Suggested solutions XT1 - XT5 (lesson 3)

EXERCISES ON XSLT

The following exercises shall be solved with XSLT based on the movie database (the file movies.xml). The database is described in chapter 10 and is available in iLearn. Try different solutions in order to compare different ways to achieve similar results! Try solutions compatible with XSLT 1, 2 or 3!

The solutions can be executed using a web browser only if they are in XSLT 1. They can be executed with BaseX (XSLT version depends on the version of Saxon in the library of BaseX). They can also be executed on the websites xsltransform.net (XSLT 2) and xslttest.appspot.com (XSLT 3). Depending on the execution method, it is important to adjust the way the XSLT and the XML are linked. Either by having a processing-instruction in the XML file pointing to the XSLT file (when using a web browser) or dynamically/automatically (when using the two websites). In BaseX, use the following XQuery statement (with the appropriate filenames):

```
let $xsl := "D:\SDXML\XSLT\file.xsl"
let $xml := "D:\SDXML\XSLT\file.xml"
return xslt:transform($xml, $xsl)
```

If the result of the XSLT is not going to be XML, use the function xslt:transform-text instead.

In BaseX it is possible to open a database instead of specifying an XML file. It is also possible to write the XSLT code in the editor instead of retrieving it from a file.

When using a web browser add the following processing-instruction to the file movies.xml:

```
<?xml-stylesheet type="text/xsl" href="file.xsl"?>
```

Alternatively, create a dummy XML file with just a root element and the processing-instruction above and then let the XSLT open the XML file movies.xml with the function document(), perhaps putting the result in a variable:

```
<xsl:variable name="doc" select="document('movies.xml')" />
```

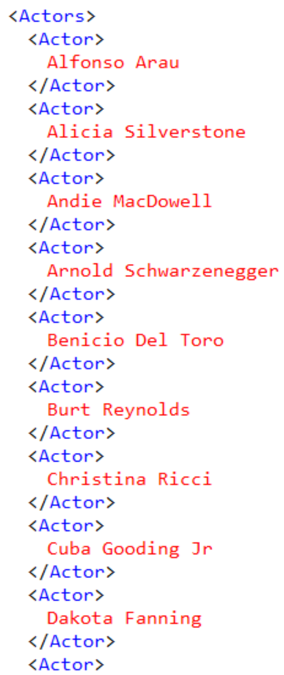
EXERCISE XT1

Desired result: An XML document with every actor as an element.

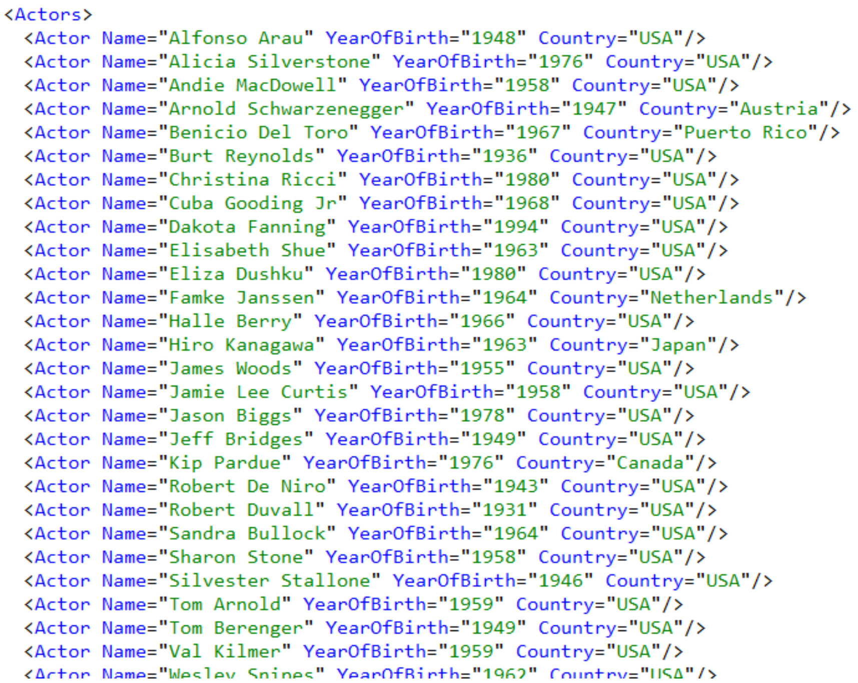
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <xsl:variable name="doc" select="document('movies.xml')" />
    <Actors>
      <xsl:for-each select="$doc//Actor[not(@Name = preceding::Actor/@Name)]">
        <xsl:sort select="@Name" order="ascending" />
        <xsl:element name="Actor"><xsl:value-of select="@Name"/></xsl:element>
      </xsl:for-each>
    </Actors>
  </xsl:template>
</xsl:transform>
```

If we want to keep the entire Actor element with its original structure instead of only putting the name as a text node, we can use <xsl:copy-of select="."/> instead of the xsl:element that creates Actor.

Result:



