

Aufgabe 1 (10 Punkte)

Beantworten Sie die Fragen in dieser Aufgabe mit einer kurzen, prägnanten Antwort.

1. Welche Eigenschaften muss ein Schlüsselkandidat erfüllen? (2 Punkte)

Aus ihm muss R ableitbar sein und er muss minimal sein.

2. Nennen Sie zwei wesentliche Konzepte von *Ruby on Rails*. (2 Punkte)

MVC, DRY, COC

3. Mit welchem XPath-Ausdruck selektieren Sie alle Knoten A in einem Dokument?

//A

4. Was zeichnet ein relationales Schema in erster Normalform aus?

Es gibt keine mengenwertigen Attribute.

5. In welcher Komplexitätsklasse in O-Notation liegt die Operation LOOKUP in einer ISAM-organisierten Datenstruktur?

$O(\log n)$

6. Wie heißt eine verzahnte Historie, die das gleiche Resultat liefert, als wenn die Transaktionen alle nacheinander ausgeführt würden?

serialisierbar

7. Wie viele Nachfolger hat jeder Knoten in einem B^* -Baum mit Parameter k höchstens?

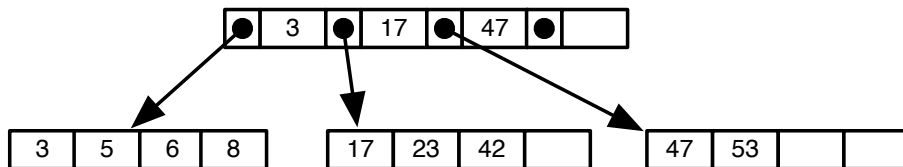
$2 \cdot k$

8. Sei F eine Menge funktionaler Abhängigkeiten. Wie nennt man die Menge aller aus F ableitbaren funktionalen Abhängigkeiten F^+ ?

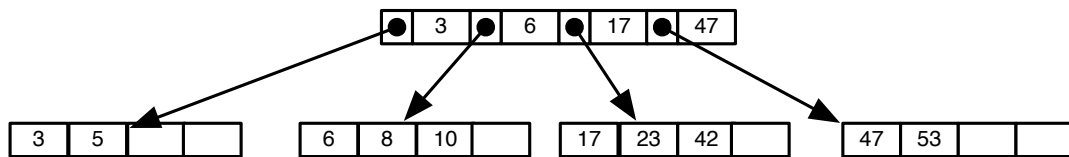
Hülle oder Closure

Aufgabe 2 (8 Punkte)

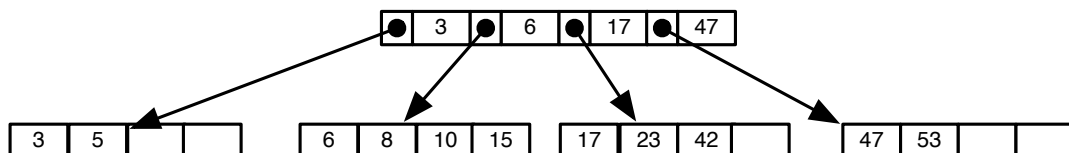
Fügen Sie in den unten dargestellten B*-Baum mit $k = 2$ die Schlüssel 10, 15 und 7 ein und zeichnen Sie den Baum nach dem Einfügen eines Schlüssels jeweils erneut.



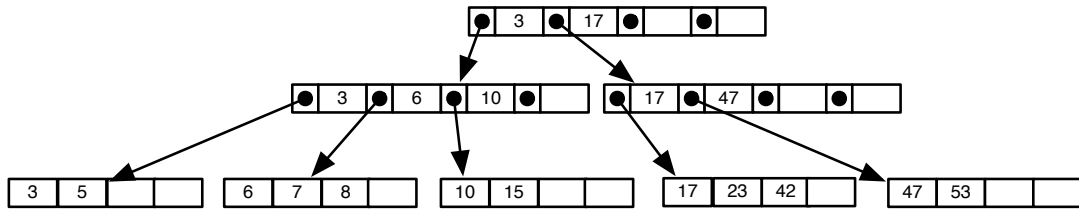
Nach Einfügen von 10:



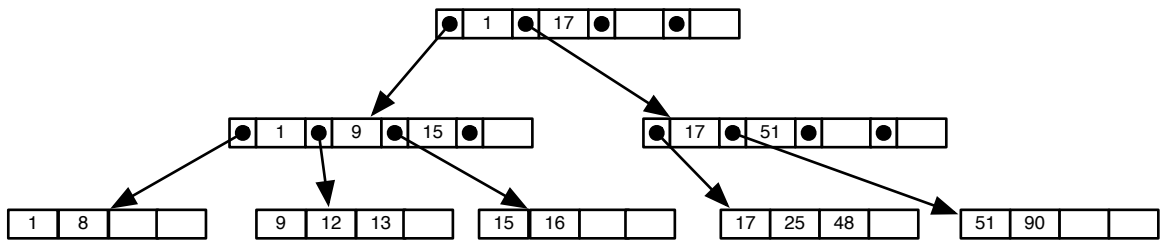
Nach Einfügen von 15:



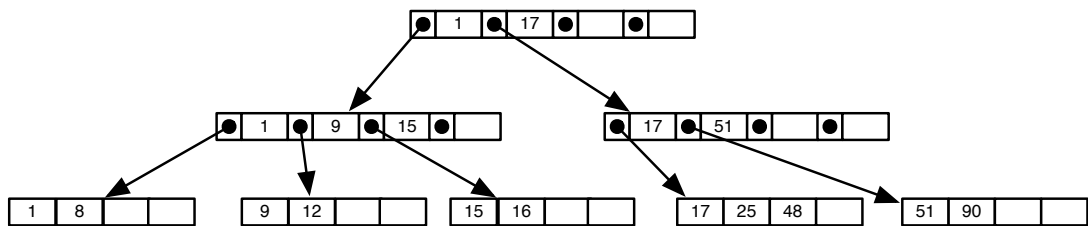
Nach Einfügen von 8:



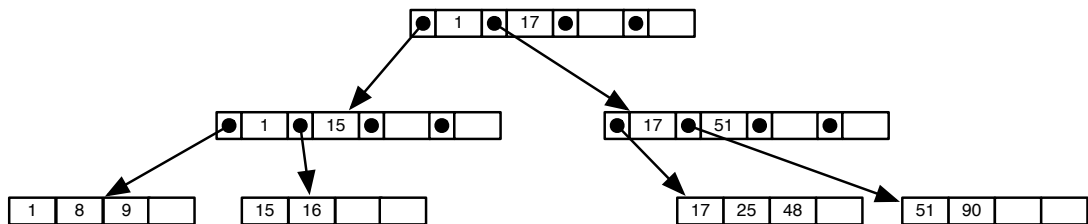
Löschen Sie aus dem folgenden, vom vorherigen unabhängigen Baum die Schlüssel 13, 12 und 90 und zeichnen Sie den B*-Baum nach jedem Löschvorgang erneut.



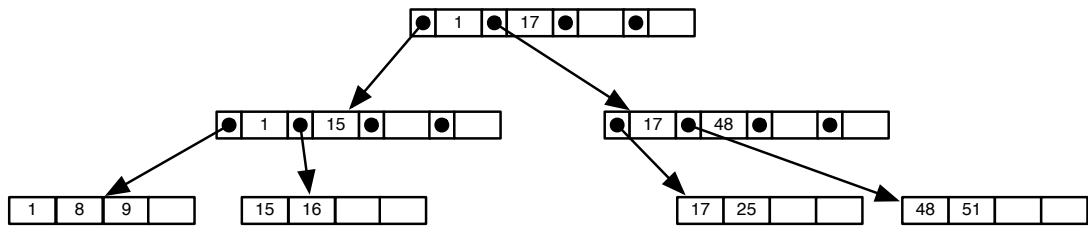
Nach Löschen von 13:



Nach Löschen von 12:

