

Aufgabe 1: Informationsmodellierung mit dem Entity-Relationship-Modell

[20 P.]

Eine nahe gelegene Raumfahrtbehörde möchte ein neues System zur Verwaltung seiner Expeditionen einführen, bei dem eine relationale Datenbank zum Einsatz kommt.

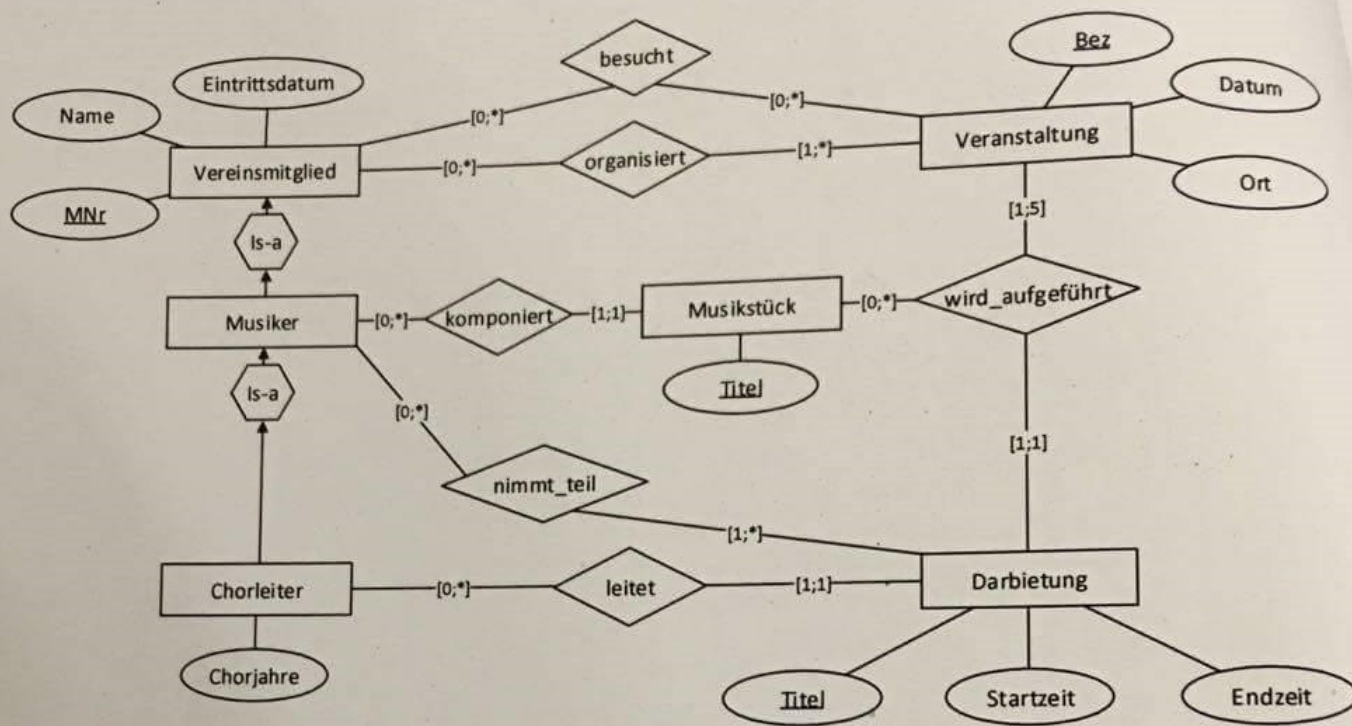
Erfassen Sie die im Folgenden beschriebenen Informationsstrukturen in einem ER-Diagramm. Beziehen Sie sich dabei genau auf die gegebene Beschreibung, ohne weiteres Wissen zu möglicherweise ähnlichen Anwendungsbereichen einfließen zu lassen. Markieren Sie in Ihrem Entwurf Primärschlüssel durch Unterstreichung und konkretisieren Sie die Abbildungstypen durch **Kardinalitätsrestriktionen** (Notation: [min;max]). Verwenden Sie unbedingt die aus der Vorlesung bekannte Notation. 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). Existenzabhängigkeiten sollen NICHT modelliert werden.

