

Beantworten Sie die Fragen in Aufgabe 1 durch Ankreuzen; pro Frage genau ein Kreuz. Ist Ihre Antwort richtig, erhalten Sie einen Punkt. Ist sie falsch, wird Ihnen ein Punkt abgezogen. Machen Sie kein Kreuz, bleibt die Punktzahl unverändert. Sie erhalten mindestens 0 Punkte.

Aufgabe 1 (10 Punkte)

	Frage	richtig	falsch
1.	Redundanz bedeutet, dass dieselben Informationen in unterschiedlichen Versionen gespeichert werden	<input type="radio"/>	<input checked="" type="radio"/>
2.	Durch Konsolidierung erhält man ein redundanzfreies globales Schema	<input checked="" type="radio"/>	<input type="radio"/>
3.	Das Gridfile ist eine zweidimensionale Suchstruktur mit variabler Gittergröße	<input type="radio"/>	<input checked="" type="radio"/>
4.	Beim Schreiben von Log-Informationen gilt das WAL-Prinzip	<input checked="" type="radio"/>	<input type="radio"/>
5.	Ein Schedule heißt serialisierbar, wenn alle Schritte einer Transaktion hintereinander ablaufen	<input type="radio"/>	<input checked="" type="radio"/>
6.	Ein <code>unique foreign key</code> modelliert eine 1:1-Beziehung	<input checked="" type="radio"/>	<input type="radio"/>
7.	Ein Relationenschema ist in 2. Normalform, wenn jedes Nichtprimär-Attribut funktional abhängig ist von jedem Schlüsselkandidaten	<input type="radio"/>	<input checked="" type="radio"/>
8.	Mit der <code>check</code> -Klausel wird in SQL statische Integrität hergestellt	<input checked="" type="radio"/>	<input type="radio"/>
9.	Beim offenen Hashing werden die Datensätze in sortierter Reihenfolge gespeichert	<input type="radio"/>	<input checked="" type="radio"/>
10.	Das Zeitstempelverfahren zur Vermeidung von Deadlocks verwendet die Time-out-Strategie	<input type="radio"/>	<input checked="" type="radio"/>

Aufgabe 2 (10 Punkte)

Gegeben seien folgende Entity- und Relationship-Typen:

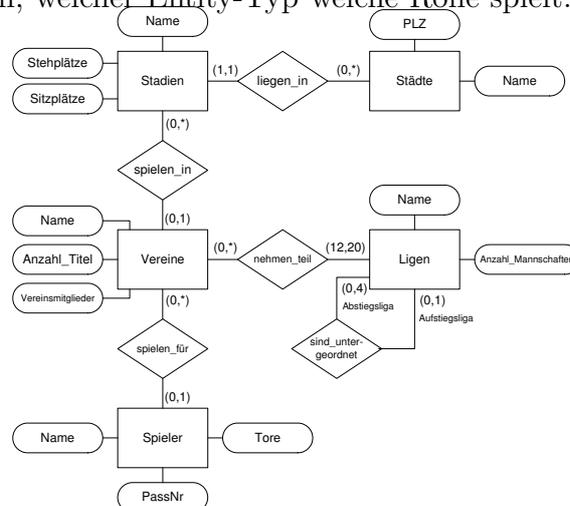
Entity-Typ	Attribute
Städte	PLZ, Name
Stadien	Name, Sitzplätze, Stehplätze
Vereine	Name, Anzahl_Titel, Vereinsmitglieder
Ligen	Name, Anzahl_Mannschaften
Spieler	Name, Tore, PassNr

Relationship-Typ
liegen_in
spielen_in
nehmen_teil
spielen_für
sind_untergeordnet

und folgende Beziehungen:

1. In einer Liga spielen immer 12 - 20 Mannschaften.
2. Ein Spieler darf in einer Saison höchstens für einen Verein spielen.
3. Ein Verein darf nur in maximal einem Stadion spielen.
4. In einem Stadion können mehrere Vereine spielen.
5. In einer Stadt können viele Stadien liegen.
6. Da ein Verein aus mehreren Mannschaften bestehen kann, kann er an mehreren Ligen teilnehmen
7. Für einen Verein können beliebig viele Spieler spielen.
8. Jeder Liga ist höchstens eine Aufstiegsliga übergeordnet, in die eine Mannschaft aufsteigen kann.
9. Jeder Liga können bis zu vier Abstiegsligen untergeordnet sein, in die eine Mannschaft absteigen kann.
10. Ein Stadion steht in genau einer Stadt.

