UNIVERSITY OF WESTMINSTER

5COSC002W DATABASE SYSTEMS

2021-2022 Tutorial 10 Querying XML documents – XQuery

Tutorial 10 Question 01: Simple XQuery statements

a) Return the modules nodes and descendants.

```
let $doc := doc("uni_marks.xml")
for $M in $doc//module
return $M
```

b) Return the module names and module leaders.

```
let $doc := doc("uni_marks.xml")
for $M in $doc//module
return ($M/modulename,$M/moduleleader)
```

c) Return the student nodes and descendants. 2 possible answers: one iterates through the student node, the other one iterates through the module node.

```
(: Answer 1 - Iterate through students :)
let $doc := doc("uni_marks.xml")
for $S in $doc//student
return $S
(: Answer 2 - Iterate through modules :)
let $doc := doc("uni_marks.xml")
for $M in $doc//module
return $M/student
```

d) Return the student first names and surnames of all students taking modules. 2 possible answers: one iterates through the student node, the other one iterates through the module node.

```
(: Answer 1 - Iterate through students :)
let $doc := doc("uni_marks.xml")
for $S in $doc//student
return ($S/sname, $S/fname)
(: Answer 2 - Iterate through modules :)
let $doc := doc("uni_marks.xml")
for $M in $doc//module
return ($M//sname, $M//fname)
```

Tutorial 10 Question 02: XQuery statements with condition on element

a) Return all the details for the module called "Database Systems".

```
let $doc := doc("uni_marks.xml")
let $modname := "Database Systems"
for $M in $doc//module
where $M/modulename = $modname
return $M
```

b) Return the surnames of the students on the "Database Systems" module. 2 possible answers: one iterates through the student node, the other one iterates through the module node.

```
(: Answer 01 - Iterate through module :)
let $doc := doc("uni_marks.xml")
```

```
let $modname := "Database Systems"
for $M in $doc//module
where $M/modulename = $modname
return $M//sname
(: Answer 02 - Iterate through students :)
let $doc := doc("uni_marks.xml")
let $modname := "Database Systems"
for $S in $doc//student
where $S/../modulename = $modname
return $S//sname
```

c) Return the details of the module and students who have scored more than 40 in the first component of the assessment.

```
let $doc := doc("uni_marks.xml")
let $passmark := 40
for $S in $doc//student
where $S/mark1 >=$passmark
return $S
```

d) Return the details of the module and students on the Database Systems module who have scored more than 40 in the first component of the assessment.

```
(: Iterate through students as you need to iterate for the condition that involves the descendent :)
let $doc := doc("uni_marks.xml")
let $passmark := 40
let $modname := "Database Systems"
for $S in $doc//student
where $S/../modulename = $modname and $S/mark1 >= $passmark
return ($S/../modulename, $S)
```

e) Return the details of the module and students on the Database Systems module who have scored more than 40 in the first component of the assessment.

```
let $doc := doc("uni_marks.xml")
let $passmark := 40
let $modname := "Database Systems"
for $S in $doc//student
where $S/../modulename = $modname and ($S/mark1 >= $passmark or $S/mark2 >= $passmark)
return $S
```

Tutorial 10 Question 03: XQuery statements with condition on attribute

a) Return the details of the students on the module identified by the code "MOD102".

```
(: Answer 01 - Iterate through students and use predicate at module level:)
let $doc := doc("uni_marks.xml")
let $modcode := "MOD102"
for $M in $doc//module
where $M[@moduleCode = $modcode]
return $M/student
(: Answer 02 - Iterate through students and use condition at module code level :)
let $doc := doc("uni_marks.xml")
let $modcode := "MOD102"
for $M in $doc//module
where $M/@moduleCode = $modcode
return $M/student
(: Answer 03 - Iterate through module :)
let $doc := doc("uni_marks.xml")
let $modcode := "MOD102"
for $S in $doc//student
where $S/../@moduleCode = $modcode
return $S
```

b) Return a list of students and their marks on the MOD102 module for the students who have scored more than 70 either on the first or second component.

```
let $doc := doc("uni_marks.xml")
let $modcode := "MOD102"
let $topmark := 70
for $S in $doc//student
where ($S/../@moduleCode = $modcode
and ($S/mark1 >= $topmark or $S/mark2 >= $topmark))
return $S
```

Tutorial 10 Question 04: XQuery statements with sequence functions

a) Write a report that retrieves the name of the module, the surname of the student, the mark scored in both components and a calculation of the final mark if every component is worth 50%.
(PTO)

```
let $doc := doc("uni_marks.xml")
for $S in $doc//student
return
<report>
{$S/../modulename}
{$S/sname}
{$S/mark1}
{$S/mark2}
<finalmark>{0.5*($S/mark1+$S/mark2)}</finalmark>
</report>
```

b) Write a report that retrieves the name of the module and the number of students on each module.

```
let $doc := doc("uni_marks.xml")
for $M in $doc//module
return
<report>
{$M/modulename}
<nbofstudents>{count($M/student)}</nbofstudents>
</report>
```

c) Write a statistical report that provides the following info: for each module, the lowest, highest and average mark on the first component, as well as the lowest, highest and average mark on the second component. Use distinct values to group by and calculate min, max and average.

```
let $doc := doc("uni_marks.xml")
for $MC in distinct-values($doc//@moduleCode)
let $minMark1 := min($doc//module[@moduleCode = $MC]//mark1)
let $maxMark1 := max($doc//module[@moduleCode = $MC]//mark1)
let $avgMark1 := avg($doc//module[@moduleCode = $MC]//mark1)
let $minMark2 := min($doc//module[@moduleCode = $MC]//mark2)
let $maxMark2 := max($doc//module[@moduleCode = $MC]//mark2)
let $avgMark2 := avg($doc//module[@moduleCode = $MC]//mark2)
return
<stats>
<modulecode>{$MC}</modulecode>
<lowestmark1>{$minMark1}</lowestmark1>
 <highestmark1>{$maxMark1}</highestmark1>
<averagemark1>{$avgMark1}</averagemark1>
<lowestmark2>{$minMark2}</lowestmark2>
<highestmark2>{$maxMark2}</highestmark2>
```

<averagemark2>{$avgMark2}</averagemark2>
</stats