XQuery Notes

XOuery

- -Used to retrieve information from XML documents
- -XQuery is a language for finding and extracting elements and attributes from XML documents.
- -Used in conjunction with Xpath

For-Let-Where-OrderBy-Return (FLWR)

- For / Let: Used for gathering nodes into sets from a series of queries.
- Where: For filtering nodes into the sets using conditions.
- Order By: For returning your sets in a particular order
- Return: How to return the identified nodes in the set.

LET

LET <variable> := <xpath expression>, <xpath xpression>

e.g let \$c:= doc("data/tcd.xml")/assessments/course/mark

```
Example LET Clause
<?xml version="1.0"?>
                          XML Source
                                                                   XQuery
<assessments>
  <student name="Smith">
                                       let $c:=
                                       doc("data/tcd.xml")/assessments/co
       <mark thecourse="4BA5"> 99
                                       urse/mark
        </mark>
                                       return
       <mark thecourse="4BA1"> 75
        </mark>
                                          t of avg course marks>
  </student>
  <course name="4BA1"</pre>
                                          </list of avg course marks>
           takenby="Smith, Jones">
     <mark>60</mark>
  </course>
  <course name="4BA5"</pre>
           takenby="Smith, Bord">
     <mark>70</mark>
  </course>
</assessments>
                                                                      Result
Curly brackets {} are used for enclosed
                                        t of avg course marks>
   expressions and indicate that the
                                            <mark>60</mark>
expression enclosed in the return clause
                                            <mark>70</mark>
  needs to be evaluated by the Xquery
                                       </list of avg course marks>
             processor
```

FOR

• FOR <variable> IN <xpath expression>, <xpath expression>, <xpath expression>

```
Example FOR Clause
<?xml version="1.0"?>
                             XML Source
                                                                            XQuery
<assessments>
                                             for $j in
doc("data/tcd.xml")/assessments/co
   <student name="Smith">
       <mark thecourse="4BA5"> 99
         </mark>
       <mark thecourse="4BA1"> 75
  </mark>
                                             ("Course Node:",$j)
   </student>
   <course name="4BA1"
          takenby="Smith, Jones">
      <mark>60</mark>
   </course>
   <course name="4BA5"
    takenby="Smith,Bord">
                                                                                 Result
      <mark>70</mark>
                                              Course Node: <course name="4BA1"
takenby="Smith, Jones">
   </course>
</assessments>
                                                 <mark>60</mark>
                                              </course>
                                              Course Node: <course name="4BA5" takenby="Smith,Bond">
       Round Brackets useful for
       grouping sequence of
                                                  <mark>70</mark>
       Operations.
                                              </course>
```

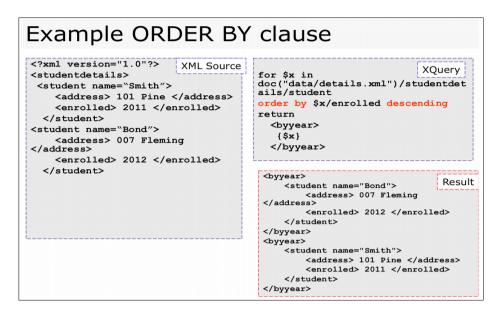
WHERE

where contains(\$j/@takenby,"Bond")

```
Example WHERE Clause
<?xml version="1.0"?>
                                                                     XQuery
<assessments>
                                        for $j in
doc("data/tcd.xml")/assessments/co
  <student name="Smith">
       <mark thecourse="4BA5"> 99
        </mark>
      <mark thecourse="4BA1"> 75
  </mark>
                                         where contains($j/@takenby, "Bond")
  </student>
                                                 <Bond courses is>
  <course name="4BA1"</pre>
                                                 {string($j/@name)}
           takenby="Smith, Jones">
                                                 </Bond_courses_is>
      <mark>60</mark>
  </course>
  <course name="4BA5"
          takenby="Smith,Bond">
     <mark>70</mark>
  </course>
</assessments>
                                                                         Result
                                          <Bond_courses_is>4BA5</Bond_courses_is>
```

