SDXML

Assignment 2

GROUP 9

XSLT

Group members.

Achyth Jagini

Ashan Lakshith Thotawatta

Kalana Sachith Mahaarachchi

Summary:

Use DB : Movie.XML
Software : Chrome Browser

Question 01

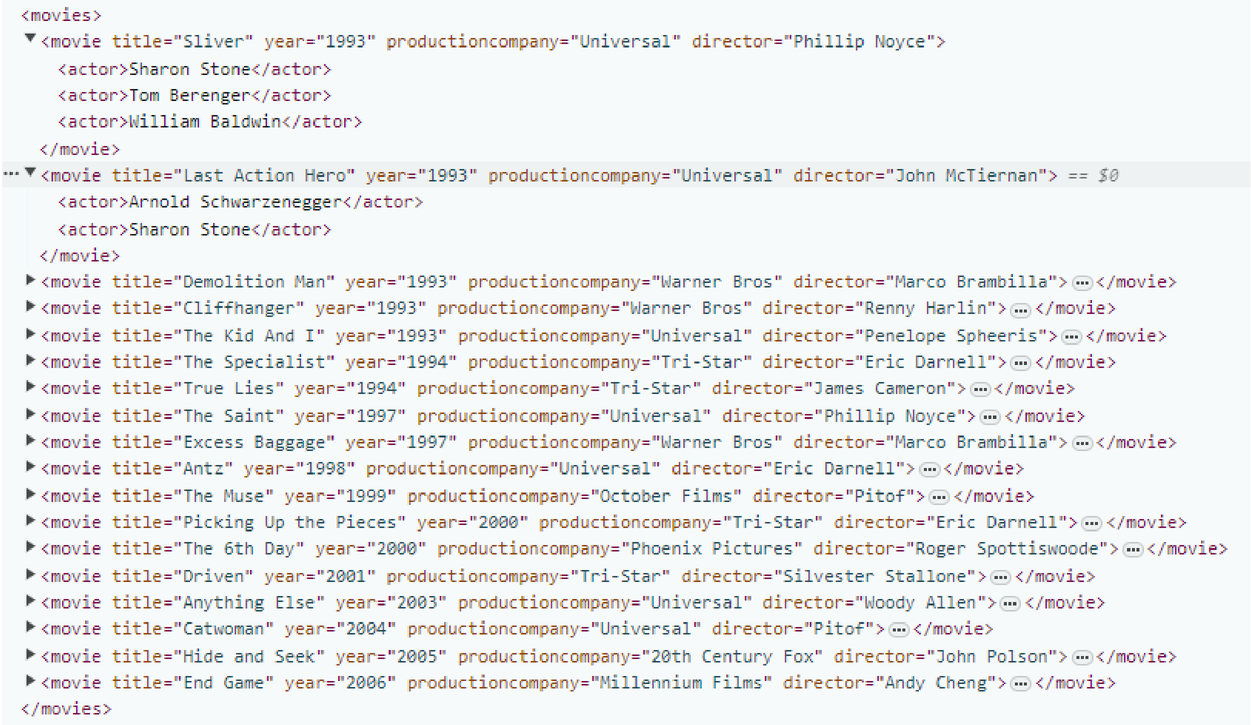
Create a new XML structure with all the movies sorted by year! The actors shall be sorted alphabetically. The result shall have the following structure:

Solution in XSLT 1.0

```
<?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:apply-templates select="$doc//Movie">
        <xsl:sort select="@Year" order="ascending"/>
      </xsl:apply-templates>
    </movies>
</xsl:template>

  <xsl:template match="Movie">
    <xsl:variable name="movie_name" select="@Title"/>
    <movie>
      <xsl:attribute name="title">
        <xsl:value-of select="@Title"/>
      </xsl:attribute>

      <xsl:attribute name="year">
        <xsl:value-of select="@Year"/>
      </xsl:attribute>

      <xsl:attribute name="productioncompany">
        <xsl:value-of select="ProductionCompany/text()"/>
      </xsl:attribute>

      <xsl:attribute name="director">
```

```
        <xsl:value-of select="Director/@Name"/>
      </xsl:attribute>

      <xsl:apply-templates select="//Movie[@Title=$movie_name]/Actor">
        <xsl:sort select="@Name" order="ascending"/>
      </xsl:apply-templates>
    </movie>
  </xsl:template>

  <xsl:template match="Actor">
    <actor>
      <xsl:value-of select="@Name"/>
    </actor>
  </xsl:template>
</xsl:transform>
```

