Exam

SDXML
Models and languages for handling semi-structured data and XML

2024-05-30

- The exam consists of three parts:
  Part 1 Theory and modeling (questions 1, 2, 3)
  Part 2 Query languages for SSD and XML (questions 4, 5)
  Part 3 XML and relational databases (question 6)

- The exam is graded based on the grading scale F/Fx/E/D/C/B/A.

- In order to pass the exam, the student must perform at a certain level on each part and at a certain level in total, thus fulfilling all three course goals (which correspond to the three exam parts). The exact levels for each grade are:

	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.

- Specify the **course code**, the **date**, the **question number** and your **anonymous examinee code** on each submitted page!

- Fill out the scanning page with the **number of submitted pages** and specify **which questions that have been answered**!

- No aids allowed.

Reference

SQL

Syntax for SQL SELECT:

```
SELECT [DISTINCT] <column list>
FROM <table list>
[WHERE <conditions>]
[GROUP BY <column list>
  [HAVING <conditions>]]
[ORDER BY <column list>]
```

Common SQL/XML functions:
XMLELEMENT, XMLATTRIBUTES, XMLFOREST, XMLAGG, XMLCONCAT, XMLTEXT, XMLEXISTS, XMLQUERY, XMLTABLE, XMLCOMMENT, XMLPI, XMLCAST

XML Schema

Common elements:
schema, element, attribute, complexType, simpleType, group, all, sequence, choice, restriction, extension, simpleContent, complexContent

XQuery

Syntax for XQuery FLWOR:

```
for <loop variables>
let <variable assignments>
where <conditions>
(group by <grouping expressions>) only in XQuery 3
order by <sorting expressions>
return <result>
```

Common XQuery functions:
distinct-values(), count(), sum(), min(), max(), avg(), empty(), exists(), not(), data()

XSLT

Common elements:
transform, output, variable, template, for-each, for-each-group, apply-templates, call-template, if, choose, when, otherwise, element, attribute, comment, processing-instruction, value-of, text, copy, copy-of, sort, param, with-param

SQL Server SQL

SQL clause: FOR XML PATH | AUTO | RAW
Relevant keywords: ELEMENTS, ROOT, TYPE
XML methods: query, value, nodes, exist, modify
XQuery functions: sql:column, sql:variable

Question 1 (6 points)

Question 1 consists of 6 terms to be explained/defined in short. The answers only need to illustrate understanding. No need for any long essays.

Explain the following terms:

1. Well-formed XML
2. PSVI
3. processing-instruction
4. JSON Schema
5. shredding
6. SVG

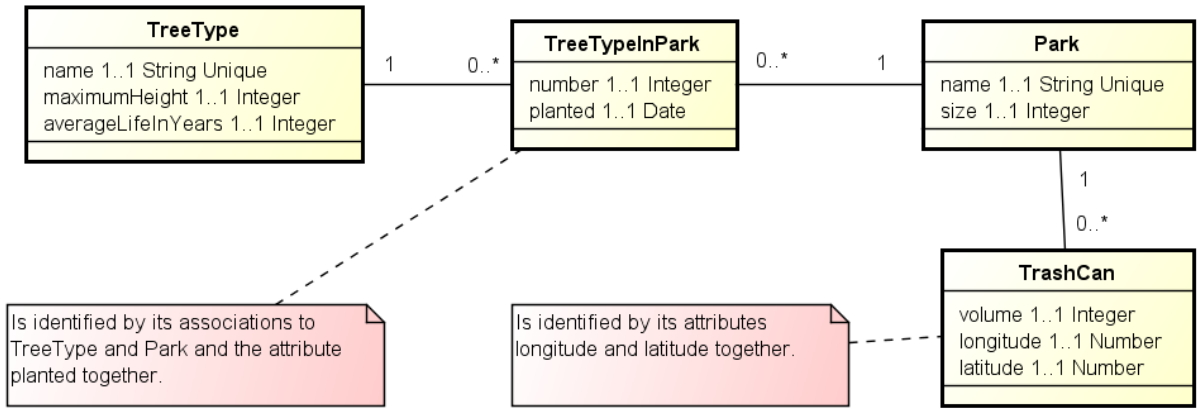
Question 2 (2 points)

Create a JSON object that conforms to the following JSON Schema!

