

Klausur “Semistructured Data and XML”
Summer Term 2025
Prof. Dr. Wolfgang May
6. August 2025, 13:30–16:30
Working Time: 120 Minutes
(carried out as a computer-based ILIAS exam)

Vorname:

Nachname:

Matrikelnummer:

Setting: The usage of saxon (with the aliases saxonValid, saxonXQ, and saxonXSL defined as in the course) and xmllint (validation error messages are better with xmllint) is recommended. Web access (e.g. for XPath/XQuery and XSLT documentation) is allowed. It was also recommended to have the slides, and a condensed self-prepared “cheat sheet” (preparation of a cheat sheet is a very effective way to work through the materials). Help by other persons or AI tools is NOT allowed.

Answers might be given in English or German (most answers are program code anyway).

In the text, the german translation is sometimes given in parentheses.

Give *all* answers via the ILIAS system.

Like in a “paper exam”, also solutions that do maybe not work (or do not work completely) can be delivered and will be graded with appropriate partial points.

For **passing** the exam, **50** points are sufficient.

	Max. Punkte	Schätzung für “4”
Aufgabe 1 (XML)	15	12
Aufgabe 2 (DTD)	12	10
Aufgabe 3 (XPath (a))	3	3
Aufgabe 4 (XPath/XQuery (b))	4	3
Aufgabe 5 (XQuery (c))	4	3
Aufgabe 6 (XPath/XQuery (d))	6	4
Aufgabe 7 (XQuery (e))	8	4
Aufgabe 8 (XPath/XQuery (f))	6	2
Aufgabe 9 (XPath/XQuery (g))	6	2
Aufgabe 10 (XQuery Update)	6	5
Aufgabe 11 (XSLT)	14	6
Aufgabe 12 (XML Consistency)	8	2
Aufgabe 13 (XQuery Update (abstract))	8	2
Summe	100	58

Note:

Project: Season Schedule for a Theater

The scenario is to manage the schedule for a theater for the upcoming season 2025/26 - the pieces, the performances for every day of the season, the stage actors and which roles they play.

1. The rooms where performances take place are the “Big Hall”, the “Small Hall”, the café and the foyer.

The Big Hall has 800 seats, and the Small Hall has 300 seats. In the café there are 100 seats, and in the foyer 200.

2. During the season, the theater performs a number of pieces, usually several times each. For every such piece, the title, the type (tragedy, comedy, opera, ...), the name of the composer(s) and writer(s), and the duration of the local staging is stored.

- The tragedy “Hamlet” by *William Shakespeare* (that was used in the lecture and exercises, see below) which has a duration of 3 hours,
- the opera “Don Carlos” by the composer *Giuseppe Verdi* (based on the play from 1787 by *Friedrich Schiller*) which has a duration of 4 hours,
- the tale “The Little Prince” by *Antoine de St. Exupéry* as a performance for children with a duration of 80 minutes,
- the opera “Orpheus in the Underworld” by *Hector Crémieux* and *Ludovic Halévy* (writers) and *Jacques Offenbach* (music) which has a duration of 90 minutes.

3. The text and stage directions of *some* plays is available in separate XML sources following the structure of the “Hamlet” play as known from the lecture, which is available at <https://www.stud.informatik.uni-goettingen.de/xml-lecture/Shakespeare/hamlet.xml>. (if you have it locally, you can also use `doc('hamlet.xml')` to access it.)

Design your main XML file in such a way that it allows to state queries that access the XML of the plays, in case that it is available.

4. Also the characters acting in the piece are listed, with their name (hint: care for compatibility with the above play XML source(s)), their gender (if applicable) and optionally a short comment:

- In “Hamlet”, the main characters are Claudius (King of Denmark), Hamlet (nephew of Claudius), Gertrude (Queen of Denmark, mother to Hamlet); the sample data also contains two minor characters: Reynaldo (servant to Polonius) and Lucianus (another nephew to Claudius).
- In “Don Carlos”, there are Philipp II (King of Spain), Elisabeth (Queen of Spain), Don Carlos (Philipp’s son and crown prince), Princess of Eboli (a Spanish aristocrat), and some more.
- In “the Little Prince”, the main characters are the pilot, the little prince, and the fox. The fox does not have a gender.
- In “Orpheus in the Underworld”, the main characters are Orpheus (a musician) and Eurydice (Orpheus’ wife).

