

## Corrigé TD XQuery

### EXERCICE 1 requêtes simples

```
{-- Affichez tous les titres de films --}
```

```
<FILMS>
{
  for
    $f in doc("data/Films.xml")//FILM
  return
    <FILM>{$f/TITRE}</FILM>
}
</FILMS>
```

```
{-- Affichez tous les titres de films triés par ordre alphabétique--}
```

```
for $e in doc("data/Films.xml")//FILM
order by $e/TITRE ascending
return $e/TITRE
```

```
{-- Les titres des films d'horreur --}
```

```
<FILMS>
{
  for
    $f in doc("data/Films.xml")//FILM
  where
    $f/GENRE="Horreur"
  return
    <FILM>{$f/TITRE}</FILM>
}
</FILMS>
```

```
{-- Le résumé d'Alien --}
```

```
for
  $f in doc("data/Films.xml")//FILM
where
  $f/TITRE="Alien"
return
  $f/RESUME
```

```
{-- nombre de films dans le document --}
```

```
for $v in doc("data/Films.xml")//FILMS
let $f := $v/FILM
return
<NombreFilms> {count($f)} </NombreFilms>
```

```
{-- Titre des films avec James Stewart et Kim Novak --}
```

```
for $f in doc("data/Films.xml")//FILM
where ($f/ROLES/ROLE/NOM="Stewart" and $f/ROLES/ROLE/PRENOM="James") and
($f/ROLES/ROLE/NOM="Novak" and $f/ROLES/ROLE/PRENOM="Kim")
return $f/TITRE/text()
```

```
{-- Quels films ont un résumé ? --}
```

```
<FILMS>
{
  for
    $f in doc("data/Films.xml")//FILM
  where
```

```

        $f/RESUME!=" "
    return
    <FILM>{$f/TITRE}</FILM>
}
</FILMS>
=====
{-- Quels films n'ont pas de résumé ? --}
<FILMS>
{
    for
        $f in doc("data/Films.xml")//FILM
    where
        not($f/RESUME!=" ")
    return
        <FILM>{$f/TITRE}</FILM>
}
</FILMS>
=====
{-- Quel rôle joue Harvey Keitel dans le film Reservoir dogs ? --}
for
    $f in doc("data/Films.xml")//FILM
where
    $f/TITRE="Reservoir dogs"
return
    for
        $r in $f/ROLES/ROLE
    where
        $r/PRENOM="Harvey" and $r/NOM="Keitel"
    return
        $r/INTITULE
=====
{-- Donnez les titres des films qui contiennent un 'A' (utiliser la fonction contains) --}
<FILMS>
{
    for
        $f in doc("data/Films.xml")//FILM
    where
        contains($f/TITRE, 'A')
    return
        <FILM>{$f/TITRE}</FILM>
}
</FILMS>

```

## EXERCICE 2a requêtes jointes

{-- Qui a mis en scène le film Eyes Wide Shut ? --}

```
<ARTISTE>
{
  for
    $f in doc("data/Films.xml")//FILM,
    $ms in doc("data/Artistes.xml")//ARTISTE
  where
    $f/MES/@idref=$ms/@id and $f/TITRE="Eyes Wide Shut"
  return
    ($ms/ACTNOM,$ms/ACTPNOM)
}
</ARTISTE>
```

{-- Les films où le metteur en scène participe comme acteur --}

```
<LISTESFILMS>
{
  for
    $ms in doc("data/Artistes.xml")//ARTISTE,
    $f in doc("Films.xml")//FILM,
    $r in $f/ROLES/ROLE
  where
    $f/MES/@idref=$ms/@id
    and $r/PRENOM=$ms/ACTPNOM
    and $r/NOM=$ms/ACTNOM
  return
    <FILM>{$f/TITRE}</FILM>
}
</LISTESFILMS>
```

## EXERCICE 2b requêtes complexes

{-- Affichez La liste des acteurs, triée par ordre alphabétique (nom et prénom), avec la liste des films dans lesquels il a joué --}

```
<ACTEURS>
{
  for
    $nom in distinct-values(document("Films.xml")//FILM/ROLES/ROLE/NOM),
    $prenom in distinct-
values(document("data/Films.xml")//FILM/ROLES/ROLE[NOM=$nom]/PRENOM)
  order by
    $nom,$prenom
  return
    <ACTEUR>
      <NOM>{$nom}</NOM>
      <PRENOM>{$prenom}</PRENOM>
      <FILMS>
        {
          for
            $f in doc("Films.xml")//FILM,
            $role in $f/ROLES/ROLE
          where
            $role/NOM=$nom and $role/PRENOM=$prenom
          return
            $f/TITRE
        }
      </FILMS>
    </ACTEUR>
}
</ACTEURS>
```

```

=====
{-- Tous les films dans lesquels a joué le même couple d'acteurs --}
<LISTESFILMS>
{
  let
    $doc:=document("data/Films.xml"),
    $actlist:=(
      for
        $nom in distinct-values($doc//FILM/ROLES/ROLE/NOM),
        $prenom in distinct-
values($doc//FILM/ROLES/ROLE[NOM=$nom]/PRENOM)
      order by
        $nom,$prenom
      return
        <ACTEUR><NOM>{$nom}</NOM><PRENOM>{$prenom}</PRENOM></ACTEUR>
    )
  for
    $a1 in $actlist,
    $a2 in $actlist
  where
    $a1 << $a2
  return (
    let
      $lst:=(
        for
          $f1 in (
            for
              $f in $doc//FILM,
              $r in $f/ROLES/ROLE
            where
              $r/PRENOM=$a1/PRENOM and $r/NOM=$a1/NOM
            return
              $f
          ),
          $f2 in (
            for
              $f in $f1,
              $r in $f/ROLES/ROLE
            where
              $r/PRENOM=$a2/PRENOM and $r/NOM=$a2/NOM
            return
              $f
          )
        return
          <FILM>{$f2/TITRE}</FILM>
      )
    return (
      if (count($lst)>=2)
      then
        <LISTEFILMS>
          <COUPLEACTEUR>{$a1,$a2}</COUPLEACTEUR>
          <FILMS>
            {$lst}
          </FILMS>
        </LISTEFILMS>
      else ( )
    )
  )
}
</LISTESFILMS>

```