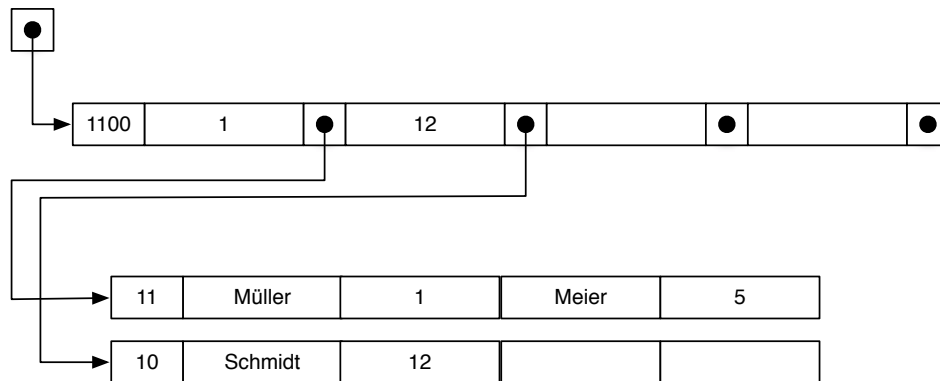


Nach Löschen von 90:

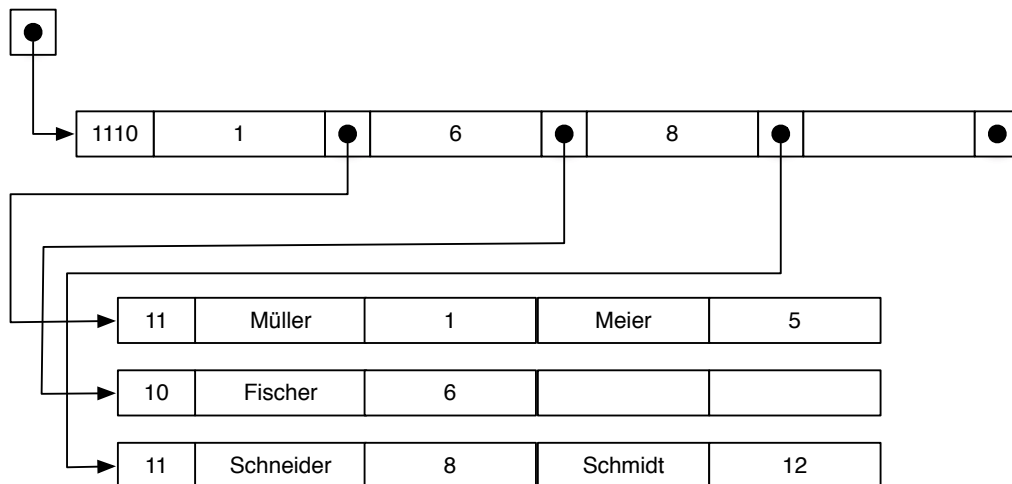


Aufgabe 3 (8 Punkte)

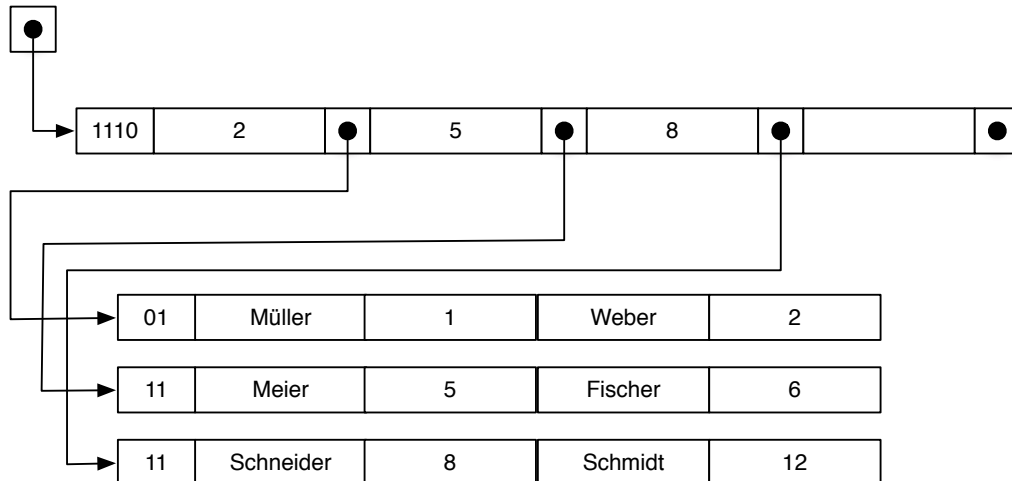
Im Folgenden ist eine Menge von Records, die mit Hilfe des ISAM-Verfahrens abgelegt wurden, in einer entsprechenden Datenstruktur dargestellt. Ein Record besteht aus einer ganzen Zahl (Schlüssel) und einem String-Wert. Gespeichert sind Namen von Personen. Dabei fasst ein Datenblock jeweils zwei Records, ein Index-Block vier Schlüssel-Adresspaare.



