

Datenbanksysteme II

Name:

Vorname:

Matrikelnummer:

Studienfach:

Wichtige Hinweise:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit (16 Seiten).
2. Es sind keine Hilfsmittel zugelassen, insbesondere keine elektronischen Geräte (z.B. Mobiltelefone). **Eingeschaltete elektronische Geräte gelten als Täuschungsversuch.**
3. Die Klausur dauert 100 Minuten.
4. Jede Aufgabe ist möglichst auf dem zugehörigen Aufgabenblatt zu bearbeiten. Nur falls eine Lösung durch selbst auf dem Blatt gemachte Korrekturen nicht mehr eindeutig erkennbar ist, darf ein separates Blatt zur Lösung verwendet werden.
5. Vermerken Sie Ihren Namen und Ihre Matrikelnummer auf jedem Aufgaben- (bzw. Lösungsblatt). Blätter ohne Namens- und Matrikelnummerangabe werden nicht bewertet.
6. Das Deckblatt sowie alle Aufgabenblätter (evtl. Lösungsblätter) sind abzugeben.

	maximale Anzahl Punkte	erreichte Anzahl Punkte
Aufgabe 1	7	
Aufgabe 2	8	
Aufgabe 3	21	
Aufgabe 4	26	
Aufgabe 5	15	
Aufgabe 6	10	
Aufgabe 7	13	
Variante A	100	

Aufgabe 1

7 Punkte

Kreuzen Sie alle zutreffenden Antworten an (Mehrfachantworten sind möglich).

Aufgabe 1 a)

2 Punkte

Welchen der folgenden Typen kann ein XPath 1.0-Ausdruck haben?

<input type="checkbox"/>	xs:date	<input type="checkbox"/>	string	<input type="checkbox"/>	number	<input type="checkbox"/>	decimal
<input type="checkbox"/>	boolean	<input type="checkbox"/>	locator	<input type="checkbox"/>	node	<input type="checkbox"/>	context
<input type="checkbox"/>	arc	<input type="checkbox"/>	node-set	<input type="checkbox"/>	namespace	<input type="checkbox"/>	text()

Aufgabe 1 b)

5 Punkte

Gegeben das XML-Dokument

```
<?xml version="1.0"?>
<a xmlns="http://pi3.informatik.uni-mannheim.de/dbs2"
  xmlns:pi3="http://pi3.informatik.uni-mannheim.de/dbs2"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <b>
    <pi3:a xmlns="http://www.uni-mannheim.de">
      <c xmlns="http://pi3.informatik.uni-mannheim.de/dbs2"
        xmlns:pi3="http://www.uni-mannheim.de" />
      <pi3:c xmlns="http://pi3.informatik.uni-mannheim.de/dbs2">
        <c/>
      </pi3:c>
    </pi3:a>
  </b>
</a>
```

Markieren Sie alle Zellen der folgenden Matrix, wenn der zugehörige lokale Name im Dokument im entsprechenden Namensraum vorkommt.

Namensraum	Lokaler Name		
	a	b	c
http://pi3.informatik.uni-mannheim.de/dbs2			
http://www.uni-mannheim.de			
http://www.w3.org/2001/XMLSchema			
In keinem Namensraum			

Aufgabe 2**8 Punkte**

Kreuzen Sie alle zutreffenden Antworten an (Mehrfachantworten sind möglich).

Aufgabe 2 a)**2 Punkte**

Eine vertikale Fragmentierung ist oft nicht disjunkt, weil

- man einen Schlüssel für die Rekonstruierbarkeit braucht
- die Fragmentierung zusätzliche Relationen zur Verwaltung benötigt
- man neue Tupel immer in alle Fragmente einfügt
- Keine der anderen Aussagen trifft zu

Aufgabe 2 b)**2 Punkte**

Eine horizontale Fragmentierung ist oft nicht disjunkt, weil

- man einen Schlüssel für die Rekonstruierbarkeit braucht
- die Fragmentierung zusätzliche Relationen zur Verwaltung benötigt
- man neue Tupel immer in alle Fragmente einfügt
- Keine der anderen Aussagen trifft zu

Aufgabe 2 c)**2 Punkte**

Welche Form der Transparenz ist in einem verteilten DBMS gegeben, wenn bei einem Update die Tupel, deren Fragmentzugehörigkeit sich geändert hat, automatisch in das richtige Fragment verschoben werden?

