

Klausur “Semistrukturierte Daten und XML”
Sommersemester 2004
Prof. Dr. Wolfgang May
27. Juli 2004, 10-12 Uhr
Bearbeitungszeit: 90 Minuten

Vorname:

Nachname:

Matrikelnummer:

Bei der Klausur sind **keine Hilfsmittel** (Skripten, Taschenrechner etc.) erlaubt. Handies müssen ausgeschaltet sein. Papier wird gestellt. Benutzen Sie nur die **ausgeteilten**, zusammengehefteten **Blätter** für Ihre Antworten. Schreiben Sie mit blauem/schwarzem Kugelschreiber, Füller etc.; Bleistift ist nicht erlaubt.

Zum **Bestehen** der Klausur sind **45** Punkte hinreichend.

- meine Note soll mit Matrikelnummer so bald wie möglich auf der Vorlesungs-Webseite veröffentlicht werden.
- meine Note soll nicht veröffentlicht werden; ich erfahre sie dann aus Munopag (bzw. für nicht im Munopag geführte Studierende: beim Abholen des Scheins).

	Max. Punkte	Erreichte Punkte
Aufgabe 1 (Allgemeines)	11	
Aufgabe 2 (XML, XPath)	20	
Aufgabe 3 (XQuery)	22	
Aufgabe 4 (XSLT)	18	
Aufgabe 5 (Verschiedenes)	19	
Summe	90	

Note:

Name:

MatNr.:

Aufgabe 1 (Allgemeines [11 Punkte])

(alle Fragen können kurz und stichpunktartig beantwortet werden)

1. Welche "Fortschritte" bedeutete die "Erfindung" des relationalen Datenmodells, der relationalen Algebra und SQL gegenüber den vorherigen, auf dem Netzwerkmodell basierenden Datenbanken? (4 P)
2. Was bedeutet "Abgeschlossenheit" einer Anfragesprache? (3 P)
3. Welche Funktionalitäten und Grundlagen haben XML-Datenbanken und relationale (und alle anderen) Datenbanken gemeinsam? (4 P)

Name:

MatNr.:

Aufgabe 2 (XML, XPath [20 Punkte])

Gegeben ist das folgende XML-Fragment (gegenüber dem Original-Hamlet der Vorlesung leicht veränderte DTD):

```
<?xml version="1.0"?>
<!DOCTYPE PLAY [
  <!ELEMENT PLAY (TITLE, PERSON+, SCENE+)>
  <!ELEMENT TITLE (#PCDATA)>
  <!ELEMENT PERSON EMPTY>
    <!ATTLIST PERSON NAME ID #REQUIRED
      DESCRIPTION CDATA #IMPLIED>
  <!ELEMENT SCENE (SPEECH+)>
  <!ELEMENT SPEECH (LINE+)>
    <!ATTLIST SPEECH SPEAKER IDREF #REQUIRED>
  <!ELEMENT LINE ANY>
  <!ELEMENT LOUD (#PCDATA)>
]
<PLAY>
  <TITLE>The Tragedy of Hamlet, Prince of Denmark</TITLE>

  <PERSON NAME="Claudius" DESCRIPTION="King of Denmark"/>
  <PERSON NAME="Hamlet" DESCRIPTION="Nephew to the present king"/>
  <PERSON NAME="Polonius" DESCRIPTION="Lord chamberlain"/>
  <PERSON NAME="Ophelia" DESCRIPTION="Daughter to Polonius"/>
  :
  <SCENE>
    <SPEECH SPEAKER="Hamlet">
      <LINE><LOUD>To be, or not to be</LOUD>: that is the question:</LINE>
      <LINE>Whether 'tis nobler in the mind to suffer</LINE>
      :
      <LINE>The fair Ophelia! Nymph, in thy orisons</LINE>
      <LINE>Be all my sins remember'd.</LINE>
    </SPEECH>
    <SPEECH SPEAKER="Ophelia">
      <LINE><LOUD>Good my lord</LOUD>,</LINE>
      <LINE>How does your honour for this many a day?</LINE>
    </SPEECH>
    <SPEECH SPEAKER="Hamlet">
      <LINE>I humbly thank you; well, well, well.</LINE>
    </SPEECH>
    :
  </SCENE>
</PLAY>
```

1. Geben Sie einen XPath-Ausdruck an, der die Beschreibung der Person "Claudius" zurückgibt. (2 P)
2. Geben Sie einen XPath-Ausdruck an, der alle Textstellen, die <LOUD> gesprochen werden, zurückgibt. (1 P)
3. Geben Sie einen XPath-Ausdruck an, der die Menge der Namen aller Personen zurückgibt, die etwas <LOUD> sprechen. (3 P)
4. Betrachten Sie den XPath-Ausdruck

```
/descendant::SPEECH[contains(line/text(),"To be or not to be")]  
/following-sibling::SPEECH[1]/@SPEAKER
```

Geben Sie an, welche(n) Knoten der Ausdruck als Ergebnis hat (2 P).

5. Geben Sie einen XPath-Ausdruck an, der die Beschreibung der Person zurückgibt, die auf Hamlets Auftritt, in dem er "To be or not to be" sagt, antwortet. (3 P)
6. Erklären Sie anhand des Ausdrucks aus Teil (4.) die Begriffe *Achse*, *Vorwärts- und Rückwärtsachse* (Sie können den Begriff "Kontextknoten" ohne separate Erklärung voraussetzen; 6 P). Geben Sie für die letzteren jeweils zwei Beispiele an (je 1.5 P).