```
<Actors>
  <Actor>
    Alfonso Arau
  </Actor>
  <Actor>
    Alicia Silverstone
  </Actor>
  <Actor>
    Andie MacDowell
  </Actor>
  <Actor>
    Arnold Schwarzenegger
  </Actor>
  <Actor>
    Benicio Del Toro
  </Actor>
  <Actor>
    Burt Reynolds
  </Actor>
  <Actor>
    Christina Ricci
  </Actor>
  <Actor>
    Cuba Gooding Jr
  </Actor>
  <Actor>
    Dakota Fanning
  </Actor>
  <Actor>
```



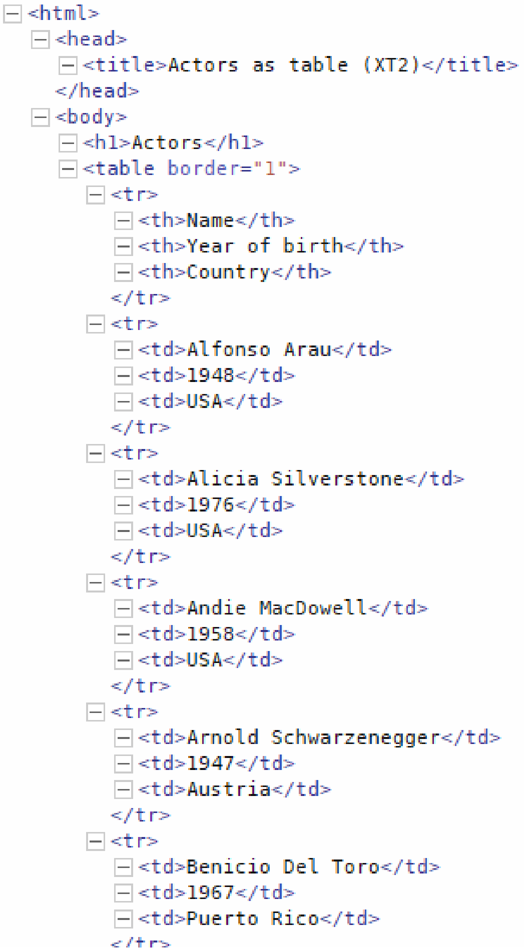
```
<Actors>
  <Actor Name="Alfonso Arau" YearOfBirth="1948" Country="USA"/>
  <Actor Name="Alicia Silverstone" YearOfBirth="1976" Country="USA"/>
  <Actor Name="Andie MacDowell" YearOfBirth="1958" Country="USA"/>
  <Actor Name="Arnold Schwarzenegger" YearOfBirth="1947" Country="Austria"/>
  <Actor Name="Benicio Del Toro" YearOfBirth="1967" Country="Puerto Rico"/>
  <Actor Name="Burt Reynolds" YearOfBirth="1936" Country="USA"/>
  <Actor Name="Christina Ricci" YearOfBirth="1980" Country="USA"/>
  <Actor Name="Cuba Gooding Jr" YearOfBirth="1968" Country="USA"/>
  <Actor Name="Dakota Fanning" YearOfBirth="1994" Country="USA"/>
  <Actor Name="Elisabeth Shue" YearOfBirth="1963" Country="USA"/>
  <Actor Name="Eliza Dushku" YearOfBirth="1980" Country="USA"/>
  <Actor Name="Famke Janssen" YearOfBirth="1964" Country="Netherlands"/>
  <Actor Name="Halle Berry" YearOfBirth="1966" Country="USA"/>
  <Actor Name="Hiro Kanagawa" YearOfBirth="1963" Country="Japan"/>
  <Actor Name="James Woods" YearOfBirth="1955" Country="USA"/>
  <Actor Name="Jamie Lee Curtis" YearOfBirth="1958" Country="USA"/>
  <Actor Name="Jason Biggs" YearOfBirth="1978" Country="USA"/>
  <Actor Name="Jeff Bridges" YearOfBirth="1949" Country="USA"/>
  <Actor Name="Kip Pardue" YearOfBirth="1976" Country="Canada"/>
  <Actor Name="Robert De Niro" YearOfBirth="1943" Country="USA"/>