ORDER BY

order by \$x/enrolled descending



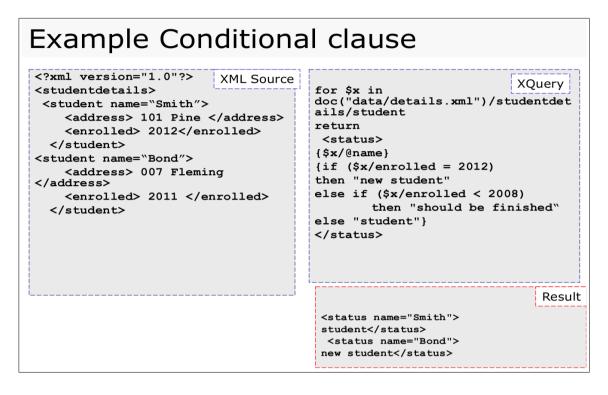
QUERYING OVER SEVERAL DOCUMENTS

-Can simultaneously access multiple documents

```
Querying over several interlinked documents
<?xml version="1.0"?>
                             XML Source
                                                                            XQuery
<assessments>
                                Tcd.xml
  <student name="Smith">
                                            for $w in
doc("data/details.xml")/studentdet
      <mark thecourse="4BA5"> 99
         </mark>
                                            ails/student,
       <mark thecourse="4BA1"> 75
         </mark>
                                            doc("data/tcd.xml")/assessments/st
  </student>
                                            udent
<course name="4BA1"
takenby="Smith,Jones">
                                            where x/\theta_n = y/\theta_n
     <mark>60</mark>
                                            return
  </course>
                                            <studentpercourse>
  <course name="4BA5"</pre>
                                                {$w/@name}
            takenby="Smith, Bond">
                                                {$w/address}
     <mark>70</mark>
                                                {$x/mark/@thecourse}
  </course>
                                            </studentpercourse>
</assessments>
                             XML Source
                             details.xml
<?xml version="1.0"?>
<studentdetails>
 <student name="Smith">
                                                                               Result
                                            <studentpercourse name="Smith"
thecourse="4BA5" thecourse="4BA1">
    <address> 101 Pine </address>
     <enrolled> 2001 </enrolled>
                                                 <address> 101 Pine </address>
  </student>
<student name="Bond">
                                            </studentpercourse>
    <address> 007 Fleming </address>
    <enrolled> 2002 </enrolled>
  </student>
```

CONDITIONAL STATEMENTS

-Can use conditional statements within your queries



QUANTIFIED EXPRESSIONS

Example Quantified Expression clauses XML Source XQuery tcd2.xml <?xml version="1.0"?> <results> <assessments> {if (every \$m in doc("data/tcd2.xml")/assessments/s <student name="Ledwidge"> <mark thecourse="4BA5"> 45 tudent satisfies \$m/mark > 60) </mark> then "excellent results" <mark thecourse="4BA1"> 55 </mark> else if (some \$t in doc("data/tcd2.xml")/assessments/s </student> <student name="ONeill"> tudent satisfies \$t/mark > 50) <mark thecourse="4BA5"> 85 </mark> </student> </assessments> then "average results" else "bad results" </results> Result <results>average results</results>

FUNCTIONS

Example Function clause

```
<?xml version="1.0"?>
                             XML Source
<assessments>
                               Tcd.xml
  <student name="Smith">
      <mark thecourse="4BA5"> 99
        </mark>
      <mark thecourse="4BA1"> 75
        </mark>
  </student>
  <course name="4BA1"</pre>
takenby="Smith, Jones">
    <mark>60</mark>
  </course>
  <course name="4BA5"</pre>
            takenby="Smith,Bond">
     <mark>70</mark>
  </course>
</assessments>
                            XML Source
                             tcd2002.xml
<?xml version="1.0"?>
<assessments>
  <student name="Ledwidge">
     <mark thecourse="4BA5"> 45
</mark>
      <mark thecourse="4BA1"> 55
</mark>
  </student>
  <student name="ONeill">
      <mark thecourse="4BA5"> 85
</mark> </student> </assessments>
```

```
XOuerv
declare function local:all students()
for $s in
doc("data/tcd.xml")/assessments/student
union
doc("data/tcd2.xml")/assessments/studen
return
<student>
          {$s/@name}
         {$s/mark/@thecourse}
</student>
<all>
{local:all_students()}
                                        Result
    <student name="Smith" thecourse="4BA5"
thecourse="4BA1"/>
<student name="Ledwidge" thecourse="4BA5"
thecourse="4BA1"/>
    <student name="ONeill" thecourse="4BA5"/>
</a11>
```

Example Function clause with param

```
<?xml version="1.0"?>
                              XML Source
<assessments>
                                Tcd.xml
  <student name="Smith">
      <mark thecourse="4BA5"> 99
         </mark>
      <mark thecourse="4BA1"> 75
         </mark>
  </student>
<course name="4BA1"
takenby="Smith,Jones">
     <mark>60</mark>
  </course>
  <course name="4BA5"</pre>
            takenby="Smith, Bond">
     <mark>70</mark>
  </course>
</assessments>
                             XML Source
<?xml version="1.0"?>
<assessments>
  <student name="Ledwidge">
      <mark thecourse="4BA5"> 45
      <mark thecourse="4BA1"> 55
  </student>
  <student name="ONeill">
      <mark thecourse="4BA5"> 85
</mark> </student> </assessments>
```