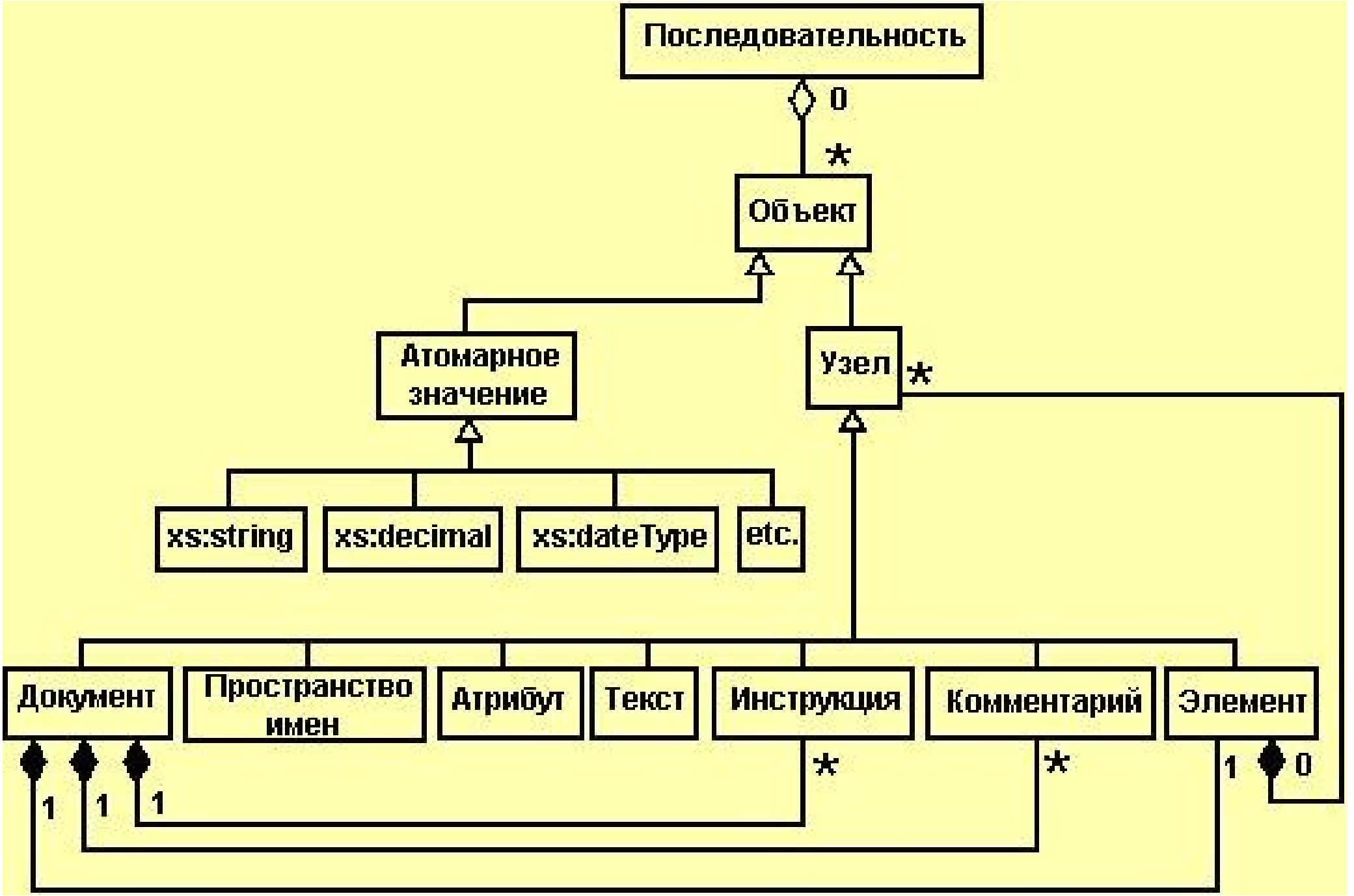
Web-технологии 3

Язык запросов XML
XML Query Language (XQuery)

Введение

- язык запросов для баз данных в формате XML
- XML как модель данных (древовидная)
- XML — стандарт обмена данными
- Взаимодействие с реляционной БД
- XML упорядочен, реляционная БД — нет
- XML разрежен, реляционная БД — нет
- Чистый язык реляционных запросов не подходит к XML данным