Antwort

Aufgabe 2 d)**2 Punkte**

Welche der folgenden Protokolle werden für die Sicherung der Transaktionseigenschaften in verteilten DBMS eingesetzt:

- 2-Phasen-Semijoin-Auswertung
- 2-Phasen-Sperrprotokoll
- 2-Phasen-Deadlock-Erkennung
- 2-Phasen-Commit-Protokoll

 Aufgabe 3

21 Punkte

Gegeben Sei das folgende Dokument:

```

1 <?xml version="1.0"?>
2 <rezepte>
3   <rezept name="Lasagne">
4     <tag>Italien</tag><tag>Pasta</tag>
5     <zutaten>
6       <zutat>Hackfleisch</zutat>
7       <zutat>Eiernudelteig</zutat>
8       <zutat>Bechamelsauce</zutat>
9     </zutaten>
10  </rezept>
11 <rezept name="Fleischleiberl">
12   <tag>Oesterreich</tag>
13   <zutaten>
14     <zutat>Hackfleisch</zutat>
15     <zutat>Zwiebeln</zutat>
16     <zutat>Knoblauch</zutat>
17   </zutaten>
18 </rezept>
19 <rezept name="Vitello tonnato">
20   <tag>Italien</tag><tag>Buffet</tag>
21   <zutaten>
22     <zutat>Thunfisch</zutat>
23     <zutat>Kalbfleisch</zutat>
24   </zutaten>
25 </rezept>
26 </rezepte>
  
```

Geben Sie für folgende XPath-1.0-Ausdrücke jeweils den Typ und das Ergebnis an. Zur Angabe von Knotenmengen geben Sie die Zeilennummern an, in der die textuelle Repräsentation der Knoten beginnt (z.B. bei Elementen die Zeile des Open-Tags).

Beispiel: Das Ergebnis von `/descendant-or-self::rezept` ist 3, 11, 19.

Aufgabe 3 a)

2 Punkte

$$\text{count}(\text{//zutat})$$

 Typ

 Ergebnis

Aufgabe 3 b)

2 Punkte

$$\text{/descendant-or-self::rezept}[1]/\text{@name}$$

 Typ

 Ergebnis

Aufgabe 3 c)

2 Punkte

```
string(//rezept[tag='Buffet']/@name)
```

Typ

Ergebnis

Aufgabe 3 d)

3 Punkte

```
count(/self::*)
```

Typ

Ergebnis

Aufgabe 3 e)

3 Punkte

```
//zutat[1]
```

Typ

Ergebnis

Aufgabe 3 f)

3 Punkte

```
count(//rezept[tag='Oesterreich' and not(descendant::zutat='Knoblauch')])
```

Typ

Ergebnis

Aufgabe 3 g)

3 Punkte

```
//rezept[tag='Oesterreich' and descendant::zutat!='Knoblauch']
```

Typ

Ergebnis

Aufgabe 3 h)

3 Punkte

```
/descendant::zutat[ancestor::* /tag='Oesterreich']
```

Typ

Ergebnis

Aufgabe 4

26 Punkte

Gegeben sei folgendes Dokument:

```
<?xml version="1.0"?>
<a id="1">
  <b id="2">
    <a id="3">not OK</a>
    <a id="4">OK</a>
    <a id="5">unknown</a>
  </b>
  <b id="6">
    <c id="7"/>
    <c id="8"/>
    <b id="9"><a id="10">OK</a></b>
  </b>
</a>
```

Vervollständigen Sie die folgenden XQuery-Anfragen nur durch Ergänzungen an den vorgesehenen Stellen (_ _ _), so daß das vorgegebene Ergebnis herauskommt. Es dürfen nicht mehr Zeichen ergänzt werden, als jeweils Unterstriche vorhanden sind, weniger sind jedoch erlaubt (_ _ _ bedeutet, daß 0, 1, 2 oder 3 Zeichen ergänzt werden dürfen).

Aufgabe 4 a)

2 Punkte

Gewünschtes Ergebnis

```
<x>6</x>
```

Anfrage

```
let $x := /descendant::b[___]
return <x> { fn:string($x/@id) } </x>
```

Das Dokument zur Wiederholung:

```
<?xml version="1.0"?>
<a id="1">
  <b id="2">
    <a id="3">not OK</a>
    <a id="4">OK</a>
    <a id="5">unknown</a>
  </b>
  <b id="6">
    <c id="7"/>
    <c id="8"/>
    <b id="9"><a id="10">OK</a></b>
  </b>
</a>
```

Aufgabe 4 b)

5 Punkte

Gewünschtes Ergebnis

```
<t><x>1</x><y>2</y></t>,
<t><x>1</x><y>6</y></t>
```

Anfrage

```
----- $x ----- //a
----- $y ----- $x___b
return <t> <x> { fn:string($x/@id) } </x>
        <y> { fn:string($y/@id) } </y>
        </t>
```