Erstellen Sie mit den obigen Entity- und Relationship-Typen ein ER-Diagramm und geben Sie für die Beziehungen zwischen den Entity-Typen den Komplexitätsgrad in der (min, max)-Notation an. Falls ein Entity-Typ an einer Relation mehr als einmal beteiligt ist, machen Sie kenntlich, welcher Entity-Typ welche Rolle spielt:



Aufgabe 3 (7 Punkte)

Das ER-Diagramm aus der letzten Aufgabe sei in folgendes relationale Schema überführt worden:

Städte: {[PLZ, Name]}

Stadien: {[Stadionname, Sitzplätze, Stehplätze]}

Vereine: {[Vereinsname, Anzahl_Titel, Vereinsmitglieder]}

Ligen: {[Liganame, Anzahl_Mannschaften]}

Spieler: {[Name, Tore, PassNr]}

liegen_in: {[Name, PLZ]}

spielen_in: {[Vereinsname, Stadionname]}

nehmen_teil: {[Vereinsname, Liganame]}

spielen_für: {[PassNr, Vereinsname]}

sind_untergeordnet: {[Aufstiegsliga, Abstiegsliga]}

Verfeinern Sie das relationale Schema soweit wie möglich und schreiben Sie es erneut auf.

Stadien: {[Stadionname, Sitzplätze, Stehplätze, PLZ]}

Vereine: {[Vereinsname, Anzahl_Titel, Vereinsmitglieder, Stadionname]}

Ligen: {[Liganame, Anzahl_Mannschaften, Aufstiegsliga]}

Spieler: {[Name, Tore, PassNr, Vereinsname]}

Städte: {[PLZ, Name]}

nehmen_teil: {[Vereinsname, Liganame]}

Aufgabe 4 (8 Punkte)

Gegeben sei das unverfeinerte Schema aus Aufgabe 3:

Städte:	{ <u>[PLZ]</u> , Name}
Stadien:	{ <u>[Stadionname]</u> , Sitzplätze, Stehplätze}
Vereine:	{ <u>[Vereinsname]</u> , Anzahl_Titel, Vereinsmitglieder}
Ligen:	{ <u>[Liganame]</u> , Anzahl_Mannschaften}
Spieler:	{Name, Tore, <u>[PassNr]</u> }
liegen_in:	{ <u>[Name]</u> , PLZ}
spielen_in:	{ <u>[Vereinsname]</u> , Stadionname}
nehmen_teil:	{ <u>[Vereinsname]</u> , <u>[Liganame]</u> }
spielen_für:	{ <u>[PassNr]</u> , Vereinsname}
sind_untergeordnet:	{Aufstiegsliga, <u>[Abstiegliga]</u> }

Formulieren Sie die folgenden Abfragen in der Relationenalgebra:

- a) Wie heißen die Spieler, die bei dem Verein 'VfL Osnabrück' spielen? (2 Punkte)

$$\Pi_{Name}(\sigma_{Vereinsname='VfL Osnabrück'}(spielen_für) \bowtie Spieler)$$

- b) Welche Vereine spielen nicht in der '1.Bundesliga'? (3 Punkte)

$$Vereine - \Pi_{sch(Vereine)}(\sigma_{Liganame='1.Bundesliga'}(nehmen_teil) \bowtie Vereine)$$

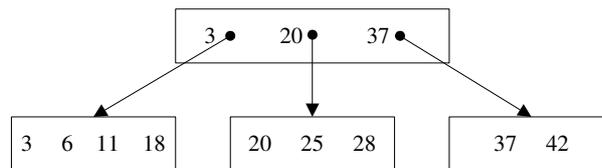
- c) In welchen Ligen spielen mindestens alle die Vereine, die im 'Westfahlenstadion' spielen? (3 Punkte)

$$nehmen_teil \div \Pi_{Vereinsname}(\sigma_{Stadionname='Westfahlenstadion'}(spielen_in))$$

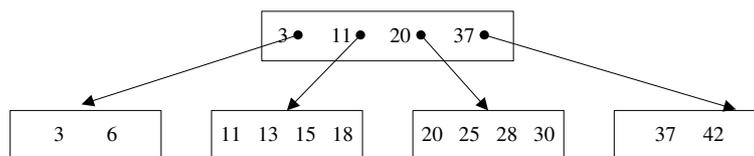
Aufgabe 5 (8 Punkte)

