

Datenbanksysteme II

Name:

Vorname:

Matrikelnummer:

Studienfach:

Unterschrift:

Wichtige Hinweise:

1. Prüfen Sie Ihr Klausurexemplar auf Vollständigkeit (18 Seiten, doppelseitig).
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	13	
Aufgabe 2	12	
Aufgabe 3	10	
Aufgabe 4	21	
Aufgabe 5	26	
Aufgabe 6	18	
Variante A	100	

 Aufgabe 1

13 Punkte

Beantworten Sie die folgenden Fragen zu verteilten DBMS.

Aufgabe 1 a)

2 Punkte

Welche Formen der Fragmentierung gibt es? Kreuzen Sie alle zutreffenden Antworten an.

- Horizontale Fragmentierung
- Totale Fragmentierung
- Kombinierte Fragmentierung
- Abgeleitete vertikale Fragmentierung
- Rekursive Fragmentierung
- Keine der oben genannten Formen

Aufgabe 1 b)

6 Punkte

Welche der folgenden Aussagen sind korrekt?

Richtig	Falsch	
		Die erste Phase im 2-Phasen-Sperrprotokoll heisst PREPARE-Phase.
		Wenn alle lokalen Transaktionen serialisierbar sind, ist die globale Transaktion serialisierbar.
		Ein Allokationsschema weist jedem Fragment genau eine Station zu.
		Eine korrekte Fragmentierung garantiert <i>immer</i> die Rekonstruierbarkeit.
		Die Modifikation eines Tupels kann die Fragmentzugehörigkeit eines anderen Tupels verändern.

Aufgabe 1 c)

5 Punkte

Zeichnen Sie das Zustandsdiagramm des Koordinators im Zwei-Phasen-Commit-Protokoll. Es reichen die Zustände und die möglichen Übergänge, die Zustandsübergänge müssen *nicht* beschriftet werden.

 Aufgabe 2

12 Punkte

Betrachten Sie folgendes Szenario in einem verteiltem Datenbanksystem. Die Relation R wird auf der Station S_R gehalten, die Relation T auf der Station S_T . Auf Station S_R wird das Ergebnis des natürlichen Verbunds (Natural Join) von R und T benötigt.

R			
A	B	C	D
1	7	2	1
2	5	3	4
3	4	3	2
4	6	2	1
5	5	7	9

T		
D	E	F
5	5	12
2	3	18
3	8	19
5	7	27
2	2	12
1	5	19

Ermitteln Sie den günstigsten Ausführungsplan für diese verteilte Anfrage, unter der Annahme, dass die Kostenfunktion die Anzahl der insgesamt versendeten Attributwerte ist. Geben Sie mindestens drei verschiedene Planalternativen an und vergleichen Sie die entstehenden Kosten.

Name: _____ Matr.Nr.: _____

 Aufgabe 3

10 Punkte

Aufgabe 3 a)

2 Punkte

Welche Möglichkeiten gibt es, in einer XML-Dokumentinstanz anzugeben, wo passende XML-Schemadokumente zu finden sind?

Aufgabe 3 b)

5 Punkte

Welche der folgenden Aussagen sind korrekt?

Richtig	Falsch	
		Ein XML-Schema besteht aus genau einem XML-Schemadokument.
		Ein XML-Schemadokument hat genau einen Target Namespace.
		Die <code>include</code> -Direktive bindet Schemadokumente mit abweichendem Target Namespace ein.
		Ein Attribut ohne Prefix ist im Default-Namespace.
		Jedes global definierte Element kann Wurzelement sein.

Aufgabe 3 c)

3 Punkte

Gegeben sei der folgende fehlerhafte Ausschnitt aus einem Schemadokument.

```
<xsd:complexType name="Kudentyp">
  <xsd:sequence>
    <xsd:element name="Kundennummer" type="KdNrTyp"
      minOccurs="1" />
    <xsd:element name="Name" type="xsd:string"
      minOccurs="1" />
    <xsd:element name="Anschrift" type="AddressTyp"
      minOccurs="1" maxOccurs="unbounded" />
    <xsd:element name="Telefon" type="TelNrTyp"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="Telefonkontakt">
  <xsd:complexContent>
    <xsd:restriction base="Kudentyp">
      <xsd:sequence>
        <xsd:element name="Kundennummer" type="KdNrTyp"
          minOccurs="1" />
        <xsd:element name="Name" type="xsd:string"
          minOccurs="1" />
        <xsd:element name="Telefon" type="TelNrTyp" />
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

Warum handelt es sich nicht um ein gültiges XML-Schemafragment?

 Aufgabe 4

21 Punkte

Gegeben sei das folgende Dokument:

```

1 <?xml version="1.0"?>
2 <buch name="Datenbanksysteme">
3   <autor>Kemper</autor>
4   <autor>Eickler</autor>
5   <kapitel nr="16" name="Verteilte Datenbanken">
6     <seiten von="443" bis="482"/>
7     <stichwort>Fragmentierung</stichwort>
8     <stichwort>2PC</stichwort>
9     <stichwort>Verteilt</stichwort>
10    <abschnitt name="Terminologie und Abgrenzung"/>
11    <abschnitt name="Entwurf verteilter Datenbanken"/>
12  </kapitel>
13  <kapitel nr="19" name="XML-Datenmodellierung und Web-Services">
14    <seiten von="535" bis="578"/>
15    <stichwort>Verteilt</stichwort>
16    <stichwort>XML</stichwort>
17    <abschnitt name="XML-Datenmodellierung"/>
18    <abschnitt name="XQuery: Eine XML-Anfragesprache"/>
19  </kapitel>
20  <index name="Index">
21    <seiten von="626" bis="640"/>
22  </index>
23 </buch>

```

Geben Sie für folgende XPath-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::autor` ist 3, 4.

Aufgabe 4 a)

2 Punkte

$$\text{fn:count}(\text{//kapitel})$$

 Typ

 Ergebnis

Aufgabe 4 b)

2 Punkte

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

 Typ

 Ergebnis

Aufgabe 4 c)

2 Punkte

```
string(//kapitel[@nr=19]/@name)
```

Typ

Ergebnis

Aufgabe 4 d)

3 Punkte

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

Typ

Ergebnis

Aufgabe 4 e)

3 Punkte

```
//abschnitt[2]
```

Typ

Ergebnis

Aufgabe 4 f)

3 Punkte

```
fn:count(//kapitel[stichwort='Verteilt' and not(descendant::stichwort='XML')])
```

Typ

Ergebnis

Aufgabe 4 g)

3 Punkte

```
//kapitel[stichwort='Verteilt' and descendant::stichwort!='XML']
```

Typ

Ergebnis

Aufgabe 4 h)

3 Punkte

```
/descendant::kapitel[seiten[@von <= 540 and 540 <= @bis]]/@name
```

Typ

Ergebnis

Aufgabe 5

26 Punkte

Gegeben sei folgendes XML Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="organigramm">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" minOccurs="1" type="xs:string" />
        <xs:element minOccurs="1" ref="person" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" minOccurs="1" type="xs:string" />
        <xs:element name="gehalt" minOccurs="0" type="xs:decimal" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="person" />
      </xs:sequence>
      <xs:attribute name="art" use="required" type="xs:string" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Dabei werden `person`-Tags geschachtelt, um zu repräsentieren, dass das Elternelement der Vorgesetzte des geschachtelten Elements ist. Ein Beispieldokument für dieses Schema ist:

```
<organigramm>
  <name>Suedfrucht AG</name>
  <person art="Manager">
    <name>Edmund</name> <gehalt>115000</gehalt>
    <person art="Manager">
      <name>Erwin</name> <gehalt>85000</gehalt>
      <person art="Arbeiter"> <name>Monika</name> </person>
    </person>
    <person art="Manager">
      <name>Horst</name> <gehalt>80000</gehalt>
      <person art="Ingenieur"> <name>Hubertus</name> <gehalt>85000</gehalt>
    </person>
  </person>
</organigramm>
```

Geben Sie XQuery-Ausdrücke für die folgenden Anfragen an. Gehen Sie davon aus, dass als Context Item die Wurzel des Dokuments gegeben ist.

Aufgabe 5 a)

3 Punkte

Geben Sie den Namen aller Manager als Sequenz von **name**-Elementen an.

Aufgabe 5 b)

3 Punkte

Geben Sie die Namen aller Personen aus, die keine Untergebenen haben, und zwar als Sequenz von Strings.

Aufgabe 5 c)

4 Punkte

Sortieren Sie die Namen der Personen absteigend nach dem Gesamtgehalt aller direkt oder indirekt Untergebenen (inkl. des eigenen Gehalts). Das Ergebnis soll als Sequenz von Strings vorliegen.

Aufgabe 5 d)

8 Punkte

Geben Sie das durchschnittliche Gehalt für jede Art von Person aus (Personen ohne Gehalt-Tag sollen bei der Durchschnittsberechnung nicht berücksichtigt werden, Personenarten bei denen keine Person ein Gehalt-Tag hat sollen als Durchschnittsgehalt 0 angeben). Verwenden Sie jeweils die Art der Person als Tag-Namen, in der Form