Name:

MatNr.:

Für die beiden folgenden Aufgaben (XQuery und XSLT) sei das folgende XML-Dokument gegeben:

```
<Munopag>
  <Dozent Name="Mueller"/>
  :
  <Student MatNo="S1" Name="Hans Schmidt"/>
  :
  <Veranstaltung Name="Informatik I">
    <Klausur Semester="WS2000" Dozent="Meier">
      <bestanden MatNo="S1">3.7</bestanden>
      <nichtbestanden MatNo="S2"/>
      <bestanden MatNo="S3">2.0</bestanden>
      <nichtbestanden MatNo="S4"/>
    </Klausur>
    <Klausur Semester="WS2001" Dozent="Mueller">
      <bestanden MatNo="S2"/>3.0</bestanden>
      <nichtbestanden MatNo="S4"/>
      <bestanden MatNo="S5">1.3</bestanden>
    </Klausur>
  :
</Veranstaltung>
<Veranstaltung Name="Informatik II">
  :
</Veranstaltung>
</Munopag>
```

Namen und Matrikelnummern seien jeweils in vernuenftiger Weise als ID und IDREF deklariert.

Name:

MatNr.:

Aufgabe 3 (XQuery [22 Punkte])

Geben Sie für die Anfragen (2)-(5) je eine XQuery-Anfrage an, die das Ergebnis in dem jeweils angegebenen Format ausgibt.

[Hinweis: bestandene Prüfungen können nicht wiederholt werden!]

1. Beschreiben Sie kurz, was XPath und XQuery miteinander zu tun haben, und was XQuery kann, was XPath nicht kann. (5 P)

2. Geben Sie alle Matrikelnummern von Studenten an, die "Informatik I" irgendwann bestanden haben (3 P):

```
<Student MatNo="matno" />
```

3. Geben Sie alle Paare von Matrikelnummern von Studenten bzw. Namen von Dozenten an, so dass der Student die "Informatik I" bei dem Dozenten bestanden hat (3 P):

```
<Ergebnis Student="matno" Dozent="doz-name" />
```

4. Geben Sie die Menge aller Paare (MatrNo eines Studenten, Name eines Dozenten) an, so dass der Student bei dem Dozenten in irgendeiner Prüfung nicht bestanden hat (Format beliebig, z.B. wie oben) (5 P).

5. Geben Sie für jeden Studenten ein Zeugnis in Form einer Liste der bestandenen Fächer mit Note aus (6 P):

```
<Zeugnis MatNo="matno">  
  <Fach Name="Veranstaltung1" Note="Note1" />  
  <Fach Name="Veranstaltung2" Note="Note2" />  
  :  
</Zeugnis>
```


Name:

MatNr.:

Aufgabe 4 (XSLT [18 Punkte])

Diese Aufgabe verwendet den Teil "Veranstaltungen" aus dem angegebenen Munopag-Beispiel.

1. Beschreiben Sie kurz, wie `<xsl:template match='pattern'>` und `<xsl:apply-templates select='xpath-ausdruck'>` zusammenarbeiten. Sie können hierzu das in der nächsten Teilaufgabe zu erstellende Beispiel verwenden. (5 P)
2. Geben Sie Templates für ein XSLT-Stylesheet an, das für eine gegebenes Klausur-Element (z.B. Informatik I in 2001) eine Notentabelle in XHTML wie folgt (wie in der Eingabe nach Matrikelnummern geordnet) ausdrückt (10 P):
 - Kopfzeile: Name der Veranstaltung sowie Semester,
 - Je eine Zeile für jeden Teilnehmer mit den Spalten "Matrikelnummer", "Note" (nicht bestanden wird mit 5.0 bewertet), sowie ein "*", falls der Teilnehmer nicht bestanden hat.

Der Aufruf der von Ihnen zu schreibenden Templates erfolgt z.B. durch

```
<xsl:template match="/">
  <xsl:apply-templates
    select="//Veranstaltung[@Name='Informatik I']
           /Klausur[@Semester='WS2001']"/>
</xsl:template>
```

3. Was müssen Sie ändern, wenn zuerst alle "bestandenen" Teilnehmer, und dann alle "nicht bestandenen" Teilnehmer aufgezählt werden sollen? (3 P)

Name:

MatNr.:

Aufgabe 5 (Verschiedenes [19 Punkte])

1. Beschreiben Sie kurz, wie eigene SimpleTypes in **XML Schema** aus existierenden Typen (z.B. `xsd:string` oder `xsd:float`) abgeleitet werden, und was man damit machen kann. Geben Sie ein Beispiel eines solchen Datentypen (mit XML Schema Quellcode). (5 P)
2. Gegeben sei die folgende Relation einer relationalen Datenbank:

Buch			
Titel	Verlag	ISBN	Preis
"XML"	"Springer"	0-815	50.00
"SQL"	"Hanser"	47-11	64.00

Geben Sie an, wie diese Tabelle *generisch* in eine XML-Repräsentation abgebildet werden kann (kurze textuelle Beschreibung der Grundideen Ihrer Abbildung, XML-Ergebnis; 6 P); geben Sie auch die DTD dazu an (4 P).

3. Angenommen, man hat einen **XLink**-fähigen Browser. Welches Verhalten zeigt eine Seite, die das folgende XHTML-Fragment enthält: (4 P)

```
<b>Die folgenden Teilgebiete werden in der Vorlesung behandelt</b>
<ul><li> ... </li>
  <li>
    <b xlink:type="simple"
      xlink:actuate="OnRequest"
      xlink:show="embed"
      xlink:href="http://ap34.inf.uni-goe.de/index.html/p[a/@name='XSLT']">
      XSLT</b>
  </li>
  :
</ul>
```