- Основан на XPath
- XPath не определяет источник данных
- XPath не формирует результат, только выборка
- XPath не объединяет данные
- XQuery получает набор элементов и возвращает набор элементов
-


Последовательность
0
*
Объект
Атомарное значение
Узел
*
xs:string
xs:decimal
xs:dateType
etc.
Документ
Пространство имен
Атрибут
Текст
Инструкция
Комментарий
Элемент
1
1
1
*
*
1
0

Выражения пути

- Основан на Xpath
- Пример XML:
- <recipe><title>Beef Parmesan with Garlic Angel Hair Pasta</title>
  <ingredient name="beef cube steak" amount="1.5" unit="pound"/>
  <ingredient name="onion, sliced into thin rings" amount="1"/>
  <ingredient name="green bell pepper, sliced in rings" amount="1"/>
  ...
- Xquery:
  doc("recipes.xml")//recipe[title="Ricotta Pie"]//ingredient[@amount]
- Результат — набор ингридиентов

Выражение FLOWR

- Это аббревиатура из первых букв операторов For-Let-Where-Order-Return
- for — связывание переменной с выражением
- let — связывание переменной с результатом
- order by — сортировка потока кортежей
- where — условие на кортежи
- return — результат выражения

Оператор for

- связывает одну или более переменных с выражениями, создавая поток кортежей.
- В потоке каждый кортеж связывает данную переменную с одним из значений, получаемых в результате вычисления выражения.
- Пример:

  Выбрать имена и фамилии всех сотрудников
- Запрос:

  for $a in doc('file:///c:/KING/kingx/king_corp.xml')//employee/name return data($a)

- Переменная XQuery - $a
- можно трактовать for как оператор цикла
- Переменная $a последовательно (в цикле) принимает все значения из потока
- функция data() извлекает данные из результата выражения

Оператор let

- связывает переменные с полным результатом вычисления выражения, добавляя эти связи к кортежам, полученным оператором for, или создавая единственный кортеж (при отсутствии оператора for).
- Пример:

  Выбрать имена и должности всех сотрудников
- Запрос:

  for $a in doc('file:///c:/KING/kingx/king_corp.xml') //employee/name   let $b:=$a/../function   return ($a, $b)

Оператор order by

- сортирует поток кортежей.
- Пример:

  Выбрать список сотрудников, упорядоченный по убыванию зарплаты
- Запрос:

  for $a in doc('file:///c:/KING/kingx/king_corp.xml')//employee order by xs:decimal(data($a/salary)) descending return ($a/name,$a/salary)
-

Оператор where

- оставляет в потоке только те кортежи, которые удовлетворяют условию, являющемуся параметром данного оператора.
- Пример:

  Выбрать всех клерков
- Запрос:

  for $a in doc('file:///c:/KING/kingx/king_corp.xml')//employee where $a/function='CLERK' return ($a/name/text(), $a/function/text())
-

Оператор return

- создает результат выражения FLWOR для данного кортежа.
- Пример:

  Выбрать имена всех клерков
- Запрос:

  for $a in doc('file:///c:/KING/kingx/king_corp.xml')//employee where $a/function='CLERK' return <clerk> { data($a/name) } </clerk>
-

Основные правила FLOWR

- for и let можно использовать несколько раз в выражении
- только одно выражение where доступно
- можно определить несколько критериев сортировки
- Объединение нескольких документов

  for $p IN document("www.irs.gov/taxpayers.xml")//person

  for $n IN document("neighbors.xml")//neighbor[ssn = $p/ssn]

  return <person> <ssn> { $p/ssn } </ssn> { $n/name } <income> { $p/income } </income> </person>

Операторы

- Список констант

  (7, 9, <thirteen/>)
- Численные интервалы

  i to j
- Выражения и функции XPath
- Объединение

  ,
- Операторы множества
- | (или union), intersect, except

Условные выражения

- Конструкция if-then-else

```
for $h in document("library.xml")//holding
return
  <holding>
    { $h/title,
      if ($h/@type = "Journal")
      then $h/editor
      else $h/author
    }
  </holding>
```

Пример

- "естественное соединение" - соединяем данные, лежащие на разных ветвях нашего дерева
- **Для каждого покупателя выбрать продавца, который его обслуживает**
- Запрос:

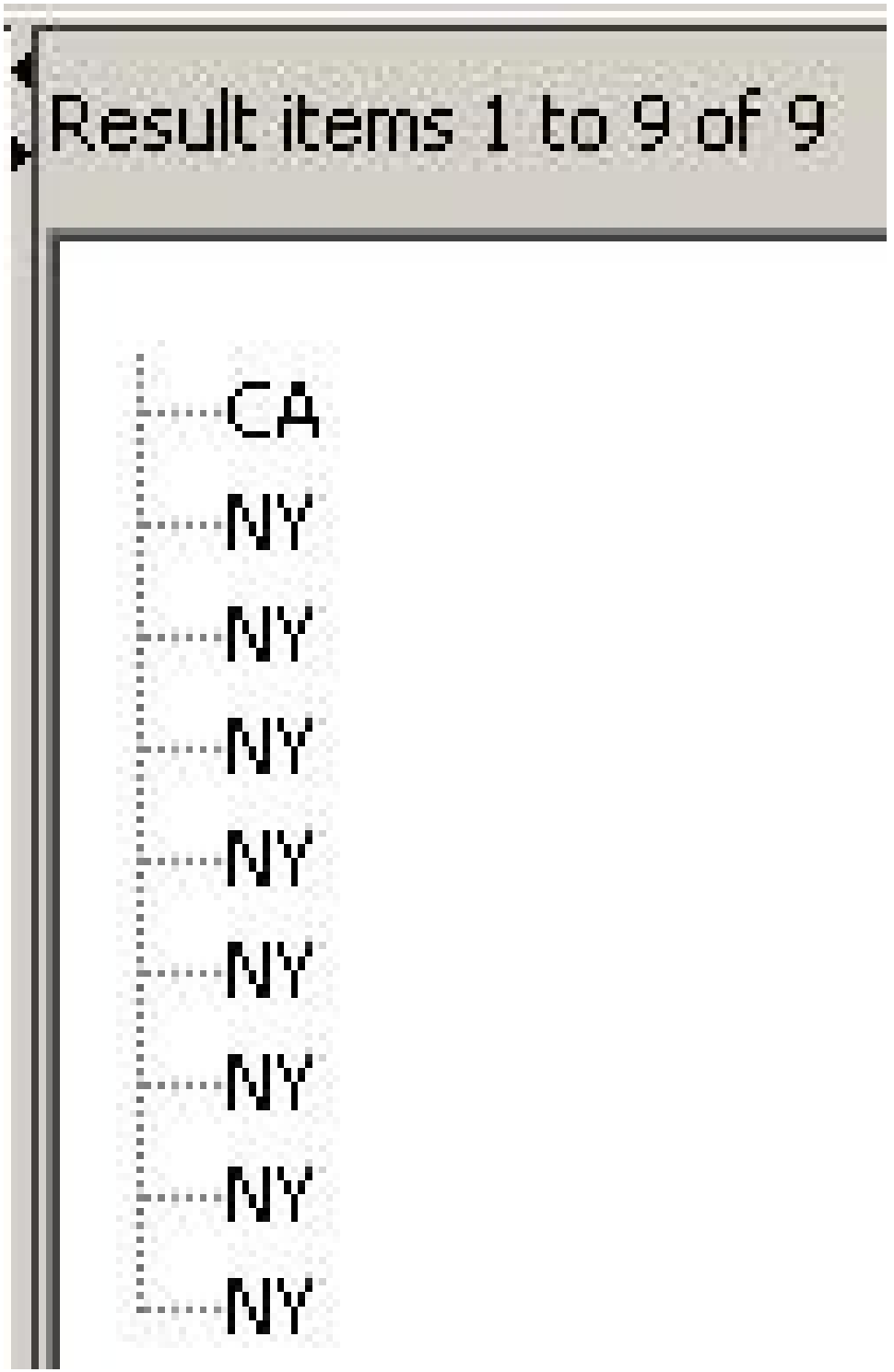
```
for $a in
doc('file:///c:/KING/kingx/king_corp.xml')//employee

for $b in
doc('file:///c:/KING/kingx/king_corp.xml')//customers
/customer

where $a/customer/@id = $b/@id

return concat($b/name,' - ',$a/name)
```