Das Dokument zur Wiederholung:

```
<?xml version="1.0"?>
<a id="1">
  <b id="2">
    <a id="3">not OK</a>
    <a id="4">OK</a>
    <a id="5">unknown</a>
  </b>
  <b id="6">
    <c id="7"/>
    <c id="8"/>
    <b id="9"><a id="10">OK</a></b>
  </b>
</a>
```

Aufgabe 4 c)

4 Punkte

Gewünschtes Ergebnis

```
<t><x>1</x><y>2</y></t>,
<t><x>3</x><y>0</y></t>,
<t><x>4</x><y>0</y></t>,
<t><x>5</x><y>0</y></t>,
<t><x>10</x><y>0</y></t>
```

Anfrage

```
----- $x ----- //a
----- $y ----- $x/b
return <t> <x> { fn:string($x/@id) } </x>
          <y> { fn:count($y) } </y>
        </t>
```

Das Dokument zur Wiederholung:

```
<?xml version="1.0"?>
<a id="1">
  <b id="2">
    <a id="3">not OK</a>
    <a id="4">OK</a>
    <a id="5">unknown</a>
  </b>
  <b id="6">
    <c id="7"/>
    <c id="8"/>
    <b id="9"><a id="10">OK</a></b>
  </b>
</a>
```

Aufgabe 4 d)

4 Punkte

Gewünschtes Ergebnis

2, 3

Anfrage

```
___ $x ___ //b___
where $x eq "OK"
return fn:count($x/_____::*)
```

Das Dokument zur Wiederholung:

```
<?xml version="1.0"?>
<a id="1">
  <b id="2">
    <a id="3">not OK</a>
    <a id="4">OK</a>
    <a id="5">unknown</a>
  </b>
  <b id="6">
    <c id="7"/>
    <c id="8"/>
    <b id="9"><a id="10">OK</a></b>
  </b>
</a>
```

Aufgabe 4 e)

5 Punkte

Gewünschtes Ergebnis

1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Anfrage

```
for $x in /**
for $y in /**
where ____ $x _____ $y _____
return fn:number($y/@id)
```

Das Dokument zur Wiederholung:

```
<?xml version="1.0"?>
<a id="1">
  <b id="2">
    <a id="3">not OK</a>
    <a id="4">OK</a>
    <a id="5">unknown</a>
  </b>
  <b id="6">
    <c id="7"/>
    <c id="8"/>
    <b id="9"><a id="10">OK</a></b>
  </b>
</a>
```

Aufgabe 4 f)

6 Punkte

Gewünschtes Ergebnis

```
<t text="not OK">1</t>,
<t text="OK">2</t>
```

Anfrage

```
let $val:=for $x in //text()
          where fn:contains($x,"OK")
          return fn:string($x)
for $t in fn:_____
return <t text="{ $t }">
_____
```

Aufgabe 5

15 Punkte

Gegeben seien die folgenden EDB-Relationen mit zugehörigem Datalog-Programm:

WeinName		Jahrgang		Rebsorte		
ID	Name	ID	Jahr	ID	Sorte	Anteil
30	Chateau Villars	30	1998	30	Cabernet Sauvignon	7
42	Quinta Hinojal	42	2003	30	Cabernet Franc	20
81	Forster Winzerverein	81	2003	30	Merlot	73
				42	Tempranillo	100
				81	Riesling	100

$cuvee(X) :- S \neq R, rebsorte(X,R,Z), rebsorte(X,S,T).$

$gleicherJahrgang(X,Y) :- jahrgang(X,Z), jahrgang(Y,Z).$

$aehnlich(X,Y) :- gleicherJahrgang(X,Y)$

$aehnlich(X,Y) :- rebsorte(X,R,Z), rebsorte(Y,R,T), Z > 20, T > 20.$

Aufgabe 5 a)

5 Punkte

Geben Sie den Abhängigkeitsgraphen des Programms an.

Aufgabe 5 b)

1 Punkte

Ist das Programm rekursiv?

ja nein

Aufgabe 5 c)

2 Punkte

Ist die Regel für *cuvee* sicher?

ja nein

Aufgabe 5 d)

2 Punkte

Ist das Programm stratifiziert?

 ja nein

Aufgabe 5 e)

5 Punkte

Welcher relationale Algebraausdruck ergibt als Ergebnis die IDB-Relation **Aehnlich**? (Die Umbenennung von Attributen ist hier mit $\Pi_{X:Y}$ bezeichnet. In der Vorlesung Datenbanksysteme I wurde hierfür auch die alternative Notation $\rho_{X \leftarrow Y}$ verwendet). Nur eine der Antworten ist richtig.

