

# XQuery

Графеева Н.Г.

2015

# XQuery

- XQuery - это язык, ориентированный на выборку XML данных
- XQuery для XML-структур подобен SQL для реляционных баз данных
- XQuery основан на XPath-выражениях
- XQuery поддерживается всеми основными производителями систем, управляющих базами данных
- XQuery рекомендован к использованию консорциумом W3C

# Пример

- for \$x in doc("books.xml")/bookstore/book  
where \$x/price>30  
order by \$x/title  
return \$x/title

# XML-ДОКУМЕНТ

- `<?xml version="1.0" encoding="UTF-8"?>`

`<bookstore>`

```
<book category="COOKING">  
  <title lang="en">Everyday Italian</title>  
  <author>Giada De Laurentiis</author>  
  <year>2005</year>  
  <price>30.00</price>  
</book>
```

```
<book category="CHILDREN">  
  <title lang="en">Harry Potter</title>  
  <author>J K. Rowling</author>  
  <year>2005</year>  
  <price>29.99</price>  
</book>
```

```
<book category="WEB">  
  <title lang="en">XQuery Kick Start</title>  
  <author>James McGovern</author>  
  <author>Per Bothner</author>  
  <author>Kurt Cagle</author>  
  <author>James Linn</author>  
  <author>Vaidyanathan Nagarajan</author>  
  <year>2003</year>  
  <price>49.99</price>  
</book>
```

```
<book category="WEB">  
  <title lang="en">Learning XML</title>  
  <author>Erik T. Ray</author>  
  <year>2003</year>  
  <price>39.95</price>  
</book>
```

`</bookstore>`

# ОСНОВЫ синтаксиса XQuery

- XQuery - case-sensitive
- Элементы XQuery должны быть правильными XML-именами (valid)
- XQuery-строки можно заключать в двойные или одиночные кавычки
- Переменные XQuery начинаются с символа \$,  
за которым следует имя xml-элемента,  
например: \$bookstore
- Комментарии XQuery выделяются двоеточием, например: (: XQuery Comment :)

# Как открыть XML-документ?

- `doc()` - открывает весь документ
- Пример использования:
- `doc("books.xml")/bookstore/book/title`
- В дальнейшем можем увидеть результат:
- `<title lang="en">Everyday Italian</title>`  
`<title lang="en">Harry Potter</title>`  
`<title lang="en">XQuery Kick Start</title>`  
`<title lang="en">Learning XML</title>`

- Пример использования (с предикатом):
- `doc("books.xml")/bookstore/book[price<30]`

# FLWOR -конструкции

- **FLWOR** - "For, Let, Where, Order by, Return"
- **for** – задает переменную для цикла
- **let** – присваивание секвенции
- **where** – задает фильтр для выбираемых данных
- **order by** – указывает порядок сортировки
- **return** – указывает выбираемые значения



# Пример (for clause, order by)

## XMLQuery

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

## Результат

```
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

# Пример (for clause)

## XQuery

- for \$x in (1 to 5)  
return <test>{\$x}</test>

## Результат

- <test>1</test>  
<test>2</test>  
<test>3</test>  
<test>4</test>  
<test>5</test>

# Пример (for clause)

## XQuery

```
for $x at $i in doc("books.xml")/bookstore/book/title  
return <book>{$i}. {data($x)}</book>
```

## Результат

- <book>1. Everyday Italian</book>  
<book>2. Harry Potter</book>  
<book>3. XQuery Kick Start</book>  
<book>4. Learning XML</book>

# Пример (for clause)

## XQuery

- for \$x in (10,20), \$y in (100,200)  
return <test>x={\$x} and y={\$y}</test>

## Результат

- <test>x=10 and y=100</test>  
<test>x=10 and y=200</test>  
<test>x=20 and y=100</test>  
<test>x=20 and y=200</test>

# Пример (let clause)

## XQuery

- `let $x := (1 to 5)`  
`return <test>{$x}</test>`

## Результат

- `<test>1 2 3 4 5</test>`

# Пример (let clause + concatenation)

## XQuery

- `let $x := (1 to 5), (3 to 7)`  
`return <test>{$x}</test>`

## Результат

`<test>1 2 3 4 5 3 4 5 6 7</test>`

# Пример (let clause + union)

## XQuery

