

**INFS 740**  
**Database Programming for the**  
**WEB**  
**XQuery**

Prof. Alex Brodsky  
GMU

# XML Example users.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<users>
  <user_tuple>
    <userid>U01</userid>
    <name>Tom Jones</name>
    <rating>B</rating>
  </user_tuple>
  <user_tuple>
    <userid>U02</userid>
    <name>Mary Doe</name>
    <rating>A</rating>
  </user_tuple>
  <user_tuple>
    <userid>U03</userid>
    <name>Dee Linqent</name>
    <rating>D</rating>
  </user_tuple>

```

```
<user_tuple>
  <userid>U04</userid>
  <name>Roger Smith</name>
  <rating>C</rating>
</user_tuple>
<user_tuple>
  <userid>U05</userid>
  <name>Jack Sprat</name>
  <rating>B</rating>
</user_tuple>
<user_tuple>
  <userid>U06</userid>
  <name>Rip Van Winkle</name>
  <rating>B</rating>
</user_tuple>
</users>

```

# XQuery Example

```
define global $users { treat as document users
    (glx:document-validate("docs/users.xml", "users")) }
<result>
{
  for $u in $users/users/user_tuple
  where ($u/rating="A")
  return
    <user_tuple>
      {$u}
    </user_tuple>
}
</result>
```

# Another Example

```
<result>
{
  for $u in $users//user_tuple,
    $i in $items//item_tuple
  where
    ($u/rating > "C" and $i/reserve_price > 1000
     and $i/offered_by = $u/userid)
  return
    <warning>
      {$u/name}
      {$u/rating}
      {$i/description}
      {$i/reserve_price}
    </warning>
}
</result>
```

# Basic Ideas Of XQuery

- Value expressions
- Sequences of values
- Use XPath expression to “navigate” on the XML tree, to get values or sequences of values

# Sequence Constructor

- Use “,” as concatenation operator
  - Example: \$u/name, \$u/id

# Loops

- Use FLWR expressions to loop through sequences

– Example

```
for $u in $users/users/user_tuple
```

```
let $n := $u/name
```

```
where ($u/rating="A")
```

```
return $n
```

# Data Types

- Literals
- Complex type: Element type using XML Schema

```
define element bid_count { element itemno,  
  element nbids { xsd:int } }
```

# Functions

```
define function bid_summary() as element*
{
  for $i in distinct-values($bids//itemno)
  let $b := $bids//bid_tuple[itemno = $i]
  return
    <bid_count>
      { $i }
      <nbids>{ count($b) }</nbids>
    </bid_count>
}
```

# More Examples

<http://www.research.avayalabs.com/user/wadler/papers/xquery-tutorial/xquery-tutorial.pdf>