Result

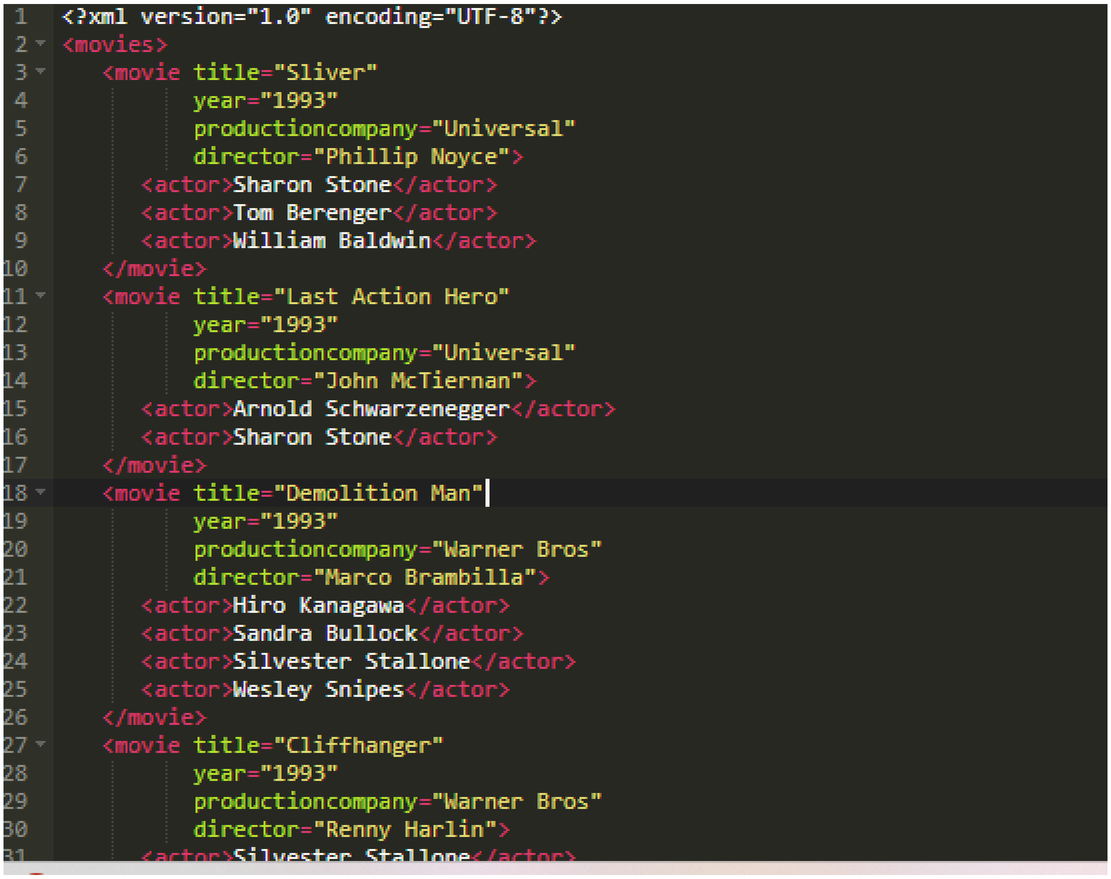


```
<movies>
 ▼<movie title="Sliver" year="1993" productioncompany="Universal" director="Phillip Noyce">
    <actor>Sharon Stone</actor>
    <actor>Tom Berenger</actor>
    <actor>William Baldwin</actor>
  </movie>
⋯▼<movie title="Last Action Hero" year="1993" productioncompany="Universal" director="John McTiernan"> == $0
    <actor>Arnold Schwarzenegger</actor>
    <actor>Sharon Stone</actor>
  </movie>
 ▶<movie title="Demolition Man" year="1993" productioncompany="Warner Bros" director="Marco Brambilla">⋯</movie>
 ▶<movie title="Cliffhanger" year="1993" productioncompany="Warner Bros" director="Renny Harlin">⋯</movie>
 ▶<movie title="The Kid And I" year="1993" productioncompany="Universal" director="Penelope Spheeris">⋯</movie>
 ▶<movie title="The Specialist" year="1994" productioncompany="Tri-Star" director="Eric Darnell">⋯</movie>
 ▶<movie title="True Lies" year="1994" productioncompany="Tri-Star" director="James Cameron">⋯</movie>
 ▶<movie title="The Saint" year="1997" productioncompany="Universal" director="Phillip Noyce">⋯</movie>
 ▶<movie title="Excess Baggage" year="1997" productioncompany="Warner Bros" director="Marco Brambilla">⋯</movie>
 ▶<movie title="Antz" year="1998" productioncompany="Universal" director="Eric Darnell">⋯</movie>
 ▶<movie title="The Muse" year="1999" productioncompany="October Films" director="Pitof">⋯</movie>
 ▶<movie title="Picking Up the Pieces" year="2000" productioncompany="Tri-Star" director="Eric Darnell">⋯</movie>
 ▶<movie title="The 6th Day" year="2000" productioncompany="Phoenix Pictures" director="Roger Spottiswoode">⋯</movie>
 ▶<movie title="Driven" year="2001" productioncompany="Tri-Star" director="Silvester Stallone">⋯</movie>
 ▶<movie title="Anything Else" year="2003" productioncompany="Universal" director="Woody Allen">⋯</movie>
 ▶<movie title="Catwoman" year="2004" productioncompany="Universal" director="Pitof">⋯</movie>
 ▶<movie title="Hide and Seek" year="2005" productioncompany="20th Century Fox" director="John Polson">⋯</movie>
 ▶<movie title="End Game" year="2006" productioncompany="Millennium Films" director="Andy Cheng">⋯</movie>
</movies>
```