  <Actor Name="Robert Duvall" YearOfBirth="1931" Country="USA"/>
  <Actor Name="Sandra Bullock" YearOfBirth="1964" Country="USA"/>
  <Actor Name="Sharon Stone" YearOfBirth="1958" Country="USA"/>
  <Actor Name="Silvester Stallone" YearOfBirth="1946" Country="USA"/>
  <Actor Name="Tom Arnold" YearOfBirth="1959" Country="USA"/>
  <Actor Name="Tom Berenger" YearOfBirth="1949" Country="USA"/>
  <Actor Name="Val Kilmer" YearOfBirth="1959" Country="USA"/>
  <Actor Name="Wesley Snipes" YearOfBirth="1962" Country="USA"/>
```

EXERCISE XT2

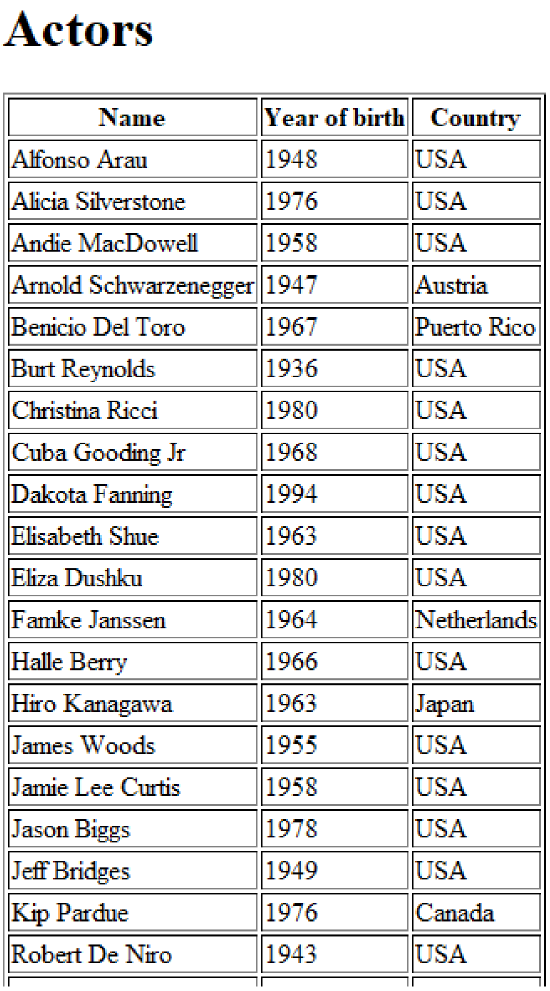
Desired result: An HTML document with all the actors as an HTML table.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html"/>
    <xsl:template match="/">
        <xsl:variable name="doc" select="document('movies.xml')" />
        <html>
            <head><title>Actors as table (XT2)</title></head>
            <body>
                <h1>Actors</h1>
                <table border="1">
                    <tr>
                        <th>Name</th>
                        <th>Year of birth</th>
                        <th>Country</th>
                    </tr>
                    <xsl:for-each select="$doc//Actor[not(@Name = preceding::Actor/@Name)]">
                        <xsl:sort select="@Name" order="ascending" />
                        <xsl:element name="tr">
                            <xsl:element name="td"><xsl:value-of select="@Name"/></xsl:element>
                            <xsl:element name="td"><xsl:value-of select="@YearOfBirth"/></xsl:element>
                            <xsl:element name="td"><xsl:value-of select="@Country"/></xsl:element>
                        </xsl:element>
                    </xsl:for-each>
                </table>
            </body>
        </html>
    </xsl:template>
</xsl:transform>
```

