Povpraševalni jezik za XML

Iztok Savnik

XQuery

- Sintaksa
  - XQL
  - XML-QL
- Delo z drevesi oz. usmerjenimi grafi
  - Podatkovni model = usmerjen označen graf

Izrazi poti

- angl. path expressions

Primeri:

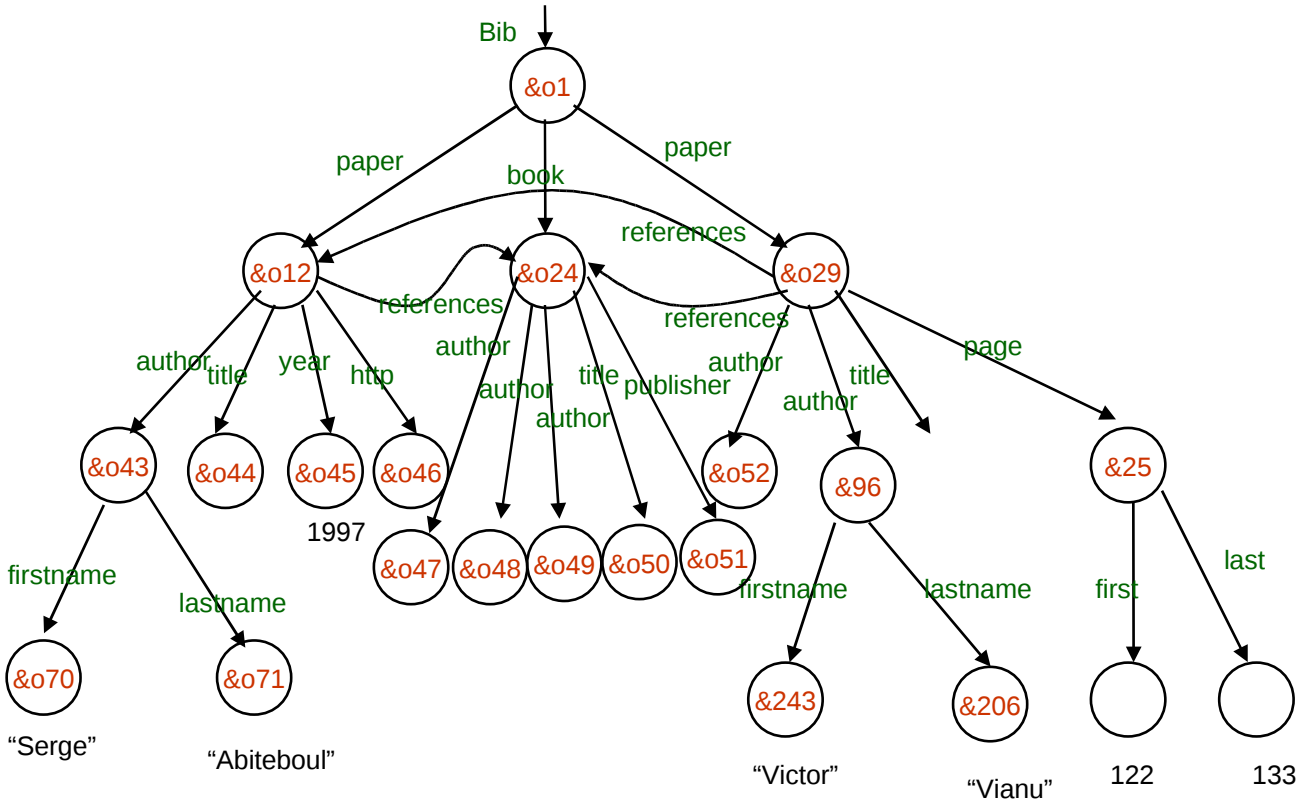
- Bib.paper
- Bib.book.publisher
- Bib.paper.author.lastname

Vrednosti izraza poti p je množica objektov.

Izrazi poti

Primer:

DB =

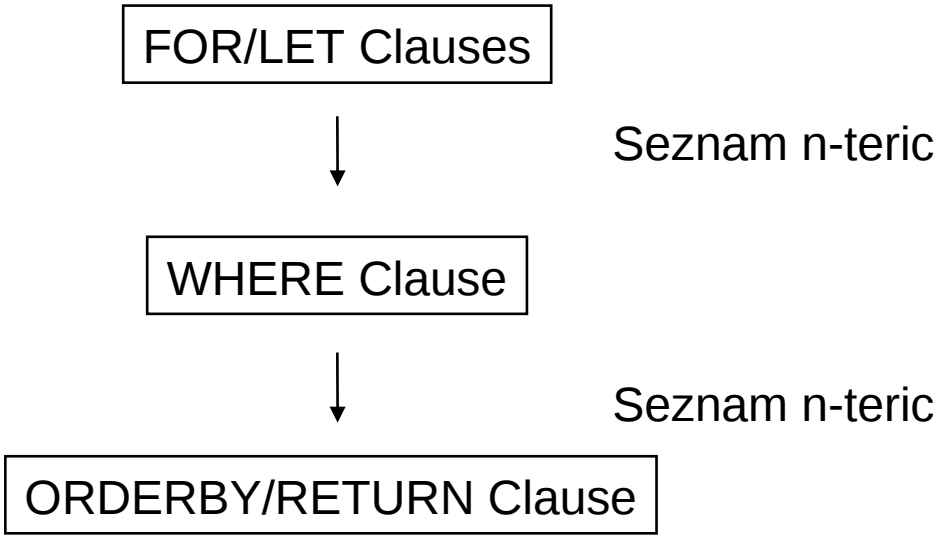



```
Bib.paper={&o12,&o29}
Bib.book.publisher={&o51}
Bib.paper.author.lastname={&o71,&206}
```

XQuery

Povzetek:

- FOR-LET-WHERE-ORDERBY-RETURN = FLWOR



FOR v.s. LET

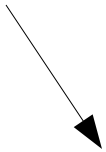
- FOR $x in expr -- poveže $x s vsako n-terico iz seznama izraza expr

- LET $x = expr -- poveže $x s celotnim seznamom izraza expr
  - Uporabno za skupne izraze in za agregacijo

FOR v.s. LET

```
for $x in document("bib.xml")/bib/book
return <result> $x </result>
```

```
let $x in document("bib.xml")/bib/book
return <result> $x </result>
```



<result> <book>...</book></result>
<result> <book>...</book></result>
<result> <book>...</book></result>
...

<result> <book>...</book>
<book>...</book>
<book>...</book>
...
</result>

Izrazi poti

- Krajši zapis
  - /bib/paper[2]/author[1]
  - /bib//author
  - paper[author/lastname=“Vianu"]
  - /bib/(paper|book)/title
- Poln zapis
  - child::bib/descendant::author
  - child::bib/descendant-or-self::*/child::author
  - parent, self, descendant-or-self, attribute

Primer 1

Poišči naslove vseh knjig, ki so bile publicirane po 1995:

```
for $x in document("bib.xml")/bib/book
where $x/year > 1995
return $x/title
```

Rezultat:

```
<title>TCP/IP Illustrated</title>
<title>Advanced Unix Programming</title>
<title>Data on the Web</title>
```

Primer 2

Poišči naslove vseh knjig, ki so dražje od 50.

```
for $x in doc("books.xml")/bib/book
where $x/price>50
orderby $x/title
return $x/title
```

Primer 2

```
<bib>
<book>
<title>TCP/IP Illustrated</title>
<author>Stevens</author>
<publisher>Addison-Wesley</publisher>
</book>
<book>
<title>Advanced Unix Programming</title>
<author>Stevens</author>
<publisher>Addison-Wesley</publisher>
</book>
<book>
<title>Data on the Web</title>
<author>Abiteboul</author>
<author>Buneman</author>
<author>Suciu</author>
<publisher>Morgan Kaufmann</publisher>
</book>
</bib>
```

Primer 3

Za vsakega avtorja knjige založbe Morgan Kaufmann, izpiši vse knjige:

```
for $a in distinct(document("bib.xml")
          /bib/book[publisher=“Morgan Kaufmann”]/author)
return <result>
       $a,
       for $t in /bib/book[author=$a]/title
       return $t
       </result>
```

Primer 3

Rezultat:

```
<result>
<author>Abiteboul</author>
<title>Data on the Web</title>
</result>
<result>
<author>Buneman</author>
<title>Data on the Web</title>
</result>
<result>
<author>Suciu</author>
<title>Data on the Web</title>
</result>
```

Primer 4

Izpiši avtorje in naslove knjig danega avtorja.

```
<authlist>
{
for $a in distinct($books//author)
order by $a
return
  <author>
  <name>
    { $a/text() }
  </name>
  <books>
    {
    for $b in $books//book[author = $a]
    order by $b/title
    return $b/title
    }
  </books>
  </author>
}
</authlist>
```

