

# Arquitectura de bases de dades

## Pràctica 2: Extensió XML

### Pregunta 1 (2 punts)

#### Enunciat

Respon cadascuna de les preguntes següents justificant breument la teva resposta.

- a) Donats els següents XML `exemple1.xml` i `exemple2.xml`, indica quin seria l'XML resultant de la seva unió si tenim en compte que tenen els següents URI:

`Example1.xml` -> `http://www.uoc.edu/subjects/adb/ns/example1#`

`Example2.xml` -> `http://www.uoc.edu/subjects/adb/ns/example2#`

I volem que s'especifiqui pels nodes conflictius, el seu *namespace*.

#### Example1.xml

```
<root>
  <movie>
    <actor>John Travolta</actor>
  </movie>
</root>
```

#### Example2.xml

```
<root>
  <movie>
    <plot> The film begins with a diner hold-up staged
    by a couple,... </plot>
  </movie>
</root>
```

- b) Digues si és certa o falsa la següent afirmació: Tot document XML que es guarda en una base de dades Oracle es troba en un objecte tipus `XMLType`.
- c) Quines clàusules podem utilitzar en una consulta XQuery? En què consisteix cadascuna d'elles?
- d) Donat l'XML següent:

```
<nodeA>
  <nodeB>
    <nodeC/>
    <nodeD/>
  </nodeB>
```

```

<nodeX>
  <nodeD>
    <nodeE/>
    <nodeD/>
    <nodeC/>
    <nodeF>
      <nodeG/>
    </nodeF>
  </nodeD>
</nodeX>
<nodeC>
  <nodeD/>
</nodeC>
</nodeA>

```

Amb quina sentència XPath obtindríem els següents nodes? Justifica la teva resposta.

```

<nodeB>
  <nodeC/>
  <nodeD/>
</nodeB>

```

## Criteris d'avaluació

Els criteris d'avaluació que s'aplicaran en la correcció d'aquesta pregunta són els següents:

- Totes les preguntes tenen el mateix pes.
- Les preguntes no contestades no penalitzen.
- Les preguntes sense argumentació no seran avaluades.
- Es valorarà la qualitat de la resposta.

## Solució

- a) Donats els següents XML exemple1.xml i exemple2.xml, indica quin seria l'XML resultant de la seva unió si tenim en compte que tenen els següents URI:

Example1.xml -> <http://www.uoc.edu/subjects/adb/ns/example1#>

Example2.xml -> <http://www.uoc.edu/subjects/adb/ns/example2#>

I volem que s'especifiqui pels nodes conflictius, el seu *namespace*.

Example1.xml

```
<root>
```

```
<movie>
  <actor>John Travolta</actor>
</movie>
</root>
```

#### Example2.xml

```
<root>
<movie>
  <plot> The film begins with a diner hold-up staged by a
    couple,... </plot>
</movie>
</root>
```

#### Resposta:

Donat que se'ns indica l'URI de cada XML i se'ns demana que indiquem el namespace, l'XML resultant seria:

```
<root
xmlns:ex1=" http://www.uoc.edu/subjects/adb/ns/example1#"
xmlns:ex2="http://www.uoc.edu/subjects/adb/ns/example2#" >
<ex1:movie>
  <actor>John Travolta</actor>
</ex1:movie>
<ex2:movie>
  <plot> The film begins with a diner hold-up staged by a
    couple,... </plot>
</ex2:movie>
</root>
```

Per a cada URI indiquem el seu alias que és el que utilitzarem a l'hora d'unir els tags amb el mateix nom.

- b) Digues si és certa o falsa la següent afirmació: Tot document XML que es guarda en una base de dades Oracle es troba en un objecte tipus XMLType.

Resposta: FALS.

Com s'explica en el mòdul, Oracle XML DB Repository és el component del gestor de bases de dades Oracle que gestiona les dades en XML i contempla dos tipus d'emmagatzemament, estructurat o no estructurat.