In der Datenbank werden u.a. Personen und Raumschiffe erfasst. Jede Person besitzt eine eindeutige Personalnummer (kurz PNr), einen Namen und eine Menge an Qualifikationen. Ein Raumschiff besitzt einen eindeutigen Namen, eine Modellnummer (kurz ModNr) und ein Baujahr. Eine Person kann Crew-Mitglied in mehreren Raumschiffen sein, wobei jede Crew-Mitgliedschaft durch einen Zeitraum und eine Funktion gekennzeichnet ist. Die Crew eines jeden Raumschiffes besteht über alle Zeiträume hinweg aus mindestens zehn Personen. Neben Personen und Raumschiffen werden auch Tätigkeiten in der Datenbank gespeichert, die jeweils über ihre Tätigkeitsnummer (kurz TNr) eindeutig identifiziert werden können und darüber hinaus eine Dauer und eine Beschreibung besitzen. Ein Raumschiff kann beliebig oft durch eine Person gewartet werden, wobei diese Person dabei eine bestimmte Tätigkeit an einem konkreten Datum ausführt. Eine Person kann beliebig viele Wartungen durchführen und eine Tätigkeit kann bei beliebig vielen Wartungen durchgeführt werden. Eine Expedition wird durch eine eindeutige ExpNummer (kurz ExpNr), ein Forschungsziel und einen Zeitraum beschrieben. Jede Expedition besitzt genau eine Person als Expeditionsleiter. Jede Person kann als Leiter beliebig vieler Expeditionen fungieren. Ein Raumschiff kann an beliebig vielen Expeditionen teilnehmen und an einer Expedition können beliebig viele Raumschiffe (jedoch mindestens eins) teilnehmen. Jeder Himmelskörper besitzt einen eindeutigen Namen und eine Position. Planeten und Monde sind besondere Himmelskörper, wobei jeder Planet beliebig viele Monde haben kann und jeder Mond zu genau einem Planeten gehört. Eine Expedition besucht einen oder mehrere Himmelskörper. Jeder Himmelskörper kann von beliebig vielen Expeditionen besucht werden.

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

[9 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. Verwenden Sie das **Partitionierungsmodell**, um die Vererbung abzubilden. Stellen Sie das resultierende DB-Schema dar, indem Sie die notwendigen Relationenschemata in der Form

$$Relation(Attribut_1, Attribut_2, \dots, Attribut_n)$$

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

$$\underline{Attr_i} \rightarrow Rel_b. \underline{Attr_j}$$

Hinweis zur Semantik von 1:n-Beziehungen: Nach dem dargestellten ER-Diagramm wird jede Darbietung von genau einem Chorleiter geleitet, während jeder Chorleiter beliebig viele Darbietungen leiten kann. Die Semantik aller anderen 1:n-Beziehungen ist entsprechend.

[12 P.]

Aufgabe 3: Relationenalgebra

Gegeben seien die folgenden Relationenschemata:

Zirkus(Name, Gründungsjahr)

Artist(SVNr, Vorname, Nachname, Zirkus → Zirkus.Name)

Darbietung(Bez, Beschreibung, Schwierigkeit)

Auftritt(Artist → Artist.SVNr, Darbietung → Darbietung.Bez, Datum, Ort)

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

[3 P.]

- a) Geben Sie eine natürlichsprachliche Beschreibung der Ergebnismenge des folgenden Relationenalgebra-Ausdrucks an.

$$\pi_{\text{Datum, Ort}}(\sigma_{\text{Gründungsjahr} > 1990}(\text{Zirkus}) \bowtie_{\text{Name} = \text{Zirkus}} \text{Artist} \bowtie_{\text{SVNr} = \text{Artist}} \text{Auftritt})$$

[3 P.]

- b) Geben Sie einen Relationenalgebra-Ausdruck an, der die SVNr aller Artisten ausgibt, die noch nie einen Auftritt hatten.

