

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Кафедра програмування

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №8

XQuery - мова запитів XML-документів

Викона(в/ла):

Ст. Прізвище Ім'я

Група

Тема: Вивчення XQuery - мови запитів XML-документів.

Мета роботи: Ознайомлення з синтаксисом мови XQuery та конструюванням запитів даних XML документа.

Теоретичний матеріал

Мова SQL, яка сьогодні є стандартом мови систем керування реляційними базами даних, була розроблена, щоб абстрагувати алгоритми, призначені для опису та перетворення даних, від деталей отримання та збереження цих даних. Це дозволяє створювати дуже складні програми транзакційної обробки даних, систем звітності, сховищ даних та бізнес-аналітики.

Потужність і гнучкість SQL зробили реляційну модель домінуючим типом даних протягом останніх десятиліть. Але, хоч попередник реляційних баз даних – ієрархічні бази, якраз і занепали через наявність сильної залежності від механізмів доступу і збереження даних, зараз вони знову широко застосовуються завдяки використанню мови XML.

XML документи представляють відносини між сутностями, використовуючи ієрархію, тобто відносини предок-нащадок. І саме застосування XQuery для роботи з даними XML в одному або декількох документах, дозволяє абстрагуватися від механізмів збереження і доступу до даних. Тобто, XQuery - це мова для пошуку та маніпулювання будь-чим, що може бути представлено як дерево і відповідає моделі даних, визначеній у [XQuery and XPath Data Model \(XDM\) 3.1](#) XQuery використовує XPath і одночасно її можна розглядати як розширення XPath. Це означає, що синтаксис XQuery схожий на XPath, і він не базується на основі XML-елементів, як XSLT.

XQuery також використовує спільну з XPath бібліотеку функцій та операторів, визначені в [XQuery and XPath Functions and Operators](#). Мова XQuery визначена W3C в [XQuery 3.1: An XML Query Language W3C Recommendation 21 March 2017](#)

Як і XPath, XQuery є підмножиною XSL (eXtensible Stylesheet Language) - мова стилів таблиць для XML.

XSL складається з чотирьох частин:

- XSLT - мова для перетворення XML-документів
- XPath - мова для навігації в XML-документах
- XSL-FO - мова для форматування XML-документів (завдяки модулю CSS3 Paged Media Module, W3C представив новий стандарт форматування документа і з 2013 року CSS3 пропонується як заміна XSL-FO)
- XQuery - мова для запитів XML-документів

Взагалі, будь-який вираз, який є синтаксично дійсним і успішно виконується в XPath 3.1 та XQuery 3.1, поверне однаковий результат в обох мовах. Є кілька винятків із цього правила:

- оскільки XQuery розширює заздалегідь визначені посилання на сутності та посилання на символи, а XPath цього не робить, вирази, що містять їх, дають різні результати на двох мовах. Наприклад, значення літерального рядка "&" є & в XQuery, і & в XPath;
- якщо ввімкнено режим сумісності XPath 1.0, XPath поводить себе по-різному від XQuery в кількох випадках, які перелічені в [XML Path Language \(XPath\), версія 3.1](#).

-

1. Основні поняття XQuery

XQuery можна використовувати для:

- Знаходження інформації для використання у веб-сервісі
- Створення зведених звітів
- Перетворення XML-даних у XHTML
- Пошуку відповідної інформації у веб-документах

1.1. Синтаксис

Деякі основні правила синтаксису:

- XQuery чутливий до регістру
- Елементи, атрибути та змінні XQuery повинні бути дійсними іменами XML
- Значення рядка XQuery може бути в одинарних або подвійних лапках
- Змінна XQuery визначається з \$, а потім ім'ям, наприклад, \$bookstore
- Коментарі XQuery обмежені (: ... :), наприклад (: XQuery Comment :)

Запит XQuery складається з одного або декількох модулів з тілом запиту, що містить вираз, значення якого є результатом запиту. Вираз представлений у граматиці XQuery символом Expr :

Expr	:: =	ExprSingle ("," ExprSingle)*
ExprSingle		FLWOR _Expr
	:: =	Quantified_Expr
		Switch_Expr
		Typeswitch_Expr
		If_Expr
		Try_CatchExpr
		Or_Expr

Оператором XQuery, який має найнижчий пріоритет є оператор кома, який використовується для об'єднання двох операндів для формування послідовності. Як показано вище, загальний вираз Expr може складатися з декількох операндів ExprSingle, розділених комами. Ім'я ExprSingle позначає вираз, який не містить оператора коми верхнього рівня (незважаючи на його ім'я, ExprSingle може означати послідовність, що містить більше одного елемента.)