Result items 1 to 32 of 32
Next
Prev.
View
xs:string = 'JOCKSPORTS - JAMES B. KING'
xs:string = 'CENTURY SHOP - DANIEL T. PETERS'
xs:string = 'FITNESS FIRST - DANIEL T. PETERS'
xs:string = 'STADIUM SPORTS - KAREN P. SHAW'
xs:string = 'REBOUND SPORTS - KAREN P. SHAW'
xs:string = 'POINT GUARD - KAREN P. SHAW'
xs:string = 'THE COLISEUM - KAREN P. SHAW'
xs:string = 'ALL SPORT - RAYMOND Y. PORTER'
xs:string = 'GOOD SPORT - RAYMOND Y. PORTER'
xs:string = 'THE POWER FORWARD - SARAH S. DUNCAN'
xs:string = 'AL AND BOB'S SPORTS - SARAH S. DUNCAN'
xs:string = 'AL'S PRO SHOP - GREGORY J. LANGE'
xs:string = 'HIT, THROW, AND RUN - GREGORY J. LANGE'
xs:string = 'WHEELS AND DEALS - LIVIA N. WEST'
xs:string = 'JUST BIKES - LIVIA N. WEST'
xs:string = 'JOE'S BIKE SHOP - LIVIA N. WEST'
xs:string = 'BOB'S SWIM, CYCLE, AND RUN - LIVIA N. WEST'
xs:string = 'EVERY MOUNTAIN - KEVIN J. ALLEN'
xs:string = 'WOMENS SPORTS - KEVIN J. ALLEN'
xs:string = 'JUST TENNIS - CYNTHIA D. WARD'
xs:string = 'SHAPE UP - CYNTHIA D. WARD'
xs:string = 'VOLLYRITE - KENNETH J. MARTIN'
xs:string = 'BOB'S FAMILY SPORTS - KENNETH J. MARTIN'
xs:string = 'K + T SPORTS - MARY A. TURNER'
xs:string = 'NORTH WOODS HEALTH AND FITNESS SUPPLY C

Пример

- Еще пример на "естественное соединение".
- **Вывести список штатов, в которых находятся покупатели, обслуживаемые отделами, расположенными в городе NEW YORK.**
- Запрос:

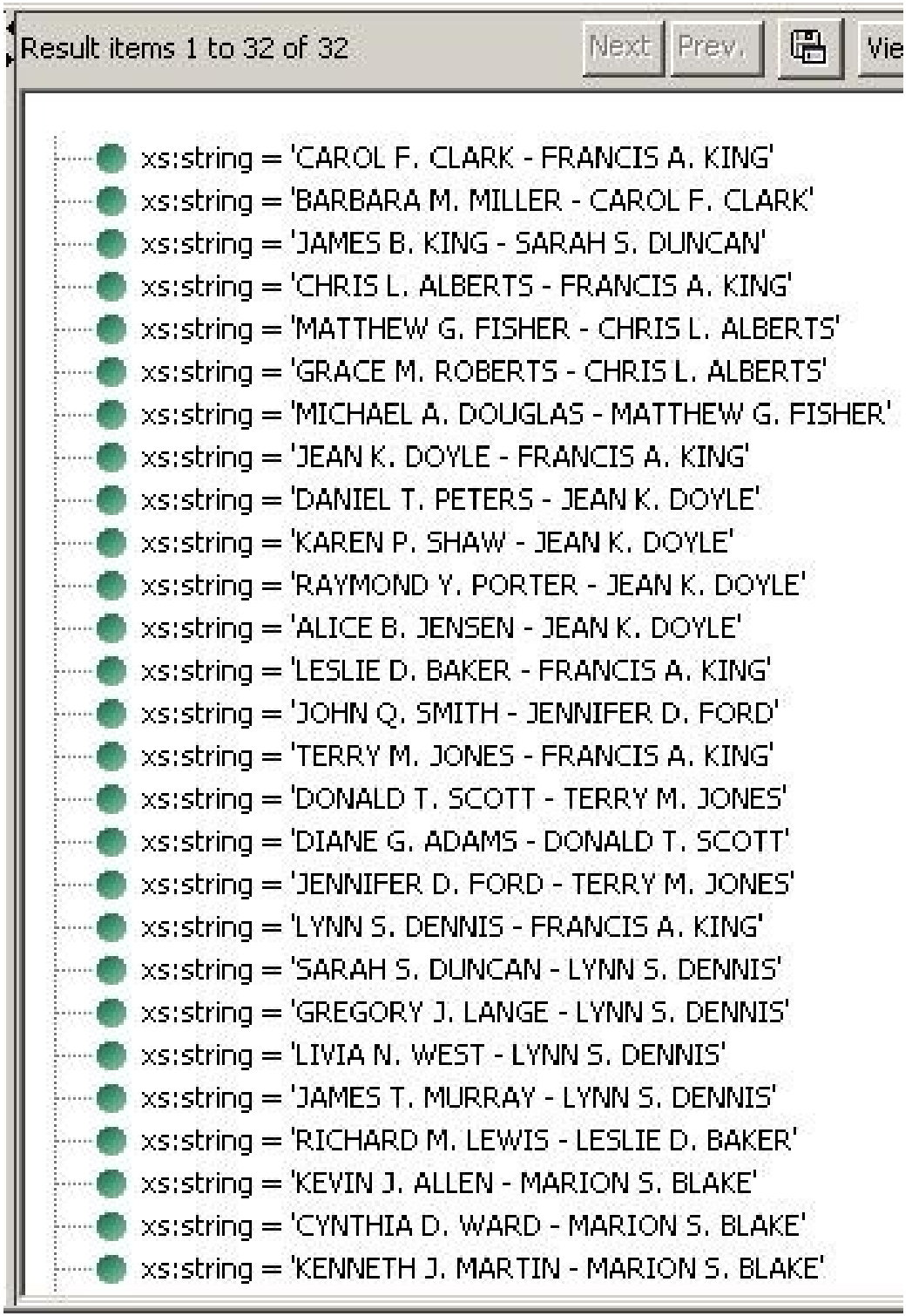
```
for $a in doc('file:///c:/KING/kingx/king_corp.xml')//
      department[location='NEW YORK']//employee/@id
for $b in doc('file:///c:/KING/kingx/king_corp.xml')//customer
where $b/salesperson=$a return $b/state/text()
```


Result items 1 to 9 of 9
CA
NY
NY
NY
NY
NY
NY
NY
NY

Пример

- Для каждого сотрудника выбрать его непосредственного начальника
- Запрос:

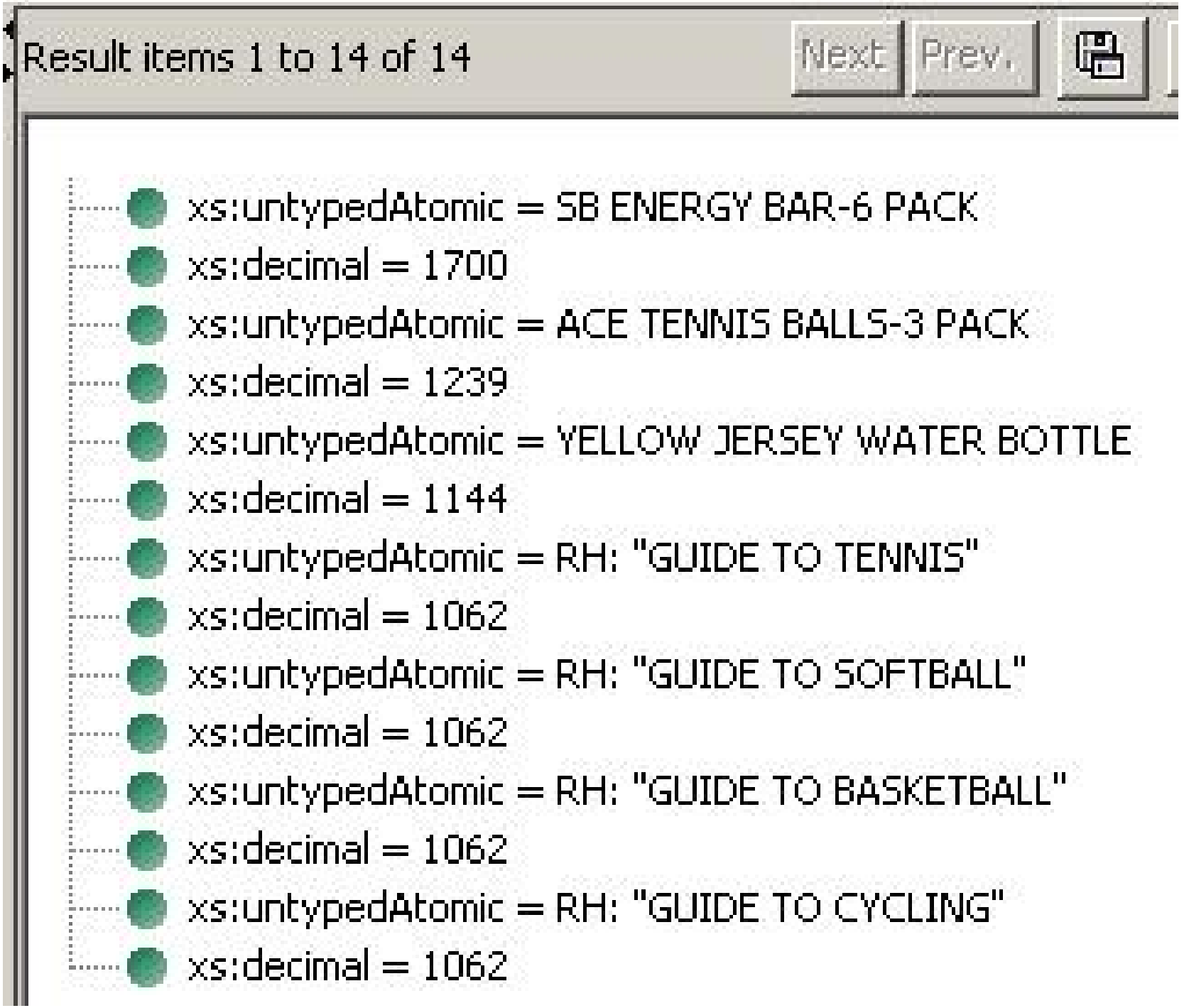
```
for $a in
doc('file:///c:/KING/kingx/king_corp.xml')//employee
for $b in
doc('file:///c:/KING/kingx/king_corp.xml')//employee
where $b/@id = $a/manager_id
return concat($a/name,' - ',$b/name)
```


Result items 1 to 32 of 32
Next
Prev.
xs:string = 'CAROL F. CLARK - FRANCIS A. KING'
xs:string = 'BARBARA M. MILLER - CAROL F. CLARK'
xs:string = 'JAMES B. KING - SARAH S. DUNCAN'
xs:string = 'CHRIS L. ALBERTS - FRANCIS A. KING'
xs:string = 'MATTHEW G. FISHER - CHRIS L. ALBERTS'
xs:string = 'GRACE M. ROBERTS - CHRIS L. ALBERTS'
xs:string = 'MICHAEL A. DOUGLAS - MATTHEW G. FISHER'
xs:string = 'JEAN K. DOYLE - FRANCIS A. KING'
xs:string = 'DANIEL T. PETERS - JEAN K. DOYLE'
xs:string = 'KAREN P. SHAW - JEAN K. DOYLE'
xs:string = 'RAYMOND Y. PORTER - JEAN K. DOYLE'
xs:string = 'ALICE B. JENSEN - JEAN K. DOYLE'
xs:string = 'LESLIE D. BAKER - FRANCIS A. KING'
xs:string = 'JOHN Q. SMITH - JENNIFER D. FORD'
xs:string = 'TERRY M. JONES - FRANCIS A. KING'
xs:string = 'DONALD T. SCOTT - TERRY M. JONES'
xs:string = 'DIANE G. ADAMS - DONALD T. SCOTT'
xs:string = 'JENNIFER D. FORD - TERRY M. JONES'
xs:string = 'LYNN S. DENNIS - FRANCIS A. KING'
xs:string = 'SARAH S. DUNCAN - LYNN S. DENNIS'
xs:string = 'GREGORY J. LANGE - LYNN S. DENNIS'
xs:string = 'LIVIA N. WEST - LYNN S. DENNIS'
xs:string = 'JAMES T. MURRAY - LYNN S. DENNIS'
xs:string = 'RICHARD M. LEWIS - LESLIE D. BAKER'
xs:string = 'KEVIN J. ALLEN - MARION S. BLAKE'
xs:string = 'CYNTHIA D. WARD - MARION S. BLAKE'
xs:string = 'KENNETH J. MARTIN - MARION S. BLAKE'

