XML – séance 10 Correctif

Soit le fichier play.dtd validant des fichiers xml de pièce de théâtre. Un exemple de pièce est donné dans le fichier hamlet.xml.

1. Pour chacune des requêtes XQuery suivantes, expliquez-les en français.

a) **let** $s:=doc("hamlet.xml")

//SPEECH[LINE="To be, or not to be: that is the question:"]

**return** <**result**>

{count($s/following-sibling::SPEECH[SPEAKER=$s/SPEAKER]/LINE)}

</**result**>

*(: Le nombre de ligne prononcée par l'auteur de "to be or not to be" après celle-ci :)*

b) (**for** $s **in** doc("hamlet.xml")//SPEECH

**let** $c:= count($s/LINE)

**order** **by** $c **ascending**

**return** <**res**>{$s/LINE}</**res**>)[last()]

*(:Donnez les lignes du Speech qui contient le plus grand nombre de lignes:)*

1. Requêtes XQuery
2. Donnez pour chaque speaker le nombre de lignes qu'il dit. Le résultat doit être trié par le nombre de lignes dites. La sortie devrait ressembler à cela :

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<result>

<nombre\_de\_ligne>

<role>HAMLET</role>

<nombre>1495</nombre>

</nombre\_de\_ligne>

<nombre\_de\_ligne>

<role>KING CLAUDIUS</role>

<nombre>550</nombre>

</nombre\_de\_ligne>  
…

</result>

<result>

{**for** $s **in** distinct-values(doc("hamlet.xml")//SPEAKER)

**let** $c:= count(doc("hamlet.xml")//SPEECH[SPEAKER=$s]/LINE)

**order** **by** $c **descending**

**return**

<**nombre\_de\_ligne**>

<role>{$s}</role>

<nombre>{$c}</nombre>

</**nombre\_de\_ligne**>

}</result>

1. Même question mais en n'affichant uniquement les speakers qui disent plus de 100 lignes

<result>{

**for** $s **in** distinct-values(doc("hamlet.xml")//SPEAKER)

**let** $c:= count(doc("hamlet.xml")//SPEECH[SPEAKER=$s]/LINE)

**order** **by** $c **descending**

**return**

<**nombre\_de\_ligne**>

<role>{$s}</role>

<nombre>{$c}</nombre>

</**nombre\_de\_ligne**>[nombre>100]

}</result>

Ou

<result>

{for $s in distinct-values(doc("hamlet.xml")//SPEAKER)

let $c:= count(doc("hamlet.xml")//LINE[preceding-sibling::SPEAKER=$s])

order by $c descending

return if ($c>100) then

<nombre\_de\_ligne>

<role>{$s}</role>

<nombre>{$c}</nombre>

</nombre\_de\_ligne>else()}

</result>

La version XPath est préférée!

1. Quelles sont les différentes phrases dites par Hamlet juste après des interventions de Marcellus ?

La sortie doit ressembler à cela :

<result>

<LINE>I am very glad to see you. Good even, sir.</LINE>

<LINE>But what, in faith, make you from Wittenberg?</LINE>

<LINE>Did you not speak to it?</LINE>

<LINE>Arm'd, say you?</LINE>

<LINE>From top to toe?</LINE>

<LINE>Then saw you not his face?</LINE>

<LINE>Hold off your hands.</LINE>

<LINE>How say you, then; would heart of man once think it?</LINE>

<LINE>But you'll be secret?</LINE>

<LINE>There's ne'er a villain dwelling in all Denmark</LINE>

<LINE>But he's an arrant knave.</LINE>

<LINE>Nay, but swear't.</LINE>

<LINE>Upon my sword.</LINE>

<LINE>Indeed, upon my sword, indeed.</LINE>

</result>

<result>

{for $s in doc("hamlet.xml")//SPEECH[SPEAKER="MARCELLUS"]/following-sibling::SPEECH[1][SPEAKER="HAMLET"]/LINE

return $s}

</result>

Ou

**for** $s **in** doc("hamlet.xml")//SPEECH[SPEAKER="MARCELLUS"]

**return** $s/following-sibling::SPEECH[1][SPEAKER="HAMLET"]/LINE

1. Pour chaque acte, donnez les différents speakers. La sortie devrait ressembler à cela :

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<ACTE>

<TITLE>ACT I</TITLE>

<SPEAKERS>

<SPEAKER>BERNARDO</SPEAKER>

<SPEAKER>FRANCISCO</SPEAKER>

<SPEAKER>HORATIO</SPEAKER>

…

</SPEAKERS>

…

</ACTE>

**for** $a **in** doc("hamlet.xml")//ACT

**return** <**ACTE**>{$a/TITLE}<**SPEAKERS**>{

**for** $s **in** distinct-values($a//SPEAKER) **return** <**SPEAKER**>{$s}</**SPEAKER**>

}</**SPEAKERS**>

</**ACTE**>

1. Affichez le nombre d'actes dans lequel apparait chaque SPEAKER (trié par nombre décroissant d’acte). Le résultat devrait ressembler à cela :

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<speakers>

<speaker>

<nom>KING CLAUDIUS</nom>

<nbActe>5</nbActe>

</speaker>

…

</speakers>

<**speakers**>{

**for** $sp **in** distinct-values(doc("hamlet.xml")//SPEAKER)

**let** $c:=count(distinct-values(doc("hamlet.xml")//ACT[SCENE//SPEAKER=$sp]))

**order** **by** $c **descending**

**return** <**speaker**><**nom**>{$sp}</**nom**><**nbActe**>{$c}</**nbActe**></**speaker**>}

</**speakers**>

1. Affichez les paires d’actes pour lesquels participent un même nombre de speakers. Le résultat devrait ressembler à cela

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<actes>

<acte1>

<TITLE>ACT III</TITLE>

</acte1>

<acte2>

<TITLE>ACT IV</TITLE>

</acte2>

<nb>15</nb>

</actes>

**for** $a1 **in** doc("hamlet.xml")//ACT,

$a2 **in** doc("hamlet.xml")//ACT

**where** $a1 <$a2 **and** count(distinct-values($a1//SPEAKER))= count(distinct-values($a2//SPEAKER))

**return** <**actes**>

<**acte1**>{$a1/TITLE}</**acte1**>

<**acte2**>{$a2/TITLE}</**acte2**>

<**nb**>{count(distinct-values($a1//SPEAKER))}</**nb**>

</**actes**>

1. Donnez les speakers qui parlent dans tous les actes où parle GUILDENSTERN. Le résultat devrait ressembler à cela :

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<speakers>

<speaker>KING CLAUDIUS</speaker>

<speaker>HAMLET</speaker>

<speaker>QUEEN GERTRUDE</speaker>

<speaker>OPHELIA</speaker>

<speaker>ROSENCRANTZ</speaker>

</speakers>

<**speakers**>{

**for** $x **in** (distinct-values(doc("hamlet.xml")//SPEAKER))

**where** **every** $y **in** (doc("hamlet.xml")//ACT[SCENE//SPEAKER="GUILDENSTERN"])

**satisfies** ($y[SCENE//SPEAKER=$x]) **and** $x!="GUILDENSTERN"

**return** <**speaker**>{$x}</**speaker**>}  
</**speakers**>