- c) Geben Sie einen Relationenalgebra-Ausdruck an, der die Schwierigkeit aller Darbietungen ausgibt, die mindestens einmal von einem Artisten des Zirkus 'Krone' aufgeführt wurden. [3 P.]

- d) Geben Sie einen Relationenalgebra-Ausdruck an, der zu dem unten angegebenen SQL-Ausdruck äquivalent ist. [3 P.]

```
SELECT DISTINCT z.Name, z.Gründungsjahr
FROM Zirkus z, Artist a, Auftritt au
WHERE z.Name = a.Zirkus
      AND au.Artist = a.SVNr
      AND au.Ort = 'Hamburg'
```

[15 P.]

Aufgabe 4: SQL

Gegeben seien die aus Aufgabe 3 bekannten Relationenschemata:

Zirkus(Name, Gründungsjahr)

Artist(SVNr, Vorname, Nachname, Zirkus → Zirkus.Name)

Darbietung(Bez, Beschreibung, Schwierigkeit)

Auftritt(Artist → Artist.SVNr, Darbietung → Darbietung.Bez, Datum, Ort)

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

[3 P.]

- i) Die Nachnamen aller Artisten ohne Duplikate und in aufsteigender Reihenfolge sortiert, die bereits einen Auftritt in Hamburg hatten.

[3 P.]

- ii) Die SVNr jedes Artisten, der schon mehr als zehn Auftritte hatte

Matrikel-Nr:

Name:

iii) Die Vor- und Nachnamen aller Artisten, die noch nie einen Auftritt in Hamburg hatten. [3 P.]

iv) Die Namen und Gründungsjahre aller Zirkusse nebst der Anzahl der bei ihnen jeweils angestellten Artisten. [3 P.]

b) Beschreiben Sie die Ergebnismenge der folgenden SQL-Anfrage in eigenen Worten. [3 P.]

```
SELECT b1.Darbietung
FROM Auftritt b1, Auftritt b2, Artist a1, Artist a2
WHERE b1.Darbietung = b2.Darbietung
AND b1.Artist = a1.SVNr
AND b2.Artist = a2.SVNr
AND a1.SVNr <> a2.SVNr
AND a1.Nachname = a2.Nachname;
```


Aufgabe 5: Änderbarkeit von Sichten

Ausgehend von der Basisrelation

Bücher(ISBN, Titel, Seitenzahl, Genre, Typ, Jahr)

seien folgende Sichten definiert:

```
CREATE VIEW Sachbücher
AS SELECT * FROM Bücher
WHERE Typ = 'Sachbuch';
```

```
CREATE VIEW Belletristik
AS SELECT * FROM Bücher
WHERE Typ = 'Belletristik';
```

```
CREATE VIEW ScienceFiction
AS SELECT * FROM Belletristik
WHERE Genre = 'SciFi';
```

```
CREATE VIEW ScienceFictionNeu
AS SELECT * FROM ScienceFiction
WHERE Jahr > 2000;
```

Da diese Sichten Änderungsoperationen erlauben, **könnten** sie alle mit dem Zusatz **WITH CASCADED CHECK OPTION** definiert werden. In der Tabelle auf Seite 15 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:

a) **UPDATE** Belletristik
 SET Jahr = 1937
 WHERE Jahr = 1936
 AND Genre = 'Horror';

b) **UPDATE** ScienceFictionNeu
 SET Jahr = 1999
 WHERE Genre = 'SciFi';

c) **UPDATE** ScienceFiction
 SET Genre = 'Romanze'
 WHERE Jahr = 2001;

d) **INSERT INTO** ScienceFiction
 VALUES (1234567, 'Essen_Morgen', 234, 'Ernährung', 'Sachbuch', 2005);

Markieren Sie im unteren Teil der 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

ja / Sicht-Liste

Operation wird für die Konfiguration zurückgewiesen

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:

Sachbücher

S

Belletristik

B

ScienceFiction

F

ScienceFictionNeu

FN

S	-	CASC	-	-
B	-	-	CASC	-
F	-	-	-	CASC
FN	-	CASC	-	-
a)	ja / B	ja / B	ja / B	ja / B
b)				
c)				
d)				

[10 P.]