5. There are the stage actors working at the theater. For every actor, the name, the gender, and the type of voice (soprano, alto, tenor, bass) is stored.
 - Alice is female and has a soprano voice.
 - Bob is male and has a tenor voice.
 - Carol is female and has an alto voice.
 - Dave is male and has a bass voice.
 - Eva is female and has also an alto voice.
 - Frank is male and has also a tenor voice.
 - Grace is female and has also a soprano voice.
 - Henry is male and has also a bass voice.
 - Michael and Nicolas are male, both are tenors.
 - and many more

6. The overall assignment for the whole season of stage actors to the roles is decided early (the actors need some time to learn their texts and to prepare) and stored. For each role, two persons are assigned:
 - “Hamlet”: Bob and Frank are assigned to prepare for the Hamlet role, Dave and Henry are assigned to the role of King Claudius; Carol and Eva are assigned to Queen Gertrude, Michael and Nicolas are each assigned to both (minor characters) Reynaldo and Lucianus.
 - “Don Carlos”: Michael and Bob are assigned to Don Carlos; Frank and Nicolas are assigned to Philipp, Grace and Eva are assigned to Princess Eboli.
 - “The Little Prince”: Eva and Grace are assigned to the Little Prince role, Dave and Michael are assigned to the Pilot role, and Nicolas and Henry are assigned to the Fox role.
 - “Orpheus in the Underworld”: Henry and Dave are assigned to the Orpheus role, Grace and Alice are assigned to the Eurydice role.

7. The theater schedule is fixed before the beginning of the season and lists for all performances the date, time, and the hall where the performance takes place. (Note that not every day there are performances; usually Friday to Sunday and some performances on weekdays).
 During the season, also the current number of the tickets sold for every performance is stored.
 Sample data:
 - The season starts on Friday, 3.10.2025, 20:00 with “Hamlet” in the *Big Hall*. Currently, there are already 543 tickets sold.
 - On Saturday, 4.10.2025 there is at 17:00 “The Little Prince” in the Big Hall for children. All 800 tickets are already booked by schools. A 20:30 “Orpheus in the Underworld” in the Big Hall. 400 tickets are sold.
 - On Sunday, 5.10.2025, there is at 20:00 “Don Carlos” in the Big Hall; 666 tickets are yet sold.

- On Saturday, 11.10.2025, there is at 17:00 again a performance of “The Little Prince”, and at 20:00 again a performance of “Hamlet”, both in the Big Hall. 300 resp. 250 tickets are yet sold.
8. For every performance, the stage actors are assigned to the roles some time in advance and are stored in the XML:
- The assignment for the “Hamlet” performance on 3.10.2025 is as follows: Hamlet: Bob; Claudius: Dave; Gertrude: Carol; and Reynaldo and Lucianus are both played (they are on stage in different acts) by Michael.
 - The assignment for the “Little Prince” performance on 4.10.2025 is: Little Prince: Eva; Pilot: Dave; Fox: Michael.
 - The assignment for the “Orpheus in the Underworld” performance on 4.10.2025 is: Orpheus: Dave; Eurydice: Alice, ...
 - The assignment for “Don Carlos” on 5.10.2025 is Don Carlos: Michael; Philipp: Frank; Princess Eboli: Eva.
 - The assignment for the “Little Prince” performance on 11.10.2025 is again Little Prince: Eva; Pilot: Dave; Fox: Michael.
 - The assignment for “Hamlet” on 11.10.2025 will be done in Exercise 10.

Design an XML structure for the above information. Use the frame given in the file `exam.xml` - it must contain significant enough data (at least 2 instances per element/attribute type).

