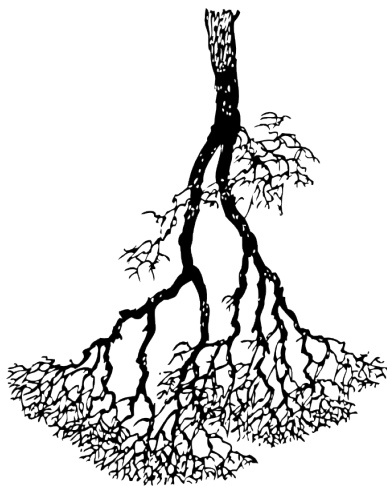




DEPARTEMENT INFORMATIQUE
DE LA FACULTE DES SCIENCES

Quentin Yeché (21520370), Yanis Allouch (21708237)

Rapport du TP N°4 : XPath & XQuery



HMIN103 — Données du web

Référent: Federico Ulliana et Pierre Pompidor

2020

Table des matières

1	XQuery : Trains (suite)	3
2	XMark	5

Introduction

L'analyse et le traitement des données consistent à les étudier afin d'en extraire des structures capables des les stocker ainsi que de les manipuler de façon la plus optimale.

L'objectif de ce quatrième TP est de se familiariser avec XQuery et l'utilisation de l'outil [BaseX](#).

Ce TP traitant de XQuery, est composé de deux exercices. Le premier concerne la base de donnée *Trains*, précédemment travaillé dans le TP 3. Le second exercice travaille sur la base *Site* (Auction).

Le TP se déroule sur papier et machine.

Durant ce TP, nous appliquerons les règles qui seront abordées dans le cours référencé [DdW3-XQuery](#).

1 XQuery : Trains (suite)

Considérez le document XML suivant, représentant des informations ferroviaires.

```
1 <gare>
2   <train numero="t5560" type="TGV">
3     <voiture numero="v1">
4       <resa numero="r17" id="u55"/>
5       <resa numero="r18" id="u52"/>
6     </voiture>
7     <voiture numero="v2"/>
8     <voiture numero="v3"/>
9     <voiture numero="v4">
10      <bar service="froid uniquement"/>
11    </voiture>
12  </train>
13  <train numero="t6731">
14    <voiture numero="v1"/> 2
15    <voiture numero="v2">
16      <resa numero="r15" id="u55"/>
17    </voiture>
18  </train>
19  <usager id="u55" nom="Jean" prenom="Dufour"/>
20  <usager id="u52" nom="Brigitte" prenom="Lefebvre"/>
21  <usager id="u56" nom="Patrick" prenom="Subiran"/>
22 </gare>
```

Donner des expressions XQuery pour les requêtes suivantes.

1. Le numéro des trains possédant une voiture-bar.

```
1 for $x in //train/@numero return $x[../descendant::bar]
```

Ou bien

```
1 for $train in /gare/train
2 where $train//bar
3 return $train/@numero
```

2. Le nom des usagers ayant au moins une réservation.

```
1 distinct-values(for $resa in //resa/@id
2 where (for $usager in //usager/@id return $usager) = $resa
3 return $resa)
```

Ou bien

```
1 for $u in //usager
2 where $u/@id=//resa/@id
3 return $u/@nom
```

3. La réservation avec le plus grand identifiant (dans l'ordre lexicographique).

```
1 for $resa in //resa
2 where $resa/@numero/substring(.,2) = max(//resa/@numero/substring(.,2))
3 return $resa
```

4. Le numéro des trains dont au moins 2 places sont réservées.

```
1 for $train in //train
2 where count($train//resa)>=2
3 return $train
```

5. Le nom des personnes ayant réservé exactement deux fois.

```
1 for $id in //usager/@id
2 where count(//resa[@id=$id])=2
3 return $id/ancestor::usager
```

6. Les usagers n'ayant effectué aucune réservation.

```
1 for $id in //usager/@id
2 where not(count(//resa[@id=$id]))
3 return $id/ancestor::usager
```

2 XMark

Donner les expressions XQuery pour les requêtes XMark disponibles [ici](#).

