Master 2 Humanités numériques – Bases de données semi-structurées

TD 7 : XQuery – Travail de synthèse shakespearien

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

Exercice 1

1. Télécharger le document XML hamlet.xml à l'URL ci-dessous. Le copier/coller sous le nom hamlet+dtd.xml.

   https://eric.univ-lyon2.fr/jdarmont/docs/hamlet.xml

2. Intégrer dans hamlet+dtd.xml une DTD à même de représenter le schéma de données du document XML.

3. Utiliser le validateur en ligne https://www.xmlvalidation.com . Copier/coller ou téléverser le fichier hamlet+dtd.xml puis valider. En cas d'erreur, corriger la DTD jusqu'à ce que le document XML soit valide.

Exercice 2

Sur la base du document XML hamlet.xml, formuler les requêtes suivantes en XQuery.

1. Titres (TITLE) de toutes les scènes (SCENE).

2. Noms de tous les personnages (PERSONA) par ordre alphabétique, y compris ceux qui sont dans un groupe (PGROUP).

3. Noms de tous les locuteurs (SPEAKER) uniques par ordre alphabétique. Un commentaire ?

4. Toutes les répliques du locuteur Hamlet.

5. Nombre d'actes (ACT), de scènes, de répliques (LINE) et d'indications de mise en scène (STAGEDIR).

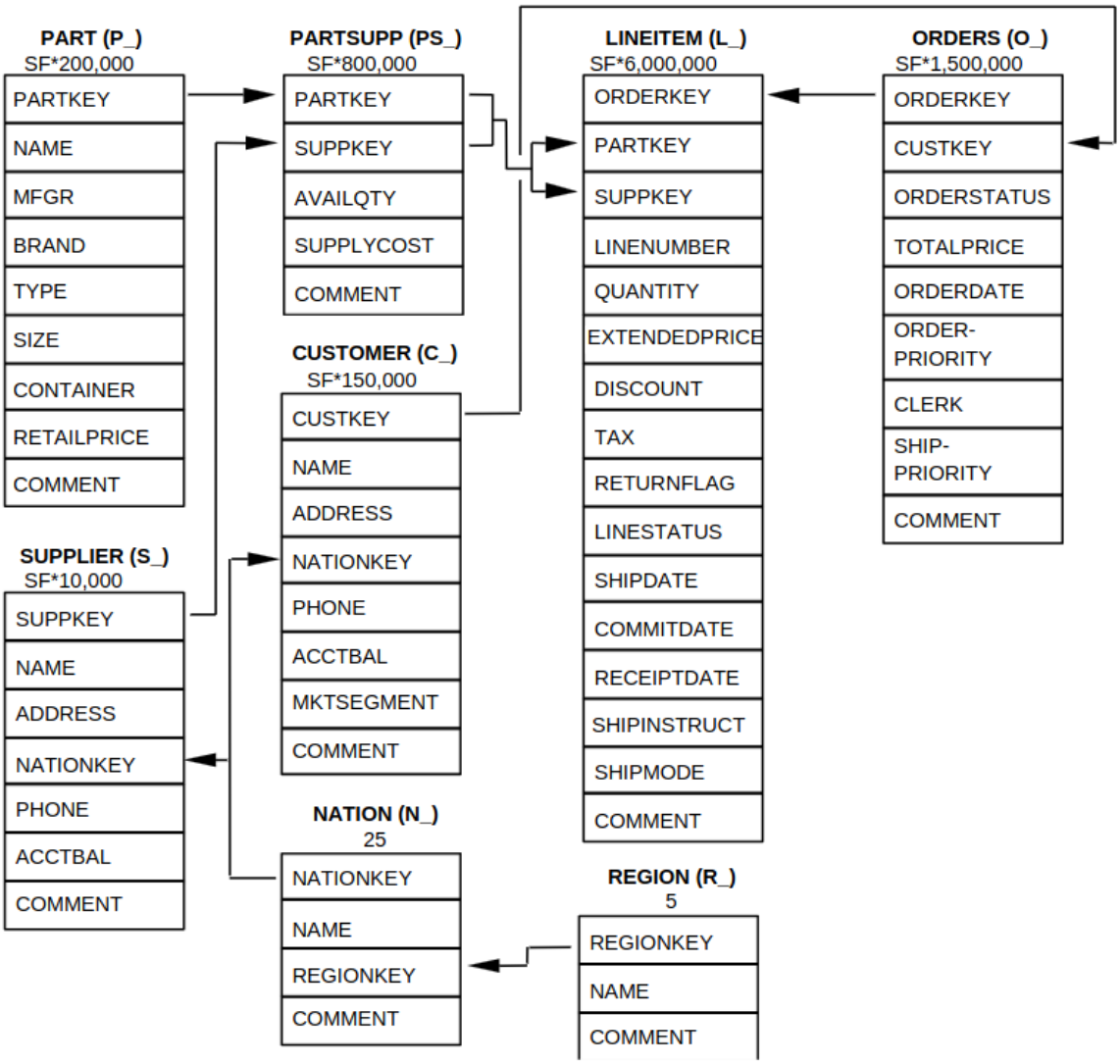
6. Nombre de répliques par scène (indiquer le titre de la scène).

