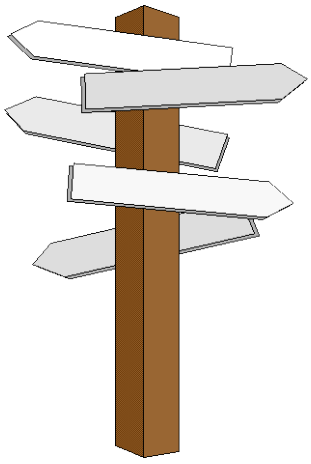
SDXML VT2024

Models and languages for handling semi-structured data and XML



COURSE INFORMATION

1 Course description

The course introduces the concepts of semi-structured data and XML and dives into many related technologies for data storage and querying with standards like XPath, XQuery and XSLT. The course discusses the relationship between relational databases and XML and covers standardized technologies like SQL/XML as well as other vendor-specific solutions from IBM, Oracle and Microsoft. During the course, XML-based languages and formats for open data are discussed. This includes XML-based languages like OOXML, ODF, RDF and SVG. The focus of the course is on XML, DTD, XML Schema, XPath, XQuery, XSLT and SQL/XML, but even JSON is presented and discussed, especially how it compares to XML.

2 Literature

MAIN BOOK

Querying XML: XQuery, XPath, and SQL-XML in context (ISBN: 9781558607118) Jim Melton, Stephen Buxton, Morgan Kaufmann, 2006

One is expected to also have a basic database book covering basic relational database concepts, SQL, normalization, data integrity, etc. For example, the book used by the Database Methodology course:
Database Systems (Thomas Connolly, Carolyn Begg), Addison-Wesley

Reading guidelines are available in the lecture slides.

OTHER RECOMMENDED BOOKS

- *An Introduction to XML and Web Technologies* (ISBN: 9780321269669) Moller, Schwartzbach, Addison-Wesley, 2006
- *Data on the Web* (ISBN: 9781558606227) Abiteboul et al, Morgan Kaufmann, 2000
- *Beginning XML Databases* (ISBN: 9780471791201) Gavin Powell, Wiley, 2007
- *XQuery - Search across a variety of XML-data* (Second Edition, ISBN: 9781491915103) Priscilla Walmsley, O'Reilly, 2015

OTHER MATERIAL

- Course information (this compendium)
- Lecture slides
- Compendiums about the different technologies and products
- Excerpts and articles
- Relevant web pages
- Other material (suggested solutions to exercises, sample databases, old exams)

3 Teachers

nikos dimitrakas (course leader)
Telephone: 08-161295 E-mail: nikos@dsv.su.se

4 Activities

The course consists of a number of lectures, lessons and seminars. Students are also expected to work with tutorials and assignments. Tutoring is available at specific times. There are also eight compulsory quizzes to be completed electronically. It is strongly recommended that students participate in all scheduled activities. Electronic assistance is provided through the course platform (iLearn) or by e-mail.

5 Computer environment

During the course, a multitude of different tools will be used for handling XML data, relational data and query languages.

The following tools will be used:

- XQuisitor
- BaseX
- Web browsers (for XSLT)
- Oracle Database 21c
- IBM DB2 11.5
- Microsoft SQL Server 2022
- Text Editor (like Notepad++) with syntax support for XML and JSON

Moreover, students are expected to have access to the web-based systems Daisy and iLearn (which is automatically granted to students registered for the course). For tutoring, the web-based tutoring system is used.

All the required software is available for free (for non-commercial use). A preconfigured virtual Windows environment is provided to each student that wants/needs one.

6 Tutoring

During the course, tutoring is available at specific times in NOD and asynchronously in iLearn. Students can ask for advice about solutions (queries, models, etc.) and also discuss anything that relates to the course content. Students can also get answers to practical questions about the course administration, the technical environment, grading, etc. Refrain from posting full solution for the assignments to the tutoring forum in iLearn though.

In the tutoring forum in iLearn, students can ask questions, but also answer each other. The teachers will monitor all forum posts and answer as soon as possible.

Tutoring in NOD is provided during scheduled times. The scheduled times for tutoring may be changed based on the need and number of students. Use the tutoring system at tutoring.dsv.su.se to get in line for help. The teachers monitor the tutoring list and will come to you as soon as it is your turn. You may sit anywhere on the student areas on floors 0, 1, 2 and 3 (mostly on floor 2).

If there is any need to meet a teacher at a different time, send an email in order to book a time. If there is a need to ask something in writing, that should not be posted in the forum in iLearn, send an email.

7 Examination

The course is divided into three examinations. One of the examinations (3.5 hec) corresponds to the written exam. The other two examinations (2.5 hec and 1.5 hec) include quizzes, assignments and seminars. Some assignments are optional in order to receive a higher grade. Some assignments are done in groups, while others are done individually.
The course grade is calculated according to the department's algorithm and is handled by the director of studies once all examinations have been completed.

Exam

Four hours written exam. No aids allowed (exception: a dictionary between English and the student's mother tongue). The exam is divided into three parts. In order to pass the exam, the student must perform at a certain level on each part and at a certain level in total. The three parts correspond to the three course goals. The grading is done according to the following principles:
E means that the student fulfills the goals at a sufficient level. For the first goal (*explain, discuss and use different technologies for managing XML data and semi-structured data in general, as well as model suitable data structures and argue about the pros and cons of different solutions*) an E implies that the student can explain some relevant techniques and terms and also model some data structures that are not perfect, but shows some basic understanding. The student moves closer to grade A by being able to explain more and in more detail, as well as being able to discuss pros and cons with different techniques and solutions and also being able to model better solutions. For the second goal (*use query languages for XML data and semi-structured data, as well as techniques for conversions between different forms of semi-structured data*) and the third goal (*use different techniques for composition of XML and for transformations between relational data and XML data*), E requires that the student can solve simple problems with some of the relevant techniques or make somewhat correct solutions (that show understanding even if they contain errors). The student moves closer to grade A by being able to solve more complex problems and with more of the relevant techniques, as well as have more correct and complete solutions.

These grading criteria are converted to the following grade thresholds:

	Whole exam	Per part
Minimums for A	92%	84%
Minimums for B	84%	73%
Minimums for C	76%	62%
Minimums for D	68%	51%
Minimums for E	60%	40%

In the unlikely event that a student achieves a very uneven result (very good result, 85% or more, in one or two parts, without reaching the minimums for E), the student will be offered the option of skipping parts during the retake. This will be denoted by the grade Fx and will be explained in the feedback to the exam. Any student that chooses to use this option will be able to receive at most the grade E. Students that have been offered this option can ignore it. When this option is used, the skipped parts are excluded from the grade calculation and the minimums for E apply to the remaining parts.

The dates and times for the exams are available in the course schedule.

Sample exams with suggested solutions and some grading comments are available in the course platform iLearn.

The three parts of the exam cover the following areas:
Part 1 (Theory and modeling): Everything about semi-structured data, XML, DTD, XML Schema, JSON, JSON Schema, XML-based languages, usages of semi-structured data and XML, representations of XML
Part 2 (Query languages for SSD and XML): Path expressions, Lorel, XPath, XQuery, XSLT
Part 3 (XML and relational databases): SQL/XML and product-specific solutions by Microsoft, Oracle and IBM

Assignments

A number of smaller assignments are to be submitted (most of them in groups, some of them individually) electronically in the course platform iLearn. The groups should consist of three students (two only if necessary). Group selection is done in iLearn, but will also be visible in Daisy later. Use the forum Groups, if necessary, to make contact with other students in order to form a group.

The assignments cover mostly the practical parts of the course, so mainly the second and third course goals. Grading is based on the principle that the compulsory assignments are sufficient for achieving enough knowledge for grade E. By doing the optional assignments, the student gains more knowledge in the covered techniques and theory and can receive all the way to grade A.

The assignments are described in the chapter Assignments.

Quizzes

There are eight quizzes that relate to the content of different lectures. They are done individually in the course platform iLearn. They cover most of the things discussed in the lectures. They should be done at the appropriate times but can also be repeated many times. Once a quiz is completed, new attempts will not affect the completion status. The quizzes are described in the chapter Quizzes.

Seminars

Three of the assignments are to be presented and discussed during three seminars. They must be submitted prior to each seminar. Otherwise, these assignments are handled in the same way as the assignments that are not presented in seminars.

8 Course development and evaluation

It is important to get good feedback about the course in order to make improvements both during the course and ahead of the next course offering. It is simple to point out a problem or offer a suggestion by sending a message in the Tutoring forum in iLearn or by email to the course leader. At the end of the course all participants will have the opportunity to fill out an anonymous course evaluation (in Daisy). It is appreciated if every student takes the time to provide some feedback (both positive and negative). A few weeks after the end of the course, a course analysis will be written and published in Daisy and in iLearn. It will also serve as a guide to changes to be made for the future.

9 Schedule/Suggested workflow

The official course schedule may change. Always consult the schedule in Daisy for the latest version. This chapter offers a suggested workflow per course week. Students should try to be on par with the expected activities. The course is 7.5 hec, which corresponds to an expected effort of 200 hours. The following shows how the course leader envisions how these 200 hours should be spent. This must of course be adjusted to each individual, since most student take other courses in parallel.