- a) Fügen Sie in die oben angegebene Struktur nacheinander die Records [8, Schneider] und [6, Fischer] ein und zeichnen Sie die resultierende Struktur anschließend erneut.



- b) Fügen Sie nun in die Struktur aus a) den Record [2, Weber] ein und löschen Sie anschließend den Eintrag mit dem Index 1. Zeichnen Sie die resultierende Struktur erneut.



Aufgabe 4 (4 Punkte)

In einer Ruby on Rails-Applikation haben Sie sich beim Erstellen Ihrer Datenbank an die Rails-Konventionen gehalten und manuell mit dem Befehl `rails generate model lecture` eine Klasse `Lecture` erstellt. In der zugehörigen Datenbanktabelle `lectures` sind die Attribute `[id,name,sws,professor_id]` gespeichert, in denen die Identifikationsnummer, der Name einer Vorlesung, die Semesterwochenstunden und ein Verweis auf einen Professor aus der Tabelle `professors` gespeichert werden kann.

Ein Model für die Professoren haben Sie bereits erstellt und wollen nun mit Hilfe des Befehls `l.professor` ein Objekt vom Typ `Professor` erhalten, dass - wenn `l` ein Objekt vom Typ `Lecture` ist - die Attribute des Professors enthält, der mit dieser Vorlesung über das entsprechende Attribut verknüpft ist. Außerdem möchten Sie mit Hilfe des Befehls `Lecture.grosse_vorlesungen` ein Array von `Lecture`-Objekten erhalten, mit allen Vorlesungen, deren Attribut `sws = 9` ist, also alle Vorlesungen mit 9 Semesterwochenstunden. Ergänzen Sie die unten angegebene Klasse, so dass die oben beschriebene Funktionalität zur Verfügung steht. *Hinweis:* `grosse_vorlesungen` ist eine Klassenmethode, die mit `def self.grosse_vorlesungen` deklariert ist (siehe unten).

```
class Lecture < ActiveRecord::Base

  belongs_to :professor

  def self.grosse_vorlesungen

    Lecture.where(:sws => 9)

  end

end
```


Aufgabe 5 (6 Punkte)

Für ein transaktionsunterstützendes Datenbanksystem kann man die so genannten *ACID*-Kriterien fordern. Nennen Sie die vier Eigenschaften und erklären Sie in maximal zwei kurzen Sätzen, was die jeweilige Eigenschaft bedeutet.

Kriterium	kurze Erklärung
Atomicity	Transaktionen werden entweder komplett oder gar nicht durchgeführt ("Alles oder Nichts").
Consistency	Während einer Transaktion darf die Datenbank gegen Konsistenz-Bedingungen verstoßen, nach Ende der Transaktion muss die Datenbank wieder in einem konsistenten Zustand sein.
Isolation	Nebenläufige Transaktionen haben keinen Einfluss aufeinander.
Durability	Änderungen erfolgreicher Transaktionen sind dauerhaft festgeschrieben (auch bei einem Absturz).

Aufgabe 6 (9 Punkte)

Gegeben sei die folgende Ausprägung einer relationalen Datenbank.

Kunden:

KdNr	Nachname	Vorname	StraßeNr	PLZ
236	Meier	Max	Geranienweg 1	10115
485	Schmidt	Hans	Adenauerplatz 4	49080
328	Fischer	Heiner	Belmer Weg 15	49069
548	Müller	Max	Baumwall 31	49076

Orte:

PLZ	Ort
49080	Osnabrück
49076	Osnabrück
10115	Berlin
49069	Osnabrück

Bestellungen:

BestellNr	KdNr	ArtikelNr	Anzahl	Preis
15	236	875	2	15.35
45	485	875	1	29.99
10115	328	567	5	49.99
15	236	733	1	199.99

Geben Sie für die folgenden Ausdrücke in relationaler Algebra in den vorgesehenen Kästen auf der nächsten Seite an, wie die Ergebnismenge für obige Ausprägung aussieht.