Lösung

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE theater SYSTEM "exam.dtd">
<theater>
  <room id="Big" name="Big Hall" seats="800"/>
  <room id="Small" name="Small Hall" seats="300"/>
  <room id="Cafe" name="Cafe" seats="100"/>
  <room id="Foyer" name="Foyer" seats="200"/>
  <piece id="Ham" title="Hamlet" type="tragedy" duration="PT3H"
    url="https://www.stud.informatik.uni-goettingen.de/xml-lecture/Shakespeare/hamlet.xml"
    <!-- other syntaxes for duration are also ok, we don't need to calculate -->
    <writer>William Shakespeare</writer>
    <character id="cl" gender="m" comment="King of Denmark"
      assigned="Dave Henry">KING CLAUDIUS</character>
    <!-- preferably use capitalization to be compatible with the plays' XMLs -->
    <character id="ham" gender="m" comment="nephew to Claudius"
      assigned="Bob Frank">HAMLET</character>
    <character id="ger" gender="f" assigned="Carol Eva">QUEEN GERTRUDE</character>
    <character id="rey" gender="m" assigned="Michael Nicolas">REYNALDO</character>
    <character id="luc" gender="m" assigned="Michael Nicolas">LUCIANUS</character>
  </piece>
  <piece id="Don" title="Don Carlos" type="opera" duration="PT4H">
    <composer>Giuseppe Verdi</composer>
    <writer>Friedrich Schiller</writer>
    <character id="ph" gender="m" comment="King of Spain"
      assigned="Dave Nicolas">PHILIPP</character>
    <character id="dc" gender="m" comment="Prince"
      assigned="Michael Bob">CARLOS</character>
    <character id="eb" gender="f" assigned="Grace Eva">PRINCESS EBOLI</character>
  </piece>
  <piece id="LP" title="The Little Prince" type="tale" duration="PT90M">
    <writer>Antoine de St. Exupery</writer>
    <character id="pi" gender="m" assigned="Dave Michael">PILOT</character>
    <character id="lp" gender="m" assigned="Eva Grace">LITTLE PRINCE</character>
    <character id="fox" assigned="Nicolas Henry">FOX</character>
  </piece>
  <piece id="Orp" title="Orpheus in the Underworld" type="opera"
    duration="PT80M">
    <composer>Jacques Offenbach</composer>
    <writer>Hector Cremieux</writer>
    <writer>Ludovic Halevy</writer>
    <character id="orp" gender="m" comment="a musician"
      assigned="Henry Dave">ORPHEUS</character>
    <character id="eur" gender="f" comment="Orpheus' wife"
      assigned="Grace Alice">EURYDICE</character>
  </piece>
  <piece id="Alis" title="Alice's Adventures in Wonderland" type="tale" duration="PT120M">
    <writer>Lewis Carroll</writer>
    <character id="ali" gender="f" assigned="Alice">ALICE</character>
    <character id="wh" gender="m" assigned="White Rabbit">WHITE RABBIT</character>
    <character id="ch" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kb" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qs" gender="m" assigned="Queen of Spades">QUEEN OF SPADES</character>
    <character id="qr" gender="m" assigned="Queen of Red Hearts">QUEEN OF RED HEARTS</character>
    <character id="chc" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc2" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh2" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh2" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc3" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh3" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh3" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
  </piece>
  <piece id="Alice" title="Alice's Adventures in Wonderland" type="tale" duration="PT120M">
    <writer>Lewis Carroll</writer>
    <character id="ali" gender="f" assigned="Alice">ALICE</character>
    <character id="wh" gender="m" assigned="White Rabbit">WHITE RABBIT</character>
    <character id="ch" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kb" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qs" gender="m" assigned="Queen of Spades">QUEEN OF SPADES</character>
    <character id="qr" gender="m" assigned="Queen of Red Hearts">QUEEN OF RED HEARTS</character>
    <character id="chc" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc2" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh2" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh2" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc3" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh3" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh3" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
