

Lena Strömbäck

Written exam in

TDDD43 Databases and Data Models - Advanced course

Location:

Date: 2010-04-08

Time: 08-12

Equipment: None

Requirements:

15 points is required for passing the exam

Approx. 20 points is required for grade 4

Approx. 25 points is required for grade 5

The results is announced within two weeks

Teacher in charge: Lena Strömbäck, 0709-39 6776, 013-28 23 24

Use one sheet for each question.

Only write on one side of each sheet.

Give relevant answers to the question.

Points can be deducted for answers that are not answers to the question.

1. XML modeling and querying (5p)

The XML document in example 1 at the end of the exam.

- a) State at least one drawback with the chosen XML model. (1p)
- b) Express “*Who is grandchild of Siv?*” as an XPath query for the second example model. (1p)
- c) Propose a new XML model for the given data that does not suffer from the drawback you stated in a). (2p)
- d) Express “*Who is grandchild of Siv?*” as an XQuery query for your suggested model. (1p)

2. RDF (4p)

Consider the data in the example above together with the following facts:

A man is a person
A woman is a person
A mother is a woman
A father is a man

- a) Give an RDF representation for the facts and the data. It is OK to state it as a set of triples, you do not need to use the correct RDF notation. (3p)
- b) Give one sample query for your model that demonstrates the benefit of using RDF and SPARQL compared to pure XML. It is OK to state your query in natural language, you do not need to use the SPARQL syntax. You must explain the benefits and why you choose the query. (1p)

3. XML storage (2p)

What is hybrid XML storage? Give one example of how it is used and discuss for which kind of data it is beneficial.

4. Data Guides (3 p)

In example 2 at the end of this exam an example data guide is given using the OEM model.

- a) Draw a minimal data guide for this data. (1 p)
- b) Draw a strong data guide for this data. (2p)

5. OODBMS modeling (1+1+1+1 = 4p)

In a distribution company deliveries are made up out of a number of parcels and a number of loose items. The parcels contain one pallet and a number of items. Items have identification numbers. All items in a parcel have the same identification number. Attached to a parcel is also a parcel card. It contains an own identification number for the parcel as well as the identification number of the items in the parcel. Deliveries to Australia have special pallets. For import restriction reasons, they can not be made of wood.

- (a) Explain inheritance and give an example based on the scenario above.
- (b) Explain composite objects and give an example based on the scenario above.
- (c) Explain operator overloading and give an example based on the scenario above.
- (d) Explain encapsulation. Why is this important?

6. Ontologies (2p)

Give 4 uses of ontologies.

7. Description logics (4p)

Define the following concepts using description logics:

C1: team for which all members are IDA-employees

C2: team for which there exists a member that is an IDA-employee

Is there an is-a relation between C1 and C2, i.e. C1 isa C2? Prove your answer using a tableau algorithm.

8. Database Integration (1+2 = 3p)

Given:

Data source 1: CarReview (model, year, review, review-date)

Data source 2: CarForSale(model, year, price, seller)

and a global schema:

Review (model, year, review)

Sale (model, year, price)

- (a) Define local as view mappings between the global schema and the content of the local sources.
- (b) Write the query: give the reviews and prices for all Saab cars from year 2000. Further, show how the bucket algorithm works for processing the query..

9. Ontology alignment (3p)

Explain what the purpose of the matchers is in the ontology alignment systems.

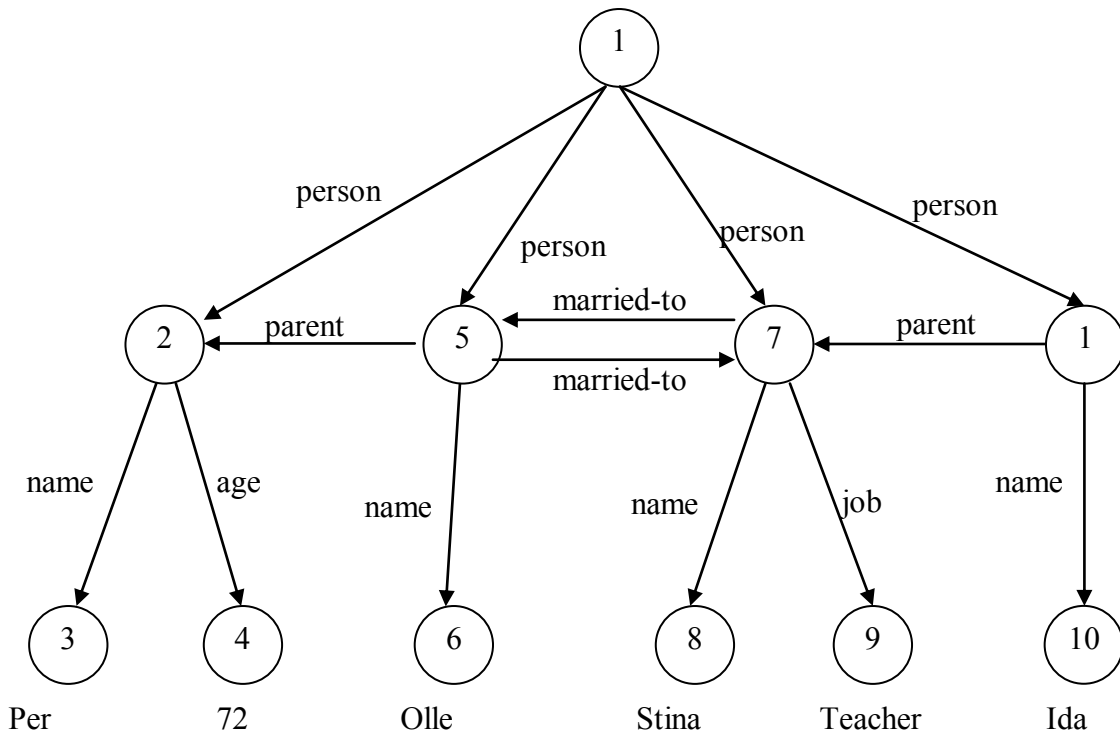
What are the different kinds of strategies for matchers. For each strategy, give an example.

```

<?xml version="1.0" encoding="UTF-8" ?>
<ancestors>
  <person name="Lena">
    <father>
      <person name="Lars" />
    </father>
    <mother>
      <person name="Siv" />
    </mother>
  </person>
  <person name="Göran">
    <father>
      <person name="Lennart" />
    </father>
    <mother>
      <person name="Gudrun" />
    </mother>
  </person>
  <person name="Ludvig">
    <father>
      <person name="Göran">
        <father>
          <person name="Lennart" />
        </father>
        <mother>
          <person name="Gudrun" />
        </mother>
      </person>
    </father>
    <mother>
      <person name="Lena">
        <father>
          <person name="Lars" />
        </father>
        <mother>
          <person name="Siv" />
        </mother>
      </person>
    </mother>
  </person>
  <person name="Filip">
    <father>
      <person name="Mats">
        <father>
          <person name="Bertil" />
        </father>
        <mother>
          <person name="Helena" />
        </mother>
      </person>
    </father>
    <mother>
      <person name="Lena">
        <father>
          <person name="Lars" />
        </father>
        <mother>
          <person name="Siv" />
        </mother>
      </person>
    </mother>
  </person>
</ancestors>

```

Example 1: Ancestors of persons



Example 2: Example data guide