$$\begin{array}{c}
 \Pi_{X,Y}(\Pi_{X:ID,R:Sorte,Z:Anteil}(Rebsorte)) \\
 \bowtie \\
 \square \quad \sigma_{Z>20 \wedge T>20}(\Pi_{Y:ID,R:Sorte,T:Anteil}(Rebsorte))) \\
 \cup \\
 \Pi_{X,Y}(\text{Gleicher Jahrgang})
 \end{array}$$

$$\begin{array}{c}
 \Pi_{X,Y}(\sigma_{Z>20 \wedge T>20}(\Pi_{X:ID,R:Sorte,Z:Anteil}(Rebsorte)) \\
 \bowtie \\
 \square \quad \Pi_{Y:ID,R:Sorte,T:Anteil}(Rebsorte))) \\
 \cup \\
 \Pi_{X,Y}(\text{Gleicher Jahrgang})
 \end{array}$$

$$\begin{array}{c}
 \Pi_{X,Y}(\sigma_{Z>20 \wedge T>20}(\Pi_{X:ID,R:Sorte,Z:Anteil}(Rebsorte)) \\
 \cup \\
 \square \quad \Pi_{Y:ID,R:Sorte,T:Anteil}(Rebsorte))) \\
 \bowtie \\
 \Pi_{X,Y}(\text{Gleicher Jahrgang})
 \end{array}$$

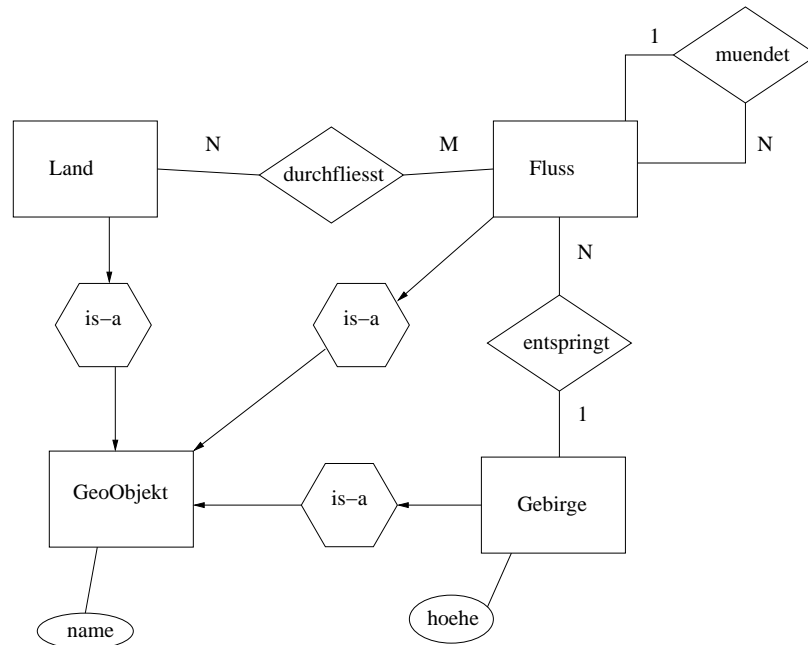
$$\begin{array}{c}
 \Pi_{X,Y}(\sigma_{Z>20}(\Pi_{X:ID,R:Sorte,Z:Anteil}(Rebsorte)) \\
 \bowtie \\
 \square \quad (\sigma_{T>20}(\Pi_{Y:ID,R:Sorte,T:Anteil}(Rebsorte)))) \\
 \bowtie \\
 \Pi_{X,Y}(\text{Gleicher Jahrgang})
 \end{array}$$

$$\begin{array}{c}
 \Pi_{X,Y}(\Pi_{X:ID,R:Sorte,Z:Anteil}(Rebsorte)) \\
 \cup \\
 \square \quad \Pi_{Y:ID,R:Sorte,T:Anteil}(Rebsorte)) \\
 \bowtie_{Z>20 \wedge T>20} \\
 \Pi_{X,Y}(\text{Gleicher Jahrgang})
 \end{array}$$

Aufgabe 6

10 Punkte

Setzen Sie folgendes ER-Diagramm in ein ODL-Schema um:



Geben Sie jeder Klasse eine Extension. Verwenden Sie die Datentypen `string` für Namen, `float` für die Höhe. Beziehungen sollen so angegeben sein, daß die inverse Richtung automatisch durch das DBMS gepflegt werden kann.

(Weiterer Platz zum Schreiben auf der Folgeseite)

Name: _____ Matr.Nr.: _____

15

(Fortsetzung Aufgabe 6)

Aufgabe 7

13 Punkte

Aufgabe 7 a)

8 Punkte

Skizzieren Sie den algebraischen Ausdruck, der aus der kanonischen Übersetzung des XPath-Ausdrucks `/descendant::A/descendant::B/child::C` resultiert.

Aufgabe 7 b)

5 Punkte

Wie sieht die gestapelte Übersetzung (*Stacked Translation*) des Ausdrucks aus Teil a) aus?