7. Nombre moyen de répliques par acte.

Exercice 3

Soit le schéma du banc d'essais TPC-H[1] exprimé en XML dans les documents suivants.

- part.xml : pièces
- supplier.xml : fournisseurs
- partsupp.xml : document « pont » entre les produits et les fournisseurs
- customer.xml : clients
- orders.xml : commandes
- lineitem.xml : lignes de facturation
- nation.xml : pays
- region.xml : continents



Dans tous ces documents, les éléments sont préfixés par l'initiale du nom du document.

Chaque document a comme racine l'élément *table*, assorti d'un attribut ID qui indique le nom du document.

Dans tous les documents, les éléments qui contiennent les données sont encadrés par les balises <T> </T> (pour l'élément *tuple* en anglais, n-uplet en français).

1 https://www.tpc.org/tpch/

L'élément qui vient immédiatement après l'élément T est systématiquement un **identifiant**.

Enfin, les flèches du diagramme figurent une **référence** depuis un document vers un autre. Un élément dans le document « source » et l'élément référencé dans le document « destination » ont le même nom, mis à part leur préfixe.

Téléchargement des documents : https://eric.univ-lyon2.fr/jdarmont/docs/TPC-H.zip

Sur cette base (de données !), formuler les requêtes suivantes en XQuery.

1. Noms de toutes les régions.

2. Nombre de n-uplets de chaque document XML.

3. Lignes de facturation dont la quantité multipliée par le prix, remise et taxe comprises : L_QUANTITY × L_EXTENDEDPRICE × (1 – L_DISCOUNT) × (1 + L_TAX), dépassent $5 000 000. Vérifier.

4. Noms des clients et de leur pays.

5. Noms des continents (par ordre alphabétique) et de leurs fournisseurs.

6. Nombre de pièces par marques (P_BRAND), triées par nombre de pièces décroissantes.

7.1. Somme des coûts d'approvisionnement (PS_SUPPLYCOST) et des quantités de pièces disponibles (PS_AVAILQTY) par fournisseur (S_NAME).

7.2. Sur la base de la question 7.1 : somme des coûts d'approvisionnement (PS_SUPPLYCOST) et des quantités de pièces disponibles (PS_AVAILQTY) par pays (N_NAME).

7.3. Sur la base de la question 7.2 : somme des coûts d'approvisionnement (PS_SUPPLYCOST) et des quantités de pièces disponibles (PS_AVAILQTY) par continent (R_NAME).

Passer d'un niveau fin (fournisseur) à des niveaux plus agrégés (pays, puis continent) permet de « dézoomer » les données et vice-versa pour « zoomer ». Plus rigoureusement, cela s'appelle un forage vers le haut (resp. vers le bas), *roll-up* et *drill-down* en anglais.

Correction Exercice 1

```
<!ELEMENT PLAY (TITLE, FM, PERSONAE, SCNDESCR, PLAYSUBT, ACT*)>
    <!ELEMENT TITLE (#PCDATA)>
        <!ATTLIST TITLE AUTHOR CDATA #IMPLIED>
    <!ELEMENT FM (P*)>
        <!ELEMENT P (#PCDATA)>
    <!ELEMENT PERSONAE (TITLE, (PERSONA | PGROUP)*)>
        <!ELEMENT PERSONA (#PCDATA)>
        <!ELEMENT PGROUP (PERSONA*, GRPDESCR)>
            <!ELEMENT GRPDESCR (#PCDATA)>
    <!ELEMENT SCNDESCR (#PCDATA)>
    <!ELEMENT PLAYSUBT (#PCDATA)>
    <!ELEMENT ACT (TITLE, SCENE*)>
        <!ELEMENT SCENE (TITLE, (STAGEDIR | SPEECH)*)>
        <!ELEMENT SPEECH (SPEAKER*, (LINE | STAGEDIR)*)>
            <!ELEMENT SPEAKER (#PCDATA)>
            <!ELEMENT LINE (#PCDATA | STAGEDIR)*>
            <!ELEMENT STAGEDIR (#PCDATA)>
```

Correction Exercice 2

```
(: 1 :)
for $title in //SCENE/TITLE
return $title

(: 2 :)
for $persona in //PERSONA
order by $persona
return $persona

(: 3 :)
for $speaker in distinct-values(//SPEAKER)
order by $speaker
return $speaker

(: 4 :)
for $speech in //SPEECH
where $speech/SPEAKER = "HAMLET"
return $speech/LINE
```

```
(: 5 :)
let    $act := count(//ACT),
       $scene := count(//SCENE),
       $line := count(//LINE),
       $stagedir := count(//STAGEDIR)
return      <res>
                  <acts>{$act}</acts>
                  <scenes>{$scene}</scenes>
                  <lines>{$line}</lines>
                  <stagedirs>{$stagedir}</stagedirs>
            </res>

(: 6 :)
for $scene in //SCENE
group by $title := $scene/TITLE
let $linecount := count($scene//LINE)
return      <scene title="{$title}">
                  <linecount>{$linecount}</linecount>
            </scene>

(: 7 :)
let $average := avg(
       for $act in //ACT
       group by $title:= $act/TITLE
       let $linecount := count($act//LINE)
       return      <act title="{$title}">
                         <linecount>{$linecount}</linecount>
                  </act>)
return <average-lines-per-act>{$average}</average-lines-per-act>
```

