

XQuery-XPath

Expression de chemin et requêtes :

Requêtes simples :

- Pour le fichier movies.xml

- 1- /movies/movie/title
- 2- /movies/movie/title/text()
- 3- /movies/movie[year>2000]/title
- 4- /movies/movie[title='Spider-Man']/summary
- 5- /movies/movie[title="Heat"]/director[last_name]/text()
- 6- /movies/movie[actor[first_name="Kirsten" and last_name="Dunst"]]/title
- 7- /movies/movie[summary]/title
- 8- /movies/movie[not(summary)]/title
- 9- /movies/movie[year<2013]/title
- 10- /movies/movie[title="Unforgiven"]/actor[first_name="Clint" and last_name="Eastwood"]/role
- 11- /movies/movie[last()]/title
- 12- /movies/movie[title="Marie Antoinette"]/preceding-sibling ::movie[1]/title
- 13- /movies/movie[contains(title,"V")]/title
- 14- /movies/movie[count(actor)=3]/title

- Pour le fichier movies_refs.xml

- 1- /movies/movie/title
- 2- /movies/movie/title/text()
- 3- /movies/movie[year>2000]/titleNf
- 4- /movies/movie[title='Spider-Man']/summaryNf
- 5- //artist[@id=//movie[title="Heat"]/director/@id]/last_name/text()
- 6- //movie[actor[@id=//artist[last_name="Dunst" and first_name="Kirsten"]/@id]]/title
- 7- /movies/movie[summary]/title
- 8- /movies/movie[not(summary)]/title
- 9- /movies/movie[year<2013]/title
- 10- //movie[actor[@id=//artist[last_name='Eastwood' and first_name='Clint']/@id] and title='Unforgiven']/actor[@id=//artist[last_name='Eastwood' and first_name='Clint']/@id]/@role
- 11- /movies/movie[last()]/title
- 12- /movies/movie[title='Marie Antoinette']/preceding-sibling ::movie[1]/title
- 13- /movies/movie[contains(title,'V')]/title
- 14- /movies/movie[count(actor)=3]/title

Requetes Xquery:

1- liste des films (titres et année) parus après 2002

```
let $ms:=doc("movies_alone.xml")  
return $ms/movies/movie[year>2000]/(title,year)
```

2- Créer une liste plate de couples titre-rôle , chaque couple étant inclus dans un élément

```
<results>{  
  let $ms:=doc("movies_alone.xml")  
  for $v in $ms/movies/movie  
  for $r in $v/actor/@role  
  return  
    <result>  
      <title>{$v/title/text()}</title>  
      <role>{$r/string()} </role>  
    </result>  
}</results>
```

3-les titres des films dans lesquels le réalisateur est également un des acteurs

```
let $ms:=doc("movies_alone.xml")  
return $ms//movie[director/@id=$ms//movie/actor/@id]/title/text()
```

4- la liste des films regroupé par genre

```
let $ms:=doc("movies_alone.xml")  
return distinct-values($ms/movies/movie/(title,genre))
```

5-pour chaque acteur dans movies_alone.xml donnez les titres des films dans lesquels il à joué :

```
<results>{
  let $ms:=doc("movies_alone.xml")
  for $r in $ms/movies/movie/actor
    for $v in $r/movies/movie/title
      return
        <result>
          <actor>{$r/actor}</actor>
          <title>{$v/title} </title>
        </result>
}</results>
```

6-

```
for $actor in distinct-values($ms //actor/@id)
let $nombre :=count($ms//movie[actor/@id=$actor]/title)
where ($nombre >=2)
return($actor,$ms//movie[actor/@id=$actor]/title)
```

7-

```
for $director in distinct-values($ms //director/@id)
return($as//artist[@id=$director],$ms//movie[director/@id=$director]/title)
```

8-

```
for $title in distinct-values($ms //movie/title)
return ($title,<actors>{$as//artist[@id= $ms //movie[title=$title]/actor/@id]}</actors>)
```

9-

```
for $i in distinct-values($ms //movie/title)
let $nombre :=count( $ms//movie[title=$i]/actor)
let $id1 := $ms//movie[title=$i]/actor[1]/@id
let $id2 := $ms//movie[title=$i]/actor[2]/@id
where ($nombre >=2)
return(<result>{<title>{$i}</title>,
<actor>{$as //artist[@id=$id1]/concat(first_name,last_name)}</actor>,
<actor>{$as //artist[@id=$id2]/concat(first_name,last_name)}</actor>,<et-al/>}</result>)
```

10-

```
let $mov := $ms //movie[year>1990 and (director/@id=$as
//artist[first_name="Clint"]/@id)]/(title,year)
for $movi in $mov
order by $movi ascending
return ($movi)
```

11-

```
let $mov := $ms //movie[year<1990 and (director/@id=$as
//artist[first_name="Clint"]/@id)]/(title,year)
for $movi in $mov
order by $movi ascending
return ($movi)
```