Solution in XSLT 2.0

```
<?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:apply-templates select="//Movie">
        <xsl:sort select="@Year" order="ascending"/>
      </xsl:apply-templates>
    </movies>
  </xsl:template>

  <xsl:template match="Movie">
    <xsl:variable name="movie_name" select="@Title"/>
    <movie>
      <xsl:attribute name="title" select="@Title"/>

      <xsl:attribute name="year" select="@Year"/>

      <xsl:attribute name="productioncompany" select="ProductionCompany/text()"/>

      <xsl:attribute name="director" select="Director/@Name"/>

      <xsl:apply-templates select="//Movie[@Title=$movie_name]/Actor">
        <xsl:sort select="@Name" order="ascending"/>
      </xsl:apply-templates>
    </movie>
  </xsl:template>

  <xsl:template match="Actor">
    <actor>
      <xsl:value-of select="@Name"/>
    </actor>
  </xsl:template>
</xsl:transform>
```

Result



```
<?xml version="1.0" encoding="UTF-8"?>
<movies>
   <movie title="Sliver"
          year="1993"
          productioncompany="Universal"
          director="Phillip Noyce">
      <actor>Sharon Stone</actor>
      <actor>Tom Berenger</actor>
      <actor>William Baldwin</actor>
   </movie>
   <movie title="Last Action Hero"
          year="1993"
          productioncompany="Universal"
          director="John McTiernan">
      <actor>Arnold Schwarzenegger</actor>
      <actor>Sharon Stone</actor>
   </movie>
   <movie title="Demolition Man"
          year="1993"
          productioncompany="Warner Bros"
          director="Marco Brambilla">
      <actor>Hiro Kanagawa</actor>
      <actor>Sandra Bullock</actor>
      <actor>Silvester Stallone</actor>
      <actor>Wesley Snipes</actor>
   </movie>
   <movie title="Cliffhanger"
          year="1993"
          productioncompany="Warner Bros"
          director="Renny Harlin">
      <actor>Silvester Stallone</actor>
```

