

Technologies XML (XML'11)

TD 4 : Requêtes XPath

Exercice 1 Pour la base de données "Dyskoteka" formuler des requêtes XPath pour :

- Afficher les titres des disques dont le prix est inférieur de 11 (il en y a 75)
`/diskotheque/disque[prix < 11]/titre/text()`
ou simplement
`//disque[prix < 11]/titre/text()`
- Afficher les titres des disques dont le prix est entre 11 et 12 (il en y a 91)
`//disque[prix >= 11 and prix <= 12]/titre/text()`
- Afficher les titres des disques sortis en 1999 (il y en a 24)
`//disque[annee = 1999]/titre/text()`
- Afficher les titres des disques sortis dans les années 90 (il y en a 292)
`//disque[annee >= 1990 and annee < 2000]/titre/text()`
- Afficher les titres des disques sortis entre 1995 et 1998 dont le prix est entre 11 et 13 (il y en a 28)
`//disque[annee >= 1995 and annee <= 1998 and prix >= 11 and prix <= 13]/titre/text()`
- Afficher les titres des disques endommagés (il y en a 173)
`//disque[etat="Endommage"]/titre/text()`
- Afficher les titres des disques endommagés, sortis en 1998 et dont le prix est inférieur à 13 (il y en a qu'un seul)
`//disque[etat="Endommage" and annee = 1998 and prix <= 13]/titre/text()`
- Afficher les titres des disques qui sont pas neufs (il y en a 631)
`//disque[etat != "Neuf"]/titre/text()`
- Afficher les titres des disques contenant des chansons Blues (il y en a 25)
`//disque[chansons/chanson/genre="Blues"]/titre/text()`
ou
`//disque[./chanson/genre="Blues"]/titre/text()`
ou encore
`//disque[./genre="Blues"]/titre/text()`
- Afficher les titres des disques contenant uniquement des chansons Blues (il y en a 20)
`//disque[not(./chanson/genre="Blues")]/titre/text()`
- Afficher les titres des chansons dont le genre est New Age (il y en a 112)
`//disque/chansons/chanson[genre="New Age"]/titre/text()`
ou
`//chanson[genre="New Age"]/titre/text()`
- Afficher les titres des chansons dont le genre est Jazz (il y en a 112)
`//chanson[genre="Jazz"]/titre/text()`
- Afficher les titres des chansons dont le genre est soit New Age soit Jazz (il y en a 224)
`//chanson[genre="Jazz"]/titre/text()`
- Afficher les noms des disques contenant la chanson "(I can't get no) Satisfaction" (il y en a 9)
`//disque[./chanson/titre="(I can't get no) Satisfaction"]/titre/text()`
ou `//chanson[titre="(I can't get no) Satisfaction"]/../../titre/text()`
- Afficher les noms de tout artiste qui interprète des chansons écrites par lui-même

```
//disque[artiste = chansons/chanson/ecrit]/artiste/text()
ou //ecrit[text() = ../../artiste/text()]/text()
```

16. Afficher le nombre total de disques (857)
`count(//disque)`
17. Afficher le nombre de disques endommagés (173)
`count(//disque[etat="Engommage"])`
18. Afficher la somme totale des prix de tous les disques (12868.3)
`sum(//disque/prix)`
19. Afficher la somme des prix de tous les disques détruits (646)
`sum(//disque[etat="Detruit"]/prix)`

Exercice 2 Pour les deux bases de données “Travaho” formuler des requêtes XPath pour (une requête pour chacune des bases de données) :

1. Afficher les noms de tous les employés
 - (a) `/travail/dept/employe/nom/text()` ou simplement `//employe/nom/text()`
 - (b) `/travail//employe/nom/text()` ou `//employe/nom/text()`
2. Afficher les noms de tous les employés dont la salaire est inférieure à 15000
 - (a) `//employe[salaire < 15000]/nom/text()`
 - (b) `//employe[salaire < 15000]/nom/text()`
3. Afficher les noms embauchés dans la première moitié de 2000
 - (a) Les dates ne sont pas bien formatées pour permettre à XPath 1.0 de formuler cette requête
 - (b) Les dates ne sont pas bien formatées pour permettre à XPath 1.0 de formuler cette requête
4. Afficher les noms des employés qui travaillent à Lille. Idem + ou à Marseille
 - (a) `//dept[ville = "Lille"]/employe/nom/text()`
 - (b) `//employe[departement/@ville = "Lille"]/nom/text()`
5. Afficher les noms des employés qui travaillent à Paris et gagnent moins de 20000
 - (a) `//dept[ville = "Paris"]/employe[salaire < 20000]/nom/text()`
 - (b) `//employe[departement/@ville = "Lille" and salaire < 20000]/nom/text()`
6. Afficher les noms des employés qui travaillent à Paris où à Lille et qui gagnent moins de 20000
 - (a) `//dept[ville = "Paris" or ville="Lille"]/employe[salaire < 20000]/nom/text()`
 - (b) `//employe[departement[@ville = "Lille" or @ville="Paris"] and salaire < 20000]/nom/text()`
7. Afficher les noms des employés qui travaillent à Paris et gagnent moins de 20000 où qui travaillent à Lille
 - (a) `//dept[ville = "Paris"]/employe[salaire < 20000]/nom/text() | //dept[ville = "Lille"]/employe/nom/text()`
La manière maligne de poser cette requête suit :
`//dept/employe[(../ville = "Paris" and salaire < 20000) or ../ville="Lille"]/nom/text()`
 - (b) `//employe[(departement/@ville = "Paris" and salaire < 20000) or (departement/@ville = "Lille")]/nom/text()`
8. Afficher les noms de tous les départements.
 - (a) `//dept/nom/text()`
 - (b) `//departement/nom/text()`
9. Afficher les noms de tous employés qui sont responsable directement de plus de deux personnes.
 - (a) Impossible de poser cette requête

- (b) `//employe[count(employe) > 2]/nom/text()`
- 10. Afficher les noms de tous employés qui sont responsable (directement où indirectement) de plus de trois personnes.
 - (a) Impossible de poser cette requête
 - (b) `//employe[count(../employe) > 3]/nom/text()`