Week 12

- Course starts on Thursday
- Lecture 1
- Read the material corresponding to lecture 1
- Read the course information compendium
- Fix all administrative issues (access to Daisy, iLearn)
- Quiz 1

Week 13

- Lecture 2
- Read the material corresponding to lecture 2
- Quiz 2
- Lecture 3
- Read the material corresponding to lecture 3
- Quiz 3
- Quiz 4
- Get an account to a virtual environment (if needed)
- Form a group
- Thursday and Friday are free

Week 14

- Monday is free
- Lecture 4
- Read the material corresponding to lecture 4
- Lecture 5
- Read the material corresponding to lecture 5
- Quiz 5
- Work with exercises for lesson 1
- Work with the compendium on XQuery
- Lesson 1
- Work with exercises for lesson 2

Week 15

- Lesson 2
- Work with assignment for seminar 1
- Submit assignment for seminar 1
- Seminar 1

Week 16

- Lecture 6
- Read the material corresponding to lecture 6
- Quiz 6
- Work with exercises for lesson 3
- Work with the compendium on XSLT
- Lesson 3
- Work with assignment for seminar 2
- Submit assignment for seminar 2
- Start with assignment 4

Week 17

- Seminar 2
- Lecture 7
- Read the material corresponding to lecture 7
- Quiz 7
- Work with exercises for lesson 4
- Lecture 8
- Work with the compendium on IBM DB2
- Lecture 9
- Work with the compendium on Oracle DB
- Start with assignment 5

Week 18

- Work with exercises for lesson 4
- Work with the compendium on IBM DB2
- Work with the compendium on Oracle DB
- Wednesday is free
- Work with assignment for seminar 3

Week 19

- Submit assignment for seminar 3
- Seminar 3
- Start with assignment 6
- Start with assignment 7
- Lecture 10
- Quiz 8
- Work with the compendium on Microsoft SQL Server
- Thursday and Friday are free

Week 20

- Start with assignment 8
- Work with assignments
- Submit assignment 4
- Submit assignment 5

Week 21

- Work with assignments
- Submit assignment 6
- Submit assignment 7
- Preparations for the exam (repetition, sample exams)

Week 22

- Work with assignments
- Submit assignment 8
- Submit optional assignments (9, 10, 11, 12)
- Preparations for the exam (repetition, sample exams)
- Exam
- Course evaluation

Later

- Submit assignments after required revisions
- Exam retake

10 Sample databases

For many of the exercises for the lectures, the tutorials, the lessons and the assignments (including the seminar assignments), several different databases will be used. They are all provided in iLearn. Some of them are relational databases with XML inside the tables, while others are a set of XML files.

The first database is an XML file with information about movies and actors.

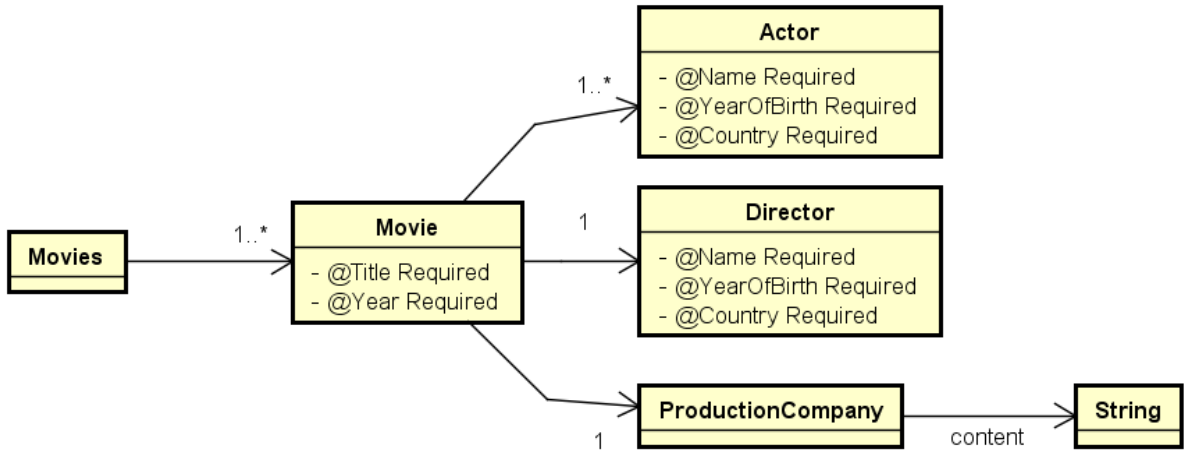
The second database is two XML files with information about books and publishers. The same database is also available as a relational database with the same data structured in a different way.

The third database is a relational database about people, their employments and their cars.

The fourth database is a relational database about courses.

The movie database

GRAPHICAL AND TEXTUAL REPRESENTATION OF THE STRUCTURE



The root element is Movies and has one or more subelements Movie. Each Movie element has the attributes Title and Year and contains one or more subelements Actor, exactly one subelement Director, and exactly one subelement ProductionCompany. The elements Actor and Director have the same structure, namely three attributes Name, YearOfBirth and Country. The element ProductionCompany has no attributes, just some textual content (a text node). The same information is also expressed in the following XML Schema and DTD.

XML SCHEMA

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="Movies" type="MoviesType"/>
    <complexType name="MoviesType">
        <sequence>
            <element name="Movie" type="MovieType" minOccurs="1" maxOccurs="unbounded" />
        </sequence>
    </complexType>
    <complexType name="MovieType">
        <sequence>
            <element name="Actor" type="PersonType" minOccurs="1" maxOccurs="unbounded" />
            <element name="Director" type="PersonType" minOccurs="1" maxOccurs="1" />
            <element name="ProductionCompany" type="string" minOccurs="1" maxOccurs="1" />
        </sequence>
        <attribute name="Title" type="string" use="required" />
        <attribute name="Year" type="integer" use="required" />
    </complexType>
    <complexType name="PersonType">
        <attribute name="Name" type="string" use="required" />
        <attribute name="YearOfBirth" type="integer" use="required" />
        <attribute name="Country" type="string" use="required" />
    </complexType>
</schema>
```

DTD

```
<!ELEMENT Movies (Movie+)>
<!ELEMENT Movie (Actor+, Director, ProductionCompany)>
<!ATTLIST Movie
        Title CDATA #REQUIRED
        Year CDATA #REQUIRED>
<!ELEMENT Actor EMPTY>
<!ATTLIST Actor
        Name CDATA #REQUIRED
        YearOfBirth CDATA #REQUIRED
        Country CDATA #REQUIRED>
<!ELEMENT Director EMPTY>
<!ATTLIST Director
        Name CDATA #REQUIRED
        YearOfBirth CDATA #REQUIRED
        Country CDATA #REQUIRED>
<!ELEMENT ProductionCompany (#PCDATA)>
```

DATABASE CONTENT

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Movies>
<Movies>
    <Movie Title="Driven" Year="2001">
        <Actor Name="Burt Reynolds" YearOfBirth="1936" Country="USA"/>
        <Actor Name="Silvester Stallone" YearOfBirth="1946" Country="USA"/>
        <Actor Name="Kip Pardue" YearOfBirth="1976" Country="Canada"/>
        <Director Name="Silvester Stallone" YearOfBirth="1946" Country="USA"/>
        <ProductionCompany>Tri-Star</ProductionCompany>
    </Movie>
    <Movie Title="Antz" Year="1998">
        <Actor Name="Woody Allen" YearOfBirth="1935" Country="USA"/>
        <Actor Name="Silvester Stallone" YearOfBirth="1946" Country="USA"/>
        <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
        <Director Name="Eric Darnell" YearOfBirth="1961" Country="Ireland"/>
        <ProductionCompany>Universal</ProductionCompany>
    </Movie>
    <Movie Title="The Muse" Year="1999">
        <Actor Name="Andie MacDowell" YearOfBirth="1958" Country="USA"/>
        <Actor Name="Jeff Bridges" YearOfBirth="1949" Country="USA"/>
        <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
        <Director Name="Pitof" YearOfBirth="1970" Country="France"/>
        <ProductionCompany>October Films</ProductionCompany>
    </Movie>
    <Movie Title="End Game" Year="2006">
        <Actor Name="James Woods" YearOfBirth="1955" Country="USA"/>
        <Actor Name="Cuba Gooding Jr" YearOfBirth="1968" Country="USA"/>
        <Director Name="Andy Cheng" YearOfBirth="1972" Country="China"/>
        <ProductionCompany>Millennium Films</ProductionCompany>
    </Movie>
    <Movie Title="Picking Up the Pieces" Year="2000">
        <Actor Name="Woody Allen" YearOfBirth="1935" Country="USA"/>
        <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
        <Actor Name="Alfonso Arau" YearOfBirth="1948" Country="USA"/>
        <Director Name="Eric Darnell" YearOfBirth="1961" Country="Ireland"/>
        <ProductionCompany>Tri-Star</ProductionCompany>
    </Movie>
```

