DMDB
WiSe 2019/20
Ticket to Class 3

Name: Moser Markus

Questions based on Jennifer Widom's videos on "well-formed XML" and on XPath. Some illustrations are reproduced from Jennifer Widom's course material.

1. XML stands for eXtensible Markup Language
2. Please complete the gaps:
   - XML is a standard for data representation and exchange (initially for ↓ on the internet)
   - The document format is similar to html,
     but the tags in xml describe content rather than how to format the data
   - Also a streaming format format. (≈ typically used in programs for emitting and consuming xml)
3. The basic constructs are
   - Tagged elements (nested)
   - Attributes (<Book Price="100"> 1984 </Book>)
   - Text.
4. Any element can have any number of attributes as long as their
   attribute names are unique.
5. If you think of an XML document as a tree, the texts form the
   leaf elements.

XML Design goals
- data transfer
- easy to write code to read/write
- document validation possible
- human readable
- supports a wide variety of apps

6. Complete the following comparison between the relational model and XML.

at least 35y.o.

	Relational	XML
Structure	Tables	Hierarchical Tree, (graph)
Schema	Fixed in advance	Flexible; "self-describing"
Queries	simple nice langs. (☺)	trickier, newer (😐)
Ordering	None. (Order by, but data is unordered)	Induced sequential order => implied
Implementation	Native	typically as Add-on (over relational DB)

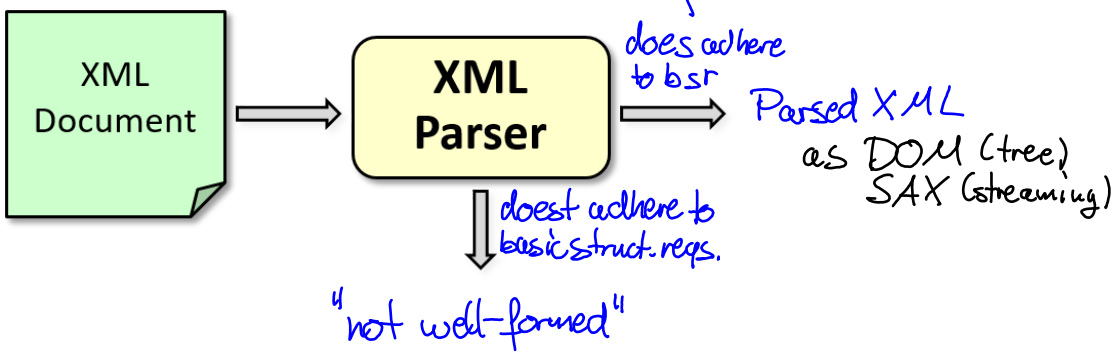
schema, count, attributes, elements can be optional

7. Well-formed XML adheres to

basic structural requirements:

- single root element
- matched tags, proper nesting
- unique attributes (names) within elements.

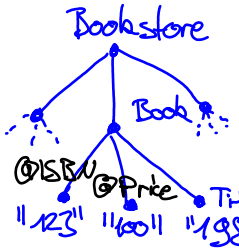
8. Label the illustration for an XML parser:



9. We can access parsed XML via two standards,

   - DOM (Document object model, tree)
   - SAX (Simple API for XML, stream model)

10. Basic constructs in XPath are



   - / root element, separator
   - name_of_element navigate to subelement with name
   - * matches anything
   - @attr_name match attribute with name
   - // matches every descendent + self
   - [c] matches condition c, ie. [@Price < 50]
   - [n] matches n-th subelement, 1-based
   - f(a,b) built-in functions

   - axe:: navigation "axes" (13 of them)

11. XPath queries operate on and return

Sequence of elements

Querying XML
- immature
- newer
- no underlying algebra
1. XPath: path expressions + conditions
2. XSLT: XPath + transformations, + output formatting
3. XQuery: XPath + full-featured Query Language
4. XLink
5. XPointer

examples
- contains(s1, s2)
  ↳ true if s1 contains s2
- name()
  ↳ tag of current element in path

- parent::
- following-sibling::
- descendents::
  ↳ without self
- self::