Zeichnen Sie den folgenden B*-Baum (k=2) jeweils

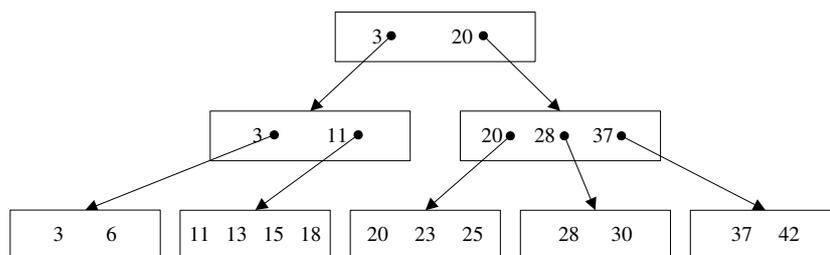
- a) nach dem Einfügen von 15, 13 und 30 (3 Punkte)
- b) nach dem Einfügen von 23 (2 Punkte)
- c) nach dem Löschen von 42 (3 Punkte)
einmal neu.



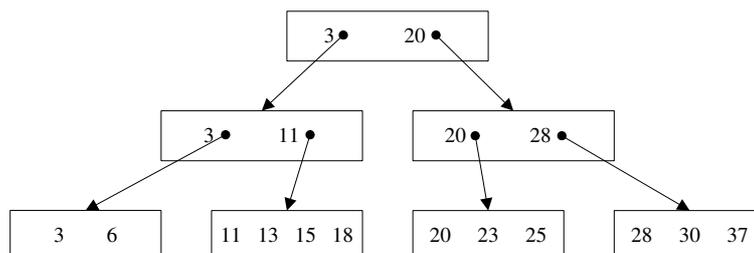
a) Nach dem Einfügen von 15, 13, 30:



b) Nach dem Einfügen von 23:



c) Nach dem Löschen von 42:



Aufgabe 6 (6 Punkte)

Sei ein B*-Baum gegeben mit $k = 10$ und drei Ebenen.

- a) Wieviele Elemente kann der Baum maximal enthalten? (1 Punkt)

8000 Elemente

- b) Wieviele Elemente kann der B*-Baum minimal enthalten, wenn bislang in den Baum nur Elemente eingefügt wurden, d.h. wenn noch keine Löschooperationen durchgeführt wurden? (3 Punkte)

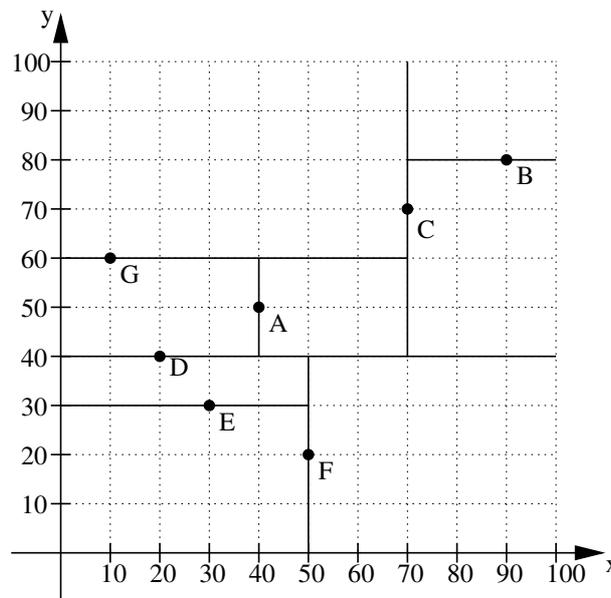
211 Elemente

- c) Wieviele Elemente kann der B*-Baum minimal enthalten, wenn bereits Löschooperationen durchgeführt wurden? (2 Punkte)

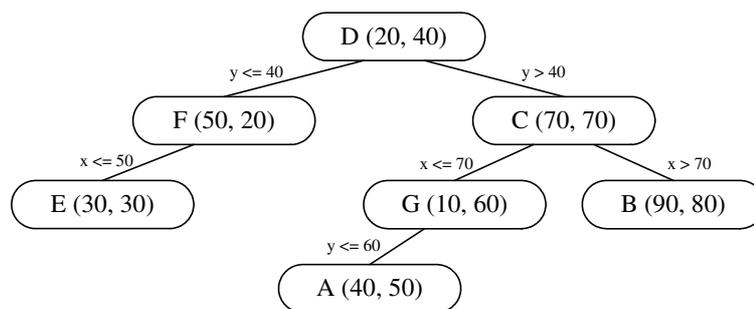
200 Elemente

Aufgabe 7 (8 Punkte)