Пример

- Определить те товары, которых сотрудник JONES может купить на свою зарплату больше 1000 штук
- Запрос:

```
for $a in doc('file:///c:/KING/kingx/king_corp.xml')
        //employee[tokenize(name,' ')[3]='JONES']/salary
for $b in doc('file:///c:/KING/kingx/king_corp.xml')
        //product/price[not(@end)]
where xs:decimal($b/min)*1000<="" pre="">
```


Result items 1 to 14 of 14
Next
Prev.
xs:untypedAtomic = SB ENERGY BAR-6 PACK
xs:decimal = 1700
xs:untypedAtomic = ACE TENNIS BALLS-3 PACK
xs:decimal = 1239
xs:untypedAtomic = YELLOW JERSEY WATER BOTTLE
xs:decimal = 1144
xs:untypedAtomic = RH: "GUIDE TO TENNIS"
xs:decimal = 1062
xs:untypedAtomic = RH: "GUIDE TO SOFTBALL"
xs:decimal = 1062
xs:untypedAtomic = RH: "GUIDE TO BASKETBALL"
xs:decimal = 1062
xs:untypedAtomic = RH: "GUIDE TO CYCLING"
xs:decimal = 1062

Пример

- **Выбрать тех сотрудников, которые имеют подчиненных**

  В этом запросе мы избавляемся от дубликатов
- Запрос:

```
for $s in {
    for $a in doc('file:///c:/KING/kingx/king_corp.xml')      //employee
    for $b in doc('file:///c:/KING/kingx/king_corp.xml')   //employee
    where $b/@id = $a/manager_id
    return $b/name
  }
return distinct-values($s/name)
```

Предыдущий пример

- использование оператора let для упрощения оператора for.
- Запрос:

```
let $x:={
  for $a in
     doc('file:///c:/KING/kingx/king_corp.xml')//employee
  for $b in
     doc('file:///c:/KING/kingx/king_corp.xml')//employee
    where $b/@id = $a/manager_id
    return $b/name
  }
for $a in $x
return distinct-values($a/name)
```

Пример

- Для каждого клерка выбрать разность между его зарплатой и средней зарплатой клерков
- В XPath этот запрос у нас не получился.
- Определяем среднюю зарплату клерков и сохраняем ее в переменной $a. Затем перебираем клерков и из зарплаты каждого вычитаем $a.
- Запрос:
- let $a := doc('file:///c:/KING/kingx/king_corp.xml')//departments /avg(//employee[function='CLERK']/salary)

  for $b in doc('file:///c:/KING/kingx/king_corp.xml')//employee[function='CLERK']

  return ($b/name/text(), $b/salary - $a)

Кванторы

- some-in-satisfies некоторые удовлетворяют выражению
- every-in-satisfies все удовлетворяют выражению
- Пример
- Выборка названий книжек, где в каждом параграфе содержится "sailing"

  for $b in document("bib.xml")//book where every $p in $b//paragraph satisfies contains($p,"sailing")

  return $b/title

Пример кванторы 1

- Выбрать тех сотрудников, которые НЕ имеют подчиненных
- Запрос:

```
let $x:=<x>{
    for $a in doc('file:///c:/KING/kingx/king_corp.xml')//employee
    for $b in doc('file:///c:/KING/kingx/king_corp.xml')//employee
    where $b/@id = $a/manager_id
    return $b/name} </x>
for $c in doc('file:///c:/KING/kingx/king_corp.xml')//employee
  where every $h in $x/name satisfies $h!=$c/name
return $c/name/text()
```

- Выбираются те, в котором имя не совпадает ни с одним элементом в списке в переменной $x

Пример кванторы 2

- Выбрать коды тех отделов, в которых есть КАКИЕ-НИБУДЬ из должностей, имеющихся в отделе 23
- Запрос:

```
let $x:= <x>{
  for $a in  doc('file:///c:/KING/kingx/king_corp.xml')//department[@id='23']
  return $a }</x>
for $c in doc('file:///c:/KING/kingx/king_corp.xml')//department[@id!='23']
 where some $h in $x//function satisfies; $h=$c//function
return $c/data(@id)
```