Question 02

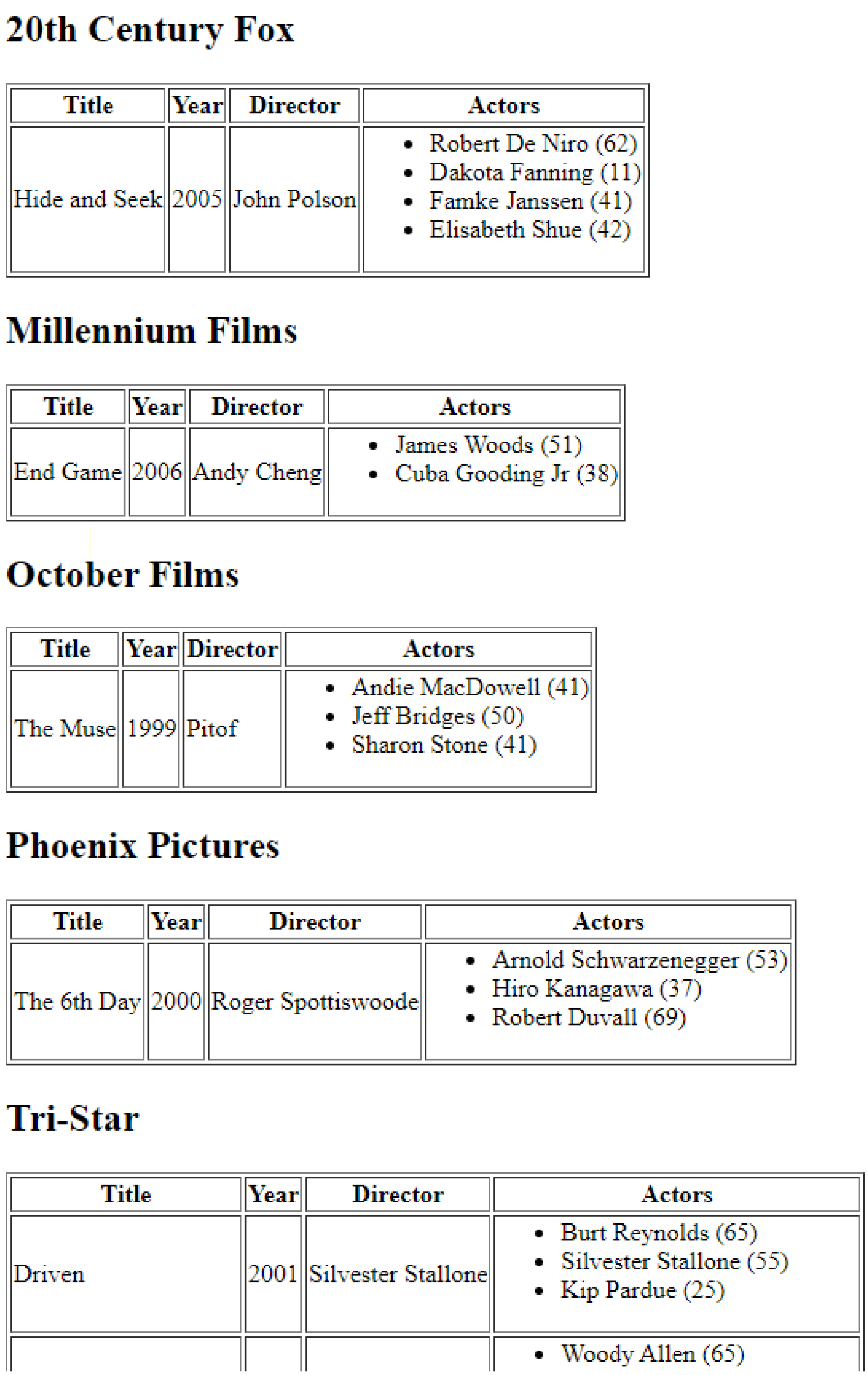
Show all the movies grouped by production company as HTML! One heading per company followed by a table with the corresponding movies. The table shall have one row with headings and one row per movie, with the following information included: Title, Year, Director's name, Actors (name and age at that time)

Solution in XSLT 1.0

```
<?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>Movies by production company</title></head>
      <body>
        <h2>Movies by production company</h2>

        <xsl:apply-templates select="$doc//ProductionCompany[not(text()=preceding::text())]">
          <xsl:sort select="text()" order="ascending"/>
        </xsl:apply-templates>

      </body>
    </html>
        </xsl:template>

  <xsl:template match="ProductionCompany">
    <xsl:variable name="prod_company_name" select="text()"/>
    <h2><xsl:value-of select="text()"/></h2>

    <table border="1">
      <tr>
        <th>Title</th>
        <th>Year</th>
        <th>Director</th>
        <th>Actors</th>
      </tr>

      <xsl:apply-templates select="//Movie[ProductionCompany=$prod_company_name]">
      </xsl:apply-templates>
    </table>

  </xsl:template>

  <xsl:template match="Movie">
    <xsl:variable name="movie_year" select="@Year"/>
    <tr>
      <td><xsl:value-of select="@Title"/></td>
      <td><xsl:value-of select="@Year"/></td>
      <td><xsl:value-of select="Director/@Name"/></td>
      <td>
        <ul>
```

```
          <xsl:apply-templates select="current()/Actor">
            <xsl:with-param name="movie_year" select="$movie_year"/>
          </xsl:apply-templates>
        </ul>
      </td>
    </tr>
  </xsl:template>

  <xsl:template match="Actor">
    <xsl:param name="movie_year"/>
    <li>
      <xsl:value-of select="@Name"/>
      (<xsl:value-of select="$movie_year - @YearOfBirth"/>)
    </li>
  </xsl:template>

</xsl:transform>
```

Result



20th Century Fox

Title	Year	Director	Actors
Hide and Seek	2005	John Polson	• Robert De Niro (62) • Dakota Fanning (11) • Famke Janssen (41) • Elisabeth Shue (42)

Millennium Films

Title	Year	Director	Actors
End Game	2006	Andy Cheng	• James Woods (51) • Cuba Gooding Jr (38)

October Films

Title	Year	Director	Actors
The Muse	1999	Pitof	• Andie MacDowell (41) • Jeff Bridges (50) • Sharon Stone (41)

Phoenix Pictures

Title	Year	Director	Actors
The 6th Day	2000	Roger Spottiswoode	• Arnold Schwarzenegger (53) • Hiro Kanagawa (37) • Robert Duvall (69)

Tri-Star

Title	Year	Director	Actors
Driven	2001	Silvester Stallone	• Burt Reynolds (65) • Silvester Stallone (55) • Kip Pardue (25)
			• Woody Allen (65)