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "http://dsv.su.se/SDXML/jsonschema/product",
  "type": "object",
  "title": "A product",
  "description": "An object representation of a product",
  "properties": {
    "name": {"type": "string"},
    "colors": {
      "type": "array",
      "items": {"type": "string"},
      "minItems": 3,
      "uniqueItems": true
    },
    "sizes": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "code": {"type": "string"},
          "price": {"type": "integer"}
        },
        "additionalProperties": false,
        "required": ["code", "price"]
      },
      "minItems": 2,
      "uniqueItems": true
    },
    "available": {"type": "boolean"}
  },
  "additionalProperties": false,
  "required": ["name", "colors", "sizes", "available"]
}
```

Question 3 (3 points)

Construct an XML document (with sample data) and a corresponding XML Schema for representing the same information as the provided conceptual model! No need to explicitly define what is unique. The solution must consider integrity rules and avoid unnecessary redundancy.



Comments about the attributes:
TreeType.maximumHeight is specified in meters.
TreeTypeInPark.number is always 1 or greater, and it specifies how many trees of the associated type that were planted in the associated park at the specific date.
Park.size is specified in square meters.
TrashCan.volume is specified in liters.

Question 4 (2 + 2 + 2 = 6 points)

Consider the XML document on the last page (after question 6) that only contains sample data in order to illustrate the structure.

a) Write XQuery for the following:
Retrieve information about every movie per genre according to the following structure!

```
<Genres>
    <Genre Name="">
        <Movie Title="" NumberOfOtherGenres=""/>
        <Movie Title="" NumberOfOtherGenres=""/>
        <Movie Title="" NumberOfOtherGenres=""/>
    </Genre>
    <Genre Name="">
        <Movie Title="" NumberOfOtherGenres=""/>
        <Movie Title="" NumberOfOtherGenres=""/>
    </Genre>
</Genres>
```

b) Write XQuery for the following:
Retrieve information about all the showings according to the following structure!

```
<Result>
    <Genre name="">
        <Showing starttime="" movie="" cinema="" hall=""/>
        <Showing starttime="" movie="" cinema="" hall=""/>
        <Showing starttime="" movie="" cinema="" hall=""/>
    </Genre>
    <Genre name="">
        <Showing starttime="" movie="" cinema="" hall=""/>
        <Showing starttime="" movie="" cinema="" hall=""/>
    </Genre>
</Result>
```

c) What is the result of the following XQuery expression with the given sample data?
element D {(//text()/ancestor::*[not(@*)]/name())[position() < 4]}

Question 5 (3 points)

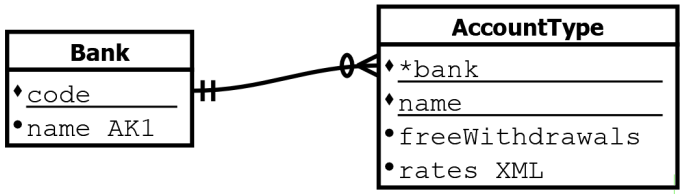
Consider the XML document on the last page (after question 6) that only contains sample data to illustrate the structure!
Write an XSLT that presents the data as html in the following fashion:

Cinemas

Cinema	Number of halls	Number of showings	Number of different movies shown
Saga	3	7	3
Scandinavia	2	5	3
Grand	3	8	4

Question 6 (2 + 2 + 2 + 2 = 8 points)

Consider the following relational database model!



Comments:
The column AccountType.rates is of the data type XML and accepts data that conform to the rules in the following DTD with root element Data. The column AccountType.freeWithdrawals is BOOLEAN. All other columns are STRING. All columns are defined as NOT NULL.