Result:



```
- <html>
  - <head>
    - <title>Actors as table (XT2)</title>
    </head>
  - <body>
    - <h1>Actors</h1>
    - <table border="1">
      - <tr>
        - <th>Name</th>
        - <th>Year of birth</th>
        - <th>Country</th>
        </tr>
      - <tr>
        - <td>Alfonso Arau</td>
        - <td>1948</td>
        - <td>USA</td>
        </tr>
      - <tr>
        - <td>Alicia Silverstone</td>
        - <td>1976</td>
        - <td>USA</td>
        </tr>
      - <tr>
        - <td>Andie MacDowell</td>
        - <td>1958</td>
        - <td>USA</td>
        </tr>
      - <tr>
        - <td>Arnold Schwarzenegger</td>
        - <td>1947</td>
        - <td>Austria</td>
        </tr>
      - <tr>
        - <td>Benicio Del Toro</td>
        - <td>1967</td>
        - <td>Puerto Rico</td>
        </tr>
```



Actors

Name	Year of birth	Country
Alfonso Arau	1948	USA
Alicia Silverstone	1976	USA
Andie MacDowell	1958	USA
Arnold Schwarzenegger	1947	Austria
Benicio Del Toro	1967	Puerto Rico
Burt Reynolds	1936	USA
Christina Ricci	1980	USA
Cuba Gooding Jr	1968	USA
Dakota Fanning	1994	USA
Elisabeth Shue	1963	USA
Eliza Dushku	1980	USA
Famke Janssen	1964	Netherlands
Halle Berry	1966	USA
Hiro Kanagawa	1963	Japan
James Woods	1955	USA
Jamie Lee Curtis	1958	USA
Jason Biggs	1978	USA
Jeff Bridges	1949	USA
Kip Pardue	1976	Canada
Robert De Niro	1943	USA

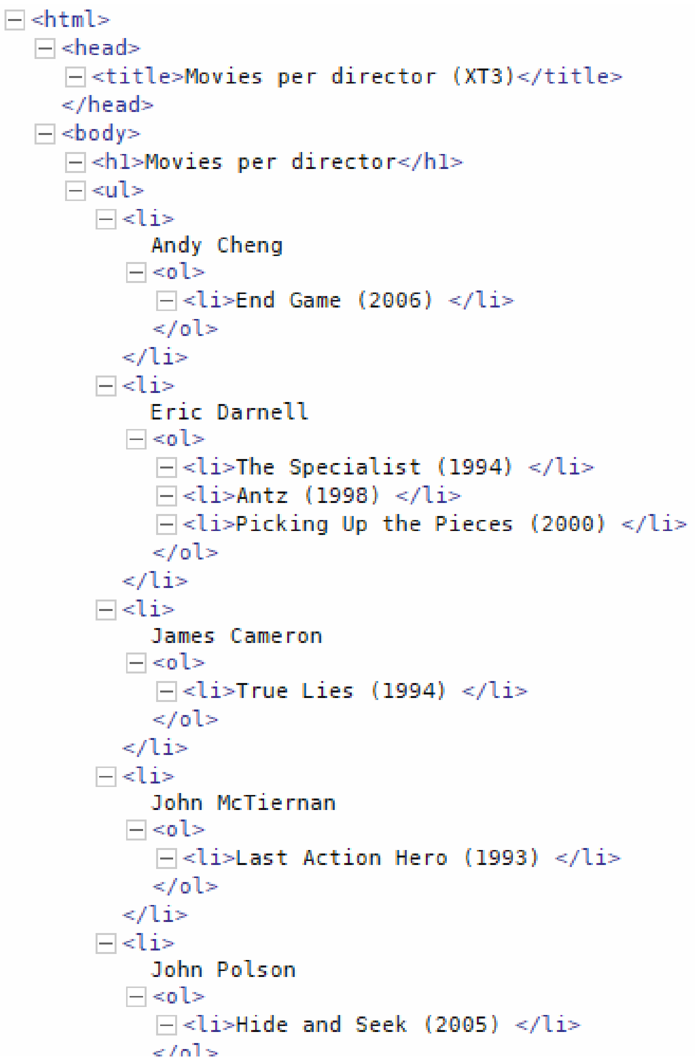
EXERCISE XT3

Desired result: An HTML document with the movies listed per director (nested lists).

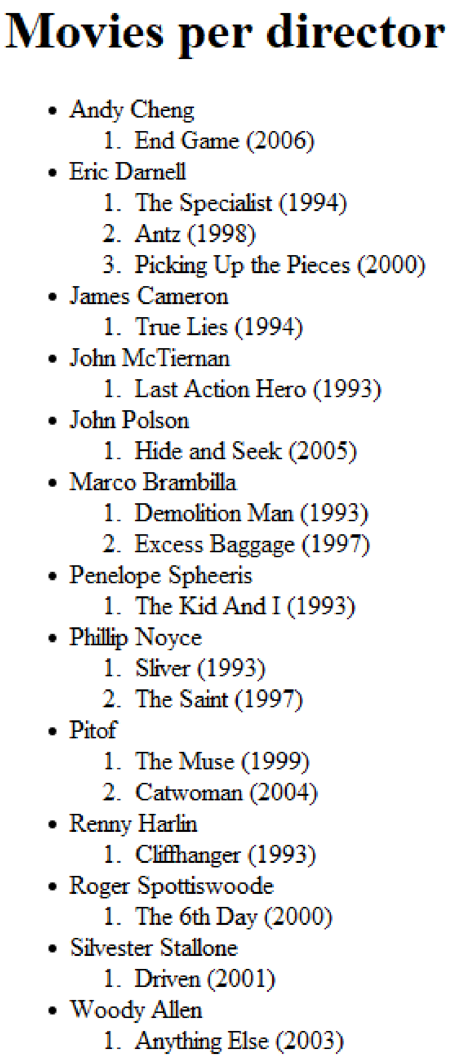
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output method="html"/>
   <xsl:variable name="doc" select="document('movies.xml')" />
   <xsl:template match="/">
      <html>
         <head><title>Movies per director (XT3)</title></head>
         <body>
            <h1>Movies per director</h1>
            <ul>
               <xsl:apply-templates select="$doc//Director[not(@Name = preceding::Director/@Name)]">
                  <xsl:sort select="@Name"/>
               </xsl:apply-templates>
            </ul>
         </body>
      </html>
   </xsl:template>
```