SOLUTION IN XSLT 2.0

```
<xsl:transform version="2.0" 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>Movies by production company> </title></head>
<body>
<h2> Movies by production company </h2>

<xsl:for-each-group select="//Movie" group-by="ProductionCompany"
>
<xsl:sort select="current-grouping-key()></h2>
```

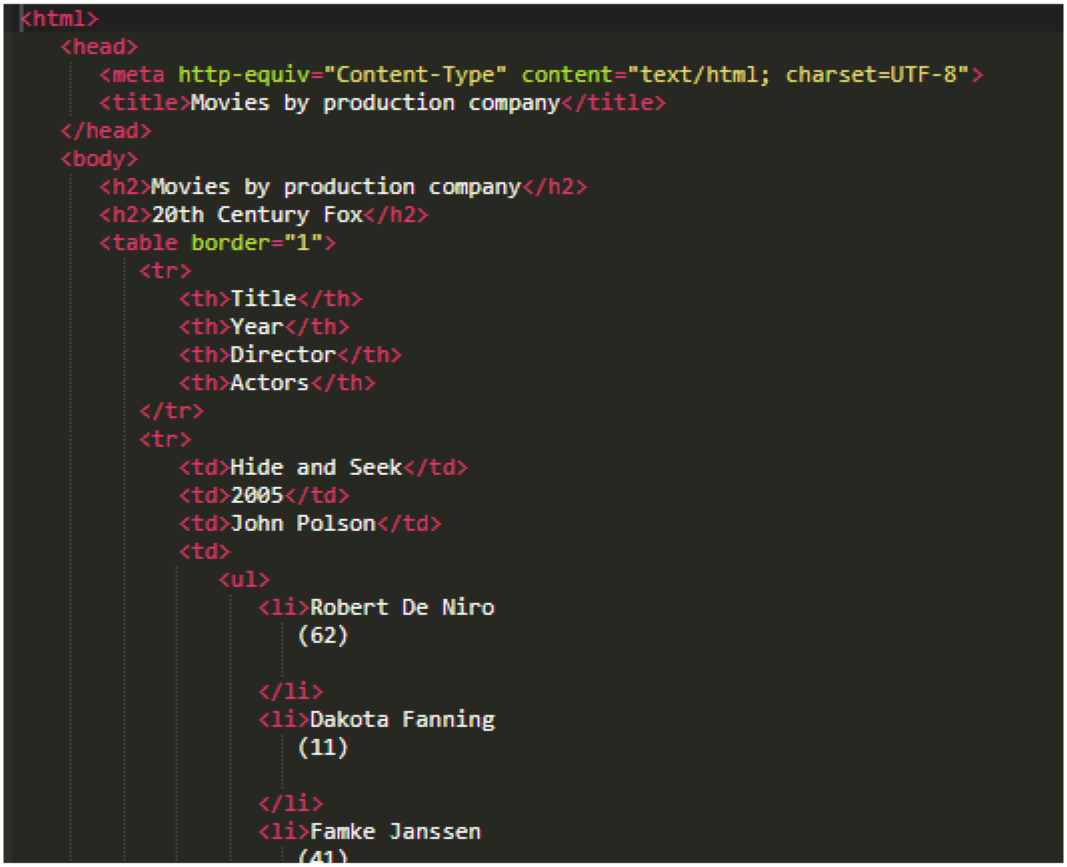
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
        <xsl:template match="/">
    <xsl:variable name="doc" select="document('../movies.xml')"/>
    <html>
      <head><title>Movies by production company</title></head>
      <body>
        <h2>Movies by production company</h2>
        <xsl:for-each-group select="//Movie" group-by="ProductionCompany">
          <xsl:sort select="current-grouping-key()" order="ascending"/>

          <h2><xsl:value-of select="current-grouping-key()"/></h2>
          <table border="1">
            <tr>
              <th>Title</th>
              <th>Year</th>
              <th>Director</th>
              <th>Actors</th>
            </tr>
            <xsl:apply-templates select="current-group()">
            </xsl:apply-templates>
          </table>
        </xsl:for-each-group>
      </body>
    </html>
        </xsl:template>

  <xsl:template match="Movie">
    <xsl:variable name="movie_year" select="@Year"/>
    <tr>
      <td><xsl:value-of select="@Title"/></td>
      <td><xsl:value-of select="@Year"/></td>
      <td><xsl:value-of select="Director/@Name"/></td>
      <td>
        <ul>
          <xsl:apply-templates select="current()/Actor">
            <xsl:with-param name="movie_year" select="$movie_year"/>
          </xsl:apply-templates>
        </ul>
      </td>
    </tr>
  </xsl:template>
```

```
<xsl:for-each-group select="//Movie" group-by="ProductionCompany">
<xsl:sort select="current-grouping-key() order="ascending"/>

