

Semi-Structured Data  
XPath, XSLT and XQUERY Lab Serie

### Exercise 1: XPATH

Use the Livre.xml file to make the following Xpath queries:

- 1- The titles of the books
- 2- Book codes
- 3- The author of the first book
- 4- The title of the 3rd book
- 5- Books that have not a year
- 6- the title of the book after the comic book
- 7- The number of books released before 2015
- 8- The title of the book does not have an author

### Exercise 2: XPATH

Let us consider the XML file cinema.xml, presented schematically below:

```
<cinema>
  <film id="1b78" type="comédie">
    <titre>Les Bronzés</titre>
    <annee>1978</annee>
    <realisateur>Patrice Leconte</realisateur>
    <role><nom>Popeye</nom><acteur>Thierry Lhermite</acteur></role>
    <role><nom>Jean-Claude Dusse</nom><acteur>Michel
Blanc</acteur></role>
  </film>
  <film id="gf94" type="comédie">
    <titre>Grosse fatigue</titre>
    <annee>1994</annee>
    <realisateur>Michel Blanc</realisateur>
    <role><nom>Patrick Olivier</nom><acteur>Michel
Blanc</acteur></role>
    <role><nom>Carole Bouquet</nom><acteur>Carole
Bouquet</acteur></role>
    <prix>Cannes, meilleur scénario</prix>
  </film>
</cinema>
```

```

    </film>
    <producteur>
      <nom>Daniel Toscan du Plantier</nom>
      <film ref="gf94"/>
    </producteur>
    <producteur>
      <nom>Yves Rousset-Rouard</nom>
      <film ref="lb78"/>
    </producteur>
  </cinema>

```

Write the following XPath queries:

1. The titles of all Items.
2. The titles of the films in which Michel Blanc plays.
3. Types of films.
4. The roles played by Michel Blanc.
5. The main actor of each film (first in the cast).
6. The name of the producer who appears just after Yves Rousset-Rouard.
7. The actors who appear before Michel Blanc in the cast of Patrice Leconte's films.
8. The directors of the films starring Michel Blanc and Thierry Lhermite.
9. The titles of the films with more than 5 actors in the cast.
10. Directors who star in at least one of their films.
11. The names of the producers of comedies.
12. Titles of films that did not win an award.
13. Actors who play a character whose name contains 'Morin'

### Exercise 3: XSLT

**XML document (bookstore.xml):**

```

<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="fiction" lang="en">
    <title>The Great Novel</title>
    <author>Jane Austen</author>
    <year>1813</year>
    <price currency="USD">29.99</price>
  </book>
  <book category="computer" lang="en">
    <title>XML Guide</title>

```

```

    <author>John Smith</author>
    <author>Mary Jones</author>
    <year>2023</year>
    <price currency="USD">49.99</price>
</book>
<book category="fiction" lang="fr">
    <title>Les Misérables</title>
    <author>Victor Hugo</author>
    <year>1862</year>
    <price currency="EUR">19.50</price>
</book>
<magazine category="tech">
    <title>Tech Monthly</title>
    <price currency="USD">5.99</price>
</magazine>
</bookstore>

```

Write an XSLT 1.0 stylesheet that produces an HTML page with the following requirements:

1. The page must have the title 'Bookstore Catalogue' and a level-1 heading 'Available Books'.
2. Display all books inside an HTML table with the columns: Title, Author(s), Year, Category, Price.
3. If a book has several authors, join them in the same cell separated by a comma.
4. Books whose price is strictly greater than 30 must be displayed with a yellow background row (use `xsl:if` and `xsl:attribute`).
5. Sort the rows by ascending year of publication.
6. Below the table, display the total number of books and the average price using `count()` and `sum()`.

Provide both the `.xsl` file and a brief description (2–3 lines) of the result.

## Exercise 4 : XSLT

**XML document (university.xml):**

```

<?xml version="1.0" encoding="UTF-8"?>
<university>
  <department name="Computer Science">
    <student id="S1" level="L3">
      <name>Ahmed Benali</name>
      <average>14.5</average>
    </student>
    <student id="S2" level="L3">
      <name>Sara Mansouri</name>
      <average>16.2</average>
    </student>
  </department>
</university>

```

```

</student>
<student id="S3" level="M1">
  <name>Karim Cherif</name>
  <average>12.8</average>
</student>
</department>
<department name="Mathematics">
  <student id="S4" level="L3">
    <name>Lina Haddad</name>
    <average>17.0</average>
  </student>
  <student id="S5" level="M1">
    <name>Omar Said</name>
    <average>11.5</average>
  </student>
</department>
</university>

```

Write an XSLT 1.0 stylesheet that produces a new XML document of the form:

```

<report>
  <department name="...">
    <passed>...</passed>      <!-- students with average >= 10 -->
    <failed>...</failed>    <!-- students with average < 10 -->
    <honors>
      <student level="...">name</student>  <!-- average >= 16 -->
      ...
    </honors>
  </department>
  ...
</report>

```

Specifications:

1. Iterate over each department using `xsl:for-each`.
2. Use `xsl:choose` / `xsl:when` to mark a student with a `<mention>` element: 'Excellent' if average  $\geq 16$ , 'Good' if  $14 \leq \text{average} < 16$ , 'Average' if  $10 \leq \text{average} < 14$ , 'Failed' otherwise.
3. Sort the students of each department by descending average using `xsl:sort`.
4. Define a global parameter `$minAverage` (default value 10) and use it instead of the literal 10 in the passed/failed test.
5. Add an attribute count on the `<passed>` and `<failed>` elements giving the number of students in each category.

**Exercise 4 : XQUERY** Let us consider the Films.xml and Artistes.xml files, which have the following structures, respectively:

```

<FILMS>
  <FILM annee="1958">
    <TITRE>Vertigo</TITRE>
    <GENRE>Drame</GENRE>
    <PAYS>USA</PAYS>
    <MES idref="3"/>
    <ROLES>
      <ROLE>
        <PRENOM>James</PRENOM><NOM>Stewart</NOM>
        <INTITULE>John Ferguson</INTITULE>
      </ROLE>
      <ROLE>
        <PRENOM>Kim</PRENOM><NOM>Novak</NOM>
        <INTITULE>Madeleine Elster</INTITULE>
      </ROLE>
    </ROLES>
    <RESUME>Scottie Ferguson, ancien inspecteur de police...</RESUME>
  </FILM>
  ...
</FILMS>

...
<ARTISTES>
  <ARTISTE id="3">
    <ARTNOM>Hitchcock</ARTNOM><ARTPNOM>Alfred</ARTPNOM>
    <ANNEENAISS>1899</ANNEENAISS>
  </ARTISTE>
  <ARTISTE id="5">
    <ARTNOM>Weaver</ARTNOM><ARTPNOM>Sigourney</ARTPNOM>
  </ARTISTE>
  ...
</ARTISTES>

```

Write the following queries in XQuery:

1. The title, genre and country for all films before 1970.
2. The roles played by Bruce Willis.
3. The roles played by Bruce Willis in the form of a role, a role containing as sub-elements, the title of the film, the name of the character.
4. The name of the director of the film Vertigo.
5. For each artist, their name and the titles of the films they directed.
6. For each film, the age of its director at the time of the film's release.
7. For each genre of film, produce a genre element with the name of the genre as an attribute and containing the titles of the films of the genre in question.

8. Artists who have acted in a film they have directed. For each artist, an artist element is produced with the full name (first name followed by last name) and film sub-elements containing the title of the film in question, with the year of the film as an attribute