```
    <xsl:template match="Director">
        <li>
            <xsl:value-of select="@Name"/>
            <ol>
                <xsl:apply-templates select="$doc//Movie[Director/@Name = current()/@Name]">
                    <xsl:sort select="@Year"/>
                </xsl:apply-templates>
            </ol>
        </li>
    </xsl:template>
    <xsl:template match="Movie">
        <li>
            <xsl:value-of select="@Title"/> (<xsl:value-of select="@Year"/>)
        </li>
    </xsl:template>
</xsl:transform>
```

Result:



```
<html>
  <head>
    <title>Movies per director (XT3)</title>
  </head>
  <body>
    <h1>Movies per director</h1>
    <ul>
      <li>
        Andy Cheng
        <ol>
          <li>End Game (2006) </li>
        </ol>
      </li>
      <li>
        Eric Darnell
        <ol>
          <li>The Specialist (1994) </li>
          <li>Antz (1998) </li>
          <li>Picking Up the Pieces (2000) </li>
        </ol>
      </li>
      <li>
        James Cameron
        <ol>
          <li>True Lies (1994) </li>
        </ol>
      </li>
      <li>
        John McTiernan
        <ol>
          <li>Last Action Hero (1993) </li>
        </ol>
      </li>
      <li>
        John Polson
        <ol>
          <li>Hide and Seek (2005) </li>
        </ol>
```



Movies per director

- Andy Cheng
  1. End Game (2006)
- Eric Darnell
  1. The Specialist (1994)
  2. Antz (1998)
  3. Picking Up the Pieces (2000)
- James Cameron
  1. True Lies (1994)
- John McTiernan
  1. Last Action Hero (1993)
- John Polson
  1. Hide and Seek (2005)
- Marco Brambilla
  1. Demolition Man (1993)
  2. Excess Baggage (1997)
- Penelope Spheeris
  1. The Kid And I (1993)
- Phillip Noyce
  1. Sliver (1993)
  2. The Saint (1997)
- Pitof
  1. The Muse (1999)
  2. Catwoman (2004)
- Renny Harlin
  1. Cliffhanger (1993)
- Roger Spottiswoode
  1. The 6th Day (2000)
- Silvester Stallone
  1. Driven (2001)
- Woody Allen
  1. Anything Else (2003)

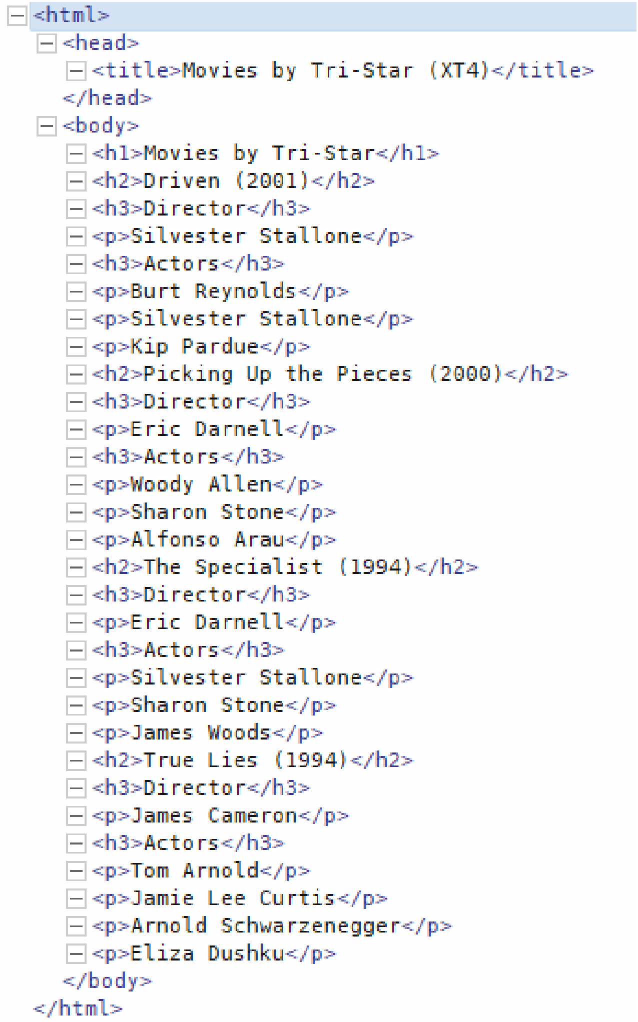
EXERCISE XT4

Desired result: An HTML document with all the movies produced by Tri-Star. Include information about each movie's director and actors.

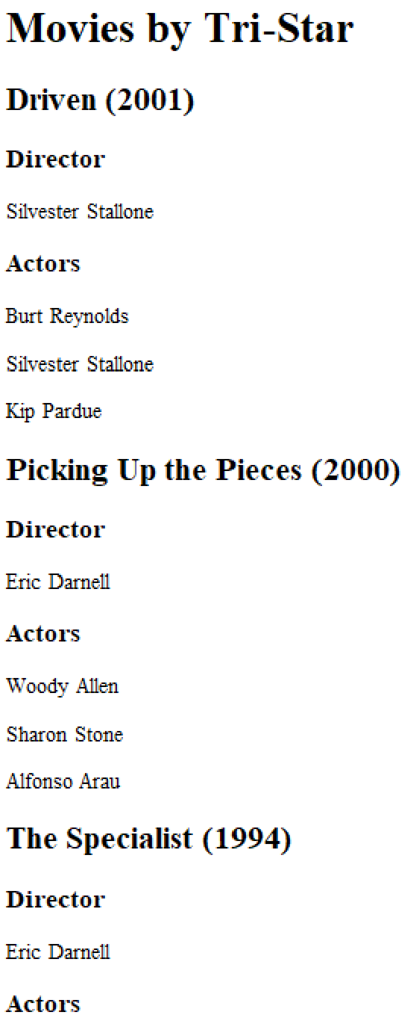
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html"/>
    <xsl:variable name="doc" select="document('movies.xml')" />
    <xsl:template match="/">
        <html>
            <head><title>Movies by Tri-Star (XT4)</title></head>
            <body>
                <h1>Movies by Tri-Star</h1>
                <xsl:apply-templates select="$doc//Movie[ProductionCompany = 'Tri-Star']" />
            </body>
        </html>
    </xsl:template>
    <xsl:template match="Movie">
        <h2><xsl:value-of select="@Title"/> (<xsl:value-of select="@Year"/>)</h2>
        <h3>Director</h3>
        <p><xsl:value-of select="Director/@Name"/></p>
        <h3>Actors</h3>
        <xsl:for-each select="Actor">
            <p><xsl:value-of select="@Name"/></p>
        </xsl:for-each>
    </xsl:template>
</xsl:transform>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="3.0">
    <xsl:output method="html"/>
    <xsl:template match="/">
        <html>
            <head><title>Movies by Tri-Star (XT4 with XSLT 3)</title></head>
            <body>
                <h1>Movies by Tri-Star</h1>
                <xsl:apply-templates select="//Movie[ProductionCompany = 'Tri-Star']" />
            </body>
        </html>
    </xsl:template>
    <xsl:template match="Movie" expand-text="yes">
        <h2>{@Title} ({@Year})</h2>
        <h3>Director</h3>
        <p>{Director/@Name}</p>
        <h3>Actors</h3>
        <xsl:for-each select="Actor">
            <p>{@Name}</p>
        </xsl:for-each>
    </xsl:template>
</xsl:transform>
```