Aufgabe 6: Optimierung

Gegeben seien die aus Aufgabe 3 bekannten Relationenschemata:

Zirkus(Name, Gründungsjahr)

Artist(SVNr, Vorname, Nachname, Zirkus → Zirkus.Name)

Darbietung(Bez, Beschreibung, Schwierigkeit)

Auftritt(Artist → Artist.SVNr, Darbietung → Darbietung.Bez, Datum, Ort)

[7 P.]

- a) Für die nachfolgende SQL-Anfrage soll eine algebraische Optimierung durchgeführt werden. Zeichnen Sie dafür als erstes den entsprechenden Operatorbaum für die vorgegebene SQL-Anfrage und optimieren Sie diesen anschließend anhand der in der Vorlesung eingeführten Regeln (Projektionen sollen dabei jedoch **nicht** nach unten gezogen werden).

[3 P.]

- b) Bewerten Sie den in Aufgabenteil a) erstellten Operatorbaum mit den Kardinalitäten der Zwischenergebnisse. (Die Anzahl der Attribute soll dabei **nicht** betrachtet werden.)

Für die zugehörige Datenbank werden folgende Kardinalitäten angenommen:

$\text{Card}(\text{Auftritt}) = 1.000$, $\text{Card}(\text{Darbietung}) = 30$.

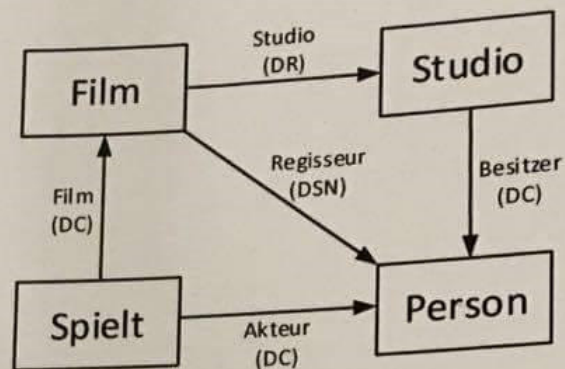
Es gibt insgesamt 50 verschiedene Orte, in denen ein Auftritt stattfindet, und es gibt insgesamt 10 verschiedene Schwierigkeitsgrade, die eine Darbietung haben kann.

```
SELECT DISTINCT d.Bez, a.Artist
FROM Auftritt a, Darbietung d
WHERE a.Darbietung = d.Bez
      AND a.Ort = 'Hamburg'
      AND d.Schwierigkeit = 5
```

[10 P.]

Aufgabe 7: Referentielle Aktionen

Gegeben sei folgender Referenzgraph für eine Filmdatenbank.



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.]

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

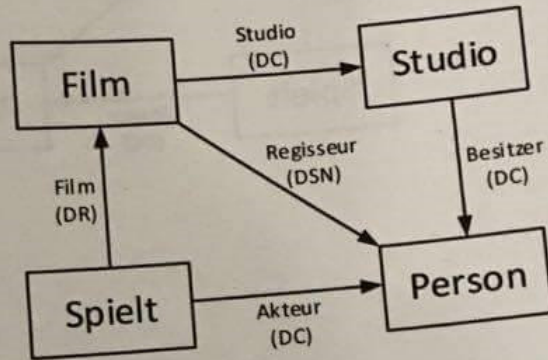
[3 P.]