Gegeben sei folgende durch die Punkte $A = (40, 50)$, $B = (90, 80)$, $C = (70, 70)$, $D = (20, 40)$, $E = (30, 30)$, $F = (50, 20)$ und $G = (10, 60)$ partitionierte Datenfläche:



- a) Zeichnen Sie den zugehörigen 2-d-Baum mit den jeweiligen Buchstaben und Schlüsseln in den Knoten und den Diskriminierungsbedingungen. (6 Punkte)



- b) Ist die Partitionierung durch die Reihenfolge, in der die Datenrecords in den 2-d-Baum eingefügt wurden, eindeutig festgelegt? (2 Punkte)

Nein

Ja

Aufgabe 8 (11 Punkte)

Gegeben seien die folgenden Relationen einer Universitätsdatenbank:

Professor: {[PersNr, Name, FBNr, Raum]}
Fachbereich: {[FBNr, FBName, DekanPersNr]}
Student: {[MatrNr, Name, Semester, FBNr]}
Pruefung: {[MatrNr, Fach, PrueferPersNr, Note]}

Anm.: die Attribute DekanPersNr und PrueferPersNr referenzieren einen Professor.

Formulieren Sie folgende Anfragen in SQL:

- a) Wie lauten die Matrikelnummern der Studenten, die eine Prüfung beim Dekan ihres Fachbereichs abgelegt haben? (2 Punkte)

```
SELECT DISTINCT s.MatrNr
FROM Pruefung p, Fachbereich f, Student s
WHERE f.DekanPersNr = p.PrueferPersNr
AND f.FBNr = s.FBNr
AND p.MatrNr = s.MatrNr
```

- b) Berechnen Sie die Durchschnittsnote für jeden Studenten. (2 Punkte)

```
SELECT MatrNr, avg(Note)
FROM Pruefung
GROUP BY MatrNr
```

- c) Wie lauten die Namen und Matrikelnummern der Studenten mit mindestens vier Semestern, die noch keine Prüfung abgelegt haben? (3 Punkte)

```
SELECT DISTINCT MatrNr, Name
FROM Student
WHERE Semester > 3
AND MatrNr NOT IN (select MatrNr from Pruefung)
```

- d) Wie lauten die Namen der Studenten, die ausschließlich Prüfungen bei Professoren ihres Fachbereichs abgelegt haben? (4 Punkte)

```
SELECT DISTINCT s1.Name
FROM Student s1, Pruefung p1
WHERE s1.MatrNr = p1.MatrNr
AND NOT EXISTS (
  SELECT * FROM Student s2, Pruefung p2, Professor prof
  WHERE s2.MatrNr = p1.MatrNr
  AND s2.MatrNr = p2.MatrNr
  AND prof.PersNr = p2.PrueferPersNr
  AND prof.FBNr != s.FBNr )
```

Aufgabe 9 (6 Punkte)

Gegeben seien die folgenden Relationen aus einer Oldtimerclub-Datenbank (Schlüssel unterstrichen):

Mitglieder: {[Mitgliedsnr, Name, Alter]}
besitzen: {[Mitgliedsnr, Kennzeichen]}
Oldtimer: {[Kennzeichen, Alter, Wert, Farbe]}

Wie lauten folgende SQL-Anfragen umgangssprachlich?

a) SQL-Anfrage:

```
SELECT DISTINCT m.Name
  FROM Mitglieder m, besitzen b, Oldtimer o
 WHERE m.Mitgliedsnr = b.Mitgliedsnr
       AND b.Kennzeichen = o.Kennzeichen
       AND m.Alter < o.Alter
```

Umgangssprachlich (3 Punkte):

Wie heißen die Clubmitglieder, die jünger sind als einer ihrer Oldtimer?

b) SQL-Anfrage:

```
SELECT m.Alter
  FROM Mitglieder m, besitzen b, Oldtimer o
 WHERE m.Mitgliedsnr = b.Mitgliedsnr
       AND b.Kennzeichen = o.Kennzeichen
       AND o.Wert = (SELECT max(Wert) FROM Oldtimer)
```

Umgangssprachlich (3 Punkte):

Wie alt sind die Mitglieder, die die Oldtimer mit maximalem Wert fahren?