Result:



```
<html>
  <head>
    <title>Movies by Tri-Star (XT4)</title>
  </head>
  <body>
    <h1>Movies by Tri-Star</h1>
    <h2>Driven (2001)</h2>
    <h3>Director</h3>
    <p>Silvester Stallone</p>
    <h3>Actors</h3>
    <p>Burt Reynolds</p>
    <p>Silvester Stallone</p>
    <p>Kip Pardue</p>
    <h2>Picking Up the Pieces (2000)</h2>
    <h3>Director</h3>
    <p>Eric Darnell</p>
    <h3>Actors</h3>
    <p>Woody Allen</p>
    <p>Sharon Stone</p>
    <p>Alfonso Arau</p>
    <h2>The Specialist (1994)</h2>
    <h3>Director</h3>
    <p>Eric Darnell</p>
    <h3>Actors</h3>
    <p>Silvester Stallone</p>
    <p>Sharon Stone</p>
    <p>James Woods</p>
    <h2>True Lies (1994)</h2>
    <h3>Director</h3>
    <p>James Cameron</p>
    <h3>Actors</h3>
    <p>Tom Arnold</p>
    <p>Jamie Lee Curtis</p>
    <p>Arnold Schwarzenegger</p>
    <p>Eliza Dushku</p>
  </body>
</html>
```



Movies by Tri-Star

Driven (2001)

Director

Silvester Stallone

Actors

Burt Reynolds

Silvester Stallone

Kip Pardue

Picking Up the Pieces (2000)