- b) Handelt es sich im vorliegenden Fall um ein sicheres Schema? 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

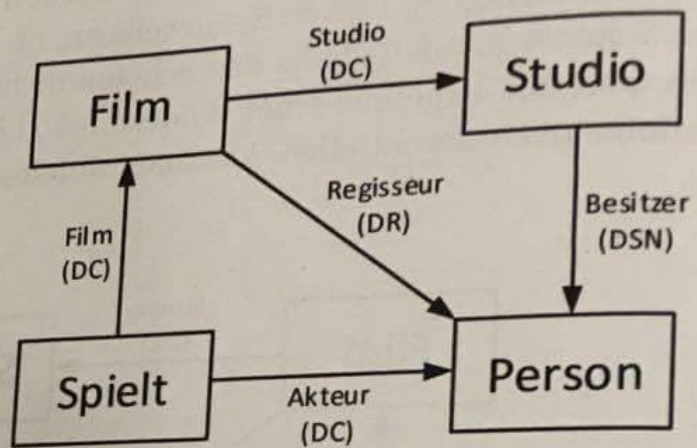
- c) Die referentiellen Aktionen des Referenzgraphen werden nun zweimal geändert (siehe nachfolgende Abbildungen). Geben Sie jeweils an, ob es sich im vorliegenden Fall um ein sicheres Schema handelt. Falls das Schema nicht sicher ist, beschreiben Sie ein Szenario, in dem das Ergebnis einer Änderungs-/Löschoperation von der Auswertungsreihenfolge der referentiellen Aktionen abhängig ist.

i)



- ☐ Ja
☐ Nein

ii)



☐ Ja

☐ Nein

[12 P.]

Aufgabe 8: Transaktionen

Gegeben sind die vier Objekte a, b, c und d , welche von den Transaktionen T_1, T_2 und T_3 gelesen bzw. geschrieben werden. Dabei bezeichnet $w_i(a)$ den Schreibzugriff der Transaktion T_i auf das Objekt a , $r_i(a)$ den Lesezugriff der Transaktion T_i auf das Objekt a und c_i das Commit der Transaktion T_i .

[8 P.]

a) 2PL-Synchronisation mit R/X-Sperren

Gegeben ist der Ablauf S_1 , der anzeigt, in welcher Reihenfolge die Operationen der drei Transaktionen T_1, T_2, T_3 beim Scheduler eines Datenbanksystems eintreffen.

$S_1 = r_1(a) \ r_2(c) \ r_1(d) \ w_1(c) \ r_3(b) \ w_1(d) \ c_1 \ r_3(c) \ r_2(b) \ c_2 \ w_3(c) \ c_3$

Bei der Ausführung von S_1 soll das RX-Sperrverfahren mit 2PL zum Einsatz kommen. Vervollständigen Sie die unten angegebene Tabelle, indem Sie die Sperranforderungen (lock) und -freigaben (unlock) der Transaktionen, deren Lese- und Schreibzugriffe (read bzw. write) und Commits (commit) eintragen. Beachten Sie, dass eine Transaktion innerhalb eines Zeitschritts nur jeweils eine Operation durchführen kann. Nutzen Sie die Spalte „Bemerkungen“ für etwaige Wartebeziehungen und Benachrichtigungen an wartende Transaktionen.

$$S_1 = r_1(a) \ r_2(c) \ r_1(d) \ w_1(c) \ r_3(b) \ w_1(d) \ c_1 \ r_3(c) \ r_2(b) \ c_2 \ w_3(c) \ c_3$$

	T ₁	T ₂	T ₃	Bemerkung
0				
1	lock(a,R)			
2	read(a)	lock(c,R)		
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

[4 P.]

- b) Gegeben ist der Ablauf S_2 , der anzeigt, in welcher Reihenfolge die Operationen der drei Transaktionen T_1, T_2, T_3 beim Scheduler eines Datenbanksystems eintreffen.

$S_2 = r_2(b) \ w_1(c) \ r_1(b) \ r_3(a) \ r_2(c) \ w_2(b) \ c_2 \ w_1(a) \ r_3(b) \ w_3(d) \ c_1 \ c_3$

[2 P.]

- (i) Welche Abhängigkeiten existieren zwischen den Operationen der drei Transaktionen innerhalb des Ablaufs S_2 ?

Nutzen Sie dabei die folgende Notation: Sind $w_1(x)$ und $r_2(x)$ voneinander abhängig und wird $w_1(x)$ zeitlich vor $r_2(x)$ ausgeführt, so wird dies durch $w_1(x) \rightarrow r_2(x)$ notiert.