1. All the items.

```
1 for $item in //item
2 return $item
```

2. The keywords in annotations of closed auctions.

```
1 for $keyword in //closed_auctions//annotation//keyword
2 return $keyword
```

3. All the keywords.

```
1 for $keyword in //keyword
2 return $keyword
```

4. The keywords in a paragraph item.

```
1 for $keyword in //keyword
2 where $keyword/ancestor::parlist
3 return $keyword
```

5. The (either North or South) American items.

```
1 for $region in //regions/child::*
2 where $region/name() = ("namerica","samerica")
3 return $region//item
```

6. The paragraph items containing a keyword.

```
1 for $paragraph in //parlist
2 where $paragraph//keyword
3 return $paragraph
```

7. The mails containing a keyword.

```
1 for $mail in //mail
2 where ($mail//keyword)
3 return $mail
```

8. The open auctions in which a certain person issued a bid before another person.

Les dates et heures étant stockées de dans des éléments différents, et le formatage des dates n'étant pas standard, nous avons décidé d'implémenter quelques fonctions pour gérer la comparaison de dates. La fonction `bidDate` prend en argument un élément bidder et renvoie le `dateTime` correspondante. La fonction `bidDateSeq` prend en argument une séquence de bidder et renvoie la séquence de `dateTime` correspondante. Le type `dateTime` peut ensuite être utilisé dans les comparaisons avec l'opérateur `<` usuel ou même les fonctions *min* et *max*.

```

1 (:
2 different dates
3 open_auction36
4 let $idp1:="person140"
5 let $idp2:="person59"
6
7 same dates, different times
8 open_auction27
9 let $idp1:="person33"
10 let $idp2:="person25"
11 :)
12
13 declare function local:bidDate($bidder as node()) as xs:dateTime{
14   let $jour:= $bidder/date/text()
15   let $tokens:= fn:tokenize($jour, '/')
16   let $jour:= concat($tokens[3], "-", $tokens[1], "-", $tokens[2])
17
18   let $heure := $bidder/time/text()
19
20   let $date:=xs:dateTime(concat($jour, "T", $heure))
21   return $date
22 };
23
24 declare function local:bidDateSeq($bidder as node()*) as (xs:dateTime)* {
25   let $ret := if ($bidder)
26   then (local:bidDate($bidder[1]), local:bidDateSeq($bidder[position() =
27     (2 to last())]))
28   else ()
29   return $ret
30 };

```

La réponse à la question 8 est donc

```

1 let $idp1:="person33"
2 let $idp2:="person25"
3 for $auction in //open_auction
4 where ($auction//bidder//@person=$idp1
5   and ($auction//bidder//@person=$idp2)
6   and local:bidDate($auction//bidder[//@person=$idp1])
7     <local:bidDate($auction//bidder[//@person=$idp2])
8 return $auction

```

9. The past bidders of a given open auction.

```

1 let $id:="open_auction36"
2 let $lastBid:= max(//open_auction[@id=$id]//local:bidDateSeq(bidder))
3 for $bidder in //open_auction[@id=$id]//bidder
4 where local:bidDate($bidder) < $lastBid
5 return ($bidder)

```

10. The items that follow, in document order, a given item.

```

1 let $idItem := "item210"
2 return //item[@id=$idItem]/following::item

```

11. The text nodes that are contained in the keywords of the description element of a given item.

```

1 let $idItem := "item210"
2 for $item in //item
3   for $keyword in $item//description//keyword

```

```

4 where $item/@id=$idItem
5 return $keyword/text()

```

12. People having an address and either a phone or a homepage.

```

1 (:considering "either or" to mean OR, not XOR:)
2 for $pers in //people/person
3 where $pers/address and ($pers/homepage or $pers/phone)
4 return $pers

```

13. People having no homepage.

```

1 for $pers in //people/person
2 where not($pers/homepage)
3 return $pers

```

14. The initial and last bidder of all open auctions.

```

1 for $auction in //open_auction
2   let $id:=$auction/@id
3   let $lastBid:= max($auction//local:bidDateSeq(bidder))
4   let $firstBid:= min($auction//local:bidDateSeq(bidder))
5   for $bidder in $auction//bidder
6 where (local:bidDate($bidder) = ($lastBid,$firstBid))
7 return ($bidder)

```

15. The open auctions having more than 5 bidders.

```

1 for $auction in //open_auction
2 where count($auction/bidder)>=5
3 return $auction/bidder

```

16. Mails sent in September.

```

1 for $mail in //mail
2 where $mail/date/substring(text(),1,2)="09"
3 return $mail

```

17. The items whose description contains the word 'gold'.

```

1 for $item in //item
2 where $item//description/contains(., "gold")
3 return $item

```