- L'emmagatzemament estructurat es basa en XMLType, un nou tipus de dades que permet indicar a la base de dades que la columna o la taula contenen informació en XML. Té com a avantatges respecte l'emmagatzematge no estructurat una millor gestió de la memòria, dels recursos i indexació en forma d'arbre per optimitzar les consultes XPath, tot i que, per contra, té un major consum de recursos a l'hora d'obtenir el document XML.
- L'emmagatzemament no estructurat proporciona un major rendiment en les tasques de recuperació de documents XML, per contra baixa el rendiment quan s'intenta treballar amb una granularitat més fina. En aquest cas, s'utilitzen objectes tipus CLOB, BLOB, BFILE o VARCHAR per guardar la informació i és difícil poder realitzar alguna optimització de les cerques o actualitzacions dels XMLs guardats en aquests formats.

- c) Quines clàusules podem utilitzar en una consulta XQuery? En què consisteix cadascuna d'elles?

Resposta:

Les consultes XQuery es componen de cinc tipus diferents de clàusules que s'anomenen FLWOR (es pronuncia *flower*) que representa **for**, **let**, **where**, **order by** i **return**. Cada expressió FLWOR retorna una seqüència de tuples:

- **for**: vincula una o més variables a expressions escrites en XPath creant una fila (tuples) que conté els valors de les variables definides al **for**.
- **let**: vincula una variable al resultat d'una expressió.
- **where**: filtra les tuples eliminant les que no compleixen les condicions indicades.
- **order by**: ordena la seqüència de tuples amb el criteri indicat.
- **return**: construeix l'XML resultant de la cerca.

- d) Donat l'XML següent:

```
<nodeA>
  <nodeB>
    <nodeC/>
    <nodeD/>
  </nodeB>
  <nodeX>
    <nodeD>
      <nodeE/>
      <nodeD/>
      <nodeC/>
      <nodeF>
        <nodeG/>
      </nodeF>
    </nodeD>
  </nodeX>
  <nodeC>
    <nodeD/>
  </nodeC>
</nodeA>
```

Amb quina sentència XPath obtindríem els següents nodes:

```
<nodeB>
  <nodeC/>
  <nodeD/>
</nodeB>
```

Resposta:

Per obtenir l'xml indicat cal executar la sentència:

```
/nodeA/nodeX/preceding-sibling::*
```

L'eix `preceding-sibling` conté tots els nodes germans que precedeixen al node actual i també, en conseqüència, tots els seus fills. Per aquest motiu al situar-nos en el nodeX, ens retorna el nodeB (i els seus fills), ja que es troba al mateix nivell que el nodeX i, per tant, són germans.

## Pregunta 2 (4 punts)

### Enunciat

Utilitzeu el fitxer XML adjunt a la pràctica (movies.xml), que conté la selecció que ha fet el responsable de programació de Netflix per incorporar una secció de clàssics a la plataforma, per a resoldre cadascun dels apartats següents:

- Proposeu una sentència XPath que retorni el nom de les pel·lícules (`title`) que han estat escrites per “Stephen King”, per tal d’incloure-les a la secció de terror.
- Proposeu una sentència XPath que busqui el node que conté la informació de la pel·lícula “Forrest Gump” i retorni el títol (`title`) i director (`director`) de la pel·lícula que trobem al node anterior (ex. Title – Director).
- Proposeu una sentència XPath que calculi el número de pel·lícules que pertanyen a més de dos gèneres (>2).
- Proposeu una sentència XPath que retorni el nom de la pel·lícula de més durada (atribut `totaltime`).

Les sentències XPath de cada apartat han d’anar acompanyades d’una explicació i justificació de la sentència.

### Criteris d’avaluació

Els criteris d’avaluació que s’aplicaran en la correcció d’aquesta pregunta són els següents:

- Cal lliurar tots els apartats per tal que l’exercici sigui avaluat.
- No es valorarà l’apartat si no hi ha argumentació o no és possible provar la sentència XPath.
- Es valorarà la qualitat de la resposta, així com l’ús dels conceptes estudiats en el mòdul 2 corresponents a l’extensió XML.
- Les respostes incorrectes no descompten.