Aufgabe 10 (4 Punkte)

Gegeben sei folgende XML-Datei, die eine Einkaufsliste repräsentiert:

```
<?xml version="1.0" ?>
<!DOCTYPE einkauf SYSTEM "einkaufsliste.dtd">
<einkauf>
  <artikel>
    <beschreibung>Milch</beschreibung>
    <menge>3</menge>
    <einzelpreis>0,55</einzelpreis>
    <gesamtpreis>1,65</gesamtpreis>
  </artikel>
  <artikel>
    <menge>8</menge>
    <einzelpreis>0,99</einzelpreis>
    <gesamtpreis>7,92</gesamtpreis>
  </artikel>
  <artikel>
    <beschreibung>Broetchen</beschreibung>
    <menge>10</menge>
    <einzelpreis>0,25</einzelpreis>
    <gesamtpreis>2,50</gesamtpreis>
  </artikel>
  <artikel>
    <beschreibung>Kiste Wasser</beschreibung>
    <einzelpreis>5,50</einzelpreis>
    <gesamtpreis>5,50</gesamtpreis>
  </artikel>
  <gesamtpreis>17,57</gesamtpreis>
</einkauf>
```

Schreiben Sie für diese XML-Datei eine DTD, gegen die die XML-Datei erfolgreich validiert werden kann. Schränken Sie dabei die Anzahl der Unterelemente sinnvoll ein:

```
<?xml version="1.0" encoding="utf-8" ?>
<!ELEMENT einkauf (artikel+, gesamtpreis) >
<!ELEMENT artikel (beschreibung?, menge?, einzelpreis, gesamtpreis) >
<!ELEMENT beschreibung (#PCDATA) >
<!ELEMENT einzelpreis (#PCDATA) >
<!ELEMENT gesamtpreis (#PCDATA) >
<!ELEMENT menge (#PCDATA) >
```

Aufgabe 11 (6 Punkte)

Gegeben sei folgende fehlerhafte XML-Datei `fehler.xml` mit Zeilennummern:

```

1 <?xml version="1,0" ?>
2 <!DOCTYPE elemente SYSTEM "fheler.dtd" >
3 <elemente>
4   <element1 oben="ja" >
5   <element2 inhalt="text">42</element2>
6   <element3 inhalt="element">
7     <element4>text</element4>
8   </element3>
9 </element>

```

und folgende DTD `fehler.dtd`:

```

<?xml encoding="UTF-8" ?>
<!ELEMENT elemente (element1|element2|element3)* >
<!ELEMENT element1 EMPTY >
<!ATTLIST element1 oben CDATA #REQUIRED>
<!ELEMENT element2 (#PCDATA) >
<!ATTLIST element2 inhalt CDATA #REQUIRED>
<!ELEMENT element3 (element4) >
<!ATTLIST element4 inhalt CDATA #IMPLIED>
<!ELEMENT element4 EMPTY >

```

Die XML-Datei enthält semantische und syntaktische Fehler. Finden Sie die Fehler und korrigieren Sie die XML-Datei, so dass sie syntaktisch korrekt ist und bzgl. der DTD erfolgreich validiert werden kann. Dabei dürfen einzelne Bestandteile, aber keine ganzen Zeilen gelöscht werden.

Notieren Sie die entsprechende Zeilennummer und die korrigierte Zeile:

Zeilennr.	korrigierte Zeile
1	<?xml version="1.0" ?>
2	<!DOCTYPE elemente SYSTEM "fehler.dtd" >
4	<element1 oben="ja" />
6	<element3>
7	<element4/>
9	</elemente>

Aufgabe 12 (6 Punkte)

Gegeben sei das Relationenschema

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

mit den funktionalen Abhängigkeiten

$$\begin{aligned} BC &\rightarrow C \\ C &\rightarrow AD \\ D &\rightarrow CE \\ E &\rightarrow BC \\ F &\rightarrow D \end{aligned}$$

- a) Nennen Sie alle Schlüsselkandidaten für R . (2 Punkte)

F ist der einzige Schlüsselkandidat:

F muss in jedem Schlüsselkandidaten enthalten sein, da es von keinem anderen Attribut funktional abhängig ist. Da F aber bereits ein Superschlüssel ist, ist F der einzige Schlüsselkandidat.