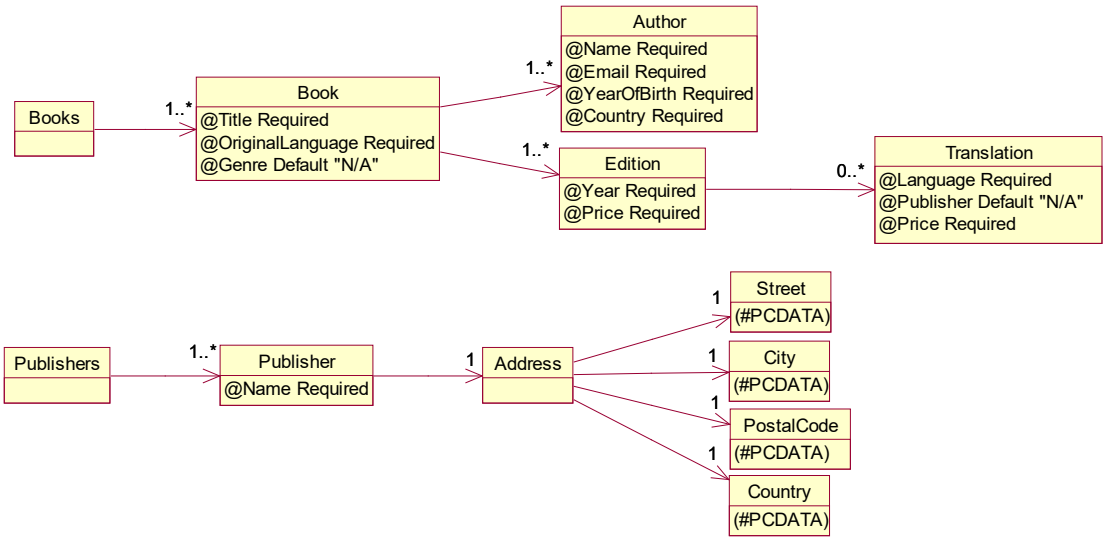
```
<Movie Title="Anything Else" Year="2003">
    <Actor Name="Woody Allen" YearOfBirth="1935" Country="USA"/>
    <Actor Name="Jason Biggs" YearOfBirth="1978" Country="USA"/>
    <Actor Name="Christina Ricci" YearOfBirth="1980" Country="USA"/>
    <Director Name="Woody Allen" YearOfBirth="1935" Country="USA"/>
    <ProductionCompany>Universal</ProductionCompany>
</Movie>
<Movie Title="Catwoman" Year="2004">
    <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
    <Actor Name="Halle Berry" YearOfBirth="1966" Country="USA"/>
    <Director Name="Pitof" YearOfBirth="1970" Country="France"/>
    <ProductionCompany>Universal</ProductionCompany>
</Movie>
<Movie Title="The Specialist" Year="1994">
    <Actor Name="Silvester Stallone" YearOfBirth="1946" Country="USA"/>
    <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
    <Actor Name="James Woods" YearOfBirth="1955" Country="USA"/>
    <Director Name="Eric Darnell" YearOfBirth="1961" Country="Ireland"/>
    <ProductionCompany>Tri-Star</ProductionCompany>
</Movie>
<Movie Title="Sliver" Year="1993">
    <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
    <Actor Name="William Baldwin" YearOfBirth="1963" Country="USA"/>
    <Actor Name="Tom Berenger" YearOfBirth="1949" Country="USA"/>
    <Director Name="Phillip Noyce" YearOfBirth="1950" Country="Australia"/>
    <ProductionCompany>Universal</ProductionCompany>
</Movie>
<Movie Title="Last Action Hero" Year="1993">
    <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
    <Actor Name="Arnold Schwarzenegger" YearOfBirth="1947" Country="Austria"/>
    <Director Name="John McTiernan" YearOfBirth="1951" Country="USA"/>
    <ProductionCompany>Universal</ProductionCompany>
</Movie>
<Movie Title="The Saint" Year="1997">
    <Actor Name="Val Kilmer" YearOfBirth="1959" Country="USA"/>
    <Actor Name="Elisabeth Shue" YearOfBirth="1963" Country="USA"/>
    <Director Name="Phillip Noyce" YearOfBirth="1950" Country="Australia"/>
    <ProductionCompany>Universal</ProductionCompany>
</Movie>
<Movie Title="Hide and Seek" Year="2005">
    <Actor Name="Robert De Niro" YearOfBirth="1943" Country="USA"/>
    <Actor Name="Dakota Fanning" YearOfBirth="1994" Country="USA"/>
    <Actor Name="Famke Janssen" YearOfBirth="1964" Country="Netherlands"/>
    <Actor Name="Elisabeth Shue" YearOfBirth="1963" Country="USA"/>
    <Director Name="John Polson" YearOfBirth="1965" Country="Australia"/>
    <ProductionCompany>20th Century Fox</ProductionCompany>
</Movie>
<Movie Title="Demolition Man" Year="1993">
    <Actor Name="Silvester Stallone" YearOfBirth="1946" Country="USA"/>
    <Actor Name="Wesley Snipes" YearOfBirth="1962" Country="USA"/>
    <Actor Name="Sandra Bullock" YearOfBirth="1964" Country="USA"/>
    <Actor Name="Hiro Kanagawa" YearOfBirth="1963" Country="Japan"/>
    <Director Name="Marco Brambilla" YearOfBirth="1960" Country="Italy"/>
    <ProductionCompany>Warner Bros</ProductionCompany>
</Movie>
```

```
    <Movie Title="Excess Baggage" Year="1997">
        <Actor Name="Alicia Silverstone" YearOfBirth="1976" Country="USA"/>
        <Actor Name="Benicio Del Toro" YearOfBirth="1967" Country="Puerto Rico"/>
        <Actor Name="Hiro Kanagawa" YearOfBirth="1963" Country="Japan"/>
        <Director Name="Marco Brambilla" YearOfBirth="1960" Country="Italy"/>
        <ProductionCompany>Warner Bros</ProductionCompany>
    </Movie>
    <Movie Title="The 6th Day" Year="2000">
        <Actor Name="Arnold Schwarzenegger" YearOfBirth="1947" Country="Austria"/>
        <Actor Name="Hiro Kanagawa" YearOfBirth="1963" Country="Japan"/>
        <Actor Name="Robert Duvall" YearOfBirth="1931" Country="USA"/>
        <Director Name="Roger Spottiswoode" YearOfBirth="1945" Country="Canada"/>
        <ProductionCompany>Phoenix Pictures</ProductionCompany>
    </Movie>
    <Movie Title="Cliffhanger" Year="1993">
        <Actor Name="Silvester Stallone" YearOfBirth="1946" Country="USA"/>
        <Director Name="Renny Harlin" YearOfBirth="1960" Country="Italy"/>
        <ProductionCompany>Warner Bros</ProductionCompany>
    </Movie>
    <Movie Title="The Kid And I" Year="1993">
        <Actor Name="Tom Arnold" YearOfBirth="1959" Country="USA"/>
        <Actor Name="Arnold Schwarzenegger" YearOfBirth="1947" Country="Austria"/>
        <Director Name="Penelope Spheeris" YearOfBirth="1945" Country="USA"/>
        <ProductionCompany>Universal</ProductionCompany>
    </Movie>
    <Movie Title="True Lies" Year="1994">
        <Actor Name="Tom Arnold" YearOfBirth="1959" Country="USA"/>
        <Actor Name="Jamie Lee Curtis" YearOfBirth="1958" Country="USA"/>
        <Actor Name="Arnold Schwarzenegger" YearOfBirth="1947" Country="Austria"/>
        <Actor Name="Eliza Dushku" YearOfBirth="1980" Country="USA"/>
        <Director Name="James Cameron" YearOfBirth="1954" Country="Canada"/>
        <ProductionCompany>Tri-Star</ProductionCompany>
    </Movie>
</Movies>
```

The book database

The database exists in two forms. As two XML files and as a relational database.

GRAPHICAL AND TEXTUAL REPRESENTATION OF THE STRUCTURE OF THE XML FILES



The same information as DTDs:

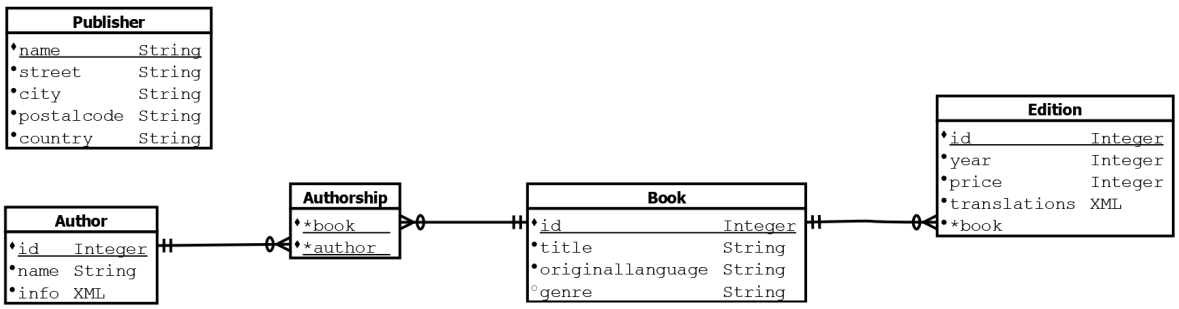
BOOKS.DTD

```
<!ELEMENT Books (Book+)>
<!ELEMENT Book (Author+, Edition+)>
<!ATTLIST Book Title CDATA #REQUIRED
        OriginalLanguage CDATA #REQUIRED
        Genre CDATA "N/A">
<!ELEMENT Edition (Translation*)>
<!ATTLIST Edition Year CDATA #REQUIRED
        Price CDATA #REQUIRED>
<!ELEMENT Translation EMPTY>
<!ATTLIST Translation Language CDATA #REQUIRED
        Publisher CDATA "N/A"
        Price CDATA #REQUIRED>
<!ELEMENT Author EMPTY>
<!ATTLIST Author Name CDATA #REQUIRED
        Email CDATA #REQUIRED
        YearOfBirth CDATA #REQUIRED
        Country CDATA #REQUIRED>
```

PUBLISHER.DTD

```
<!ELEMENT Publishers (Publisher+)>
<!ELEMENT Publisher (Address)>
<!ATTLIST Publisher Name CDATA #REQUIRED>
<!ELEMENT Address (Street, City, PostalCode, Country)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT PostalCode (#PCDATA)>
<!ELEMENT Country (#PCDATA)>
```

GRAPHICAL REPRESENTATION OF THE STRUCTURE OF THE RELATIONAL DATABASE



There are two columns that contain XML: Author.info and Edition.translations. The content of these columns complies to the following XML Schemas:

XML SCHEMA FOR AUTHOR.INFO

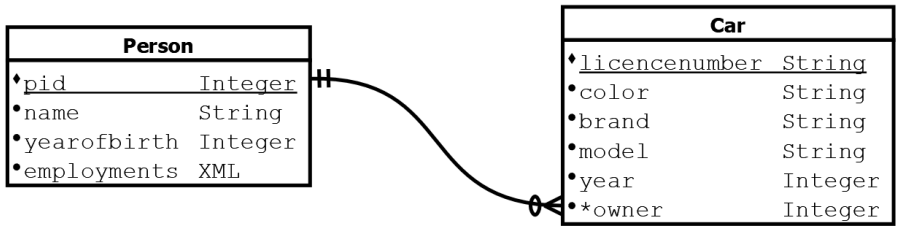
```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="Info" type="InfoType"/>
    <complexType name="InfoType">
        <all>
            <element name="Email" type="string"/>
            <element name="YearOfBirth" type="integer"/>
            <element name="Country" type="string"/>
        </all>
    </complexType>
</schema>
```

XML SCHEMA FOR EDITION.TRANSLATIONS

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="Translations">
        <complexType>
            <sequence>
                <element name="Translation" minOccurs="0" maxOccurs="unbounded">
                    <complexType>
                        <attribute name="Language" type="string" use="required"/>
                        <attribute name="Publisher" type="string" default="N/A"/>
                        <attribute name="Price" type="integer" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

The car database

GRAPHICAL REPRESENTATION OF THE STRUCTURE OF THE RELATIONAL DATABASE



The column Person.employments contains XML documents according to the following XML Schema:

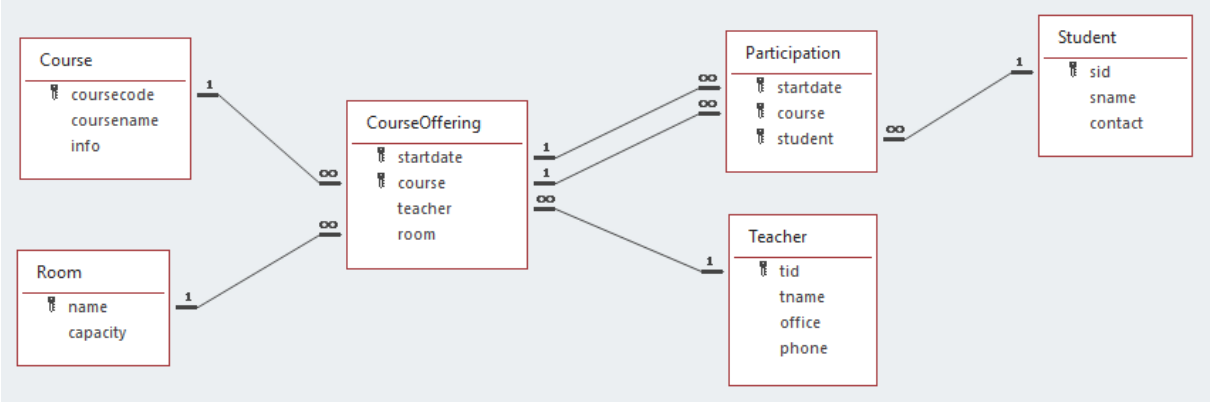
XML SCHEMA FOR PERSON.EMPLOYMENTS

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="root"/>
        <complexType>
            <sequence>
                <element name="employment" type="EmploymentType"
                        minOccurs="0" maxOccurs="unbounded" />
            </sequence>
        </complexType>
    </element>
```

```
    <complexType name="EmploymentType">
        <attribute name="startdate" type="date" use="required" />
        <attribute name="enddate" type="date" use="optional" />
        <attribute name="employer" type="string" use="required" />
    </complexType>
</schema>
```

The course database

GRAPHICAL REPRESENTATION OF THE STRUCTURE OF THE RELATIONAL DATABASE



There are two columns that contain XML: Course.info and Student.contact. The content of these columns complies to the following XML Schemas:

XML SCHEMA FOR COURSE.INFO

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns="http://dsv.su.se/SDXML/coursedb/course"
                targetNamespace="http://dsv.su.se/SDXML/coursedb/course">
    <xs:element name="info">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="length" type="xs:integer" />
                <xs:element name="price" type="xs:integer" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

XML SCHEMA FOR STUDENT.CONTACT

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns="http://dsv.su.se/SDXML/coursedb/student"
                targetNamespace="http://dsv.su.se/SDXML/coursedb/student">
    <xs:element name="contact" type="ContactType"/>
    <xs:complexType name="ContactType">
        <xs:sequence>
            <xs:element name="phone" type="PhoneType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="address" type="AddressType"/>
        </xs:sequence>
    </xs:complexType>
```

```
    <xs:complexType name="PhoneType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="type" type="xs:string" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="AddressType">
        <xs:sequence>
            <xs:element name="street" type="xs:string"/>
            <xs:element name="number" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="code" type="xs:string" use="required"/>
        <xs:attribute name="city" type="xs:string" use="required"/>
    </xs:complexType>
</xs:schema>
```

DATABASE CONTENT

Course

coursecode	coursename	info
DBM1	Database Methodology	<info><length>2</length><price>2800</price></info>
Java1	Java, basic course	<info><length>5</length><price>6700</price></info>
Java2	Java, advanced course	<info><length>4</length><price>6000</price></info>
LDBD	Logical Database Design	<info><length>4</length><price>6000</price></info>
Log1	Logic	<info><length>3</length><price>4500</price></info>
PDBD	Physical Database Design	<info><length>5</length><price>7200</price></info>

Student

sid	sname	contact
1	Bo Dahl	<contact><phone type="home">163578</phone><phone type="mobile">073564889</phone><address code="16102" city="Bromma"><street>Ahlgatan</street><number>6</number></address></contact>
2	Ann Stål	<contact><phone type="home">373789</phone><phone type="mobile">073148964</phone><address code="16429" city="Kista"><street>Lindvägen</street><number>3</number></address></contact>
3	Ebba Ryd	<contact><address code="16107" city="Bromma"><street>Ankvägen</street><number>4</number></address></contact>
4	Robert Ahl	<contact><phone type="home">123435</phone><phone type="mobile">073552491</phone><address code="16425" city="Kista"><street>Ekvägen</street><number>1</number></address></contact>
5	Lars Holm	<contact><phone type="home">203045</phone><address code="16966" city="Solna"><street>Skolgatan</street><number>3</number></address></contact>
6	Siw Björk	<contact><phone type="home">452678</phone><address code="16431" city="Kista"><street>Bokvägen</street><number>2</number></address></contact>
7	Sigge Ehn	<contact><phone type="home">245578</phone><phone type="mobile">073691934</phone><address code="16429" city="Kista"><street>Bokvägen</street><number>24</number></address></contact>
8	Kurt Grahn	<contact><phone type="home">192292</phone><phone type="mobile">073318175</phone><address code="19735" city="Bro"><street>Byvägen</street><number>112</number></address></contact>
9	Eva Jung	<contact><phone type="mobile">073559185</phone><address code="16966" city="Solna"><street>Storgatan</street><number>5</number></address></contact>
10	Lola Frid	<contact><phone type="home">723384</phone><phone type="mobile">073292612</phone><address code="18754" city="Täby"><street>Lillgatan</street><number>3</number></address></contact>

CourseOffering

startdate	course	teacher	room
2008-03-06	Java1	2	Tellus
2008-04-02	DBM1	5	Sirius
2008-04-16	Java2	2	Orion
2008-04-16	Log1	3	Jupiter
2008-05-06	LDBD	5	Sirius
2008-05-06	Java1	2	Jupiter
2008-05-09	FDBD	4	Jupiter
2008-09-02	Java1	3	Tellus
2008-09-02	LDBD	4	Orion
2009-01-22	Java2	2	Orion
2009-01-28	DBM1	5	Jupiter

Room

name	capacity
Jupiter	12
Orion	24
Sirius	16
Tellus	32

Teacher

tid	tnamn	office	phone
1	Anders Ödman	634	151576
2	Bo Åkerman	604	151526
3	Carl Nordin	603	151553
4	Lena Svensson	605	151556
5	Sofia Wilsson	622	151585

Participation

startdate	course	student
2008-03-06	Java1	1
2008-04-02	DBM1	1
2008-04-16	Java2	7
2008-04-16	Log1	4
2008-05-06	LDBD	7
2008-05-06	Java1	2
2008-05-09	FDBD	8
2008-09-02	Java1	1
2008-09-02	LDBD	8
2009-01-22	Java2	2
2009-01-28	DBM1	3
2008-03-06	Java1	9
2008-04-02	DBM1	2
2008-04-16	Java2	5
2008-04-16	Log1	5
2008-05-06	LDBD	1
2008-05-06	Java1	6
2008-05-09	FDBD	6
2008-09-02	Java1	9
2008-09-02	LDBD	6
2009-01-22	Java2	9
2009-01-28	DBM1	5
2008-03-06	Java1	8
2008-04-02	DBM1	10
2008-04-16	Java2	6
2008-03-06	Java1	10
2008-04-02	DBM1	3
2008-03-06	Java1	3
2009-01-22	Java2	10

11 Lesson exercises

During four lessons we will solve together exercises in the different areas discussed in previous lectures. The exercises are described in this chapter. It can be a good idea to take a look at the exercises in order to be familiar with them, perhaps even have tried to solve them, though that is not a requirement. We will solve them together on the whiteboard or using a computer. Solutions will be available in iLearn after each lesson.