<h2> <xsl:value-of select="current-grouping-key()"> </h2>
- the Production Company

<table border="1">
<tr> <th>Title</th> <th>Year</th> <th>Director</th> <th>Actors</th>
</tr>
<xsl:apply-templates select="current-group()">
```

```
  <xsl:template match="Actor">
    <xsl:param name="movie_year"/>
    <li>
      <xsl:value-of select="@Name"/>
      (<xsl:value-of select="$movie_year - @YearOfBirth"/>)
    </li>
  </xsl:template>

</xsl:transform>
```

```
</xsl:template>
</xsl:transform>
```

RESULT



```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Movies by production company</title>
    </head>
    <body>
        <h2>Movies by production company</h2>
        <h2>20th Century Fox</h2>
        <table border="1">
            <tr>
                <th>Title</th>
                <th>Year</th>
                <th>Director</th>
                <th>Actors</th>
            </tr>
            <tr>
                <td>Hide and Seek</td>
                <td>2005</td>
                <td>John Polson</td>
                <td>
                    <ul>
                        <li>Robert De Niro
                            (62)

                        </li>
                        <li>Dakota Fanning
                            (11)

                        </li>
                        <li>Famke Janssen
                            (41)
```

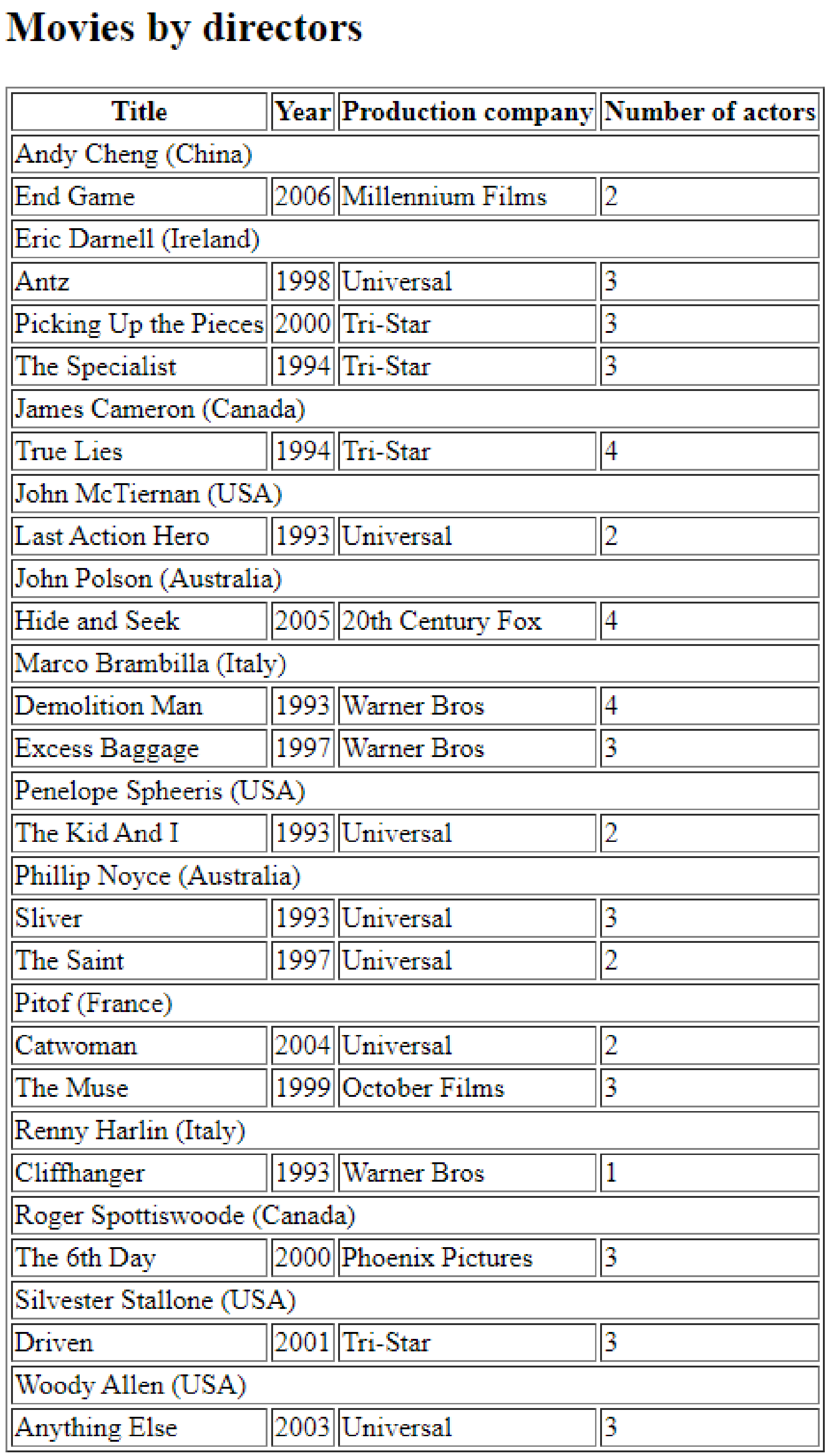
Question 03

Show all the movies grouped by director as HTML! One big table with one heading row per director with name and country together in one cell and followed by one row per movie with four columns: Title, Year, Production company, Number of actors. Make sure the table is filled with cells correctly by using the html attribute colspan. The directors and the movies shall be sorted alphabetically.

Solution in XSLT 1.0

