M1 Informatique – Année 2018-2019
Bases de données avancées
TD n° 6 (XQuery) : Requêtes XPath
J. Darmont (http://eric.univ-lyon2.fr/~jdarmont/), 27/08/18

Exercice 1 : Création de base de données XML sous BaseX

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.
   http://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.
   http://basex.org/products/download/all-downloads/
3. Créer une nouvelle base de données uniquement avec le document books.xml (menu Database/New).

Exercice 2 : Requêtes XPath

Créer un nouveau fichier dans la fenêtre de droite de BaseX pour écrire une requête. Exécuter la requête grâce à CTRL + ENTER ou en cliquant sur la flèche verte. 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 4e 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 3 : Plus de requêtes XPath !

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.

http://eric.univ-lyon2.fr/~jdarmont/docs/auctions.xml

Formuler ensuite les requêtes suivantes avec XPath.

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

[1] Utiliser la fonction `normalize-space()`.

Correction

```
(: Exercice 2 :)

(: 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)]


(: Exercice 3 :)

 (: 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[normalize-space(memory) = ""]

(: 10 :)
max(//auction_info/number(normalize-space(replace(substring-after(current_bid,
"$"), ",", ""))))
```