Correction Exercice 3

```
(: 1 :)
for $r in /table[@ID = "region"]/T
return <region>{data($r/R_NAME)}</region>

(: 2 version force brute :)
let $p := count(/table[@ID="part"]/T)
let $s := count(/table[@ID="supplier"]/T)
let $ps := count(/table[@ID="partsupp"]/T)
let $c := count(/table[@ID="customer"]/T)
let $o := count(/table[@ID="orders"]/T)
let $l := count(/table[@ID="lineitem"]/T)
let $n := count(/table[@ID="nation"]/T)
let $r := count(/table[@ID="region"]/T)
return      <res>
                  <part>{$p}</part>
                  <supplier>{$s}</supplier>
                  <partsupp>{$ps}</partsupp>
                  <customer>{$c}</customer>
```

```
                <orders>{$o}</orders>
                <lineitem>{$l}</lineitem>
                <nation>{$n}</nation>
                <region>{$r}</region>
        </res>

(: 2 version algorithmique :)
for $i in /table/@ID
let $c := count(/table[@ID = $i]/T)
order by number($c) descending (: optionnel, mais bien utile :)
return <res><id>{data($i)}</id><compte>{$c}</compte></res>

(: 3 :)
for $l in /table[@ID = "lineitem"]/T
let $totalprice :=
        $l/L_QUANTITY * $l/L_EXTENDEDPRICE * (1 - $l/L_DISCOUNT) * (1 + $l/L_TAX)
where $totalprice > 5000000
return $l

(: 4 :)
for     $c in /table[@ID = "customer"]/T,
        $n in /table[@ID = "nation"]/T
where $c/C_NATIONKEY = $n/N_NATIONKEY
return      <res>
                <client>{data($c/C_NAME)}</client>
                <pays>{data($n/N_NAME)}</pays>
            </res>

(: 5 :)
for     $s in /table[@ID = "supplier"]/T,
        $n in /table[@ID = "nation"]/T,
        $r in /table[@ID = "region"]/T
where       $s/S_NATIONKEY = $n/N_NATIONKEY
and         $n/N_REGIONKEY = $r/R_REGIONKEY
order by $r/R_NAME
return      <res>
                <continent>{data($r/R_NAME)}</continent>
                <fournisseur>{data($s/S_NAME)}</fournisseur>
            </res>

(: 6 :)
for $p in /table[@ID = "part"]/T
group by $g := $p/P_BRAND
let $c := count($p/P_PARTKEY)
order by number($c) descending
return      <res marque="{$g}">
                <nb_pieces>{$c}</nb_pieces>
            </res>
```

```
(: 7.1 :)
for    $ps in /table[@ID = "partsupp"]/T,
       $s in /table[@ID = "supplier"]/T
where $s/S_SUPPKEY = $ps/PS_SUPPKEY
group by $g := $s/S_NAME
let    $cost := sum($ps/PS_SUPPLYCOST),
       $qty := sum($ps/PS_AVAILQTY)
return      <res fournisseur="{$g}">
                  <totalcost>{$cost}</totalcost>
                  <totalqty>{$qty}</totalqty>
            </res>

(: 7.2 :)
for    $ps in /table[@ID = "partsupp"]/T,
       $s in /table[@ID = "supplier"]/T,
       $n in /table[@ID = "nation"]/T
where       $s/S_SUPPKEY = $ps/PS_SUPPKEY
and         $s/S_NATIONKEY and $n/N_NATIONKEY
group by $g := $n/N_NAME
let    $cost := sum($ps/PS_SUPPLYCOST),
       $qty := sum($ps/PS_AVAILQTY)
return      <res pays="{$g}">
                  <totalcost>{$cost}</totalcost>
                  <totalqty>{$qty}</totalqty>
            </res>

(: 7.3 :)
for    $ps in /table[@ID = "partsupp"]/T,
       $s in /table[@ID = "supplier"]/T,
       $n in /table[@ID = "nation"]/T,
       $r in /table[@ID = "region"]/T
where       $s/S_SUPPKEY = $ps/PS_SUPPKEY
and         $s/S_NATIONKEY and $n/N_NATIONKEY
and         $n/N_REGIONKEY = $r/R_REGIONKEY
group by $g := $r/R_NAME
let    $cost := sum($ps/PS_SUPPLYCOST),
       $qty := sum($ps/PS_AVAILQTY)
return      <res continent="{$g}">
                  <totalcost>{$cost}</totalcost>
                  <totalqty>{$qty}</totalqty>
            </res>
```