```
<!ELEMENT Data (Period+)>
<!ELEMENT Period (InterestRate+)>
<!ATTLIST Period StartDate CDATA #REQUIRED EndDate CDATA #IMPLIED >
<!ELEMENT InterestRate EMPTY>
<!ATTLIST InterestRate MinimumBalance CDATA #REQUIRED Rate CDATA #REQUIRED>
```

There is always at most one Period without an EndDate and no two periods may overlap. No two InterestRate elements inside the same Period may have the same MinimumBalance. The values of the attributes StartDate and EndDate are always valid dates. The values of MinimumBalance and Rate are always valid numbers.

Write **standard SQL** for the following:

a) Retrieve information about the highest ever interest rate per bank! The result shall have one row per bank and the following columns: Code, Name, HighestRateEver.

b) Retrieve information about all the account types that offer free withdrawals! The result shall have the following structure:

```
<Result>
    <Bank Name="" Code="">
        <AccountType Name=""/>
    </Bank>
    <Bank Name="" Code="">
        <AccountType Name=""/>
        <AccountType Name=""/>
    </Bank>
</Result>
```

c) Which account types offer currently at least 4% in interest rate if you have a balance of 50000? The result shall have the following structure:

```
<Result>
    <AccountType bankcode="" bankname="" name=""/>
    <AccountType bankcode="" bankname="" name=""/>
    <AccouctType bankcode="" bankname="" name=""/>
</Result>
```

Write **SQL Server SQL** for the following:

d) Retrieve information about each bank according to the following structure:

```
<Result>
    <Bank code="" name="" numberOfActiveAccountTypes=""/>
    <Bank code="" name="" numberOfActiveAccountTypes=""/>
</Result>
```

Definition: An account type is considered to be active if it has a rate period without an end date.

XML document for question 4 and question 5

```
<?xml version="1.0" encoding="UTF-8" ?>
<MoviesToday>
    <Movie title="Civil War" length="109">
        <Genre>Action</Genre>
        <Genre>Thriller</Genre>
        <Cinema name="Saga">
            <Showing starttime="14:30" hall="1" ticketprice="110"/>
            <Showing starttime="16:30" hall="2" ticketprice="90"/>
            <Showing starttime="17:50" hall="1" ticketprice="110"/>
        </Cinema>
        <Cinema name="Scandinavia">
            <Showing starttime="16:00" hall="1" ticketprice="110"/>
            <Showing starttime="18:20" hall="1" ticketprice="120"/>
        </Cinema>
        <Cinema name="Grand">
            <Showing starttime="22:10" hall="1" ticketprice="120"/>
        </Cinema>
    </Movie>
    <Movie title="Furiosa" length="158">
        <Genre>Action</Genre>
        <Cinema name="Saga">
            <Showing starttime="14:40" hall="3" ticketprice="110"/>
            <Showing starttime="18:00" hall="3" ticketprice="120"/>
            <Showing starttime="21:50" hall="1" ticketprice="120"/>
        </Cinema>
        <Cinema name="Scandinavia">
            <Showing starttime="17:00" hall="2" ticketprice="120"/>
            <Showing starttime="19:50" hall="2" ticketprice="120"/>
        </Cinema>
        <Cinema name="Grand">
            <Showing starttime="16:00" hall="1" ticketprice="120"/>
            <Showing starttime="19:20" hall="1" ticketprice="120"/>
        </Cinema>
    </Movie>
    <Movie title="Abigail" length="94">
        <Genre>Comedy</Genre>
        <Genre>Horror</Genre>
        <Genre>Fantasy</Genre>
        <Cinema name="Saga">
            <Showing starttime="14:20" hall="2" ticketprice="90"/>
        </Cinema>
        <Cinema name="Scandinavia">
            <Showing starttime="21:00" hall="1" ticketprice="100"/>
        </Cinema>
        <Cinema name="Grand">
            <Showing starttime="16:00" hall="3" ticketprice="90"/>
            <Showing starttime="18:20" hall="3" ticketprice="100"/>
            <Showing starttime="21:10" hall="3" ticketprice="100"/>
        </Cinema>
    </Movie>
    <Movie title="Whitney" length="97">
        <Genre>Drama</Genre>
        <Genre>Music</Genre>
        <Cinema name="Grand">
            <Showing starttime="15:20" hall="2" ticketprice="100"/>
            <Showing starttime="22:10" hall="2" ticketprice="100"/>
        </Cinema>
    </Movie>
</MoviesToday>
```