  </piece>
  <piece id="Alice2" title="Alice's Adventures in Wonderland" type="tale" duration="PT120M">
    <writer>Lewis Carroll</writer>
    <character id="ali" gender="f" assigned="Alice">ALICE</character>
    <character id="wh" gender="m" assigned="White Rabbit">WHITE RABBIT</character>
    <character id="ch" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kb" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qs" gender="m" assigned="Queen of Spades">QUEEN OF SPADES</character>
    <character id="qr" gender="m" assigned="Queen of Red Hearts">QUEEN OF RED HEARTS</character>
    <character id="chc" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc2" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh2" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh2" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc3" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh3" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh3" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
  </piece>
  <piece id="Alice3" title="Alice's Adventures in Wonderland" type="tale" duration="PT120M">
    <writer>Lewis Carroll</writer>
    <character id="ali" gender="f" assigned="Alice">ALICE</character>
    <character id="wh" gender="m" assigned="White Rabbit">WHITE RABBIT</character>
    <character id="ch" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kb" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qs" gender="m" assigned="Queen of Spades">QUEEN OF SPADES</character>
    <character id="qr" gender="m" assigned="Queen of Red Hearts">QUEEN OF RED HEARTS</character>
    <character id="chc" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc2" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh2" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh2" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc3" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh3" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh3" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
  </piece>
  <piece id="Alice4" title="Alice's Adventures in Wonderland" type="tale" duration="PT120M">
    <writer>Lewis Carroll</writer>
    <character id="ali" gender="f" assigned="Alice">ALICE</character>
    <character id="wh" gender="m" assigned="White Rabbit">WHITE RABBIT</character>
    <character id="ch" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kb" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qs" gender="m" assigned="Queen of Spades">QUEEN OF SPADES</character>
    <character id="qr" gender="m" assigned="Queen of Red Hearts">QUEEN OF RED HEARTS</character>
    <character id="chc" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc2" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh2" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh2" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc3" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh3" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh3" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
  </piece>
  <piece id="Alice5" title="Alice's Adventures in Wonderland" type="tale" duration="PT120M">
    <writer>Lewis Carroll</writer>
    <character id="ali" gender="f" assigned="Alice">ALICE</character>
    <character id="wh" gender="m" assigned="White Rabbit">WHITE RABBIT</character>
    <character id="ch" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kb" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qs" gender="m" assigned="Queen of Spades">QUEEN OF SPADES</character>
    <character id="qr" gender="m" assigned="Queen of Red Hearts">QUEEN OF RED HEARTS</character>
    <character id="chc" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh" gender="m" assigned="Queen of Hearts">QUEEN OF HEARTS</character>
    <character id="chc2" gender="m" assigned="Cheshire Cat">CHESHIRE CAT</character>
    <character id="kbh2" gender="m" assigned="King of Hearts">KING OF HEARTS</character>
    <character id="qsh2" gender="m" assigned="
```