- a) $\Pi_{Ort}(\sigma_{Vorname='Max'}(Kunden) \bowtie Orte)$
- b) $\Pi_{BestellNr}(\sigma_{BestellNr=PLZ}(Kunden \times Bestellungen))$
- c) $\Pi_{Nachname}((Kunden \bowtie Bestellungen) \div \Pi_{ArtikelNr}(\sigma_{ArtikelNr=875 \vee ArtikelNr=733}(Kunden \bowtie Bestellungen)))$

d) Schreiben Sie nun einen Ausdruck in relationaler Algebra selbst, der auf Basis des oben angegebenen *Schemas* alle Nachnamen von Kunden liefert, die in Osnabrück wohnen und mindestens eine Bestellung aufgegeben haben. Notieren Sie den Ausdruck auf der nächsten Seite.

Geben Sie nachfolgend die Lösungen zur den Aufgaben der vorherigen Seite an:

a) $\Pi_{Ort}(\sigma_{Vorname='Max'}(Kunden) \bowtie Orte)$

{[Osnabrück], [Berlin]}

b) $\Pi_{BestellNr}(\sigma_{BestellNr=PLZ}(Kunden \times Bestellungen))$

{[10115]}

c) $\Pi_{Nachname}((Kunden \bowtie Bestellungen) \div$
 $\Pi_{ArtikelNr}(\sigma_{ArtikelNr=875 \vee ArtikelNr=733}(Kunden \bowtie Bestellungen)))$

{}

d) Eigener Ausdruck laut vorheriger Seite:

$\Pi_{Nachname}(\sigma_{Ort=Osnabrück}(Kunden \bowtie Bestellungen \bowtie Orte))$

Aufgabe 7 (17 Punkte)

Betrachten Sie das folgende Muster, nach dem ein Unternehmen Rechnungen stellt. Das Rechnungswesen des Unternehmens soll nun in einer relationalen Datenbank verwaltet werden. Zeichnen Sie auf der nächsten Seite ein ER-Diagramm für den Entwurf des relationalen Datenbanksystems. Achten Sie dabei auf folgende Gegebenheiten: An einer Adresse können mehrere Kunden wohnen und zu einem Kunden sollen mehrere Adressen erfasst werden können. Ein Artikel hat immer den gleichen Preis, es finden keine nachträglichen Preisänderungen statt. Zu einem Artikel gehört ein bestimmter Mehrwertsteuersatz. Die eventuell anfallenden Versandkosten werden als Artikel gespeichert. Achten Sie darauf, dass Sie mit Hilfe Ihres Entwurfs alle (mit den grauen Kästen) markierten Angaben auf der Rechnung erfassen und die Rechnung so aus der Ausprägung Ihrer Datenbank später wieder erzeugen können. Speichern Sie nicht zu viel und achten Sie darauf, keine mengenwertigen Attribute zuzulassen, nutzen Sie keine schwachen Entitäten.

Supershop Zambrowsky

Supershop Zambrowsky, Hauptstraße 5, 12345 Hauptstadt

Hauptstraße 5
12345 Hauptstadt
Tel 01234 / 12 34 56 78
Fax 01234 / 12 34 56 87
info@super-zambrowsky.de
www.super-zambrowsky.de

Herr
Mark Sulkop
Am Drehwurm 20

48602 Spielhausen

Rechnung

Rechnungsnummer 2007061234
Rechnungsdatum 12.06.2007

Wir lieferten Ihnen:

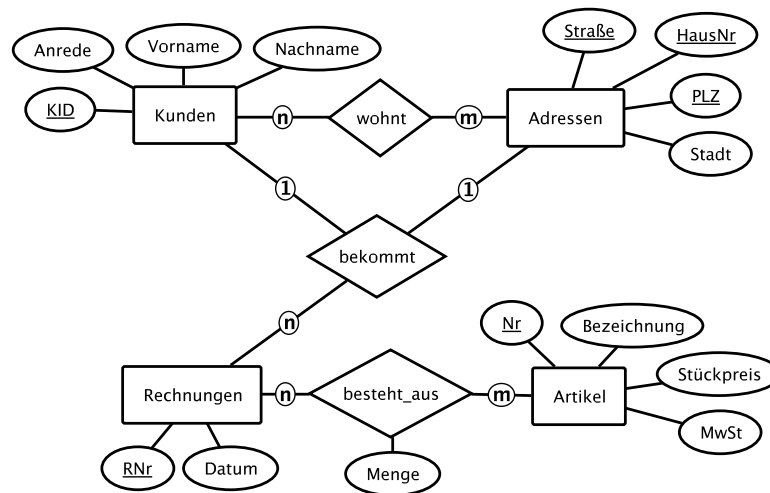
Menge	Artikel-Nr.	Bezeichnung	Einzelpreis €	Gesamt €	MWSt.
2	S-SB018	Sitzbank Gloria	119,00 €	238,00 €	1
3	K-S2073	Kartoffelsalat (5000 g)	3,75 €	11,25 €	2
1	X-0030	Versandkosten Sperrgut	15,00 €	15,00 €	1
Rechnungsbetrag				264,25 €	
			Mehrwertsteuer 1	19%	40,39 €
			Mehrwertsteuer 2	7%	0,74 €
			Nettobetrag		223,12 €

Zahlungsart: Vorkasse
Gesamtbetrag dankend erhalten.

Sofern nicht anders angegeben ist Rechnungsdatum gleich Lieferdatum.
Die Lieferung erfolgt ausschließlich zu unseren Allgemeinen Geschäftsbedingungen (AGB).

Geschäftsführer:	USt.-IdNr. DE123456789	Kleinbank
Dieter Zambrowsky	HRB Spielhausen 123 456	BLZ 001 001 01

Das ER-Diagramm zur Aufgabe (10 Punkte):



Überführen Sie Ihr ER-Diagramm in ein (noch unverfeinertes) relationales Schema (4 Punkte):

Kunden: {[KID, Anrede, Name, Vorname]}

Adressen: {[Straße, HausNr, PLZ, Stadt]}

Rechnungen: {[RNr, Datum]}

Artikel: {[Nr, Bezeichnung, Stückpreis]}

wohnt: {[KID, Straße, HausNr, PLZ]}

bekommt: {[KID, RNr, Straße, HausNr, PLZ]}

besteht_aus: {[RNr, Nr, Menge]}

Verfeinern Sie nun das Schema. Notieren Sie nur die geänderten Relationen und welche Relationen eliminiert werden können. (3 Punkte):

Kunden: {[KID, Anrede, Name, Vorname]}

Adressen: {[Straße, HausNr, PLZ, Stadt]}

Rechnungen: {[RNr, Datum, KID, Straße, HausNr, PLZ]}

Artikel: {[Nr, Bezeichnung, Stückpreis]}

wohnt: {[KID, Straße, HausNr, PLZ]}

besteht_aus: {[RNr, Nr, Menge]}

Aufgabe 8 (13 Punkte)

Gegeben seien die folgenden Relationen einer Universitätsdatenbank:

Studenten: {[MatrNr, Name, Semester]}

Dozenten: {[PersNr, Name]}

Vorlesungen: {[VorlNr, Titel, SWS, PersNr]}

hören: {[MatrNr, VorlNr]}

Formulieren Sie folgende Anfragen in SQL. *Hinweis:* Der lesende Dozent einer Vorlesung wird mit PersNr referenziert.

- a) Wie heißen die Studenten, die die Vorlesung "Selber Atmen" hören? (2 Punkte)

```
select s.Name from
  Studenten s,
  hoeren h,
  Vorlesungen v
where
  v.Titel = "Selber Atmen" and
  v.VorlNr = h.VorlNr and
  h.MatrNr = s.MatrNr
```

- b) Für wie viele Semesterwochenstunden (SWS) hat der Student "Meier" insgesamt Vorlesungen gehört? (2 Punkte)

```
select sum(v.SWS) from
  Vorlesungen v,
  Studenten s,
  hoeren h
where
  s.Name = "Meier" and
  h.MatrNr = s.MatrNr and
  v.VorlNr = h.VorlNr
```

- c) Wie heißen die Studenten, die keine Vorlesung bei "Prof. Sokrates" gehört haben? (5 Punkte)