Lesson 1 & 2 - Data structures, Lorel, XPath, XQuery

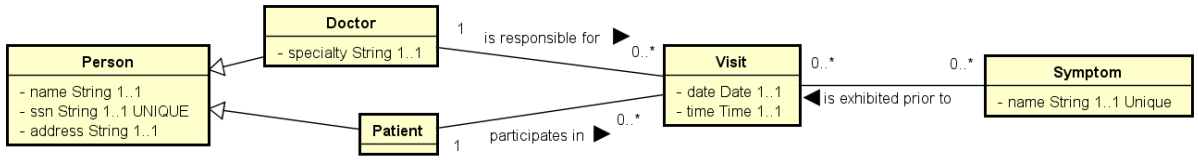
GOAL

To understand how SSD graphs, SSD expressions, XML and JSON represent information.
To understand how to design a suitable data structure from a conceptual model.
To understand pros and cons of different structures when it comes to redundancy and integrity.
To understand how to navigate semi-structured data and how to use XPath to navigate XML structures.
To understand how XQuery can be used to query XML data and construct XML as a result.
To understand how XQuery can be used to modify an existing XML structure.

Understand how SSD graphs, SSD expressions, XML and JSON represent information.
how to design a data structure from a conceptual model.
pros and cons of different structures when it comes to redundancy and integrity.
How to navigate semi-structured data and use Xpath to navigate XML structures.
How Xquery used to query XML data and construct XML as a result

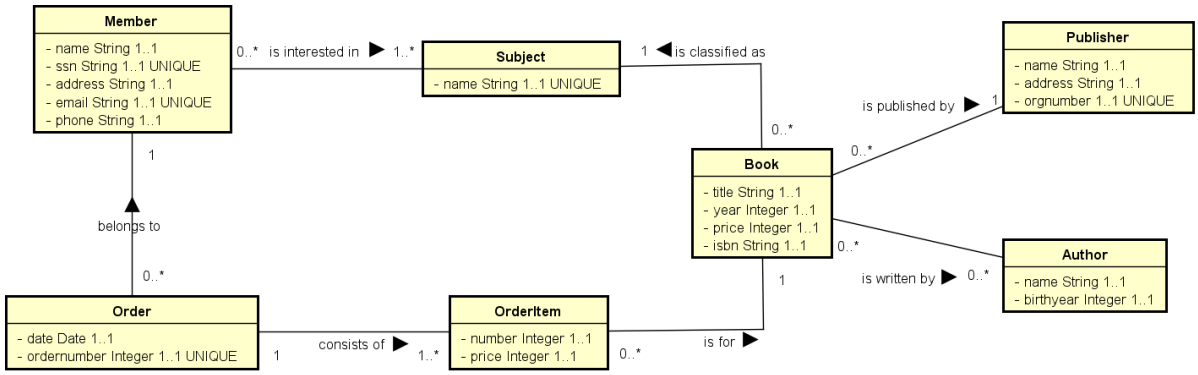
EXERCISE SD1

Create suitable structures in SSD, XML and JSON for the following conceptual model! Do not think about how a relational database would turn out! Think of the possibilities and limitations of semi-structured data structures!



EXERCISE SD2

Create suitable structures in SSD, XML and JSON for the following conceptual model!



EXERCISE SD3

Convert the movie database (the XML document) into an SSD graph and an SSD expression! The first two movies should be enough.

Write Lorel statements for the following:

1. Retrieve the name of the production company of the movie "True Lies"!
2. Retrieve the name of the actors of the movie "Sliver"!
3. Retrieve the title and the director's name for movies where the director is also one of the actors!

EXERCISE SD4

Convert the movie database (the XML document) into JSON! The first two movies should be enough. Construct a corresponding JSON Schema!

EXERCISES IN XQUERY

The following exercises shall be solved with XQuery based on the movie database (the file movies.xml). The database is described in chapter 10 and is available in iLearn. The exercises can be interpreted in different ways, so, try different solutions with different result structures.

EXERCISE XQ1

Retrieve all the directors!

EXERCISE XQ2

Retrieve all the actors from USA that are part of the movie Driven!

EXERCISE XQ3

Show the number of actors per movie!

EXERCISE XQ4

Find the movies that have a director that is an actor in the movie!

EXERCISE XQ5

Show the movies per production company!

EXERCISE XQ6

Show the movies per actor!

EXERCISE XQ7
Show the age of each actor per movie!

EXERCISE XQ8
Show the titles of movies without Sharon Stone!

EXERCISE XQ9
Show the movies that only have actors from USA!

EXERCISE XQ10
Find all the actors that Eric Darnell has directed!

EXERCISE XQ11
Retrieve all the people and create an attribute that has the value "director", "actor" or "both"!

EXERCISE XQ12
What are "Driven", "Pitof", "Universal" and "Canada"? Show if they are attribute values of text nodes and for what element!

EXERCISE XQ13
Fix the year of the movie "The Kid And I" to 2005!

EXERCISE XQ14
Add Joe Mantegna (1947, USA) as an actor to the movie "The Kid And I"!

EXERCISE XQ15
Fix Sylvester Stallone's name! Sylvester instead of Silvester.

EXERCISE XQ16
Restructure the data with minimal redundancy!
Would the previous exercises be easier or harder to solve against the new structure?

Lesson 3 - XSLT

GOAL

To understand how XSLT can be used to transform XML using recursion and iteration.
To understand how XSLT templates can be called.
To understand how output can be created with and without XSLT instructions.

EXERCISES ON XSLT

The following exercises shall be solved with XSLT based on the movie database (the file movies.xml). The database is described in chapter 10 and is available in iLearn. Try different solutions in order to compare different ways to achieve similar results! Try solutions compatible with XSLT 1, 2 or 3!

EXERCISE XT1
Desired result: An XML document with every actor as an element.

EXERCISE XT2
Desired result: An HTML document with all the actors as an HTML table.

EXERCISE XT3
Desired result: An HTML document with the movies listed per director (nested lists).

EXERCISE XT4
Desired result: An HTML document with all the movies produced by Tri-Star. Include information about each movie's director and actors.

EXERCISE XT5
Desired result: An XML document with all the movies grouped by production company and year. Include information about each movie's director and the actors grouped by country.

Lesson 4 - SQL/XML

GOAL

To understand how SQL/XML publishing functions can be used to transform the result of a SELECT statement into XML.
To understand how to use XQuery inside SQL to query and retrieve data stored as XML.
To understand how XQuery can be used to modify XML data stored inside tables.

EXERCISES IN SQL/XML

The following exercises shall be solved with standard SQL based on the course database. The database is described in chapter 10 and is available in iLearn. Try different solutions in order to compare different ways to achieve similar results! Use the DB2 version of the database for testing!

EXERCISE SX1

Retrieve the name of the courses that have used the room Orion! One element per course.

EXERCISE SX2

How many course offerings has each student participated in? Show the name of the student and the number of course offerings! One element per student.

EXERCISE SX3

Retrieve information about every course offering! Include the course name, start date, room name and teacher name! One element per course offering.

EXERCISE SX4

Retrieve information about every teacher! One element per teacher inside a root element.

EXERCISE SX5

Which rooms have been used by each teacher? One element per teacher with one subelement per used room.

EXERCISE SX6

Which teachers has each student met? One element per student with one subelement per teacher. Everything inside a root element.

EXERCISE SX7

How many rooms and how many teachers are associated to each course? One element per course inside a root element.

EXERCISE SX8

Which rooms have been used by four-week courses? Show the name and the capacity!

EXERCISE SX9

Which students have a mobile phone? Show the name and the mobile number.

EXERCISE SX10

How many phone numbers does each student have? Show the name and number!

EXERCISE SX11

Show the students grouped by city! One element per city, one subelement per student. Include the name and all phone numbers!

EXERCISE SX12

Retrieve the phone numbers of Robert Ahl! Two columns: type and number.

EXERCISE SX13

Lars Holm got a mobile! The number is 070919293. Update the database!

EXERCISE SX14

All course prices must be raised will 100 SEK per week. Update the database!

12 Quizzes

Eight quizzes must be completed individually during the course. They are available in the course platform iLearn. All quizzes require that every question is answered correctly. Multiple attempts are possible. Here is some information about the quizzes.

Quiz 1 - XML

GOAL

To repeat the theory from the lecture about XML and the different models and node types.

TASK

The quiz should be completed after lecture 1 and before lecture 2.

Quiz 2 - DTD & XML Schema

GOAL

To repeat the theory from the lecture about DTD and XML Schema.

TASK

The quiz should be completed after lecture 2 and before lecture 3.

Quiz 3 - JSON & JSON Schema

GOAL

To repeat the theory from the lecture about JSON and JSON Schema.

TASK

The quiz should be completed after lecture 2 and before lecture 3.

Quiz 4 - Semi-structured data, XML-based languages, usages

GOAL

To repeat the theory from the lectures about semi-structured data, XML-based languages and usages of semi-structured data and XML.

TASK

The quiz should be completed after lecture 3 and before lecture 4.

Quiz 5 - XQuery & XPath

GOAL

To repeat the theory from the lectures about XPath and XQuery.
To practice evaluating some basic XPath expressions and XQuery statements.

TASK

The quiz should be completed after lecture 5 and before lesson 1.

Quiz 6 - XSLT

GOAL

To repeat the theory from the lecture about XSLT.

TASK

The quiz should be completed after lecture 6 and before lesson 3.

Quiz 7 - SQL/XML

GOAL

To repeat the theory from the lecture about SQL/XML.
To practice evaluating SQL statements containing some basic SQL/XML functions.

TASK

The quiz should be completed after lecture 7 and before lecture 8.

Quiz 8 - SQL Server & SQLXML

GOAL

To repeat the theory from the lecture about SQL Server's support for XML, including SQLXML.

TASK

The quiz should be completed after lecture 10.