```

    <character id="lp" gender="m" assigned="Eva Grace">LITTLE PRINCE</character>
    <character id="fox" assigned="Nicolas Henry">FOX</character>
</piece>
<piece id="Orp" title="Orpheus in the Underworld" type="opera"
    duration="PT80M">
    <composer>Jacques Offenbach</composer>
    <writer>Hector Cremieux</writer>
    <writer>Ludovic Halevy</writer>
    <character id="orp" gender="m" comment="a musician"
        assigned="Henry Dave">ORPHEUS</character>
    <character id="eur" gender="f" comment="Orpheus' wife"
        assigned="Grace Alice">EURYDICE</character>
</piece>
<actor name="Alice" gender="f" voice="soprano"/>
<actor name="Bob" gender="m" voice="tenor"/>
<actor name="Carol" gender="f" voice="alto"/>
<actor name="Dave" gender="m" voice="bass"/>
<actor name="Eva" gender="f" voice="alto"/>
<actor name="Frank" gender="m" voice="tenor"/>
<actor name="Grace" gender="f" voice="soprano"/>
<actor name="Henry" gender="m" voice="bass"/>
<actor name="Michael" gender="m" voice="tenor"/>
<actor name="Nicolas" gender="m" voice="tenor"/>
<performance date="2025-10-03" time="20:00" piece="Ham" room="Big" tickets="543">
    <staged role="ham" actor="Bob"/>
    <staged role="cl" actor="Dave"/>
    <staged role="cl" actor="Bob"/> <!-- for debugging Q4, does not change other answers
    <staged role="ger" actor="Carol"/>
    <staged role="rey" actor="Michael"/>
    <staged role="luc" actor="Michael"/>
</performance>
<performance date="2025-10-04" time="17:00" piece="LP" room="Big" tickets="800">
    <staged role="pi" actor="Dave"/>
    <staged role="fox" actor="Michael"/>
    <staged role="lp" actor="Eva"/>
</performance>
<performance date="2025-10-04" time="20:30" piece="Orp" room="Big" tickets="400">
    <staged role="orp" actor="Dave"/>
    <staged role="eur" actor="Alice"/>
</performance>
<performance date="2025-10-05" time="20:00" piece="Don" room="Big" tickets="666">
    <staged role="dc" actor="Michael"/>
    <staged role="ph" actor="Frank"/>
    <staged role="eb" actor="Eva"/>
</performance>
<performance date="2025-10-11" time="17:00" piece="LP" room="Big" tickets="300">
    <staged role="pi" actor="Dave"/>
    <staged role="fox" actor="Michael"/>
    <staged role="lp" actor="Eva"/>
</performance>
<performance date="2025-10-11" time="20:00" piece="Ham" room="Big" tickets="250">
</performance>
</theater>

```

Note that here, also XInclude can be used for the PLAY text:

```
<piece id="Ham" title="Hamlet" type="tragedy" duration="PT3H">
  <stagetext>
    <xi:include href="hamlet.xml"/>
  </stagetext>
  :
</piece>
```

Add it to the DTD (only the xi:include needed, not the PLAY part of the DTD):

```
<!ELEMENT stagetext (xi:include)>
<!ELEMENT xi:include EMPTY>
<!--LIST xi:include href CDATA #REQUIRED
           parse CDATA #IMPLIED-->
```

Then run saxon with the -xi option

```
saxonXQ -xi -s:exam.xml query5.xq
```

and use path expressions like

```
piece[@title="Hamlet"]/stagetext//SPEECH[SPEAKER="KING CLAUDIUS"]
```

to navigate inside it.

Exercise 2 (DTD [12 Points])

Give the DTD for your document developed in Exercise 1, use the file `exam.dtd`.
Use one of the calls

```
xmllint -load dtd -valid --noblanks -noout exam.xml
saxonValid.bat -s:exam.xml
```

for validating it (note that xmllint provides better error messages).
Upload or copy-and-paste the DTD from the file `exam.dtd` afterwards here:

Lösung

```

<!ELEMENT theater (room+, piece*, actor*, performance*)>
<!ELEMENT room EMPTY>
<!ATTLIST room id ID #REQUIRED
              name CDATA #REQUIRED
              seats CDATA #REQUIRED>
<!ELEMENT piece ((composer|writer)+, character*)>
<!ATTLIST piece id ID #REQUIRED
              title CDATA #REQUIRED
              type CDATA #REQUIRED
              duration CDATA #REQUIRED
              url CDATA #IMPLIED>
<!ELEMENT composer (#PCDATA)>
<!ELEMENT writer (#PCDATA)>
<!ELEMENT character (#PCDATA)>
<!ATTLIST character id ID #REQUIRED
                  gender CDATA #IMPLIED
                  comment CDATA #IMPLIED
                  assigned IDREFS #IMPLIED>
<!ELEMENT actor EMPTY>
<!ATTLIST actor name ID #REQUIRED
            gender CDATA #REQUIRED
            voice CDATA #REQUIRED>
<!ELEMENT performance (staged*)>
<!ATTLIST performance date CDATA #REQUIRED
                    time CDATA #IMPLIED
                    piece IDREF #REQUIRED
                    room IDREF #REQUIRED
                    tickets CDATA #IMPLIED>
<!ELEMENT staged EMPTY>
<!ATTLIST staged role IDREF #REQUIRED
              actor IDREF #REQUIRED>

```

Exercise 3 (XPath (a) [3 Points])

Use your `exam.xml` XML file as a basis for solving this and the following exercises. None of the results should contain duplicates.

Give an XPath query that returns the *names* of those characters where at least one actor is staged for who has a bass voice.

Write the query string in the file `query1.xq` and call it with

