen)



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

[15 P.]

[11 P.]

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

**Anmerkung:** Begründen Sie in den folgenden Teilaufgaben Ihr jeweiliges Ergebnis durch einen Rechenweg.

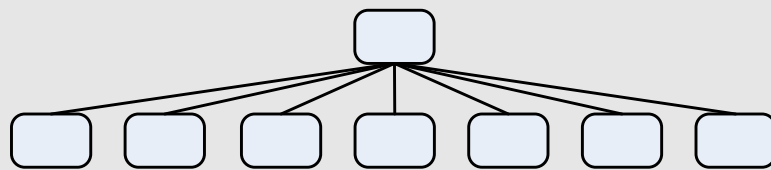
[6 P.]

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

Ein maximal gefüllter B-Baum der Klasse  $\tau(3, 2)$  hat 48 Datensätze, ein minimal gefüllter B\*-Baum der Klasse  $\tau(2, 3, 3)$  dagegen nur 18 Datensätze. Differenz: 30 Datensätze

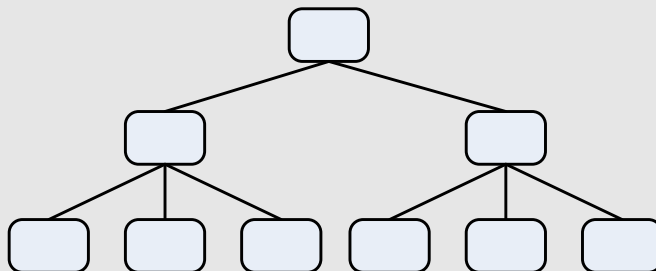
maximal gefüllter B-Baum:

(1 Wurzel + 7 Blattknoten = 8 Knoten mit je 6 Datensätzen)



minimal gefüllter B\*-Baum:

(1 Wurzel (1 Eintrag) + 2 innere Knoten (je 2 Einträge) + 6 Blattknoten = 6 Knoten mit je 3 Datensätzen)



[3 P.]

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

Bei dem B-Baum muss man 8 (Wurzel + 7 Blätter) verschiedene Knoten (Seiten) lesen. Beim B\*-Baum sind es ebenfalls 8 Knoten (Seiten) (einmal von Wurzel zu linken äusseren Blatt, danach über den *sequence set* alle Blätter).

- (3) Wenn man wahlfrei einen Datensatz aus einem der beiden Bäume [2 P.] lesen möchte, wieviel Knoten (Seiten) muss man im jeweiligen Baum maximal lesen?

Bei dem B-Baum muss man maximal 2 Knoten (Seiten) lesen, beim B\*-Baum hingegen immer 3 Knoten (Seiten).

- b) Gegeben sei ein **B\*-Baum** der Klasse  $\tau(k, k^*, h)$ . Innerhalb des zugehörigen (fiktiven) DBS wird vereinfachend eine Seitengröße von 800B (L=800B) angenommen. Die weiteren Kenngrößen sind dabei wie folgt festgelegt: [4 P.]

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

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

Begründen Sie Ihr Ergebnis durch einen Rechenweg.

Hinweis: Die oben genannten Kenngrößen haben folgende Bedeutung:

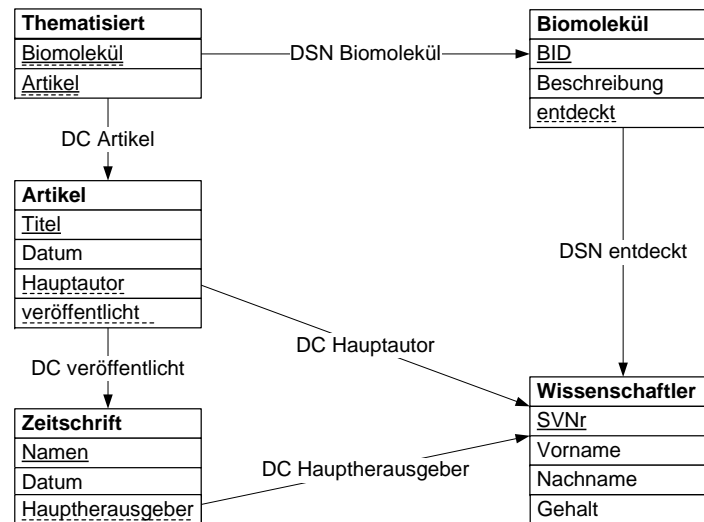
|       |                                               |
|-------|-----------------------------------------------|
| $l_M$ | die Länge der Seiteninformationen einer Seite |
| $l_D$ | die Länge eines Datensatzes                   |
| $l_K$ | die Länge eines Schlüsselwertes               |
| $l_P$ | die Länge eines Zeigers                       |