[2 P.]

- (ii) Ist der Ablauf seriell, serialisierbar oder nicht serialisierbar? Begründen Sie die Antwort (z.B. mit Hilfe der Abhängigkeiten).

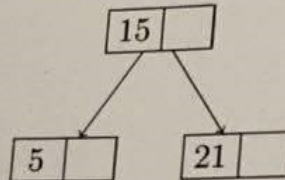
[10 P.]

Aufgabe 9: B-Bäume und B*-Bäume

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

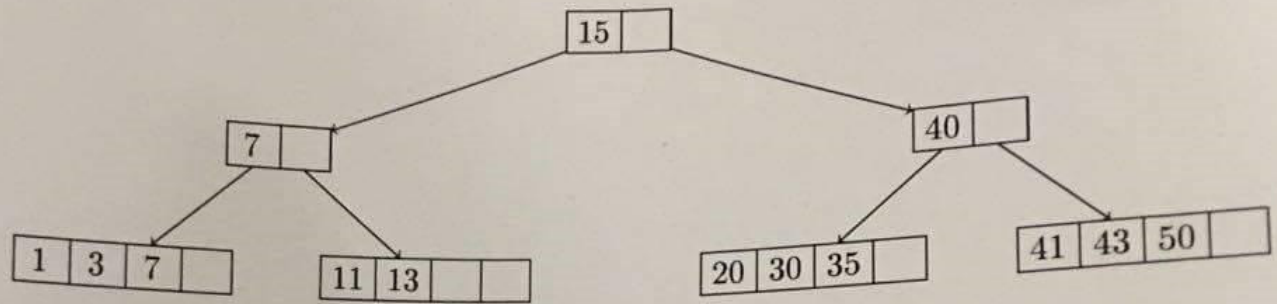
[5 P.]

- 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 **9, 23, 16, 27** und **12** in dieser Reihenfolge ein. Nennen Sie jeweils die durchgeführten Maßnahmen (Splitten, einfaches Einfügen) und zeichnen Sie den Baum nach jedem Split-Vorgang neu.



[5 P.]

- b) Löschen Sie aus dem unten abgebildeten **B*-Baum** der Klasse $\tau(1, 2, h)$ die Datensätze mit den Schlüsselwerten **3, 35, 13, 7** und **30** (in dieser Reihenfolge). Geben Sie jeweils kurz an, welche konkrete Maßnahme Sie durchgeführt haben (Mischen, Ausgleichen, einfaches Löschen) und zeichnen Sie den Baum nach jedem Mischen und Ausgleichen neu. Für Ausgleichs- und Mischoperationen sollen nur direkt benachbarte Geschwisterknoten (bevorzugt der rechte) herangezogen werden.



[10 P.]

Aufgabe 10: Berechnungen in B- und B*-Bäumen

[2 P.]

- a) Gegeben ist ein B*-Baum der Klasse $\tau(4, 2, h^*)$ mit genau 100 vollbesetzten Blattknoten.

i) Wieviele Datensätze enthält der Baum?

[3 P.]

ii) Wieviele innere Knoten hat der Baum höchstens?

Hinweis: Überlegen Sie sich hierzu den Aufbau ausgehend von der Blattebene.

- iii) Wie groß ist eine Seite (Seitengröße L), wenn jeder Blattknoten genau einer Seite entspricht und für die anderen Kenngrößen die unten aufgelisteten Werte gelten? [3 P.]

l_M (Länge der Seiteninformationen)	$= 4B$
l_D (Länge eines Datensatzes)	$= 145B$
l_K (Länge eines Schlüsselwertes)	$= 5B$
l_P (Länge eines Zeigers)	$= 6B$

- b) Gegeben sei der nachfolgend vereinfacht dargestellte B-Baum. Wieviele unterschiedliche Knoten (Seiten) müssen gelesen werden, um alle Datensätze d_i mit Schlüsselwert $k_i > 18$ auszulesen? [2 P.]