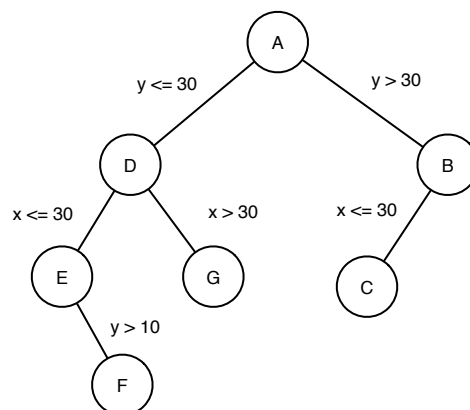
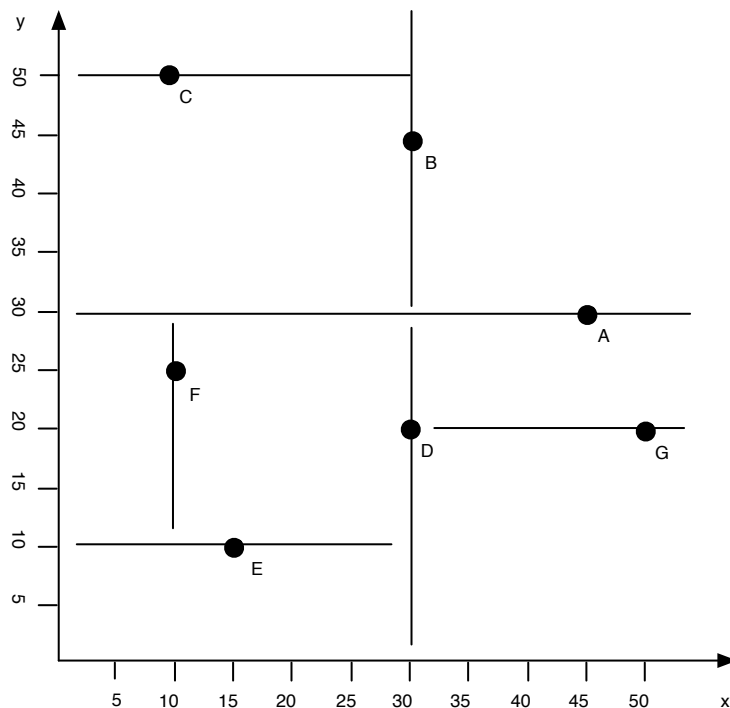
```
select s.Name from
    Studenten s
where NOT EXISTS
    (select v.VorlNr FROM
        Vorlesungen v,
        Dozenten d,
        hoeren h
        where
            s.MatrNr = h.MatrNr and
            h.VorlNr = v.VorlNr and
            d.PersNr = v.PersNr and
            d.Name = "Prof. Sokrates")
```

- d) Liefen Sie die Namen aller Dozenten, sortiert nach der Anzahl der Studenten, die bei dem jeweiligen Dozenten Vorlesungen gehört haben. (4 Punkte)

```
select    d.Name from
        Dozenten d,
        hoeren h,
        Vorlesungen v
where
        d.PersNr = v.PersNr
        h.VorlNr = v.VorlNr
group by
        d.PersNr
order by
        count(*) asc
```

Aufgabe 9 (6 Punkte)

Gegeben seien die 2-dimensionalen Schlüssel $A = (45, 30)$, $B = (30, 45)$, $C = (10, 50)$, $D = (30, 20)$, $E = (15, 10)$, $F = (10, 25)$, $G = (50, 20)$. Die Schlüssel wurden in die unten dargestellte Fläche eingefügt. Partitionieren Sie die Fläche anhand des in der Veranstaltung vorgestellten k-d-Baum-Verfahrens beginnend mit einem horizontalen Split und von A bis G der Reihe nach. Zeichnen Sie außerdem den zugehörigen k-d-Baum. Notieren Sie am Baum auch die verwendeten Diskriminierungsbedingungen.