### Solució

Utilitzeu el fitxer XML adjunt a la pràctica, que descriu les diferents Denominacions d’Origen de Vi de Catalunya per a resoldre cadascun dels apartats següents:

- Proposeu una sentència XPath que retorni el nom de les pel·lícules (`title`) que han estat escrites per “Stephen King”, per tal d’incloure-les a la secció de terror.

### Sentència XPath

```
doc("movies.xml")//movies/movie/writers/writer[contains(text(),"Stephen King")]//../../title/text()
```

### Resultat

The Shawshank Redemption  
The Shining

S'avalua el text dels nodes writer situant-nos al node, amb la comanda `[contains(text(),"Stephen King")]` i recuperem el nom de la pel·lícula agafant el text del camp `title`.

b) Proposeu una sentència XPath que busqui el node que conté la informació de la pel·lícula "Forrest Gump" i retorni el títol (`title`) i director (`director`) de la pel·lícula que trobem al node anterior (ex. Title – Director).

### Sentència XPath

```
doc("movies.xml")//movies/movie[contains(title,"Forrest Gump")]//preceding::movie[position()=1]/concat(title/text(), '- ', director/text())
```

### Resultat

Seven Samurai-Akira Kurosawa

### Solució

De manera similar a l'apartat anterior, busquem la pel·lícula que es digui "Forrest Gump" `/movie[contains(title,"Forrest Gump")]` i ens col·loquem al node anterior mitjançant: `/preceding::movie[position()=1]/` i, un cop ens hem situat, només ens cal obtenir el nom de la pel·lícula i del director.

c) Proposeu una sentència XPath que calculi el número de pel·lícules que pertanyen a més de dos gèneres (>2).

### Sentència XPath

```
count(doc("movies.xml")//movies/movie/genres[count(genre)>2])
```

### Resultat

29

### Solució

Primer obtenim les pel·lícules que tenen més d'un gènere comptant `genres[count(genre)` i posteriorment es compten els nodes que compleixen aquesta condició amb la funció `count`.

d) Proposeu una sentència XPath que retorni el nom de la pel·lícula de més durada (atribut `totaltime`).

Sentència XPath

```
doc("movies.xml")//movies/movie/runtime[@totaltime=max(doc("movies.xml")
//movies/movie/runtime/@totaltime)]/../title/text()
```

Resultat

Saving Private Ryan

Solució

L'ús de la funció MAX ens permet obtenir el valor màxim de l'atribut `totaltime` entre tots els nodes. Un cop es troba el node, s'obté el seu títol per mostrar-ne el resultat.

## Pregunta 3 (2 punts)

### Enunciat

Proporcioneu una sentència XQuery que, utilitzant el fitxer XML adjunt a la pràctica (`movies.xml`), retorni un resum de les pel·lícules que estan tipificades amb el gènere familiars (`genre=family`). El format de sortida ha de ser el següent, de manera que només es retorni el nom de la pel·lícula, la trama (`plot`) i el número de guionistes (`writer`):

```
<movies>
<family-movie>
  <title>movie title </title>
  <plot>movie plot</description>
  <num-writers>xx</num-writers>
</family-movie>
<family-movie>
  ...
</family-movie>
....
</movies>
```

Doneu una explicació i justificació de la sentència XQuery realitzada.



## Criteris d'avaluació

Els criteris d'avaluació que s'aplicaran en la correcció d'aquesta pregunta són els següents:

- No es valorarà l'apartat si no hi ha argumentació, no és possible provar la consulta XQuery o el format de la resposta no s'ajusta al demanat.
- Es valorarà la qualitat de la resposta, així com l'ús dels conceptes estudiats en el mòdul 2 corresponents a l'extensió XML.

