Introduction to Databases
Written Exam winter term 2003/2004

1. Conceptual Modelling (7 points)
A database system for museums has to be developed. It should manage the following information:

- A museum can be uniquely identified by its name. Furthermore, it should be possible to store its address (consisting of zip code, city and street).
- We want to distinguish (a bit over-simplified) between public museums that are associated with a government agency and have a budget per year, and private museums that get their money from private investors (with name and amount of contribution).
- A museum shows exhibitions that can be uniquely identified by their title. Furthermore, their start and end date must be stored.
- An exhibition consists of several works of art, that can be uniquely identified by a global registration number. Besides this, the creator and the year of its creation must be saved.
- In addition to this general information about works of art we want to store characteristics of the following more special pieces:
  - Paintings (height and width of the picture), sculptures (height and weight) and items on loan (the loaner’s name, amount of payment)
- A museum presents at least one exhibition, an exhibition takes places at exactly one museum.
- A work of art can be (over time) part of arbitrarily many (or no) exhibition(s). An exhibition consists of at least 5 pieces of art.

Design an ER-diagram that models this requirements scenario. Use the (min,max) notation to constrain the cardinalities of relationships.
2. Relational Databases (15 points)
Consider the following ER-diagram that shows the conceptual model of an information system for the music industry.

a) Map this diagram to a relational database schema. Use the formal notation presented in the lecture and tutorials including intra- and interrelational dependencies. (5 points)

b) Formulate the following query in tuple relational calculus:
   *List all bandleaders, who publish a solo album at a record label that is located at Munich.*
   (3 points)

c) Describe the following query in SQL:
   *List the names of record labels and the names of bands together with the number of albums that the band has published at this record label before 1980. Only those bands should be considered, that are not lead by an artist who has published a solo album.*
   (4 points)

d) Give an expression in Relational Algebra that calculates the result of the following query:
   *List all names of bands that have published at the record label ‘Funky records’ and consist only of artists who have not published a solo album.*
   (3 points)
3. Functional Dependencies and Normal Forms (9 points)
Consider the following relation schema of a book store database:
BOOKS = {{title, author, type, listprice, affiliation, publisher},
           {title → publisher, type;
            type → listprice;
            author → affiliation}}

a) Identify the key of BOOKS and prove your claim. (2 points)
b) In which normal form is BOOKS? Why? (2 points)
c) Transform BOOKS by using the decomposition algorithm into a normalised relational database schema. Does the resulting decomposition have the dependency preservation property? If not, which dependency is lost? (5 points)

4. Object Oriented Databases (8 points)
Consider the following ER-diagram that models an information system around a car dealer and his customers. Car dealers can decide to join their forces in a purchasing association, i.e. a group of dealers that purchase cars from the manufacturer as a group.

a) Transfer the above ER-diagram into an UML structure diagram. (5 points)
b) Explain briefly the difference between the following types of object relationships. Give for each of these types two classes of your structure diagram, that are in a relationship which could be reasonably modelled by this type. (3 points)
- association
- aggregation
- composition
5. XML and XML Schema (7 points)
Consider the XML Schema document given in the appendix which fixes the format of XML documents that describe a collection of lectures. These lectures can be part of several courses and can end with an oral or written examination.

a) Decide for both XML document in the appendix, whether it is valid according to the given schema. For the nonvalid document(s), give all reasons why they violate the schema constraints. (3 points)

b) Describe the following query in XQuery:
List all lectures with an oral examination grouped by their instructors in the following format:

```xml
<lectures>
  <instructor name="INSTRUCTOR_NAME" email="INSTRUCTOR_EMAIL">
    <lecture lang="LECTURE_LANG">LECTURE_TITLE</lecture>
  </instructor>
</lectures>
```

The instructors should be sorted in alphabetic order. Formulate your query against an XML document “lectures.xml” that is considered to be valid according to the given schema. (4 points)

6. Distributed Databases (4 points)
Explain briefly the differences between the following levels of “distribution transparency“:

- Fragmentation transparency
- Location transparency
- Local mapping transparency
- No transparency
Appendix

XML Schema document

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="lectures">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="lecture" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title" type="xs:string" minOccurs="1" maxOccurs="1"/>
              <xs:element name="course" type="xs:string" minOccurs="1" maxOccurs="unbounded"/>
              <xs:element name="instructor" type="personType" minOccurs="1" maxOccurs="1"/>
              <xs:element name="tutorialInstructor" type="personType" minOccurs="1" maxOccurs="3"/>
              <xs:element name="examInfo" minOccurs="1" maxOccurs="1">
                <xs:complexType>
                  <xs:choice>
                    <xs:element name="oralExamDate" type="xs:date" minOccurs="1"/>
                    <xs:element name="writtenExamDate" type="xs:date" minOccurs="1"/>
                  </xs:choice>
                </xs:complexType>
              </xs:element>
              <xs:attribute name="lang" type="xs:string" use="optional" default="EN"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

XML document 1:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<lectures>
  <lecture>
    <title>Bioinformatics</title>
    <instructor>
      <name>Prof. T. Berlage</name>
      <roomNumber>FIT 1032</roomNumber>
      <eMail>thomas.berlage@fit.fraunhofer.de</eMail>
    </instructor>
    <examInfo>
      <writtenExamDate>2004-03-01</writtenExamDate>
      <oralExamDate>2004-03-05</oralExamDate>
    </examInfo>
    <course>Introduction to Databases</course>
  </lecture>
</lectures>
```

XML document 2:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<lectures lang="EN">
  <lecture>
    <title>Introduction to Databases</title>
    <course>SSE</course>
    <course>Informatik Diplom</course>
    <instructor>
      <name>Prof. Dr. M. Jarke</name>
      <roomNumber>6230</roomNumber>
      <eMail>jarke@i5.informatik.rwth-aachen.de</eMail>
    </instructor>
    <tutorialInstructor>
      <name>Aida Jertila</name>
      <roomNumber>6239</roomNumber>
      <eMail>jertila@i5.informatik.rwth-aachen.de</eMail>
    </tutorialInstructor>
    <examInfo>
      <writtenExamDate>2004-02-18</writtenExamDate>
    </examInfo>
  </lecture>
</lectures>
```