```
<?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>Movies by directors</title>
      </head>
      <body>
        <h2>Movies by directors</h2>

        <table border="1">
          <xsl:variable name="unq_directors"
select="$doc//Director[not(@Name=preceding::Director/@Name)]"/>
          <tr>
            <th>Title</th>
            <th>Year</th>
            <th>Production company</th>
            <th>Number of actors</th>
          </tr>

          <xsl:apply-templates select="$unq_directors">
            <xsl:sort select="current()/@Name" order="ascending"/>
          </xsl:apply-templates>

        </table>
      </body>
    </html>
        </xsl:template>

  <xsl:template match="Director">
    <xsl:variable name="director_name" select="@Name"/>
    <tr>
      <td colspan="4">
        <xsl:value-of select="@Name"/>
        (<xsl:value-of select="@Country"/>)
      </td>
    </tr>

    <xsl:apply-templates select="//Movie[Director/@Name=$director_name]">
      <xsl:sort select="current()/@Title" order="ascending"/>
    </xsl:apply-templates>
  </xsl:template>

  <xsl:template match="Movie">
    <tr>
      <td>
        <xsl:value-of select="@Title"/>
      </td>
      <td>
        <xsl:value-of select="@Year"/>
      </td>
      <td>
        <xsl:value-of select="ProductionCompany"/>
      </td>
      <td>
        <xsl:value-of select="count(current()/Actor)"/>
      </td>
    </tr>
  </xsl:template>

</xsl:transform>
```

Result



Movies by directors

Title	Year	Production company	Number of actors
Andy Cheng (China)			
End Game	2006	Millennium Films	2
Eric Darnell (Ireland)			
Antz	1998	Universal	3
Picking Up the Pieces	2000	Tri-Star	3
The Specialist	1994	Tri-Star	3
James Cameron (Canada)			
True Lies	1994	Tri-Star	4
John McTiernan (USA)			
Last Action Hero	1993	Universal	2
John Polson (Australia)			
Hide and Seek	2005	20th Century Fox	4
Marco Brambilla (Italy)			
Demolition Man	1993	Warner Bros	4
Excess Baggage	1997	Warner Bros	3
Penelope Spheeris (USA)			
The Kid And I	1993	Universal	2
Phillip Noyce (Australia)			
Sliver	1993	Universal	3
The Saint	1997	Universal	2
Pitof (France)			
Catwoman	2004	Universal	2
The Muse	1999	October Films	3
Renny Harlin (Italy)			
Cliffhanger	1993	Warner Bros	1
Roger Spottiswoode (Canada)			
The 6th Day	2000	Phoenix Pictures	3
Silvester Stallone (USA)			
Driven	2001	Tri-Star	3
Woody Allen (USA)			
Anything Else	2003	Universal	3

Solution in XSLT 2.0

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>
        <xsl:template match="/">
    <html>
      <head><title>Movies by directors</title></head>
      <body>
        <h2>Movies by directors</h2>

        <table border="1">
          <tr>
            <th>Title</th>
            <th>Year</th>
            <th>Production company</th>
            <th>Number of actors</th>
          </tr>
```

table with headers for title, year, production company and number of actors.