## Solució

### Sentència XQuery

```
<movies>
{
  for $do in doc("movies.xml")/movies/movie
  where $do/genres/genre/text() = "Family"
  return
  <family-movie>
    <name>{$do/title/text()}</name>
    <plot>{$do/plot/text()}</plot>
    <num-writers>{count($do/writers/writer)}</num-writers>
  </family-movie>
}
</movies>
```

### Resposta

```
<movies>
  <family-movie>
    <name>It's a Wonderful Life</name>
    <plot>An angel helps a compassionate but despairingly frustrated businessman by
showing what life would have been like if he never existed.</plot>
    <writers>2</writers>
  </family-movie>
  <family-movie>
    <name>Spirited Away</name>
    <plot>In the middle of her family's move to the suburbs, a sullen 10-year-old girl
wanders into a world ruled by gods, witches, and monsters; where humans are changed
into animals; and a bathhouse for these creatures.</plot>
    <writers>1</writers>
  </family-movie>
</movies>
```

### Solució

Obtenim un cursor amb totes les pel·lícules tipificades amb el gènere de pel·lícules familiars (utilitzant el `where` per filtrar el resultat). Posteriorment obtenim el títol, la trama i comptem (amb la funció `count`) el número de guionistes per a poder-los mostrar en

l'XML de resum que se'ns demana.

## Pregunta 4 (2 punts)

### Enunciat

Proporcioneu una sentència XQuery que, utilitzant el fitxer XML adjunt a la pràctica (*movies.xml*), retorni els diferents gèneres de pel·lícules i indiqui, per a cada gènere, les diferents pel·lícules classificades amb aquest gènere (noteu que una pel·lícula pot estar tipificada com de diversos gèneres). El format de sortida ha de seguir el que es marca a continuació i ha d'estar ordenat pel nom de gènere:

```
<movies-genres>
  <genre type="genre 1" >
    <movie>Title 1</movie>
    <movie>Title 2</movie>
    ...
  </genre>
  <genre type="genre 2" >
    <movie>Title 1</movie>
    <movie>Title 3</movie>
    ...
  </genre>
  ...
</movies-genres>
```

Doneu una explicació i justificació de la sentència XQuery realitzada.

### Criteris d'avaluació

Els criteris d'avaluació que s'aplicaran en la correcció d'aquesta pregunta són els següents:

- No es valorarà l'apartat si no hi ha argumentació, no és possible provar la consulta XQuery o el format de la resposta no s'ajusta al demanat.
- Es valorarà la qualitat de la resposta, així com l'ús dels conceptes estudiats en el mòdul 2 corresponents a l'extensió XML.

### Solució

Sentència XQuery

```
<movies-genres>
{
for $genre in
  (distinct-values(doc("movies.xml")//genres/*))
order by $genre
```

```

return
<genre type="{ $genre }">
{
  for $movie in (doc("movies.xml")/movies/movie)
  where $movie/genres/genre=$genre
  return
    <movie>{$movie/title/text()}</movie>
}
</genre>
}
</movies-genres>

```

## Resultat

```

<movies-genres>
  <genre type="Action">
    <movie>The Dark Knight</movie>
    <movie>The Lord of the Rings: The Return of the King</movie>
    <movie>Star Wars: Episode V - The Empire Strikes Back</movie>
    <movie>The Lord of the Rings: The Fellowship of the Ring</movie>
    <movie>Inception</movie>
    <movie>Star Wars</movie>
    <movie>Seven Samurai</movie>
    <movie>The Matrix</movie>
    <movie>The Lord of the Rings: The Two Towers</movie>
    <movie>Raiders of the Lost Ark</movie>
    <movie>Terminator 2: Judgment Day</movie>
    <movie>Saving Private Ryan</movie>
    <movie>North by Northwest</movie>
  </genre>
  <genre type="Adventure">
    <movie>The Good, the Bad and the Ugly</movie>
    <movie>The Lord of the Rings: The Return of the King</movie>
    <movie>Star Wars: Episode V - The Empire Strikes Back</movie>
    <movie>The Lord of the Rings: The Fellowship of the Ring</movie>
    <movie>Inception</movie>
    <movie>Star Wars</movie>
    <movie>The Matrix</movie>
    <movie>The Lord of the Rings: The Two Towers</movie>
    <movie>Once Upon a Time in the West</movie>
    <movie>Raiders of the Lost Ark</movie>
    <movie>Spirited Away</movie>
    <movie>North by Northwest</movie>
    <movie>Back to the Future</movie>
  </genre>
  <genre type="Animation">
    <movie>Spirited Away</movie>
  </genre>
  <genre type="Biography">
    <movie>Schindler's List</movie>
    <movie>Goodfellas</movie>
  </genre>

