

<b>Name:</b>	<b>Studiengang:</b> <input type="checkbox"/> B.A.   <input type="checkbox"/> M.A.
<b>Vorname:</b>	<b>Matrikelnummer:</b>
<b>Studienfächer:</b>	<b>Fachsemester:</b>

**Allgemeine Hinweise:**

1. Überprüfen Sie bitte, ob Sie alle Seiten der Klausurangabe vollständig erhalten haben (Gesamtzahl: 9)
2. **Bearbeitungszeit: 60 Minuten**, maximal erreichbare **Punktzahl: 94**. Die jeweils erreichbare Punktzahl ist bei jeder Frage angegeben. Bitte teilen Sie Ihre Arbeitszeit entsprechend ein.
3. Denken Sie daran, die Daten oben einzutragen, **bevor** Sie mit der Bearbeitung beginnen.
4. Verwenden Sie für die Beantwortung aller Fragen diese Klausurangabe. Sie können bei Bedarf zusätzliche Blätter anfordern. Wenden Sie sich dafür bitte an die Aufsichtsführenden. Bitte geben Sie in jedem Fall an, auf welche Frage sich die Lösung jeweils bezieht. Bei Multiple-Choice-Fragen treffen Sie bitte die Auswahl Ihrer Antworten ebenfalls auf der Klausurangabe.
5. Benutzen Sie keine Bleistifte, keine rot schreibenden Stifte und kein TippEx, o.ä.
6. Zugelassene Hilfsmittel: **KEINE**
7. Mobiltelefone sowie Computer am Arbeitsplatz - auch ausgeschaltet - sind **nicht zugelassen**.
8. Bitte legen Sie Lichtbildausweis und Studierendenausweis gut sichtbar vor sich, damit Ihre Identität möglichst störungsfrei überprüft werden kann.
9. Geben Sie keine mehrdeutigen (oder mehrere) Lösungen an. In solchen Fällen wird stets die Lösung mit der geringeren Punktzahl gewertet. Eine richtige und eine falsche Lösung ergeben also 0 Punkte.
10. Wenden Sie sich bei Unklarheiten in den Aufgabenstellungen immer an die Aufsichtsführenden. Hinweise und Hilfestellungen werden dann, falls erforderlich, offiziell für alle Teilnehmer durchgegeben.

**TEIL I) Relationale Datenbankmanagement-Systeme**

**I.1 Wahr oder falsch? (24 Punkte)**

Beurteilen Sie für jede der folgenden Aussagen, ob diese Wahr oder Falsch ist. Bitte beachten Sie: Falsch angekreuzte Antworten führen zu Punktabzug, wobei eine Punktzahl kleiner als Null nicht erreicht werden kann.

**Wahr**

**Falsch**

- |                          |   |                          |
|--------------------------|---|--------------------------|
| <input type="checkbox"/> | Relationen sind immer mindestens in erster Normalform.  | <input type="checkbox"/> |
| <input type="checkbox"/> | Assoziative Entitäten können zur Abbildung von M:N-Beziehungen genutzt werden.                | <input type="checkbox"/> |
| <input type="checkbox"/> | Sog. Strong Entities in ER-Diagrammen können unabhängig von anderen Entitätstypen existieren. | <input type="checkbox"/> |
| <input type="checkbox"/> | Primärschlüssel identifizieren jede Zeile einer Relation eindeutig.                           | <input type="checkbox"/> |
| <input type="checkbox"/> | Primärschlüsselattribute dürfen NULL sein.  | <input type="checkbox"/> |
| <input type="checkbox"/> | Fremdschlüsselattribute dürfen NULL sein.   | <input type="checkbox"/> |
| <input type="checkbox"/> | Relationen in zweiter Normalform verbieten transitive Abhängigkeiten.                         | <input type="checkbox"/> |
| <input type="checkbox"/> | CREATE TABLE-Statements sind Bestandteil der DML.   | <input type="checkbox"/> |
| <input type="checkbox"/> | Policies (a.k.a. Permissions) schützen die Zeilen.  | <input type="checkbox"/> |
| <input type="checkbox"/> | Ein DBMS nutzt Locks, um Konflikte während einer Transaktion zu vermeiden.                    | <input type="checkbox"/> |
| <input type="checkbox"/> | Aus mehreren Attributen zusammengesetzte Primärschlüssel sind nicht erlaubt.                  | <input type="checkbox"/> |
| <input type="checkbox"/> | In ER-Modellen werden Primärschlüssel durch Unterstreichen gekennzeichnet.                    | <input type="checkbox"/> |

### I.2 Transaktionen (6 Punkte)

Relation „bands“			
band_id [PK] serial	name character varying(50)	anzahl_mitglieder integer	gruendungsjahr integer
1	Queen	4	1970
2	Die Ärzte	3	1982
3	Green Day	3	1989
4	Rammstein	6	1994

Gegeben sei folgende Transaktion, die auf obige Tabelle „bands“ angewandt wird.

```
BEGIN TRANSACTION;  
UPDATE bands SET gruendungsjahr = 1987 WHERE bands.name = 'Green Day';  
DELETE FROM bands WHERE anzahl_mitglieder > 3;  
COMMIT;
```

Bitte kreuzen Sie diejenige(n) Antwort(en) an, die Ihrer Meinung nach zutreffen. Falsch angekreuzte Antworten führen zu Punktabzug, wobei eine Punktzahl kleiner als Null nicht erreicht werden kann.