```
let $x := (1 to 5) union (3 to 7)  
return <test>{$x}</test>
```

## Результат

```
<test>1 2 3 4 5 6 7</test>
```

# Пример (let clause + intersect)

## XQuery

```
let $x := (1 to 5) intersect (3 to 7)  
return <test>{$x}</test>
```

## Результат

```
<test>3 4 5</test>
```



# Пример (let clause + except)

## XQuery

```
let $x := (1 to 5) except (3 to 7)  
return <test>{$x}</test>
```

## Результат

```
<test>1 2</test>
```

# Пример (let clause + reverse function)

## XQuery

```
let $x := reverse((1 to 5))  
return <test>{$x}</test>
```

## Результат

```
<test>5 4 3 2 1</test>
```

# Пример (data function)

## XMLQuery

```
for $x in doc("books.xml")/bookstore/book/title  
order by $x  
return {data($x)}
```

## Результат

Everyday Italian  
Harry Potter  
Learning XML  
XQuery Kick Start

# Пример (distinct-values function)

## XMLQuery

```
for $x in
  doc("books.xml")/bookstore/book/year
order by $x descending
return {distinct-values(data($x))}
```

## Результат

```
2005
2003
```

# Агрегатные функции

- min
- max
- avg
- count
- sum

# Пример

## XMLQuery

```
let $x := doc("books.xml")/bookstore/book
return
  <info>
    <count>{count($x)}</count>
    <avg_price>{avg($x/price)}</avg_price>
  </info>
```

## Результат

```
<info>
  <count>4</count>
  <avg_price>37.4825</avg_price>
</info>
```

# Условный оператор (if – then-else)

## XMLQuery

```
for $x in doc("books.xml")/bookstore/book
return if ($x/@category="CHILDREN")
then <child>{data($x/title)}</child>
else <adult>{data($x/title)}</adult>
```

## Результат

- <adult>Everyday Italian</adult>  
<child>Harry Potter</child>  
<adult>Learning XML</adult>  
<adult>XQuery Kick Start</adult>

# Пример (добавление HTML)

## XMLQuery

```
<html>
<body>
<h1>Bookstore</h1>
<ul>
{
for $x in doc("books.xml")/bookstore/book
order by $x/title
return <li>{data($x/title)}. Category:
{data($x/@category)}</li>
}
</ul>
</body>
</html>
```

## Результат

```
<html>
<body>
<h1>Bookstore</h1>
<ul>
<li>Everyday Italian. Category: COOKING</li>
<li>Harry Potter. Category: CHILDREN</li>
<li>Learning XML. Category: WEB</li>
<li>XQuery Kick Start. Category: WEB</li>
</ul>
</body>
</html>
```



# Пример (XQuery + ORACLE)

- SELECT xtab.count, xtab.firstname, xtab.lastname
- FROM employees,
- XMLTable('for \$i at \$count in /Employees/Employee
- return
- <P0>
- <count>{\$count}</count>
- <firstname>{\$i/firstname}</firstname>
- <lastname>{\$i/lastname}</lastname>
- </P0>
- ,
- PASSING data
- COLUMNS
- count number PATH '/P0/count' ,
- firstname VARCHAR2(32) PATH '/P0/firstname' ,
- lastname VARCHAR2(32) PATH '/P0/lastname') xtab
- where id = 1

SQL Commands - Windows Internet Explorer

http://apex.oracle....

Файл Правка Вид Избранное Сервис Справка

Избранное SQL Commands

SQL Workshop SQL Commands

Rows 10 Save Run