Primer 4

```
<authlist>
<author>
  <name>Abiteboul</name>
  <books>
  <title>Data on the Web</title>
  </books>
  </author>
<author>
  <name>Buneman</name>
  <books>
  <title>Data on the Web</title>
  </books>
</author>
<author>
  <name>Stevens</name>
  <books>
  <title>TCP/IP Illustrated</title>
  <title>Advanced Unix Programming</title>
  </books>
</author>
<author>
  <name>Suciu</name>
  <books>
  <title>Data on the Web</title>
  </books>
</author>
</authlist>
```

Primer 5

Poišči vse oddelke, ki imajo več kot 10 zaposlenih.
Izpiši št. oddelka, število zaposlenih in povprečno plačo.

```
for $d in doc("depts.xml")//deptno
let $e := doc("emps.xml")//emp[deptno = $d]
where count($e) >= 10
order by avg($e/salary) descending
return
  <big-dept>
    {
    $d,
    <headcount>{count($e)}</headcount>,
    <avgsal>{avg($e/salary)}</avgsal>
    }
</big-dept>
```

Povezanost: XQuery in XML

- Sintaksa vprašanj je zasnovana tako, da se lahko vključijo v XML
- XML dokumenti imajo komponente, ki so poizvedbe
  - Pri interpretaciji XML dokumenta jih ovrednotimo
- Dokumenti so tako “relativni” glede na okolje kamor so prenešeni oz. uporabljeni

Agregacijske funkcije

Izpiši vse založnike, ki so izdali več kot 100 knjig.

```
<big_publishers>
   for $p in distinct(document("bib.xml")//publisher)
   let $b := document("bib.xml")/book[publisher = $p]
   where count($b) > 100
   return $p
</big_publishers>
```

count = agregacijska funkcija, ki vrne število elementov

Agregacijske funkcije (2)

Poišči knjige, ki so dražje od povprečne cene knjig:

```
LET $a=avg(document("bib.xml")/bib/book/price)
FOR $b in document("bib.xml")/bib/book
WHERE $b/price > $a
RETURN $b
```

for v.s. let

FOR

- Poveži spremenljivke vozlišča → iteracija

LET

- Poveži spremenljivko s kolekcijo → ena vrednost

Kolekcije v Xquery (1)

- Urejene in neurejene kolekcije
  - /bib/book/author = urejena kolekcija
  - Distinct(/bib/book/author) = neurejena kolekcija
- `let $a = /bib/book` → $a je kolekcija
- `$b/author` → kolekcija (več avtorjev...)

```
return <result>$b/author</result>
```

Rezultat:

```
<result> <author>...</author>
         <author>...</author>
         <author>...</author>
         ...
</result>
```

Kolekcije v Xquery (2)

Kaj s kolekcijami v izrazih ?

- $b/price ▭ seznam n cen

- $b/price * 0.7 ▭ seznam n števil ??
- $b/price * $b/quantity ▭ seznam n x m števil ??

  – Operacije je veljavna samo, če ima seznam en sam element.
  – Atomizacija

- $book1/author eq "Kennedy" – primerjava vrednosti
- $book1/author = "Kennedy" – splošno primerjanje

Sortiranje v XQuery

```
<publisher_list>
  for $p in distinct(document("bib.xml")//publisher)
  orderby  $p
  return
    <publisher>
      <name> $p/text() </name> ,
      for $b in document("bib.xml")//book[publisher = $p]
      orderby $b/price descending
      return
        <book>
        $b/title,
        $b/price
        </book>
    </publisher>
</publisher_list>
```

If-Then-Else

```
for $h in //holding
orderby $h/title
return <holding>
          $h/title,
          if $h/@type = "Journal"
          then $h/editor
          else $h/author
       </holding>
```

Eksistenčni kvantifikator

```
for $b in //book
where some $p IN $b//para satisfies
   contains($p, "sailing")
   and contains($p, "windsurfing")
return $b/title
```

Univerzalni kvantifikator

```
for $b in //book
where every $p IN $b//para satisfies
   contains($p, "sailing")
return $b/title
```

Group-By v XQuery

```
for $y in document("http://www.bn.com")
          /bib/book/@year
let $b = document("http://www.bn.com")
         /bib/book[@year=$y and
                   publisher="Morgan Kaufmann"]
where count($b)>10
return <year> $y </year>
```

Ekvivalenten SQL :

```
select year
from Bib
where Bib.publisher="Morgan Kaufmann"
groupby year
having count(*) > 10
```

Viri

- http://www.w3.org/TR/xquery/
- R. Ramakrishnan, J. Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2004.
- David J. Malan, XML with Java, Java Servlet, and JSP, Course CS E-259, Harvard University, 2007