```
saxonXQ.bat -s:exam.xml query1.xq
```

Copy-and-paste the query from `query1.xq` afterwards here:

Lösung

```

//staged[id(@actor)/@voice="bass"]/id(@role)/text()
(: Claudius, Pilot, Orpheus :)

```

Exercise 4 (XPath/XQuery (b) [4 Points])

Give an XPath query or an XQuery query that returns the *names* of those actors who are not staged in any performance on the weekend from 3.10.2025 until 5.10.2025.

Write the query string in the file query2.xq and call it with

```
saxonXQ.bat -s:exam.xml query2.xq
```

Copy-and-paste the query from query2.xq afterwards here:

Lösung

```
//actor[not (@name =  
  //performance[@date>='2025-10-03' and @date<='2025-10-05']  
    /staged/@actor)]  
  /@name/string()  
(: Grace, Henry, Michael :)
```

```
for $a in //actor  
where not (//performance[@date>='2025-10-03' and @date<='2025-10-05'  
    and ./staged/@actor=$a/@name])  
return $a/@name/string()
```

Exercise 5 (XQuery (c) [4 Points])

Give an XQuery query that for each case, where in some performance an actor plays a role of a character that has a different gender, returns the date of the performance, the title of the piece, the name of the actor, and the role in some reasonable XML markup. Copy-and-paste the query from query3.xq afterwards here:

Lösung

```
for $s in //staged[id(@role)/@gender != id(@actor)/@gender]  
return  
<result date='{ $s/../../@date}' piece='{ $s/../../id(@piece)/@title}'  
actor='{ $s/@actor}' role='{ $s/id(@role)/string()}' />  
(: eva, little prince, 4.10., and eva, little prince, 11.10. :)  
(: fox, if no gender is seen as different from male :)
```

Exercise 6 (XPath/XQuery (d) [6 Points])

Give an XQuery query that outputs for each piece the total number of tickets sold for it during this season, the number of performances that have been given, and the number of performances where all tickets have been sold.

Copy-and-paste the query from query4.xq afterwards here:

Lösung

```
for $p in //piece  
return <result piece="{ $p/@title}"  
  sum="{sum(//performance[@piece=$p/@id]/@tickets)}"  
  num="{count(//performance[@piece=$p/@id])}"  
  soldout=  
    "{count(//performance[@piece=$p/@id and @tickets=id(@room)/@seats])}" />
```

Exercise 7 (XQuery (e) [8 Points])

Give an XQuery query that checks that no collisions in the casting of a performance occur: a stage actor must not be casted for two roles that occur on the stage (here: have

speeches) in the same scene (other multi-occurrences are allowed - the actor must then change his costume in-between). Concretely, output all actors and pairs of roles they are casted for where there would be a co-occurrence in some scene in the form

```
<conflict piece="..." date="..." actor="..." role1="..." role2="..."
          act="..." scene="..."/> .
```

Obviously (and hopefully), the result is empty for the sample data. Describe how you can validate whether your query would correctly detect such cases (1P).

Copy-and-paste the query from query5.xq afterwards here:

Lösung