```
SELECT xtab.count, xtab.firstname, xtab.lastname
FROM employees,
XMLTable('for $i at $count in /Employees/Employee
return
  <P0>
    <count>{$count}</count>
    <firstname>{$i/firstname}</firstname>
    <lastname>{$i/lastname}</lastname>
  </P0>
```

Results Explain Describe Saved SQL History

COUNT	FIRSTNAME	LASTNAME
1	John	Watson
2	Sherlock	Homes
3	Jim	Moriarty
4	Mycroft	Holmes

4 rows returned in 0.07 seconds [Download](#)

Интернет 100%

# Упражнение

- Добавьте к предыдущему запросу поля age и email.

# Пример (XQuery + ORACLE)

- SELECT xtab.count, xtab.firstname, xtab.lastname
- FROM employees,
- XMLTable('for \$i at \$count in /Employees/Employee
- order by \$i/lastname
- return
- <P0>
- <count>{\$count}</count>
- <lastname>{\$i/lastname}</lastname>
- <firstname>{\$i/firstname}</firstname>
- </P0>
- '
- PASSING data
- COLUMNS
- count number PATH '/P0/count' ,
- lastname VARCHAR2(32) PATH '/P0/lastname' ,
- firstname VARCHAR2(32) PATH '/P0/firstname') xtab
- where id = 1

SQL Commands - Windows Internet Explorer

http://apex.oracle....

Файл Правка Вид Избранное Сервис Справка

Избранное SQL Commands

# ORACLE Application Express

Home Application Builder SQL Workshop Team Development

SQL Workshop SQL Commands

Rows 10 Save Run

```
SELECT xtab.count, xtab.firstname, xtab.lastname
FROM employees,
XMLTable('for $i at $count in /Employees/Employee
order by $i/lastname
return
<P0>
<count>{$count}</count>
<lastname>{$i/lastname}</lastname>
<firstname>{$i/firstname}</firstname>
</P0>
'
PASSING data
```

Results Explain Describe Saved SQL History

COUNT	FIRSTNAME	LASTNAME
4	Mycroft	Holmes
2	Sherlock	Homes
3	Jim	Moriarty
1	John	Watson

4 rows returned in 0.07 seconds [Download](#)

Интернет 100%

# Упражнение

- Выведите упорядоченный по фамилии список сотрудников, но с номерами 1,2,3 и т.п.

# Пример(XQuery + ORACLE)

- SELECT xtab.count
- FROM employees,
- XMLTable('let \$i := /Employees/Employee
- return <P0><count>{count(\$i)}</count></P0>'
- PASSING data
- COLUMNS count number PATH '/P0/count' ) xtab
- WHERE id = 1

SQL Commands - Windows Internet Explorer

http://apex.oracle....

Google

Файл Правка Вид Избранное Сервис Справка

Избранное SQL ...

# ORACLE Application Express

Home Application Builder SQL Workshop Team Development

SQL Workshop SQL Commands

Rows 10 Save Run

```
SELECT xtab.count|
FROM employees,
XMLTable('let $i := /Employees/Employee
return <P0><count>{count($i)}</count></P0>'
PASSING data
COLUMNS count number PATH '/P0/count' ) xtab
WHERE id = 1
```

Results Explain Describe Saved SQL History

COUNT
4

1 rows returned in 0.04 seconds [Download](#)

Интернет 100%



# Упражнение

- Выведите минимальный, максимальный и средний возраст сотрудников.

# Задание 11(8)

- Создайте приложение с отчетами, в которых будут выведены:
  - 1) фамилии студентов и все отметки (список должен быть упорядочен по фамилиям студентов);
  - 2) фамилии студентов и средние баллы (список должен быть упорядочен по среднему баллу);
  - 3) все названия предметов, которые читаются на различных IT кафедрах (список должен быть упорядочен по кафедрам, а затем по названиям предметов);
  - 4) названия предметов, которые читаются на каждой IT кафедре (т.е. пересечение).
  
- Ссылку на приложение с отчетами отправьте по адресу:
  - N.Grafeeva@spbu.ru
  - Тема - DB\_Application\_2015\_job11
  - *Примечание: задание должно быть отправлено в течение 14 дней. За более позднее отправление будут сниматься штрафные баллы ( по баллу за каждые две недели).*

# Полезные ссылки

- <http://www.w3schools.com/xqury>