13 Assignments

Multiple assignments must be submitted during the course. This chapter describes the different assignments. All assignments are submitted in iLearn. There is a specific assignment box for each assignment. Revised solutions are also to be submitted to the same box.

Common instructions for all assignments:
The submission must include the solutions to all exercises/questions/tasks.
The submission must include execution results (as text and/or rendered) when relevant.
The names of the group members must be specified for group submissions (as a confirmation that the group consists of the right students).
Each submission must be one file.
Feedback is provided as an XML file that students may inspect in a way of their choice.
If a submission is graded as "Revision needed", a new submission will become possible in the same box. The new submission should be done as soon as possible.

Grading of the submissions is done continuously. Do not wait in order to submit just before the deadline.

Assignment 1 - Seminar 1: XQuery

Required for a passing grade
Assignment 1 must be submitted before seminar 1 in order to be presented and discussed during the seminar.

GOAL

To practice using XPath and XQuery to navigate and query XML.
To reflect on similarities and differences between different solutions.
To practice using XQuery to modify existing XML data.

TASK

Solve the following with XQuery. Use the movie database! Submit at least one solution that is compatible with XQuery 1 and one solution using XQuery 3 features. Your submission shall therefore have ten solutions.

1. Show the number of different people that have worked for each production company! One element per production company with attributes for the name and number. Everything inside a root element.

2. Show the number of actors per country in each movie! The result shall have the following structure:

```
<result>
    <movie title="">
        <country name="" numberofactors="" />
        <country name="" numberofactors="" />
    </movie>
    <movie title="">
        <country name="" numberofactors="" />
        <country name="" numberofactors="" />
    </movie>
</result>
```

3. Show which actors each director has directed at least twice! Only directors with at least one such actor shall be included in the result. The result shall have the following structure:

```
<Result>
    <Director Name="" Country="" YearOfBirth="">
        <Actor Name="" Country="" />
        <Actor Name="" Country="" />
    </Director>
    <Director Name="" Country="" YearOfBirth="">
        <Actor Name="" Country="" />
        <Actor Name="" Country="" />
    </Director>
</Result>
```

```
Director
  Actor
  Actor
Director
  Actor
  Actor
Director
  Actor
```

4. Show all the movies grouped by year according to the following structure:

```
<Result>
    <Movies Year="" NumberOfMovies="">
        <Movie>
            <Title>title</Title>
            <Director>director's name</Director>
            <Actors>number of actors</Actors>
        </Movie>
        <Movie>
            <Title>title</Title>
            <Director>director's name</Director>
            <Actors>number of actors</Actors>
        </Movie>
    </Movies>
    <Movies Year="" NumberOfMovies="">
        <Movie>
            <Title>title</Title>
            <Director>director's name</Director>
            <Actors>number of actors</Actors>
        </Movie>
    </Movies>
</Result>
```

5. Change the title of "Picking Up the Pieces" to "Picking up all the Pieces" and change the name of the production company "Warner Bros" to "Warner Brothers" (for all relevant movies).

Assignment 2 - Seminar 2: XSLT

Required for a passing grade
Assignment 2 must be submitted before seminar 2 in order to be presented and discussed during the seminar.

GOAL

To practice using XSLT templates.
To experiment with different XSLT techniques for constructing the result.
To compare the possibilities of different XSLT versions.

TASK

Solve the following with XSLT. Use the movie database! Submit at least one solution that is compatible with XSLT 1 and one solution using XSLT 2/3 features. Your submission shall therefore have six solutions.

1. Create a new XML structure with all the movies sorted by year! The actors shall be sorted alphabetically. The result shall have the following structure:

```
<movies>
    <movie title="" year="" productioncompany="" director="the director's name">
        <actor>the actor's name</actor>
        <actor>the actor's name</actor>
    </movie>
    <movie title="" year="" productioncompany="" director="the director's name">
        <actor>the actor's name</actor>
        <actor>the actor's name</actor>
    </movie>
</movies>
```

2. Show all the movies grouped by production company as HTML! One heading per company followed by a table with the corresponding movies. The table shall have one row with headings and one row per movie, with the following information included: Title, Year, Director's name, Actors (name and age at that time).

3. Show all the movies grouped by director as HTML! One big table with one heading row per director with name and country together in one cell and followed by one row per movie with four columns: Title, Year, Production company, Number of actors. Make sure the table is filled with cells correctly by using the html attribute colspan. The directors and the movies shall be sorted alphabetically.

All movies grouped by director as HTML . One big table one heading row per director with name and country in one cell followed by one row per movie with 4 columns - Title ,Year,Production company,Number of actors.

Assignment 3 - Seminar 3: SQL/XML

Required for a passing grade
Assignment 3 must be submitted before seminar 3 in order to be presented and discussed during the seminar.

GOAL

To practice using SQL that generates XML.
To practice using XQuery inside SQL to query and shred XML.
To understand how to combine SQL and XQuery in order to query and modify XML data.
To understand how working with XML in SQL can influence what is possible.

TASK

Solve the following with SQL. Use the course database! The solutions must be compatible with IBM DB2 or Oracle Database, or both.

1. Retrieve the rooms that have been visited by students from Kista! The result shall have the following structure:

```
<Result>
    <Student Name="">
        <Room>the room name</Room>
        <Room>the room name</Room>
    </Student>
    <Student Name="">
        <Room>the room name</Room>
        <Room>the room name</Room>
    </Student>
</Result>
```

2. How many students have participated in each course offering? The result shall have the following structure:

```
<Result>
    <CourseOffering CourseCode="" CourseName="" StartDate="" Room="">
        <Teacher>the teacher's name</Teacher>
        <NumberOfStudents>number</NumberOfStudents>
    </CourseOffering>
    <CourseOffering CourseCode="" CourseName="" StartDate="" Room="">
        <Teacher>the teacher's name</Teacher>
        <NumberOfStudents>number</NumberOfStudents>
    </CourseOffering>
</Result>
```

3. Show the course offerings grouped by course! The result shall have the following structure:

```
<Result>
    <Course Code="" Name="">
        <CourseOffering StartDate="" Room="" Teacher="" NumberOfStudents=""/>
        <CourseOffering StartDate="" Room="" Teacher="" NumberOfStudents=""/>
    </Course>
    <Course Code="" Name="">
        <CourseOffering StartDate="" Room="" Teacher="" NumberOfStudents=""/>
        <CourseOffering StartDate="" Room="" Teacher="" NumberOfStudents=""/>
    </Course>
</Result>
```

4. Show the following information about each course: Code, Name, Length, Price, Number of course offerings. The result shall have five columns and one row per course.

5. Show the following information about each student: ID, Name, City, Mobile number, Number of taken courses. The result shall have five columns and one row per student.

6. Get the total cost per student based on their participations! All students shall be included in the result. The result shall have the following structure:

```
<Result>
    <Student ID="" Name="" City="" NumberOfParticipations="" TotalCost="" />
    <Student ID="" Name="" City="" NumberOfParticipations="" TotalCost="" />
    <Student ID="" Name="" City="" NumberOfParticipations="" TotalCost="" />
</Result>
```

Assignment 4 - XQuery

Required for a passing grade

GOAL

To practice using XPath and XQuery.
To practice using DTD.
To compare how XPath and XQuery may sometimes be used to achieve the same effect.
To understand how to handle possible data redundancy in XQuery.

TASK

Solve the following with XQuery! Use the book database (XML files)! The solutions may be compatible with XQuery 1, but they don't have to.

1. Retrieve all authors per genre! The result must comply with the following DTD:
   <!ELEMENT Result (Genre*)>
   <!ELEMENT Genre (Author*)>
   <!ATTLIST Genre Name CDATA #REQUIRED>
   <!ELEMENT Author (#PCDATA)>
   The text node of the element Author shall contain the name of the author.

2. Which books have been translated to Russian and at least one more language? The result must comply with the following DTD:
   <!ELEMENT Result (Book*)>
   <!ELEMENT Book EMPTY>
   <!ATTLIST Book Title CDATA #REQUIRED
   OriginalLanguage CDATA #REQUIRED
   NumberOfOtherLanguages CDATA #REQUIRED>
   The attribute NumberOfOtherLanguages specifies the number of languages other than Russian that the book has been translated to.

3. The translations of which books has each publisher published? The result must comply with the following DTD:
   <!ELEMENT Result (Publisher*)>
   <!ELEMENT Publisher (Book*)>
   <!ATTLIST Publisher Name CDATA #REQUIRED
   Country CDATA #REQUIRED>
   <!ELEMENT Book EMPTY>
   <!ATTLIST Book Title CDATA #REQUIRED
   Genre CDATA #IMPLIED>

4. The French translation of the 1999 edition of the book "Archeology in Egypt" was published by "ABC International", not by "KLC". Correct the error in the database!

5. The author Sam Davis is actually called Samuel Davies. Make the appropriate change in the database!

Assignment 5 - XSLT

Required for a passing grade

GOAL

To practice using XSLT.
To compare different ways of iteration in XSLT.
To experience the limitations of XSLT 1 and XPath 1.

TASK

Solve the following with XSLT. Use the book database (XML files)! The solutions must be compatible with XSLT 1. Solutions using XSLT 2 or 3 features may also be submitted in order to get feedback, but only the XSLT 1 solutions are required for a passing grade.

1. Show all the books by genre! The result shall be an HTML page with each genre as a heading followed by a list of books where the title and the earliest edition's year are shown. Books with no genre may be included or not, depending on your preference and whether the default value is loaded from the DTD or not.