- Nachdem die Transaktion ausgeführt wurde, fehlen die Bands mit den IDs 1 und 4.
- An der Datenbank werden keine Änderungen vorgenommen, da am Ende der Transaktion `END TRANSACTION;` stehen müsste.
- Nachdem die Transaktion ausgeführt wurde, findet man bei Green Day 1987 als Gründungsjahr.
- Nach dem UPDATE-Statement wird sofort ein ROLLBACK ausgeführt.

### I.3 SQL (12 Punkte)

Gegeben seien die folgenden beiden Relationen „bands“ und „alben“.

band_id [PK] serial	name character varying(50)	anzahl_mitglieder integer	gruendungsjahr integer
1	Queen	4	1970
2	Die Ärzte	3	1982
3	Green Day	3	1989
4	Rammstein	6	1994

album_id [PK] serial	name character varying(50)	erscheinungsjahr integer	anzahl_tracks integer	gesamtdauer double precision	refband integer
1	A Night at the Opera	1975	12	43.18	1
2	Jazz ist anders	2007	16	50.78	2
3	Innuendo	1991	12	53.72	1
4	Rosenrot	2005	11	48	4
5	21st Century Breakdown	2009	18	69.17	3
6	Mutter	2001	12	44.92	4
7	Le Frisur	1996	17	45.12	2
8	American Idiot	2004	13	57.22	3

Erstellen Sie eine PostgreSQL-Abfrage, die die Namen der Bands sowie deren Gründungsjahr ausgibt.

Erstellen Sie eine PostgreSQL-Abfrage, die den Namen jedes Albums, das ab dem Jahr 2000 veröffentlicht wurde, den Namen der Band, die das Album produziert hat, das Erscheinungsjahr und die Anzahl der auf dem Album befindlichen Tracks ausgibt.

Erstellen Sie eine PostgreSQL-Abfrage, die für jede Band (spezifiziert durch deren Namen) die Gesamtdauer aller in der Datenbank gespeicherten Alben ausgibt.

#### **I.4 EER-Diagramm (10 Punkte)**

Erstellen Sie ein EER-Diagramm für folgendes Szenario.

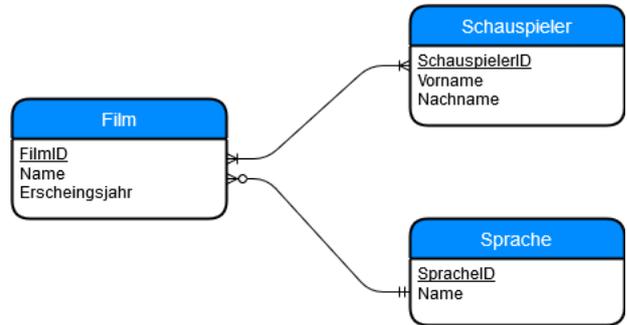
Für eine Firma, die E-Scooter verleiht, soll eine Datenbank eingerichtet werden. Dabei sollen unter anderem Daten zu den einzelnen Scootern, den Läden, von denen aus sie verliehen werden, und den Kunden gespeichert werden. Die einzelnen Scooter besitzen eine Seriennummer und haben unterschiedliche Farben. Kunden werden spezifiziert durch Vor- und Nachnamen sowie deren Adresse. Eine Adresse besteht aus Straße, Hausnummer, Postleitzahl und Ort. Ein Laden hat einen Namen sowie eine Adresse. Ein Scooter kann von mehreren Kunden entliehen und ein Kunde kann mehrere Scooter ausleihen. Der Zeitpunkt der Ausleihe soll dabei festgehalten werden. Jeder Kunde kann nur eine einzige Adresse angeben, während unter einer Adresse mehrere oder gar keine Kunden zu finden sind. Analog besitzt jeder Laden genau eine Adresse. Hinter einer Adresse können sich jedoch mehrere oder gar keine Läden verbergen. Jeder Scooter kann genau einem Laden zugeordnet werden. Ein Laden besitzt mindestens einen Scooter.

**I.5 Transformation eines ER-Diagramms in das relationale Schema (5 Punkte)**

Transformieren Sie das nachstehende ER-Diagramm in das relationale Schema. Die Ergebnisrelationen sollen in dritter Normalform vorliegen.

