

# Data Bases II

## XQuery (again)

Michele Beretta

[michele.beretta@unibg.it](mailto:michele.beretta@unibg.it)



## Exam from 12/19/2019

In a class hierarchy, a subclass extending the parent classes inherits all the attributes they define. Note that a parent class can not extend a child class. The following DTD describes the structure of a hierarchy of classes and the list of attributes each class defines.

```
<!ELEMENT repo (class+, attributes+)>
<!ELEMENT class (subclass*)>
<!ATTLIST class id #PCDATA #required>
<!ELEMENT subclass EMPTY>
<!ATTLIST subclass id #PCDATA #required>
<!ELEMENT attributes (attribute+)>
<!ATTLIST attributes classid #PCDATA #required> <!--id of (sub)class element-->
<!ELEMENT attribute (#PCDATA)>
```

Formulate in XQuery:

1. For each leaf subclass, list all its attributes considering also the ones it inherits.
2. Find the list of attributes belonging to all the leaf subclasses (for each root class).

1. For each leaf subclass, list all its attributes considering also the ones it inherits.

```
for $s in //subclass
let $c := $s/../class
return (
  <result>
    $s
    <attributes>
      { for $a in //attribute
        where $a/@classid = $s/id or $a/@classid = $c/@classid
        return $a/attribute }
    </attributes>
  </result>
)
```

2. Find the list of attributes belonging to all the leaf subclasses (for each root class).

```
for $c in //class
return (
  <group>
    $c
    <attributes>
      { for $a in //attribute[@classid = $c/subclass/@id]
        where count($c/subclass) = count($c/subclass[@id = $a/@classid])
        return $a }
      </attributes>
    </group>
  )
```

## Exam from 20/12/2021

The `forest.xml` document stores a forest of N-ary Trees. Each node in the tree is characterized by a weight. The following DTD details the structure of the document.

```
<!ELEMENT forest (node*)>
<!ELEMENT node (node*)>
<!ATTLIST node weight CDATA #REQUIRED>
```

1. Write an XQuery program to extract the heaviest tree. The weight of a tree equals the sum of the weights of all the nodes belonging to the tree. (Hint: you can use `xs:string(hash:sha256($node))` to generate a string that uniquely identifies a given node)
2. The width of an N-ary Tree is the number of nodes present at any level, whichever is the parent of the node. Write a recursive XQuery program printing the maximum width of each Tree.

1. Write an XQuery program to extract the heaviest tree. The weight of a tree equals the sum of the weights of all the nodes belonging to the tree. (Hint: you can use `xs:string(hash:sha256($node))` to generate a string that uniquely identifies a given node)

```
declare function local:weight($node as element()) as xs:decimal() {
  return node/@weight + sum( for $n in $node/node
                             return local:weight($n) )
}

for $tree in /forest/node
return (
  <weight>
    <tree>{xs:string(hash:sha256($tree))}</tree>
    <value>{local:weight($tree)}</value>
  </weight>
)
```

2. The width of an N-ary Tree is the number of nodes present at any level, whichever is the parent of the node. Write a recursive XQuery program printing the maximum width of each Tree.

```
declare function local:width($nodes as element(*) as xs:decimal() {
  if (empty($nodes)) then return 0
  else return max(count($nodes), local:width($nodes/node))
}

for $tree in /forest/node
return (
  <width>
    <tree>{xs:string(hash:sha256($tree))}</tree>
    <value>{local:width($tree)}</value>
  </width>
)
```

## Exam from 22/12/2022

The Crown Jewels is a magnificent collection of royal regalia, precious jewels, and gemstones, which has been stored and displayed at the Tower of London since 1661. The following DTD stores information about the jewels in the collection. Unspecified elements only contain PCDATA.

```
<!ELEMENT Collection (masterpiece*) >  
<!ELEMENT masterpiece (name, description, gemstone*) >  
<!ELEMENT gemstone (identifier, type, value) >
```

1. Write an XQUERY program to extract the masterpiece with the largest number of gemstones.
2. Write an XQUERY program to extract all the masterpieces without a Ruby (a ruby is a type of gemstone).
3. Write an XQUERY function to compute the value of each masterpiece containing gemstones.

1. Write an XQUERY program to extract the masterpiece with the largest number of gemstones.

```
for $m in //masterpiece
where count($m/gemstone) = max(for $m in //masterpiece
                               return count($m/gemstone))
return $m
```

2. Write an XQUERY program to extract all the masterpieces without a Ruby (a ruby is a type of gemstone).

```
for $m in //masterpiece
where not($m = //masterpiece[gemstone/type = "Ruby"])
return $m
```

3. Write an XQUERY function to compute the value of each masterpiece containing gemstones.

```
declare function local:value($masterpiece as element()) as xs:decimal() {  
  if (empty($masterpiece/gemstone)) then  
    return 0  
  else  
    return sum($masterpiece/gemstone/value)  
}
```