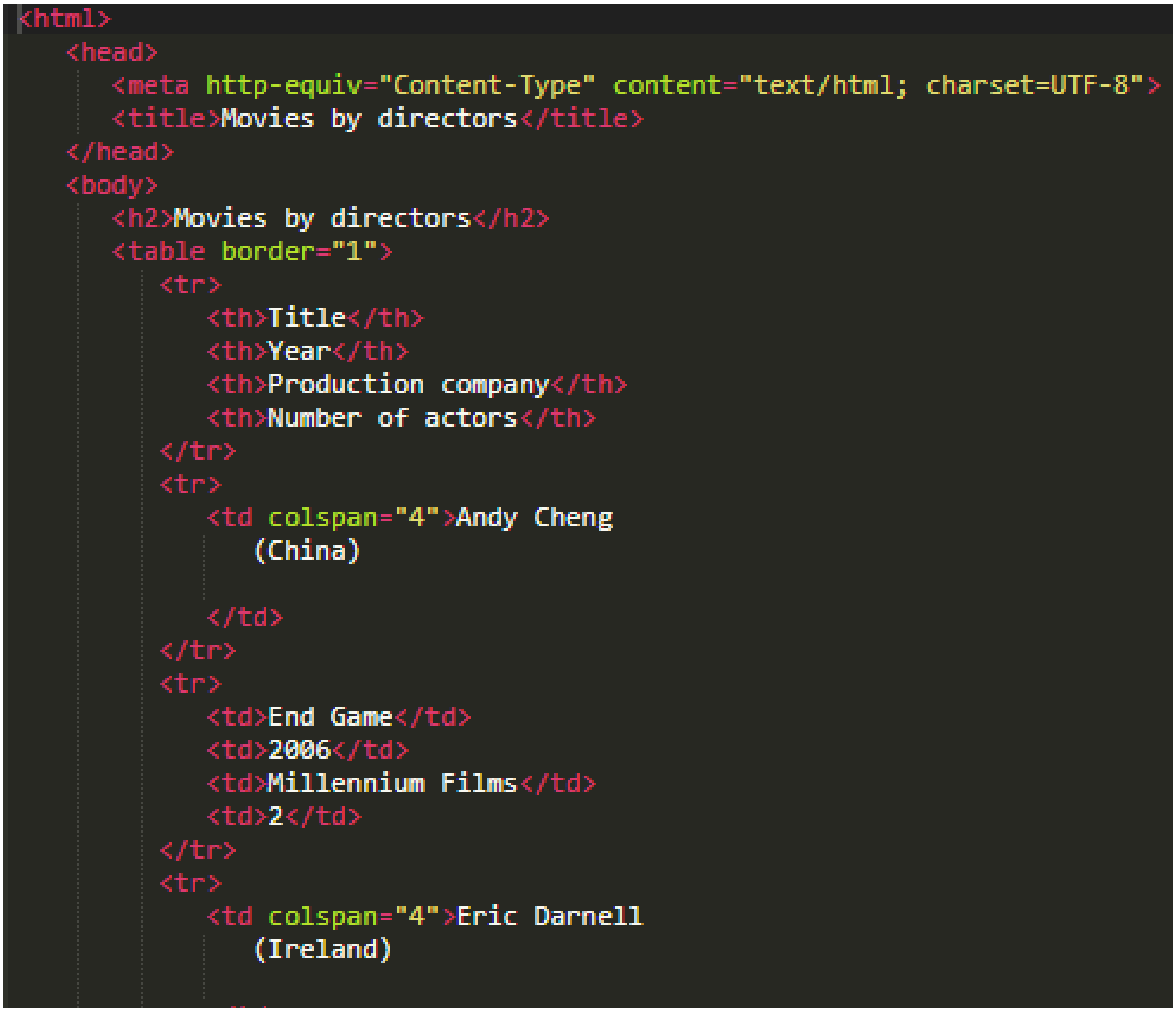
```
          <xsl:for-each-group select="//Movie" group-by="Director/@Name">
            <xsl:sort select="current-grouping-key()" order="ascending"/>
            <tr>
              <td colspan="4">
                <xsl:value-of select="current-grouping-key()"/>
                (<xsl:value-of select="distinct-values(//Director[@Name=current-grouping-key()]/@Country)"/>)
              </td>
            </tr>
            <xsl:apply-templates select="current-group()">
              <xsl:sort select="current()/@Title" order="ascending"/>
            </xsl:apply-templates>
          </xsl:for-each-group>
```

director's name- current-grouping-key()

xsl:sort select="current()/@Title" order="ascending"

```
        </table>
      </body>
    </html>
        </xsl:template>

  <xsl:template match="Movie">
    <tr>
      <td>
        <xsl:value-of select="@Title"/>
      </td>
      <td>
        <xsl:value-of select="@Year"/>
      </td>
      <td>
        <xsl:value-of select="ProductionCompany"/>
      </td>
      <td>
        <xsl:value-of select="count(current()/Actor)"/>
      </td>
    </tr>
  </xsl:template>

</xsl:transform>
```

Result



```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Movies by directors</title>
    </head>
    <body>
        <h2>Movies by directors</h2>
        <table border="1">
            <tr>
                <th>Title</th>
                <th>Year</th>
                <th>Production company</th>
                <th>Number of actors</th>
            </tr>
            <tr>
                <td colspan="4">Andy Cheng
                    (China)

                </td>
            </tr>
            <tr>
                <td>End Game</td>
                <td>2006</td>
                <td>Millennium Films</td>
                <td>2</td>
            </tr>
            <tr>
                <td colspan="4">Eric Darnell
                    (Ireland)
```