```

```

<movie>The Pianist</movie>
</genre>
<genre type="Comedy">
  <movie>City Lights</movie>
  <movie>Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb</movie>
  <movie>Modern Times</movie>
  <movie>Back to the Future</movie>
  <movie>Life Is Beautiful</movie>
</genre>
<genre type="Crime">
  <movie>The Shawshank Redemption</movie>
  <movie>The Godfather</movie>
  <movie>The Godfather: Part II</movie>
  <movie>Pulp Fiction</movie>
  <movie>The Dark Knight</movie>
  <movie>Goodfellas</movie>
  <movie>City of God</movie>
  <movie>Se7en</movie>
  <movie>The Silence of the Lambs</movie>
  <movie>The Usual Suspects</movie>
  <movie>Léon: The Professional</movie>
  <movie>American History X</movie>
  <movie>North by Northwest</movie>
  <movie>The Departed</movie>
  <movie>M</movie>
</genre>
<genre type="Drama">
  <movie>The Shawshank Redemption</movie>
  <movie>The Godfather</movie>
  <movie>The Godfather: Part II</movie>
  <movie>Pulp Fiction</movie>
  <movie>The Dark Knight</movie>
  <movie>12 Angry Men</movie>
  <movie>Schindler's List</movie>
  <movie>Fight Club</movie>
  <movie>One Flew Over the Cuckoo's Nest</movie>
  <movie>Goodfellas</movie>
  <movie>Seven Samurai</movie>
  <movie>Forrest Gump</movie>
  <movie>City of God</movie>
  <movie>The Silence of the Lambs</movie>
  <movie>Casablanca</movie>
  <movie>It's a Wonderful Life</movie>
  <movie>Léon: The Professional</movie>
  <movie>Sunset Blvd.</movie>
  <movie>American History X</movie>
  <movie>Apocalypse Now</movie>
  <movie>Saving Private Ryan</movie>
  <movie>City Lights</movie>
  <movie>Modern Times</movie>

```

```

<movie>Life Is Beautiful</movie>
<movie>The Pianist</movie>
<movie>Citizen Kane</movie>
<movie>The Departed</movie>
<movie>M</movie>
</genre>
<genre type="Family">
  <movie>It's a Wonderful Life</movie>
  <movie>Spirited Away</movie>
</genre>
<genre type="Fantasy">
  <movie>The Lord of the Rings: The Return of the King</movie>
  <movie>The Lord of the Rings: The Fellowship of the Ring</movie>
  <movie>Star Wars</movie>
  <movie>The Lord of the Rings: The Two Towers</movie>
  <movie>It's a Wonderful Life</movie>
  <movie>Spirited Away</movie>
</genre>
<genre type="Film-Noir">
  <movie>Sunset Blvd.</movie>
</genre>
<genre type="History">
  <movie>Schindler's List</movie>
  <movie>The Pianist</movie>
</genre>
<genre type="Horror">
  <movie>Psycho</movie>
  <movie>Alien</movie>
  <movie>The Shining</movie>
</genre>
<genre type="Mystery">
  <movie>Inception</movie>
  <movie>Se7en</movie>
  <movie>The Usual Suspects</movie>
  <movie>Rear Window</movie>
  <movie>Psycho</movie>
  <movie>Memento</movie>
  <movie>North by Northwest</movie>
  <movie>The Shining</movie>
  <movie>Citizen Kane</movie>
</genre>
<genre type="Romance">
  <movie>Forrest Gump</movie>
  <movie>Casablanca</movie>
  <movie>City Lights</movie>
  <movie>Life Is Beautiful</movie>
</genre>
<genre type="Sci-Fi">
  <movie>Star Wars: Episode V - The Empire Strikes Back</movie>
  <movie>Inception</movie>
  <movie>Star Wars</movie>