```
<gehaelter>  
  <Manager>90000</Manager>  
  <Arbeiter>0</Arbeiter>  
  <Ingenieur>85000</Ingenieur>  
</gehaelter>
```

Aufgabe 5 e)

8 Punkte

Erzeugen Sie ein Dokument, in dem alle Personen direkte Kinder des Dokumentelements sind. Ein Attribut **boss** an den Personenelementen soll, wenn vorhanden, den Namen des Vorgesetzten enthalten.

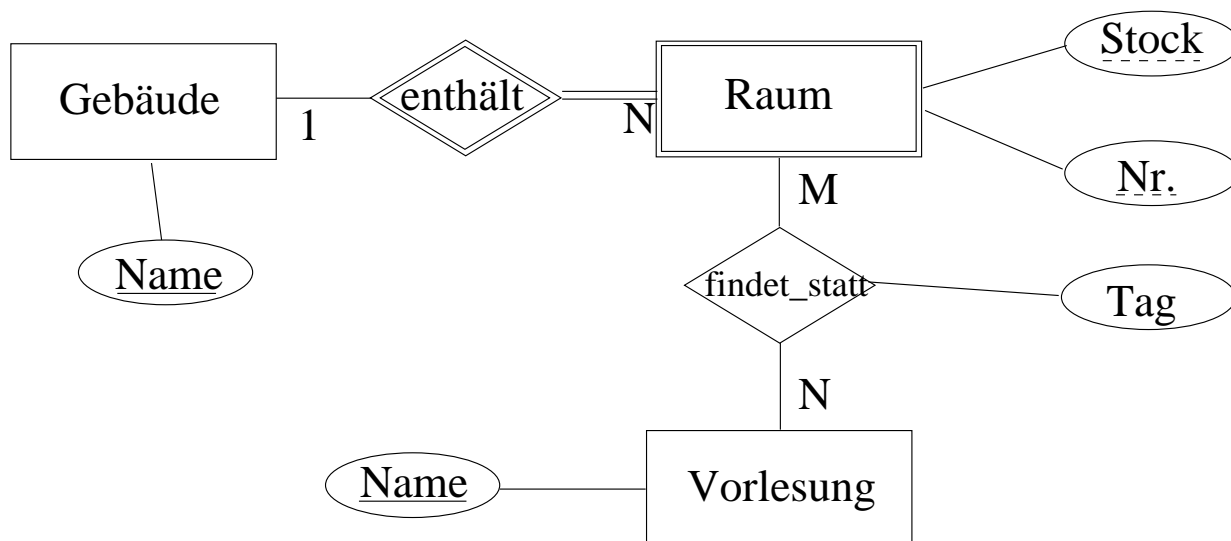
Für das Beispieldokument soll folgendes Ergebnis herauskommen:

```
<organigramm>
  <name>Suedfrucht AG</name>
  <person art="Manager"> <name>Edmund</name> <gehalt>115000</gehalt>
</person>
  <person art="Manager" boss="Edmund"> <name>Erwin</name> <gehalt>85000</gehalt>
</person>
  <person art="Arbeiter" boss="Erwin"> <name>Monika</name>
</person>
  <person art="Manager" boss="Edmund"> <name>Horst</name> <gehalt>70000</gehalt>
</person>
  <person art="Ingenieur" boss="Horst"> <name>Hubertus</name> <gehalt>85000</gehalt>
</person>
</organigramm>
```

Aufgabe 6

18 Punkte

Folgendes ER-Diagramm stellt ein konzeptuelles Modell für die Raumplanung von Vorlesungen dar. Das Attribut Stock bezeichnet das Stockwerk als eine ganze Zahl (0 für Erdgeschoss, negative Zahlen für Kellergeschosse). Die Raumnummer ist eine positive ganze Zahl. Der Tag ist der Wochentag, an dem die Vorlesung stattfindet, als ganze Zahl von 1-7.



Erstellen Sie ein XML-Schema für dieses konzeptuelle Modell. Stellen Sie sicher, dass man auch Dokumente validieren kann, die nur Gebäude oder nur Vorlesungen enthalten. Der Target Namespace des Schemas soll sein <http://raumplanung.org>.

Hilfestellung: Der Namensraum von XML Schema ist <http://www.w3.org/2003/XMLSchema>. Um Schreibarbeit zu vermeiden, verwenden Sie folgende abgekürzten Tag-Namen aus dem XML-Schema-Namensraum, wenn benötigt:

Tag	Abkürzung	Tag	Abkürzung
element	el	sequence	seq
attribute	attr	choice	cho
complexType	cT	extension	ext
simpleType	sT	global	glo

Name: _____ Matr.Nr.: _____