```
for $p in //performance[id(@piece)/@url]
for $s1 in $p/staged,
    $s2 in $p/staged[@actor = $s1/@actor and @role < $s1/@role]
let $conflicts :=
    doc($p/id(@piece)/@url)//SCENE[SPEECH[SPEAKER=$s1/id(@role)/text()]
    and SPEECH[SPEAKER=$s2/id(@role)/text()]]
where $conflicts
return
    for $c in $conflicts
    return
        <conflict piece="{ $p/id(@piece)/@title}" date="{ $p/@date}"
            name="{ $s1/@actor}" role1="{ $s1/id(@role)}" role2="{ $s2/id(@role)}"
            act="{ $c/./TITLE}" scene="{ $c/TITLE}" />
```

Check: just add a casting to the first performance of Hamlet where the Hamlet actor also plays King Claudius. Then, there must be conflicts detected.

Exercise 8 (XPath/XQuery (f) [6 Points])

Give an XQuery query that returns the names of those stage actors that are casted for every day in October where the theater shows some performance.

Copy-and-paste the query from query6.xq afterwards here:

Lösung

```
for $a in //actor
where every
    $date in //performance[@date>='2025-10-01' and @date<='2025-10-31']/@date
satisfies
    //performance[@date=$date and staged[@actor=$a/@name]]
return $a/@name/string()
(: Michael :)
```

```

let $octDates :=
  distinct-values(//performance[@date>='2025-10-01'
                                and @date<='2025-10-31']/@date)
for $a in //actor
where count(distinct-values(
  //performance[@date>='2025-10-01' and @date<='2025-10-31' and
    staged[@actor=$a/@name]]/@date)) =
  count($octDates)
return $a/@name/string()
(: Michael :)

```

Exercise 9 (XPath/XQuery (g) [6 Points])

Give an XQuery query that does the following: for a given date (e.g., the 2025-10-04, where two performances are given in the sample data) outputs an HTML printout that lists the performances of that day with the actual staging of the characters for this day. So the visitors can look up who are the actors that they see in the performance.

Copy-and-paste the query from query7.xq afterwards here:

Lösung

```

declare variable $date external;    (: or use current-date() :)
<html><body>
  { for $p in //performance[@date=$date]
    return ( <h1> {$p/@time/string()} {$p/id(@room)/@name/string()} :
      {$p/id(@piece)/@title/string()} </h1>,
      for $s in $p/staged
        return ($s/id(@role)/text(), ":", $s/@actor/string(), <br/>) )
  }
</body></html>
(: saxonXQ -s:exam.xml date=2025-10-04 query7.xq :)

```

Exercise 10 (XQuery Update [6 Points])

For the “Hamlet” performance on 11.10.2025, the staging of each role should be the actor who is listed to be prepared for this role (cf. item 6 in the project description), but did *not* play it in the “Hamlet” performance on 3.10.2025.

Give an XQuery Update statement that adds the appropriate XML contents to your file. (not necessary to execute, recall that updates can only be executed if you have the Saxon Enterprise Edition installed)

Lösung

```

delete nodes //performance[date="2025-10-11" and
                             id(@piece)/@title="Hamlet"]/staged
(: to be on the safe side in case something was already stored there :)
insert nodes
  for $s in //performance[date="2025-10-03" and id(@piece)="Hamlet"]/staged
  return
    <staged role="{ $s/@role }"
      actor="{ //character[@id=$s/@id]/id(@assigned)[@id != $s/@actor]/@id }"/>
into //performance[date="2025-10-11" and id(@piece)="Hamlet"]

```

Exercise 11 (XSLT [14 Points])

Extend the given XSLT stylesheet frame `exam.xsl` to an XSLT stylesheet that generates HTML cheatsheets for the actors: Given a play (it should be usable for the general case - although we have the play text only for Hamlet), it should do the following for each character:

For each speech of the respective character, output (i) who is the last speaker before, and the last line of his speech, and (ii) then, the speech. If there is no other speech directly before, e.g., at the beginning of a scene, output “at begin of” and the SCENE’s title instead.

Use the following call to execute it:

```
saxonXSL.bat -s:exam.xml exam.xsl play=Hamlet
```

Copy-and-paste the XSLT stylesheet from `exam.xsl` afterwards here:

Lösung

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="2.0">
  <xsl:param name="play"/>
  <xsl:template match="/">
    <xsl:apply-templates select="//piece[@title=$play]/character"/>
  </xsl:template>
  <xsl:template match="character">
    <html><body>
      <h1> Cheatsheet for character: <xsl:value-of select="."/> </h1>
      <xsl:variable name="url" select="//piece[@title=$play]/@url"/>
      <xsl:apply-templates select="doc($url)//SPEECH[SPEAKER=current()]" />
    </body></html>
  </xsl:template>
  <xsl:template match="SPEECH">
    <xsl:choose>
      <xsl:when test="preceding-sibling::SPEECH[1]">
        after <xsl:value-of select="preceding-sibling::SPEECH[1]/SPEAKER/text()"/> :
        <xsl:value-of select="preceding-sibling::SPEECH[1]/LINE[last()]/text()"/>
        <br/>
      </xsl:when>
      <xsl:otherwise>
        at begin of <xsl:value-of select="preceding-sibling::TITLE/text()"/> :
      </xsl:otherwise>
    </xsl:choose>
    ME as <xsl:value-of select="SPEAKER"/> :
    <xsl:for-each select='LINE'>
      <xsl:value-of select="."/> <br/>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Exercise 12 (XML Consistency [8 Points])

Your XML designed in Exercise 1 probably does not guarantee that in the staging only actors can be casted for a role where they are also assigned to prepare for at the beginning of the season (see item 6 in the project description). (If your XML guarantees this already, you have already successfully solved this exercise).

Sketch (text, with XML and DTD fragments) how this could be guaranteed by a refined XML+DTD design.

Lösung Replace the character/@assigned lists by pairs that have an ID, and refer to these IDs for the actual casting:

```
<character id="ham" ...>
  <assigned id="ham-bob" actor="bob"/>
  <assigned id="ham-frank" actor="frank"/>
</character>
<staged ref="ham-bob"/>
DTD:
<!ELEMENT character (assigned*)>
<!ELEMENT assigned EMPTY>
<!ATTLIST assigned ID id #REQUIRED
                  actor IDREF #REQUIRED>
<!ELEMENT staged EMPTY>
<!ATTLIST staged ref IDREF #REQUIRED>
```

- Note that this cannot prevent to add the assignment of Eva as Little Prince (id "lp-eva") to the casting of a Hamlet performance.
- Note that this is an example of *reification* in XML to create a "thing" from a relationship to be able to refer to this thing.
- Exercise: how would this be done in a relational database using foreign keys?

Exercise 13 (XQuery Update (abstract) [8 Points])

Consider an XQuery Update statement with a rather simple for-clause like

```
for $e in some-xpath-expression
return (
  insert node <bla/> after $e/xxx,
  delete node $e/@yyy
)
```

where *some-xpath-expression* is formed only with */*, *//*, and simple filters of the form *[@attr=value]*, and where every such *e*-element has at most one *xxx* subelement. Give an XSLT stylesheet pattern that executes the same updates on the XML tree.

Lösung

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="2.0">
  <xsl:template match="*|node()">
    <xsl:copy>
      <xsl:apply-templates select = "@*|node()">
        <!-- all attrs and children, including text nodes -->
      </xsl:apply-templates>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="some-xpath-expression">
    <xsl:copy>
      <xsl:apply-templates select = "@*[name()!='yyy']">
      <xsl:apply-templates select = "node()">
    </xsl:apply-templates>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="some-xpath-expression/xxx">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
    <bla/>
  </xsl:template>
</xsl:stylesheet>

```

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="2.0">
  <xsl:template match="*|@*">
    <xsl:copy>
      <xsl:apply-templates select = "@*|node()">
        <!-- all children, including text nodes -->
      </xsl:apply-templates>
    </xsl:copy>
  </xsl:template>
  <xsl:template match="some-xpath-expression/@yyy">
    <!-- nothing. delete it. -->
  </xsl:template>
  <xsl:template match="some-xpath-expression/xxx">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
    <bla/>
  </xsl:template>
</xsl:stylesheet>

```

The following frames can be used:

- Frame for XML file exam.xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE put-name-here SYSTEM "exam.dtd">
  to be extended here

```

- Frame for XML stylesheet `exam.xsl`:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="2.0">
  to be extended here
</xsl:stylesheet>
```