Director

Eric Darnell

Actors

Woody Allen

Sharon Stone

Alfonso Arau

The Specialist (1994)

Director

Eric Darnell

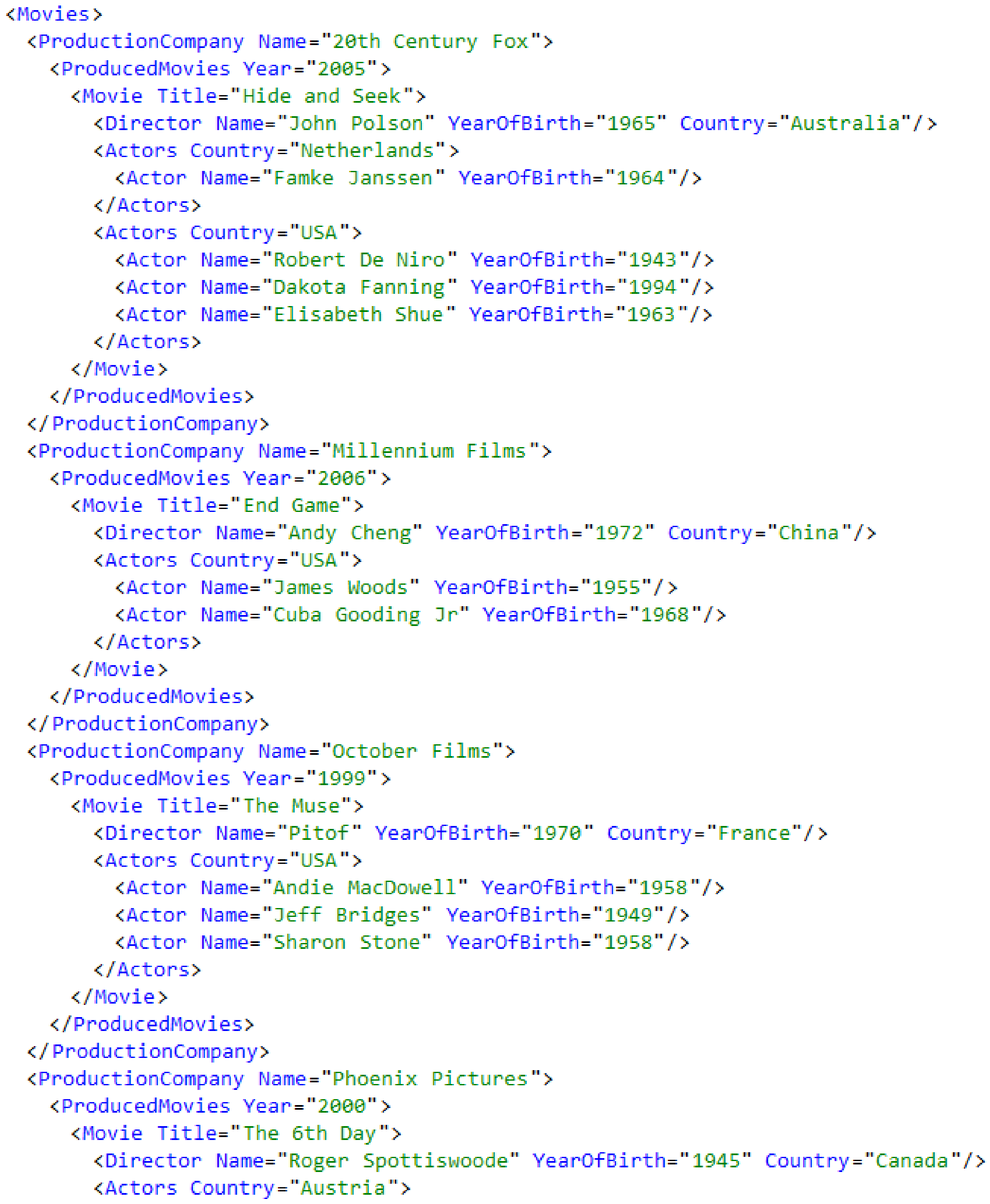
Actors

EXERCISE XT5

Desired result: An XML document with all the movies grouped by production company and year. Include information about each movie's director and the actors grouped by country.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="xml"/>
    <xsl:template match="/">
        <xsl:variable name="doc" select="document('movies.xml')"/>
        <Movies>
            <xsl:for-each select="$doc//ProductionCompany[not(text() =
preceding::ProductionCompany/text())]">
                <xsl:sort select="." order="ascending"/>
                <xsl:variable name="pc" select="."/>
                <ProductionCompany Name="{text()}">
                    <xsl:for-each select="$doc//Movie[ProductionCompany = $pc and not(@Year =
preceding::Movie[ProductionCompany = $pc]/@Year)]/@Year">
                        <ProducedMovies Year="{.}">
                            <xsl:apply-templates select="$doc//Movie[ProductionCompany = $pc and
@Year = current()]"/>
                        </ProducedMovies>
                    </xsl:for-each>
                </ProductionCompany>
            </xsl:for-each>
        </Movies>
    </xsl:template>
    <xsl:template match="Movie">
        <xsl:variable name="m" select="."/>
        <Movie>
            <xsl:copy-of select="@Title"/>
            <xsl:copy-of select="Director"/>
            <xsl:for-each select="Actor[not(@Country = preceding-
sibling::Actor/@Country)]/@Country">
                <xsl:sort select="." order="ascending"/>
                <Actors Country="{.}">
                    <xsl:apply-templates select="$m/Actor[@Country = current()]"/>
                </Actors>
            </xsl:for-each>
        </Movie>
    </xsl:template>
    <xsl:template match="Actor">
        <Actor>
            <xsl:copy-of select="@Name"/>
            <xsl:copy-of select="@YearOfBirth"/>
        </Actor>
    </xsl:template>
</xsl:transform>
```

Result:



```
<Movies>
  <ProductionCompany Name="20th Century Fox">
    <ProducedMovies Year="2005">
      <Movie Title="Hide and Seek">
        <Director Name="John Polson" YearOfBirth="1965" Country="Australia"/>
        <Actors Country="Netherlands">
          <Actor Name="Famke Janssen" YearOfBirth="1964"/>
        </Actors>
        <Actors Country="USA">
          <Actor Name="Robert De Niro" YearOfBirth="1943"/>
          <Actor Name="Dakota Fanning" YearOfBirth="1994"/>
          <Actor Name="Elisabeth Shue" YearOfBirth="1963"/>
        </Actors>
      </Movie>
    </ProducedMovies>
  </ProductionCompany>
  <ProductionCompany Name="Millennium Films">
    <ProducedMovies Year="2006">