- Оператором let создается список должностей отдела 23.
- Затем выбираются те, у которых какая-нибудь должность входит в список в переменной $x

Пример

- Выбрать имена и зарплаты 10 самых высокооплачиваемых сотрудников фирмы (без учета комиссионных)
- Запрос:

```
let $x:=<x>{
 for $a in doc('file:///c:/KING/kingx/king_corp.xml')//employee
 order by xs:decimal($a/salary) descending
 return <y>{$a/name, $a/salary}</y> }</x>
for $b in $x/y[position()<=10]
return concat($b/name,' - ',$b/salary)
```

- запрос представляет некоторую трудность в SQL

Конструктор элементов

- Можно строить новые элементы

```
<employee empid="{$id}">
  <name>{$name}</name>
  {$job}
  <deptno>{$deptno}</deptno>
  <salary>{$SGMLspecialist+100000}</salary>
</employee>
```

Пример конструктора

- Составить отчет по работе отделов продаж, в котором указать:
- id отдела
- город
- список покупателей
- общую сумму продаж

```
1 <report> {
2   let $x:=<x>{
3     for $a in doc('file:///c:/KING/kingx/king_corp.xml')
4         //department
5     where $a/name='SALES'
6     return $a
7     }</x>
8   for $a in $x//department
9   return <department>{$a/@id} {
10     $a/location,
11     <customers> {
12       for $b in doc('file:///c:/KING/kingx/king_corp.xml')
13          //customers/customer
14       for $c in $a//employee
15       where $c/data(@id)=$b/data(salesperson)
16       return $b/name
17       } </customers>,
```

```
18      <total> {
19        let $y:=<y>{
20          for $b in doc('file:///c:/KING/kingx/king_corp.xml')
21            //customers/customer
22          for $c in $a//employee
23          where $c/data(@id)=$b/data(salesperson)
24          return <t>{sum($b/order/total)}</t>
25          }</y>
26        for $d in $y return sum($d//t)
27        }</total>
28      } </department>
29 }</report>
```

```
<report>
  <department id="13">
    <location>NEW YORK</location>
    <customers>
      <name>STADIUM SPORTS</name>
      <name>REBOUND SPORTS</name>
      <name>POINT GUARD</name>
      <name>THE COLISEUM</name>
      <name>ALL SPORT</name>
      <name>GOOD SPORT</name>
      <name>CENTURY SHOP</name>
      <name>FITNESS FIRST</name>
      </customers>
    <total>44715.649999999994</total>
    </department>
  <department id="23">
    <location>DALLAS</location>
    <customers>
      <name>THE POWER FORWARD</name>
      <name>AL AND BOB'S SPORTS</name>
      <name>AL'S PRO SHOP</name>
      <name>HIT, THROW, AND RUN</name>
      <name>WHEELS AND DEALS</name>
      <name>JUST BIKES</name>
      <name>JOE'S BIKE SHOP</name>
      <name>BOB'S SWIM, CYCLE, AND RUN</name>
```

Вычисляемый конструктор

- используются ключевые слова для создания элементов и атрибутов
- Значения элементов и атрибутов задаются либо константами, либо выражениями, в том числе, это могут быть выражения FLOWR.
- Пример:

```
element employee {
  attribute id {"2265"},
  element name {"Edsger W. Dejkstra"},
  element salary {"4500.00"},
  element commission {"1000"},
  element hire_date {"1967-01-10"},
  element function {"RESEARCH WORKER"},
  element manager_id {"7250"}
}
```

Функции

- Использование в рекурсивных выражениях
- **Выбрать всех прямых начальников сотрудницы DIANE G. ADAMS**
- Запрос:
- 1 declare namespace mySpace = "http://khpi-iip/derev"
- 2 define function mySpace:getManager($a as element(employee)*)
- 3 as element(employee)* {
- 4   for $b in doc('file:///c:/KING/kingx/king_corp.xml')//employee
- 5   where $b/@id = $a/manager_id
- 6   return ($a, $b, mySpace:getManager($b))
- 7 }
- 8 for $a in doc('file:///c:/KING/kingx/king_corp.xml')//employee[name='DIANE G. ADAMS']
- 9 return mySpace:getManager($a)/name/text()