Verwenden Sie zur Darstellung der Relationen folgende Syntax:

Name\_der\_Relation (Primärschlüsselattribut, Attribut, Fremdschlüsselattribut)



**TEIL II) Auszeichnungssprachen**

**II.1 Wahr oder falsch (10 Punkte)**

Beurteilen Sie für jede der folgenden Aussagen, ob diese Wahr oder Falsch ist. Bitte beachten Sie: Falsch angekreuzte Antworten führen zu Punktabzug, wobei eine Punktzahl kleiner als Null nicht erreicht werden kann.

Wahr		Falsch
<input type="checkbox"/>	XML-Dateien müssen stets wohlgeformt und valide sein.	<input type="checkbox"/>
<input type="checkbox"/>	Leere Elemente dürfen Attribute enthalten.	<input type="checkbox"/>
<input type="checkbox"/>	In jeder XML-Datei gibt es mindestens einen Knoten, der keine Eltern hat.	<input type="checkbox"/>
<input type="checkbox"/>	In wohlgeformten XML-Dokumenten müssen unter anderem die Elemente korrekt verschachtelt werden.	<input type="checkbox"/>
<input type="checkbox"/>	XML-Dateien weisen eine baumartige Struktur auf.	<input type="checkbox"/>

## II.2 XML-Schema (5 Punkte)

Gegeben sei das nachstehende XML-Dokument.

```
<klausuren>
  <klausur>
    <fach>Daten effizient speichern und verarbeiten</fach>
    <datum>2019-08-09</datum>
  </klausur>
  <klausur>
    <fach>Übung zu Daten effizient speichern und verarbeiten</fach>
    <datum>2019-08-08</datum>
  </klausur>
</klausuren>
```

Ergänzen Sie die folgende XML-Schema-Datei so, dass diese zur Validierung dieses XML-Dokuments geeignet ist. Verwenden Sie die am besten geeigneten Datentypen. Bitte beachten Sie: Falsche Angaben führen zu Punktabzug, wobei eine Punktzahl kleiner als Null nicht erreicht werden kann.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="_____">
    <xs:complexType>
      _____
      <xs:element name="klausur" _____="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="fach" type="_____" />
            <xs:element name="datum" type="_____" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      _____
    </xs:complexType>
  </xs:element>
</xs:schema>
```

### II.3 JSON (5 Punkte)

In nachstehendem JSON-Snippet werden die Daten von Testpersonen repräsentiert. Wie sähe das Snippet syntaktisch korrekt aus?

```
1  {  
2      testpersonen:  
3          {  
4              "vorname": "Alice",  
5              "nachname": "Scott",  
6              "alter": 40  
7              "geschlecht": "weiblich"  
8          },  
9          {  
10             "vorname": "Bernhard",  
11             "nachname": "Brot",  
12             "alter": 35,  
13             "geschlecht": "männlich"  
14         }  
15     }  
16 }
```

### II.4 XQuery (5 Punkte)

Gegeben sei das nachstehende XML-Dokument „studierende.xml“.

```
<studierende>  
  <student>  
    <vorname>Max</vorname>  
    <nachname>Mustermann</nachname>  
    <adresse>  
      <plz>93053</plz>  
      <ort>Regensburg</ort>  
      <strasse>Beispielweg</strasse>  
      <hausnummer>3</hausnummer>  
    </adresse>  
  </student>  
  <student>  
    <vorname>Michael</vorname>  
    <nachname>Mustermann</nachname>  
    <adresse>  
      <plz>93053</plz>  
      <ort>Regensburg</ort>  
      <strasse>Beispielweg</strasse>  
      <hausnummer>6</hausnummer>  
    </adresse>  
  </student>  
</studierende>
```

Welche Ausgabe erzeugt folgender FLWOR-Ausdruck?

```
for $b in doc("studierende.xml")//adresse  
return <adressen>{$b}</adressen>
```

### **TEIL III) JAVA und RDBMS bzw. Auszeichnungssprachen**

#### **III.1 Eigenschaften von Transaktionen (8 Punkte)**

Was verbirgt sich hinter dem Akronym ACID in Zusammenhang mit Transaktionen? Nennen und erläutern Sie kurz diese Eigenschaften.

#### **III.2 JDBC (4 Punkte)**

Welche Aussage(n) bezüglich JDBC trifft/treffen zu? Bitte beachten Sie: Falsch angekreuzte Antworten führen zu Punktabzug, wobei eine Punktzahl kleiner als Null nicht erreicht werden kann.

- Der Cursor im ResultSet zeigt zu Beginn vor die erste Ergebniszeile.
- PreparedStatements bieten die Möglichkeit, Parameter zu übergeben.
- Ein ResultSet-Cursor kann als Datenbank-Cursor betrachtet werden.