      <Movie Title="End Game">
        <Director Name="Andy Cheng" YearOfBirth="1972" Country="China"/>
        <Actors Country="USA">
          <Actor Name="James Woods" YearOfBirth="1955"/>
          <Actor Name="Cuba Gooding Jr" YearOfBirth="1968"/>
        </Actors>
      </Movie>
    </ProducedMovies>
  </ProductionCompany>
  <ProductionCompany Name="October Films">
    <ProducedMovies Year="1999">
      <Movie Title="The Muse">
        <Director Name="Pitof" YearOfBirth="1970" Country="France"/>
        <Actors Country="USA">
          <Actor Name="Andie MacDowell" YearOfBirth="1958"/>
          <Actor Name="Jeff Bridges" YearOfBirth="1949"/>
          <Actor Name="Sharon Stone" YearOfBirth="1958"/>
        </Actors>
      </Movie>
    </ProducedMovies>
  </ProductionCompany>
  <ProductionCompany Name="Phoenix Pictures">
    <ProducedMovies Year="2000">
      <Movie Title="The 6th Day">
        <Director Name="Roger Spottiswoode" YearOfBirth="1945" Country="Canada"/>
        <Actors Country="Austria">
```

The previous solution got too complicated because of the limitations of XSLT 1. If we have access to for-each-group (XSLT 2) we can get a solution like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output method="xml" indent="yes"/>
    <xsl:template match="/">
        <Movies>
            <xsl:for-each-group select="//Movie" group-by="ProductionCompany">
                <xsl:sort select="current-grouping-key()" order="ascending"/>
                <ProductionCompany Name="{current-grouping-key()}">
                    <xsl:for-each-group select="current-group()" group-by="@Year">
                        <xsl:sort select="current-grouping-key()" order="ascending"/>
                        <ProducedMovies Year="{current-grouping-key()}">
                            <xsl:apply-templates select="current-group()"/>
                        </ProducedMovies>
                    </xsl:for-each-group>
                </ProductionCompany>
            </xsl:for-each-group>
        </Movies>
    </xsl:template>
    <xsl:template match="Movie">
        <Movie>
            <xsl:copy-of select="@Title"/>
            <xsl:copy-of select="Director"/>
            <xsl:for-each-group select="Actor" group-by="@Country">
                <xsl:sort select="current-grouping-key()" order="ascending"/>
                <Actors Country="{current-grouping-key()}">
                    <xsl:apply-templates select="current-group()"/>
                </Actors>
            </xsl:for-each-group>
        </Movie>
    </xsl:template>
    <xsl:template match="Actor">
        <Actor>
            <xsl:copy-of select="@Name"/>
            <xsl:copy-of select="@YearOfBirth"/>
        </Actor>
    </xsl:template>
</xsl:transform>
```