$$\begin{aligned}k &= \left\lfloor \frac{L - l_M - l_P}{2 \cdot (l_K + l_P)} \right\rfloor = \left\lfloor \frac{800B - 4B - 6B}{2 \cdot (5B + 6B)} \right\rfloor = \left\lfloor \frac{790B}{22B} \right\rfloor \simeq [35, 9] = 35 \\k^* &= \left\lfloor \frac{L - l_M - 2 \cdot l_P}{2 \cdot (l_K + l_D)} \right\rfloor = \left\lfloor \frac{800B - 4B - 12B}{2 \cdot (5B + 160B)} \right\rfloor = \left\lfloor \frac{784B}{330B} \right\rfloor \simeq [2, 376] = 2\end{aligned}$$

## Aufgabe 10: Referentielle Aktionen

[12 P.]

Gegeben sei folgender Referenzgraph für eine Biomoleküldatenbank.



Die referentiellen Aktionen der Fremdschlüssel in den Abbildungen sind jeweils abgekürzt mit 'DC' für 'ON DELETE CASCADE', 'DR' für 'ON DELETE RESTRICT' bzw. 'DSN' für 'ON DELETE SET NULL'.

[1 P.]

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

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

[1 P.]

2. Handelt es sich im vorliegenden Fall um ein sicheres Schema?

- Ja  
 Nein

Die theoretisch einzig mögliche Stelle für eine Reihenfolgenabhängigkeit im Schema existiert beim Löschen eines Wissenschaftlers, da beim Kaskadieren des Löschens auf zwei Wegen bis zum Artikel kaskadiert werden kann; Einmal über die Zeitschrift und einmal auf direktem Wege. **Das vorliegende Schema ist sicher**, da auf beiden Wegen keine blockierenden referentiellen Aktionen wie beispielsweise 'ON DELETE RESTRICT' verwendet werden.



3. Die Fremdschlüssel der Relation 'Artikel' wurden durch folgende SQL-DDL-Anweisung erzeugt: [5 P.]

```
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
);
```

Für den Fremdschlüssel 'fkVeröffentlicht' aus der Relation 'Artikel' soll nun die referentielle Aktion von 'ON DELETE CASCADE' auf „ON DELETE RESTRICT“ geändert werden:

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

Siehe auch die Folien zur Schemaevolution, z.B. Folie K5/F106:

```
ALTER TABLE Artikel DROP CONSTRAINT fkVeröffentlicht;

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

4. Handelt es sich bei dem geänderten Schema um ein sicheres Schema? [5 P.]  
 Falls das Schema nicht sicher ist, beschreiben Sie ein Szenario, in dem das Ergebnis einer Änderungs-/Löschoperation von der Auswertungsreihenfolge der referentiellen Aktionen abhängig ist.

Ja

Nein

**Das vorliegende Schema ist jetzt nicht mehr sicher**, da beim Löschen eines Wissenschaftlers die Löschoperation direkt und über 'Zeitschrift' zu 'Artikel' propagiert wird. Dies führt zu **reihenfolgeabhängigen Ergebnissen**, wenn ein Wissenschaftler Hauptherausgeber einer Zeitschrift ist, in der nur Artikel veröffentlicht wurden, bei denen der zu löschende Wissenschaftler Hauptautor ist:

- Wird zunächst der direkte Weg beschritten, wird der Artikel gelöscht. Da kein Artikel mehr die Zeitschrift des Wissenschaftlers referenziert, kann nun auch die Zeitschrift gelöscht werden.
- Wird zunächst der Weg über 'Zeitschrift' beschritten, wird das Löschen der Zeitschrift jedoch verweigert, da die zu löschende Zeitschrift noch von einem Artikel referenziert wird.