

Master 2 Humanités numériques – Bases de données semi-structurées TD 4 : XQuery – Requêtes XPath

J. Darmont – <https://eric.univ-lyon2.fr/jdarmont/>

Préambule

Soit un catalogue d'ouvrages stocké dans un document XML dont la structure hiérarchique est donnée ci-dessous. @ marque un attribut, * indique une multiplicité « plusieurs » et ? une multiplicité « zéro ou un »

```
catalog
  book*
    @id
    author
    title
    genre
    price
    publish_date
    description
    onsale?
```

1. Télécharger le document XML suivant et le stocker localement.
<https://eric.univ-lyon2.fr/jdarmont/docs/books.xml>
2. Lancer BaseX depuis le menu Démarrer de Windows ou grâce au paquetage JAR disponible à l'adresse suivante.
<https://basex.org/download/>
3. Créer une nouvelle base de données uniquement avec le document books.xml (menu Database/New).

Exercice 1

Les requêtes XPath doivent être écrites dans l'éditeur (fenêtre en haut au centre, par défaut). Pour exécuter, une requête, utiliser les touches **Ctrl + Entrée** ou cliquer sur le triangle vert. Formuler les requêtes suivantes à l'aide d'expressions de chemin uniquement ! Pas de requêtes FLWOR pour l'instant.

1. Éléments *book* du document books.xml. Spécifier le chemin complet des éléments *book*.
2. Même question sans spécifier le chemin complet des éléments *book*.
3. Titres de tous les livres.

4. Identifiants de tous les livres.
5. Caractéristiques du 4^e livre.
6. Titres des 5 premiers livres.
7. Titres des livres dont le genre est « Computer ».
8. Livres de genre « Computer » et coûtant moins de 40 €.
9. Livres actuellement en solde (c'est-à-dire, qui possèdent un élément *onsale*).
10. Titres et descriptions des livres écrits par Eva Corets.
11. Prix moyen des livres.
12. Titres des livres publiés en 2001.
13. Livre le plus cher du catalogue.

Exercice 2

Télécharger l'extrait de données d'enchères réelles à l'adresse ci-dessous, créer dans BaseX une nouvelle base de données avec ce document et visualiser sa structure.

<https://eric.univ-lyon2.fr/jdarmont/docs/auctions.xml>

Formuler ensuite les requêtes suivantes avec XPath.

1. Nombre d'enchères (*listing*).
2. Nombre d'enchérisseurs (*bidder_name*).
3. Nombre d'enchérisseurs distincts.
4. Noms et évaluations (*seller_rating*) de tous les vendeurs (*seller_name*).
5. Évaluation la plus basse (ne pas tenir compte des nouveaux vendeurs).
6. Noms des vendeurs ayant l'évaluation la plus basse.
7. Objets avec un processeur (CPU) Celeron.
8. Nombre moyen d'enchères (*num_bids*).
9. Information à propos des objets sans élément mémoire (*memory*) spécifié.
10. Plus haute enchère (*current_bid*).

Correction Exercice 1

(: 1 :)

```
doc("books.xml")/catalog/book
```

(: 2 :)

```
//book
```

(: 3 :)

```
//title
```

(: 4 :)

```
//data(@id)
```

(: 5 :)

```
//book[4]/*
```

(: 6 :)

```
//book[position() <= 5]/title
```

(: 7 :)

```
//book[genre = "Computer"]/title
```

(: 8 :)

```
//book[genre = "Computer" and price < 40]
```

(: 9 :)

```
//book[onsale]
```

(: 10 :)

```
//book[author = "Corets, Eva"]/title | //book[author = "Corets, Eva"]/description
```

(: 11 :)

```
avg(//price)
```

(: 12 :)

```
//book[year-from-date(publish_date) = 2001]/title
```

(: 13 :)

```
//book[price=max(//price)]
```

Correction Exercice 2

(: 1 :)

```
count(//listing)
```

(: 2 :)

```
count(//bidder_name)
```

(: 3 :)

count(distinct-values(//bidder_name))

(: 4 :)

//seller_info

(: 5 :)

min(//seller_info[normalize-space(seller_rating) != "new"]/seller_rating)

(: 6 :)

//seller_info[normalize-space(seller_rating) != "new" and seller_rating =
min(//seller_info[normalize-space(seller_rating) != "new"]/seller_rating)]/seller_name

(: 7 :)

//item_info[contains(cpu, "Celeron")]

(: 8 :)

avg(//num_bids)

(: 9 :)

//item_info[memory = ""]

(: 10 :)

max(//auction_info/number(replace(substring-after(current_bid, "\$"), ",", "")))