- b) R ist in der 1. Normalform. Ist R auch in der 2. Normalform? Begründen Sie Ihre Antwort. (2 Punkte)

Da F als einziger Schlüsselkandidat einelementig ist, sind alle anderen Attribute voll funktional abhängig von F , daher ist R in 2. Normalform.

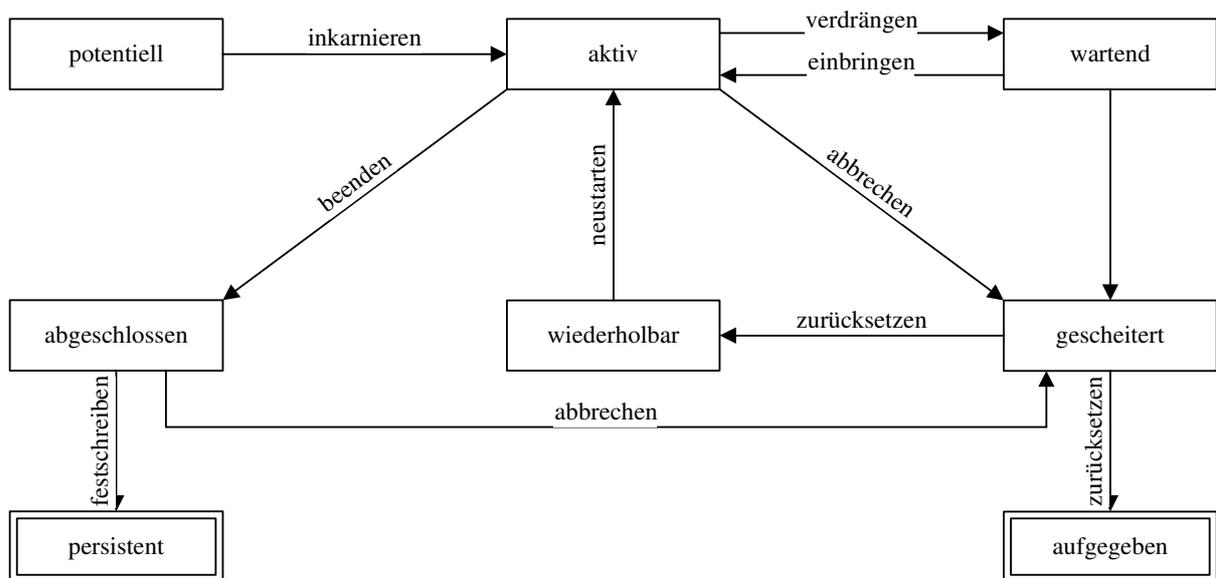
- c) Warum ist R nicht in der 3. Normalform? (2 Punkte)

Beispielsweise ist E von F transitiv abhängig:

$$F \xrightarrow{D} E$$

Aufgabe 13 (4 Punkte)

Das folgende Diagramm enthält die Zustandsübergänge einer Transaktion. Benennen Sie die möglichen Zustände:



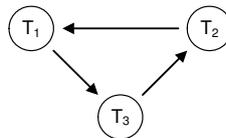
Aufgabe 14 (6 Punkte)

Betrachten Sie die drei Schedules mit den Transaktionen T_1, T_2, 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_2 : lockX(a)	T_2 : lockX(b)	T_1 : lockX(c)
T_3 : lockX(c)	T_3 : lockX(b)	T_2 : lockX(a)
T_3 : lockX(a)	T_1 : lockX(a)	T_3 : lockX(b)
T_1 : lockX(b)	T_1 : lockX(b)	T_2 : unlockX(a)
T_2 : lockX(b)	T_2 : unlockX(b)	T_2 : lockX(c)
T_1 : lockX(c)	T_3 : lockX(a)	T_1 : lockX(b)
...

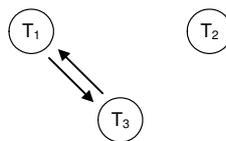
Zeichnen Sie den Wartegraphen zur Situation am Ende eines jeden Schedules. Liegt zu diesem Zeitpunkt ein Deadlock vor?

Schedule 1: Wartegraph:



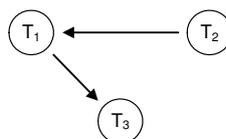
Deadlock? Ja Nein

Schedule 2: Wartegraph:



Deadlock? Ja Nein

Schedule 3: Wartegraph:



Deadlock? Ja Nein