Aufgabe 10 (9 Punkte)

Gegeben sei ein abstraktes Relationenschema:

$$R = \{A, B, C, D, E\}$$

Es gelten die folgenden funktionalen Abhängigkeiten:

$$\begin{aligned}BC &\rightarrow D \\D &\rightarrow E \\AB &\rightarrow C \\ADC &\rightarrow E \\DE &\rightarrow C\end{aligned}$$

a) Geben Sie α^+ für alle $\alpha \in \{BC, D, AB, ADC, DE\}$ an. (3 Punkte)

$$\begin{aligned}(BC)^+ &= BCDE \\D^+ &= CDE \\(AB)^+ &= ABCDE \\(ADC)^+ &= ACDE \\(DE)^+ &= CDE\end{aligned}$$

b) Wie lauten die Schlüsselkandidaten für R ? Begründen Sie in maximal zwei Sätzen warum es keine anderen Schlüsselkandidaten geben kann. (2 Punkte)

Nur AB ist Schlüsselkandidat. A und B sind nicht ableitbar, jeder andere Schlüssel, der A und B enthält wäre also nicht minimal.

b) Sei R in der 1. Normalform. Ist R auch in der 2. Normalform? Geben Sie eine kurze Begründung. (2 Punkte)

Die Nicht-Primärattribute C, D und E sind alle voll funktional abhängig vom einzigen Schlüsselkandidaten AB , R ist also in der 2. Normalform.

c) Ist R auch in der 3. Normalform? Begründen Sie auch hier Ihre Antwort. (2 Punkte)

$AB \rightarrow D \rightarrow E$, es gilt aber nicht $D \rightarrow AB$, E ist also transitiv über D abhängig vom Schlüsselkandidaten AB . R ist also nicht in der dritten Normalform.

Aufgabe 11 (4 Punkte)

Zeigen Sie nur unter Verwendung der *Armstrong-Axiome*, dass die Regel der *Vereinigung* für funktionale Abhängigkeiten gilt.

Zu zeigen ist: $\alpha \rightarrow \beta \wedge \alpha \rightarrow \gamma \Rightarrow \alpha \rightarrow \beta\gamma$

Beweis: Verstärkung von $\alpha \rightarrow \beta$ mit α :

$$\alpha \rightarrow \alpha\beta \Leftrightarrow \alpha \rightarrow \alpha\beta$$

Verstärkung von $\alpha \rightarrow \gamma$ mit β :

$$\alpha\beta \rightarrow \gamma\beta \Leftrightarrow \alpha\beta \rightarrow \beta\gamma$$

Wegen der Transitivität folgt:

$$\alpha \rightarrow \alpha\beta \wedge \alpha\beta \rightarrow \beta\gamma \Rightarrow \alpha \rightarrow \beta\gamma$$

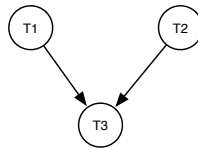
Aufgabe 12 (6 Punkte)

Betrachten Sie die drei Schedules mit den Transaktionen T_1 , T_2 und T_3 :

Schedule 1	Schedule 2	Schedule 3
T_1 : BOT	T_1 : BOT	T_1 : BOT
T_2 : BOT	T_2 : BOT	T_2 : BOT
T_3 : BOT	T_3 : BOT	T_3 : BOT
T_3 : lockX(a)	T_1 : lockX(a)	T_2 : lockX(a)
T_2 : lockX(c)	T_3 : lockX(b)	T_3 : lockX(a)
T_1 : lockX(a)	T_2 : lockX(c)	T_1 : lockX(c)
T_3 : lockX(b)	T_3 : lockX(a)	T_2 : lockX(c)
T_2 : lockX(b)	T_2 : lockX(b)	T_1 : unlock(c)
T_3 : lockX(d)	T_1 : lockX(c)	T_1 : lockX(b)

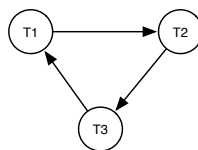
Zeichnen Sie den Wartegraphen zur Situation am Ende eines jeden Schedules. Liegt zu diesem Zeitpunkt ein Deadlock vor?

Schedule 1: Wartegraph:



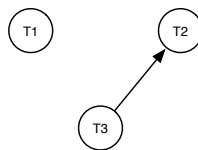
Deadlock? Ja Nein

Schedule 2: Wartegraph:



Deadlock? Ja Nein

Schedule 3: Wartegraph:



Deadlock? Ja Nein