```

```

<movie>The Matrix</movie>
<movie>Terminator 2: Judgment Day</movie>
<movie>Dr. Strangelove or: How I Learned to Stop Worrying and Love the
Bomb</movie>
<movie>Alien</movie>
<movie>Back to the Future</movie>
</genre>
<genre type="Thriller">
<movie>Pulp Fiction</movie>
<movie>The Dark Knight</movie>
<movie>Inception</movie>
<movie>Goodfellas</movie>
<movie>Se7en</movie>
<movie>The Silence of the Lambs</movie>
<movie>The Usual Suspects</movie>
<movie>Rear Window</movie>
<movie>Psycho</movie>
<movie>Léon: The Professional</movie>
<movie>Terminator 2: Judgment Day</movie>
<movie>Memento</movie>
<movie>North by Northwest</movie>
<movie>The Departed</movie>
<movie>M</movie>
</genre>
<genre type="War">
<movie>Schindler's List</movie>
<movie>Casablanca</movie>
<movie>Apocalypse Now</movie>
<movie>Saving Private Ryan</movie>
<movie>Dr. Strangelove or: How I Learned to Stop Worrying and Love the
Bomb</movie>
<movie>Life Is Beautiful</movie>
<movie>The Pianist</movie>
</genre>
<genre type="Western">
<movie>The Good, the Bad and the Ugly</movie>
<movie>Once Upon a Time in the West</movie>
</genre>
</movies-genres>

```

### Solució

En un cursor, obtenim tots els gèneres de les pel·lícules i els ordenem (`order by`). Per evitar les repeticions, agafem els valors únics utilitzant la sentència `distinct-values`. Després obtenim un altre cursor amb totes les pel·lícules que tenen el gènere obtingut en el pas previ i escrivim el títol de la pel·lícula a l'XML de sortida.

## Recursos

Per tal de resoldre aquesta pràctica caldrà utilitzar els recursos que s'enumeren a continuació:

- Mòdul 2 de l'assignatura Arquitectura de Bases de Dades (únicament la secció 2 que fa referència l'extensió XML)
- Conjunt de dades movies.xml
- Per resoldre la pràctica podeu utilitzar el processador BaseX (XPath, XQuery processor). El trobareu a : <http://files.basex.org/releases/8.6.7/BaseX867.jar>

## Criteris de valoració

A l'enunciat de cada exercici s'indica el valor del mateix sobre la puntuació total de la pràctica. Aquesta activitat representa el 50% de la nota de pràctiques de l'assignatura i la seva realització és **obligatòria** per tal de superar l'assignatura.

No s'acceptaran ni es tindran en compte els lliuraments realitzats fora dels terminis indicats al calendari de l'aula.

## Format i data de lliurament

Caldrà lliurar, a través de la bústia de lliuraments de l'assignatura de l'aula, un arxiu .zip que contingui els fitxers següents:

- Un document PDF que contingui les respostes dels exercicis 1, 2, 3 i 4.
- Un fitxer de text per cada una de les sentències generades en els exercicis 2, 3 i 4. Assegureu-vos que aquests siguin executables, que no esteu utilitzant variables d'entorn i que la crida al fitxer xml sigui doc ("movies.xml").

Tots els documents lliurats hauran de contenir el nom i cognoms de l'alumne, altrament no seran avaluats.

- Lliurament individual: Cal incloure el nom i cognoms a cadascun dels fitxers que lliureu, i cal indicar que es tracta de la PR2.
- Lliurament en parella: Cal incloure el nom i cognom dels dos estudiants a cadascun dels fitxers que lliureu, i cal indicar que es tracta de la PR2. Únicament un dels components ha de fer el lliurament al registre d'avaluació continuada (campus).

La data límit per lliurar la PR2 és el **10/11/2019**.