2. Translations of which books has each publisher published? The result must comply with the following DTD:
```
<!ELEMENT Result (Publisher*)>
<!ELEMENT Publisher (Book*)>
<!ATTLIST Publisher Name CDATA #REQUIRED
                    Country CDATA #REQUIRED>
<!ELEMENT Book EMPTY>
<!ATTLIST Book Title CDATA #REQUIRED
               Genre CDATA #IMPLIED>
```

3. Show all the books grouped by original language! The result shall be an HTML page with a table where every language has one or more rows, equal to the number of books. The languages shall be sorted and the books shall be sorted by title! The result shall look like this (depending on whether default values are retrieved from the DTD):

Books by original language **Books by original language**

Language	Title	Genre
English	Archeology in Egypt	Educational
	Contact	Science Fiction
	Database Systems in Practice	Educational
	European History	Educational
	Misty Nights	Thriller
	Music Now and Before	Educational
	Musical Instruments	Educational
	Oceanography for Dummies	Educational
	Oceans on Earth	Educational
	The Beach House	Novel
	The Fifth Star	Novel
	The Fourth Star	Science Fiction
French	Encore une fois	
	Le chateau de mon pere	
Swedish	Dödliga Data	Thriller
	Midsommar i Lund	Novel
	Våren vid sjön	Novel

Language	Title	Genre
English	Archeology in Egypt	Educational
	Contact	Science Fiction
	Database Systems in Practice	Educational
	European History	Educational
	Misty Nights	Thriller
	Music Now and Before	Educational
	Musical Instruments	Educational
	Oceanography for Dummies	Educational
	Oceans on Earth	Educational
	The Beach House	Novel
	The Fifth Star	Novel
	The Fourth Star	Science Fiction
French	Encore une fois	N/A
	Le chateau de mon pere	N/A
Swedish	Dödliga Data	Thriller
	Midsommar i Lund	Novel
	Våren vid sjön	Novel

Tip: rowspan

Assignment 6 - DB2 & XML

Required for grade E

GOAL

To practice using SQL/XML.
To practice using XQuery including XQuery Update Facility.
To compare the standard to a product specific implementation.

TASK

Solve the following with **SQL** in IBM DB2! Use the book database!

1. The translations of which books has each publisher published? The result must comply with the following DTD:

```
<!ELEMENT Result (Publisher*)>
<!ELEMENT Publisher (Book*)>
<!ATTLIST Publisher Name CDATA #REQUIRED
                    Country CDATA #REQUIRED>
<!ELEMENT Book EMPTY>
<!ATTLIST Book Title CDATA #REQUIRED
               Genre CDATA #IMPLIED>
```

2. Retrieve all the authors and their books! The result shall be an XML document that complies with the following XML Schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="AllAuthors">
        <complexType>
            <sequence>
                <element name="Author" maxOccurs="unbounded">
                    <complexType>
                        <all minOccurs="0">
                            <element name="Book" maxOccurs="unbounded">
                                <complexType>
                                    <attribute name="Title" type="string" use="required"/>
                                    <attribute name="OriginalLanguage" type="string" use="required"/>
                                    <attribute name="Genre" type="string" />
                                </complexType>
                            </element>
                        </all>
                        <attribute name="Name" type="string" use="required"/>
                        <attribute name="Country" type="string" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

3. Retrieve the name and country of authors of books that have been translated to Russian! The result shall have two columns.

4. Retrieve the following information for each book: title, original language, genre, number of editions, number of languages the book is available in, number of authors, year of the earliest edition. The result shall have seven columns (and one row per book).

5. Retrieve information about all the publishers with at least one book (one translation) and the languages of the publisher (the languages this publisher has at least one translation in)! The result shall comply with the following XML Schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="Publishers">
        <complexType>
            <sequence>
                <element name="Publisher" minOccurs="0" maxOccurs="unbounded">
                    <complexType>
                        <sequence>
                            <element name="Language" maxOccurs="unbounded" type="string"/>
                        </sequence>
                        <attribute name="Name" type="string" use="required"/>
                        <attribute name="Country" type="string" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

6. The French translation of the 1999 edition of the book "Archeology in Egypt" was published by "ABC International", not by "KLC". Correct the error in the database!

7. The name of the publisher "Addison" is supposed to be "Éditions Addison". Correct the error in the database!

Solve the following with **XQuery** in IBM DB2! Use the book database!

8. Retrieve information about the books in the database! The result shall have the following structure:

```
<Result>
    <Language Name="" NumberOfBooks="" />
    <Language Name="" NumberOfBooks="" />
    <Language Name="" NumberOfBooks="" />
</Result>
```

The attribute NumberOfBooks specifies how many books that have that language as their original language. The translations are not relevant here.

Assignment 7 - Oracle DB & XML

Required for grade E

GOAL

To practice using SQL/XML.
To practice using XQuery including XQuery Update Facility.
To compare the standard to a product specific implementation.

TASK

Solve the following with **SQL** in Oracle Database! Use the book database!

1. The translations of which books has each publisher published? The result must comply with the following DTD:
```
<!ELEMENT Result (Publisher*)>
<!ELEMENT Publisher (Book*)>
<!ATTLIST Publisher Name CDATA #REQUIRED
                    Country CDATA #REQUIRED>
<!ELEMENT Book EMPTY>
<!ATTLIST Book Title CDATA #REQUIRED
               Genre CDATA #IMPLIED>
```

2. Retrieve all the authors and their books! The result shall be an XML document that complies with the following XML Schema:
```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="AllAuthors">
        <complexType>
            <sequence>
                <element name="Author" maxOccurs="unbounded">
                    <complexType>
                        <all minOccurs="0">
                            <element name="Book" maxOccurs="unbounded">
                                <complexType>
                                    <attribute name="Title" type="string" use="required"/>
                                    <attribute name="OriginalLanguage" type="string" use="required"/>
                                    <attribute name="Genre" type="string" />
                                </complexType>
                            </element>
                        </all>
                        <attribute name="Name" type="string" use="required"/>
                        <attribute name="Country" type="string" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

3. Retrieve the name and country of authors of books that have been translated to Russian! The result shall have two columns.

4. Retrieve the following information for each book: title, original language, genre, number of editions, number of languages the book is available in, number of authors, year of the earliest edition. The result shall have seven columns (and one row per book).

5. Retrieve information about all the publishers with at least one book (one translation) and the languages of the publisher (the languages this publisher has at least one translation in)! The result shall comply with the following XML Schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="Publishers">
        <complexType>
            <sequence>
                <element name="Publisher" minOccurs="0" maxOccurs="unbounded">
                    <complexType>
                        <sequence>
                            <element name="Language" maxOccurs="unbounded" type="string"/>
                        </sequence>
                        <attribute name="Name" type="string" use="required"/>
                        <attribute name="Country" type="string" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

6. The French translation of the 1999 edition of the book "Archeology in Egypt" was published by "ABC International", not by "KLC". Correct the error in the database!

7. The name of the publisher "Addison" is supposed to be "Éditions Addison". Correct the error in the database!

Solve the following with **XQuery** in Oracle Database! Use the book database!

8. Retrieve information about the books in the database! The result shall have the following structure:

```
<Result>
    <Language Name="" NumberOfBooks="" />
    <Language Name="" NumberOfBooks="" />
    <Language Name="" NumberOfBooks="" />
</Result>
```

The attribute NumberOfBooks specifies how many books that have that language as their original language. The translations are not relevant here.

Assignment 8 - SQL Server & XML

Required for grade E

GOAL

To practice using SQLXML, the alternative to SQL/XML.
To compare the standard to a product specific implementation.
To reflect on the differences between SQL/XML and SQLXML and identify pros and cons with each of the alternatives.

TASK

Solve the following with **SQL** in Microsoft SQL Server! Use the book database!

1. The translations of which books has each publisher published? The result must comply with the following DTD:
```
<!ELEMENT Result (Publisher*)>
<!ELEMENT Publisher (Book*)>
<!ATTLIST Publisher Name CDATA #REQUIRED
                    Country CDATA #REQUIRED>
<!ELEMENT Book EMPTY>
<!ATTLIST Book Title CDATA #REQUIRED
               Genre CDATA #IMPLIED>
```

2. Retrieve all the authors and their books! The result shall be an XML document that complies with the following XML Schema:
```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="AllAuthors">
        <complexType>
            <sequence>
                <element name="Author" maxOccurs="unbounded">
                    <complexType>
                        <all minOccurs="0">
                            <element name="Book" maxOccurs="unbounded">
                                <complexType>
                                    <attribute name="Title" type="string" use="required"/>
                                    <attribute name="OriginalLanguage" type="string" use="required"/>
                                    <attribute name="Genre" type="string" />
                                </complexType>
                            </element>
                        </all>
                        <attribute name="Name" type="string" use="required"/>
                        <attribute name="Country" type="string" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

3. Retrieve the name and country of authors of books that have been translated to Russian! The result shall have two columns.

4. Retrieve the following information for each book: title, original language, genre, number of editions, number of languages the book is available in, number of authors, year of the earliest edition. The result shall have seven columns (and one row per book).

5. Retrieve information about all the publishers with at least one book (one translation) and the languages of the publisher (the languages this publisher has at least one translation in)! The result shall comply with the following XML Schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="Publishers">
        <complexType>
            <sequence>
                <element name="Publisher" minOccurs="0" maxOccurs="unbounded">
                    <complexType>
                        <sequence>
                            <element name="Language" maxOccurs="unbounded" type="string"/>
                        </sequence>
                        <attribute name="Name" type="string" use="required"/>
                        <attribute name="Country" type="string" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

6. The French translation of the 1999 edition of the book "Archeology in Egypt" was published by "ABC International", not by "KLC". Correct the error in the database!

