Exercice 1

Question 1

declare variable $bib := "biblio.xml" ;

<bib>

{

for $b in doc($bib)//book

where $b/publisher = "Addison-Wesley" and

$b/@year > 1991

return

<book year="{$b/@year}">

{$b/title}

</book>

}

</bib>

**réponse:**

la requete genere du xml qui contient les titres de livres publiés par

Addison-Wesley vers les années > 1991

let $bib := "biblio.xml"

return

<results>

{

for $b in doc($bib)//book,

$t in $b/title,

$a in $b/author

return

<result>

{$t,$a}

</result>

}

</results>

**réponse:**

la requete renvoie le titre et autheur de livres

let $bib := "biblio.xml"

return

<results> {

let $a := doc($bib)//author

for $last in distinct-values($a/last),

$first in distinct-values($a[last=$last]/first)

order by $last, $first

return

<result><author> <last>{$last}</last>

<first>{$first}</first> </author>

{ for $b in doc($bib)/bib/book

where some $ba in $b/author

satisfies ($ba/last = $last and

$ba/first=$first)

return $b/title }

</result> }

</results>

Réponse :

elle crée un xml qui liste pour chaque auteur son nom et prénom ainsi qu'une liste de titre de livre qu'il a écrit seul ou en collaboration.

declare function local:books-by-author ($root, $last, $first) {

for $b in $root/book

where some $ba in $b/author

satisfies ($ba/last = $last and

$ba/first=$first)

return $b/title

};

let $bib := "biblio.xml"

return

<results> {

let $a := doc($bib)//author

for $last in distinct-values($a/last),

$first in distinct-values($a[last=$last]/first)

order by $last, $first

return <result>

<author> <last>{$last}</last>

<first>{$first}</first> </author>

{local:books-by-author(doc($bib)/bib, $last, $first)}

</result>

}

</results>

Réponse :

Même chose que la precedente par contre avec une fonction

let $bib := "biblio.xml"

return

<bib>

{

for $b in doc($bib)//book

where count($b/author) > 0

return

<book>

{$b/title}

{ for $a in $b/author[position() <= 2]

return $a }

{ if (count($b/author) > 2)

then <et-al/> else () }

</book>

}

</bib>

Réponse :

Requete renvoie les titres de livres de deux premiers auteurs

Et s’il y a plus de deux auteurs un <et-al/> est ajouté.

declare function local:books-by-author ($root, $last, $first) {

for $b in $root/book

where some $ba in $b/author

satisfies ($ba/last = $last and

$ba/first=$first)

return $b/title

};

let $bib := "biblio.xml"

return

<results> {

let $a := doc($bib)//author

for $last in distinct-values($a/last),

$first in distinct-values($a[last=$last]/first)

order by $last, $first

return <result>

<author> <last>{$last}</last>

<first>{$first}</first> </author>

<number> {count(local:books-by-author(doc($bib)/bib,$last, $first))} </number>

</result>

}

</results>

Réponse :

Requete revoie le nombre de livres par auteur

let $bib := "biblio.xml"

return

<data> {

for $year in distinct-values(doc($bib)//book/@year)

let $avg := avg(doc($bib)//book[@year=$year]/price/text())

return <year value="{$year}" avgprice="{$avg}"/>

}

</data>

Réponse :

La requete renvoie la moyenne de prix pour chaque année

Exercice 2

Question 1

xquery version "1.0";

declare function local:calculerSurface($m){

for $s in $m/\*/\*/@surface-m2

return $s

};

declare variable $maisons := doc("maisons.xml");

<html>

<body>

<table border="1">

<thead><tr><th>Maisons</th><th>Surface(m2)</th></tr></thead>

<tbody>

{

for $maison in $maisons//maison

return

<tr>

<td>Maison {data($maison/@id)}</td>

<td>{sum(local:calculerSurface($maison))}</td>

</tr>

}

</tbody>

</table>

</body>

</html>

Exercice 3

Question 1

Réponse :

let $p\_o:= "plant\_order.xml";

let $p\_f:= "plant\_families.xml";

let $p\_c:= "plant\_catalog.xml";

return

<results>

{

for $b in doc($p\_o)//COMMON

if ($b="Bloodroot")

then for $a in doc($p\_f)//NAME

if( $a="Papaveraceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Bloodroot"))

return <FAMILY>{$a} </FAMILY>

if ($b="Hepatica")

then for $a in doc($p\_f)//NAME

if( $a="Ranunculaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Hepatica"))

return <FAMILY>{$a} </FAMILY>

if ($b="Phlox, Blue")

then for $a in doc($p\_f)//NAME

if( $a="Polemoniaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Phlox, Blue"))

return <FAMILY>{$a} </FAMILY>

if ($b="Hepatica")

then for $a in doc($p\_f)//NAME

if( $a="Ranunculaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Hepatica"))

return <FAMILY>{$a} </FAMILY>

if ($b="Cinquefoil")

then for $a in doc($p\_f)//NAME

if( $a="Rosaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Cinquefoil"))

return <FAMILY>{$a} </FAMILY>

if ($b="Gentian")

then for $a in doc($p\_f)//NAME

if( $a="Gentianaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Gentian"))

return <FAMILY>{$a} </FAMILY>

}

</results>

Question 2

let $p\_c:= "plant\_catalog.xml";

return

<results>

{

for $b in doc($p\_c)//PLANT

order by $b/LIGHT

}

</results>

Question 3

let $p\_c:= "plant\_catalog.xml";

return

<results>

{

let $p\_o:= "plant\_order.xml";

let $p\_f:= "plant\_families.xml";

let $p\_c:= "plant\_catalog.xml";

for $b in doc($p\_c)//PLANT

order by $b/LIGHT and $b/COMMON

{

for $b in doc($p\_o)//COMMON

if ($b="Bloodroot")

then for $a in doc($p\_f)//NAME

if( $a="Papaveraceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Bloodroot"))

return <FAMILY>{$a} </FAMILY>

if ($b="Hepatica")

then for $a in doc($p\_f)//NAME

if( $a="Ranunculaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Hepatica"))

return <FAMILY>{$a} </FAMILY>

if ($b="Phlox, Blue")

then for $a in doc($p\_f)//NAME

if( $a="Polemoniaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Phlox, Blue"))

return <FAMILY>{$a} </FAMILY>

if ($b="Hepatica")

then for $a in doc($p\_f)//NAME

if( $a="Ranunculaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Hepatica"))

return <FAMILY>{$a} </FAMILY>

if ($b="Cinquefoil")

then for $a in doc($p\_f)//NAME

if( $a="Rosaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Cinquefoil"))

return <FAMILY>{$a} </FAMILY>

if ($b="Gentian")

then for $a in doc($p\_f)//NAME

if( $a="Gentianaceae")

then for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Gentian"))

return <FAMILY>{$a} </FAMILY>

}

</results>

Question 4

let $p\_o:= "plant\_order.xml";

let $p\_f:= "plant\_families.xml";

let $p\_c:= "plant\_catalog.xml";

return

<results>

{

declare variable $som as xs:double := 0.0;

for $b in doc($p\_o)//COMMON

if ($b="Bloodroot")

then let $a:=doc($p\_o)[COMMON="Bloodroot"]/QUANTITY

for $c in doc($p\_c)

where some $p in doc($p\_c)//COMMON satisfies

(contains($p,"Bloodroot"))

let $r:=doc($p\_c)/PRICE

$som=$som+( $a\*$r)

return $som

(: $som , variable global on repete cette mettre instruction pour les autres plantes

A la dernière plante on fait le return <PRICE>$som</PRICE> :)

</results>