7. The name of the publisher "Addison" is supposed to be "Éditions Addison". Correct the error in the database!

Assignment 9 - Optional assignment for grade D

Required for grade D (on top of everything required for E)

GOAL

To improve the skills in using XML Query languages and SQL.
To practice in designing semi-structured data structures using XML and JSON.
To practice in specifying schemas for XML and JSON.

TASK

Solve the following with XQuery! Use the book database (XML files)! The solution may be compatible with XQuery 1, but it doesn't have to.

1. Show all the authors and all the publishers per country! The result must be an XML document with Countries as its root element and comply to the following DTD:
   <!ELEMENT Countries (Country*)>
   <!ELEMENT Country ((Publisher+, Author*) | Author+)>
   <!ELEMENT Publisher EMPTY>
   <!ELEMENT Author EMPTY>
   <!ATTLIST Publisher name CDATA #REQUIRED>
   <!ATTLIST Author name CDATA #REQUIRED>

Solve the following with **SQL** in IBM DB2 and in Microsoft SQL Server! Use the book database!

2. Which languages are each author's books available in? The result must comply with the following XML Schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="AllAuthors">
        <complexType>
            <sequence>
                <element name="Author" minOccurs="0" maxOccurs="unbounded">
                    <complexType>
                        <sequence minOccurs="0" maxOccurs="unbounded">
                            <element name="Language" type="string"/>
                        </sequence>
                        <attribute name="Name" type="string" use="required"/>
                        <attribute name="Country" type="string" use="required"/>
                        <attribute name="NumberOfBooks" type="integer" use="required"/>
                    </complexType>
                </element>
            </sequence>
        </complexType>
    </element>
</schema>
```

   Both the original language of each book and the translation languages are relevant here.

3. Design an XML structure and a JSON object for the following description. The solution must contain the following:
   An XML document with sample data (at least two buildings).
   An XML Schema document corresponding to the XML document.
   A JSON array with the same sample data as the XML document.
   A JSON Schema for the JSON array.

   Every building has a unique name. In each building there are floors (0 for the ground floor, negative numbers for sublevels and positive number for higher floors). On each floor there are zero or more people whose name and age we know. Some buildings have an elevator.

Assignment 10 - Optional assignment for grade C

Required for grade C (on top of everything required for D)

GOAL

To improve the skills in using XML query languages and SQL.
To expand the knowledge about the evolution of XML query languages.

TASK

1. Read through the relevant articles (available under "Relevant web pages" and "Files" in iLearn) and write a reflection/discussion (approximately one A4 page, at most two A4 pages) about one of the following topics. You may of course use more sources than the ones provided in iLearn.
   A. What are the possible pros and cons with designing a specialized query language for a specific XML-based language (like RDF XML, RSS, MathML, SVG)? Are there many such query languages? Why so?
   B. JSON is a popular format that shares similarities with XML, but is also different in many ways. Since JSON is not XML, the query languages available for XML are not suitable for JSON. Or are they? Is there a need for a specialized query language for JSON? Discuss pros and cons with a specialized JSON query language (like JSONiq) versus reusing/modifying XML query languages for working with JSON!

Solve the following with XQuery! Use the book database (XML files)! The solution may be compatible with XQuery 1, but it doesn't have to.

2. Retrieve all the books (title and year of earliest edition) according to the following rules: If a book's genre is thriller or science fiction and it has at least one translation or at least two editions, then it is very interesting. If it is a thriller or science fiction but with only one edition and no translations, then it is interesting. If it is educational, then it is not interesting. If it is not thriller, not science fiction and not educational, it is a maybe if its original language is Swedish, otherwise it is not interesting. The result shall have the following structure:
   <BookRanking>
   <VeryInteresting>*book title (year)*</VeryInteresting>
   <VeryInteresting>*book title (year)*</VeryInteresting>
   <VeryInteresting>*book title (year)*</VeryInteresting>
   <Interesting>*book title (year)*</Interesting>
   <Interesting>*book title (year)*</Interesting>
   <Maybe>*book title (year)*</Maybe>
   <Maybe>*book title (year)*</Maybe>
   <Maybe>*book title (year)*</Maybe>
   <NotInteresting>*book title (year)*</NotInteresting>
   <NotInteresting>*book title (year)*</NotInteresting>
   <NotInteresting>*book title (year)*</NotInteresting>
   </BookRanking>
   The books must be ordered from the most interesting to the least interesting and then by title.

Solve the following with **SQL** in Oracle Database and in Microsoft SQL Server! Use the book database!

3. How many publishers are there in each city? Show the cities grouped by country! The result must comply with the following XML Schema:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema">
    <element name="PublishersPerCity">
        <complexType>
            <sequence minOccurs="0" maxOccurs="unbounded">
                <element name="Counrty" type="CountryType"/>
            </sequence>
        </complexType>
    </element>
    <complexType name="CountryType">
        <sequence maxOccurs="unbounded">
            <element name="City" type="CityType"/>
        </sequence>
        <attribute name="Name" type="string" use="required"/>
    </complexType>
    <complexType name="CityType">
        <attribute name="Name" type="string" use="required"/>
        <attribute name="NumberOfPublishers" type="positiveInteger" use="required"/>
    </complexType>
</schema>
```

Assignment 11 - Optional assignment for grade B

Required for grade B (on top of everything required for C)

GOAL

To improve the skills in using XML Query languages and SQL.
To try combining SQL and XSLT to create more dynamic solutions.

TASK

Solve the following with **SQL** in IBM DB2 and in Microsoft SQL Server! Use the car database!

1. Who is employed by each employer right now? Include the employer name, the employee name, the employment length so far in days and the number of cars the employee owns! The result must be an XML document with CurrentEmployees as its root element and comply with the following DTD:

```
<!ELEMENT CurrentEmployees (Company*)>
<!ELEMENT Company (Employee+)>
<!ELEMENT Employee EMPTY>
<!ATTLIST Company Name CDATA #REQUIRED>
<!ATTLIST Employee Name CDATA #REQUIRED
                   EmploymentLength CDATA #REQUIRED
                   NumberOfCars CDATA #REQUIRED >
```

Solve the following with **SQL** in Oracle Database and in Microsoft SQL Server! Use the car database!

2. Which company has never employed a person that owns a NISSAN? The result shall have one column and one row per qualified company.

Solve the following in IBM DB2 or Oracle Database!

3. Create a new database with two tables:
   MovieXML (xmldata XML)
   MovieXSLT (name String Primary Key, xsltdata XML)
   Populate the first table with one row with the content of the XML file movies.xml!
   Populate the second table with three rows corresponding to the following three descriptions of transformations to be done with XSLT. Each row must have a unique name (perhaps A, B and C) and an XSLT document. Since only XSLT 1 is supported in IBM DB2 and Oracle Database, the three XSLT documents must be compatible with XSLT 1 and the limitations of the selected product.
   A. Produce an HTML page with a table with two columns with headings and one row per production company! The first column has the company name and the second column has the number of movies by that company.
   B. Produce an HTML page with a list of countries and for each country a nested list with the actors! Show the countries sorted by name and the actors sorted by name! For each actor include the name and birth year!
   C. Produce an HTML page with information about the directors! Sort the directors by name and show the number of movies they have directed and the number of different actors they have worked with!

   Write three SQL statements that return the transformation result from applying each of the XSLT documents to the content of the XML document in the table MovieXML!
   Save each result as an HTML file in order to open it in a web browser and see the HTML page rendered.
   Submit the three XSLT documents (the content of the table MovieXSLT), the three SQL statements and their respective result (the HTML code), as well as the rendered version of each HTML (screenshot from a web browser).

Assignment 12 - Optional assignment for grade A

Required for grade A (on top of everything required for B)
This assignment can only be submitted individually.

GOAL

To improve the skills in using XML Query languages and SQL.
To get a deeper sense for the XML-based language SVG.
To experiment with a more abstract problem that combines modelling, SQL, XQuery and SVG.

TASK

Solve the following in IBM DB2 or Oracle Database or Microsoft SQL Server! You will need to have some basic understanding of the XML-based language SVG. Use the specification and the tutorial!

Expand the following database model with more columns and/or tables as you see fit for the following purpose. The database may not have XML stored in any columns (not as XML and not as strings). The database must contain information to represent SVG shapes for circles, rectangles, ellipses and lines as well as text. Only the most basic properties must be supported: coordinates, dimensions, colors, transparency, fonts).

Given database model (to be expanded):

Drawing (ID PRIMARY KEY, name STRING UNIQUE)

SVGObject (ID PRIMARY KEY)

DrawingObject (drawingID, svgobjectID, placementorder INTEGER,
PRIMARY KEY (drawingID, svgobjectID),
UNIQUE (drawingID, placementorder))

DrawingObject.drawingID is a foreign key to Drawing.ID
DrawingObject.svgobjectID is a foreign key to SVGObject.ID

Each row in the table DrawingObject specifies that an SVGObject is included in a Drawing and in what order (lowest order number is in the back, highest order number is in the front).

Populate the database with at least two drawings and at least two SVG objects of each type (at least two circles, at least two rectangles, at least two ellipses, at least two lines and at least two texts)! Make sure to differentiate their properties (different sizes, colors, coordinates, etc.)! Populate the table DrawingObject so that each drawing has a few objects, all objects are included in at least one drawing and some objects are included in more than one drawing!
Write an SQL statement for each drawing name that returns the drawing as an SVG document! Save the SVG document as an XML file and open it in a web browser to view it rendered!

Submit the following:

- The complete database model after all your additions and an explanation of your changes.
- The content of all the tables.
- The SQL statements and their result for each drawing. The result must be included in two forms: XML and rendered in a web browser.
- If you have used views, all view definitions as CREATE VIEW statements.