ExprSingle використовується в різних місцях, де вираз не може містити коми верхнього рівня. Наприклад, кожен з аргументів виклику функції повинен бути ExprSingle, оскільки коми використовуються для розділення аргументів виклику функції.

Після коми, виразами, які мають нижчий пріоритет є [FLWORExpr](#), [QuantifiedExpr](#), [SwitchExpr](#), [TypeswitchExpr](#), [IfExpr](#), [TryCatchExpr](#), і [OrExpr](#).

Для ознайомлення з синтаксисом та основними поняттями XQuery скористаємося прикладом наступного XML документа, де перелічені актори та фільми з їх участю, а також деякі атрибути цих фільмів :

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <video_template>
    <title/>
    <genre>
      <choice>action</choice>
      <choice>comedy</choice>
      <choice>drama</choice>
      <choice>musical</choice>
    </genre>
```

```

<rating>
  <choice>G</choice>
  <choice>PG</choice>
  <choice>PG-13</choice>
  <choice>R</choice>
  <choice>NC-17</choice>
</rating>
<user_rating>
  <choice>1</choice>
  <choice>2</choice>
  <choice>3</choice>
</user_rating>
<summary/>
<year/>
<director/>
<studio/>
<runtime/>
<vhs/>
<vhs_stock/>
<dvd/>
<dvd_stock/>
</video_template>
<actors>
  <actor id="000000015">Anderson, Jeff</actor>
  <actor id="00000000f">Bonet, Lisa</actor>
  <actor id="000000006">Ford, Harrison</actor>
  <actor id="000000018">Ghigliotti, Marilyn</actor>
  <actor id="00000000c">Hackman, Gene</actor>
  <actor id="000000003">Jones, Tommy Lee</actor>
  <actor id="000000009">Smith, Will</actor>
  <actor id="00000001b">Spoonhauer, Lisa</actor>
  <actor id="000000012">Voight, John</actor>
</actors>

<videos>
  <video id="id1235">
    <title>The Fugitive</title>
    <genre>action</genre>
    <rating>PG-13</rating>
    <summary>
      Tommy Lee Jones and Harrison Ford are the hunter and the hunted in this fast-
      paced story of a falsely convicted man who escapes to find his wife's true
      killer.
    </summary>
    <year>1997</year>
    <director>Andrew Davis</director>
    <studio>Warner</studio>
    <user_rating>2</user_rating>
    <runtime>110</runtime>
    <actorRef>000000003</actorRef>
    <actorRef>000000006</actorRef>
    <vhs>13.99</vhs>
    <vhs_stock>206</vhs_stock>
    <dvd>14.99</dvd>
    <dvd_stock>125</dvd_stock>
  </video>
  <video id="id1244">
    <title>Enemy of the State</title>
    <genre>action</genre>
    <rating>R</rating>
    <summary>
      After a chance meeting with an old pal, Robert Deal finds himself in
      posession of a disk that contains evidence of an assassination plot by the
      NSA.
    </summary>
    <year>1999</year>
    <director>Tony Scott</director>
    <studio>Buena vista</studio>
  </video>

```

```

<user_rating>3</user_rating>
<runtime>113</runtime>
<actorRef>00000009</actorRef>
<actorRef>0000000c</actorRef>
<actorRef>0000000f</actorRef>
<actorRef>00000012</actorRef>
<vhs>16.99</vhs>
<vhs_stock>188</vhs_stock>
<dvd>29.99</dvd>
<dvd_stock>353</dvd_stock>
</video>
<video id="id1243">
  <title>Clerks</title>
  <genre>comedy</genre>
  <rating>R</rating>
  <summary>
    Spend a day at work with two friends and their quirky customers at a video
    shop and convenience store.
  </summary>
  <year>1999</year>
  <director>Kevin Smith</director>
  <studio>Buena vista</studio>
  <user_rating>1</user_rating>
  <runtime>97</runtime>
  <actorRef>00000015</actorRef>
  <actorRef>00000018</actorRef>
  <actorRef>0000001b</actorRef>
  <vhs>16.99</vhs>
  <vhs_stock>188</vhs_stock>
</video>
</videos>
</result>

```

Припустимо, ми хочемо знайти заголовки всіх відео з актором, ім'я якого - Ліза. XPath запит виглядатиме так:

```
//video[actorRef=//actors/actor[ends-with(., 'Lisa')]/@id]/title
```

Тобто його можна прочитати зліва направо як:

- Виберіть усі <video> елементи на будь-якому рівні
- Виберіть ті, у яких є actorRef елемент, значення якого дорівнює одному із наведених нижче значень:
 - Виберіть усі <actors>елементи на будь-якому рівні
 - Виберіть усі їх <actor>дочірні елементи
 - Виберіть елемент лише в тому випадку, якщо його значення закінчується на "Lisa"
 - Виберіть значення id атрибута
- Виберіть <title>дочірній елемент цих вибраних <video> елементів

Результат:

```

<title>Enemy of the State</title>
<title>Clerks</title>

```

Як бачимо, на такому рівні складності синтаксис XPath стає досить неприємним. Для такого роду запитів та для будь-якого складнішого синтаксису, власне, і підходить XQuery.

1.2. XQuery FLWOR вирази

Використовуючи SQL для останнього прикладу, фактично потрібно виконати з'єднання двох таблиць, таблиці відео та таблиці акторів. Це не зовсім те саме в XML, оскільки дані є ієрархічними, а не табличними, але XQuery дозволяє писати запити приєднання аналогічно звичному підходу SQL. Еквівалент виразу SELECT SQL називається виразом FLWOR, названим його п'ятьма пунктами: *for*, *let*, *where*, *order by*, *return*.

Приклад, як вираз FLWOR видає той самий результат:

```
let $doc := .
for $v in $doc//video,
  $a in $doc//actors/actor
where ends-with($a, 'Lisa') and $v/actorRef = $a/@id
return $v/title
```

Розберемо цей вираз FLWOR:

- Let просто оголошує змінну. Це оголошення включено, тому що можна по-різному встановити цю змінну; наприклад, можна ініціалізувати її як `doc('videos.xml')` або як результат якогось складного запиту, який знаходить певний документ у базі даних.
- for визначає дві змінні діапазону: одна стосується всіх відео, інша - всіх акторів. У сукупності вираз FLWOR обробляє всі можливі пари відео та акторів.
- Потім у where вибираються ті пари, які нас насправді цікавлять. Нас цікавить лише те, що в цьому відео є актор, і лише якщо ім'я актора закінчується на "Ліза".
- Нарешті return повідомляє системі, яку інформацію ми хочемо отримати. У цьому випадку ми хочемо назву відео.

Якщо уважно розглянути код запиту, можна помітити одну особливість XPath. А саме - більшість відеороликів матимуть більше одного актора. Тому вираз `$v/actorRef` вибирає кілька елементів. Правила для "=" оператора в XPath (а отже, і в XQuery) полягають у тому, що він порівнює *все* зліва з *усім* правим і повертає істину, якщо є хоча б одна відповідність. Фактично це робить неявне з'єднання. Можна уникнути використання цієї функції та написати запит у більш класичній реляційній формі як:

```
let $doc := .
for $v in $doc//video,
  $va in $v/actorRef,
  $a in $doc//actors/actor
where ends-with($a, 'Lisa') and $va eq $a/@id
return $v/title
```

Цього разу використали інший оператор рівності "**eq**", який дотримується більш звичайних правил, ніж = : він суворо порівнює *одне* значення зліва з *одним* значенням праворуч.

Таким чином у XQuery є два способи порівняння значень.

1. Загальні порівняння: =, !=, <, <=, >, >=
2. Порівняння величин: eq, ne, lt, le, gt, ge

Різниця між двома методами порівняння показана вище.

Щоб отримати результати у відсортованому порядку використовується **order by**. Припустимо, потрібно, щоб відео були в порядку їх виходу:

```
let $doc := .
for $v in $doc//video,
  $a in $doc//actors/actor
where ends-with($a, 'Lisa') and $v/actorRef = $a/@id
order by $v/year
return $v/title
```

for прив'язує змінну до кожного елемента, повернутого у виразі і призводить до ітерації.

Щоб зафіксувати певну кількість ітерацій, використовується ключове слово **to**:

```
for $x in (1 to 3)
return <test>{$x}</test>
```

Результат:

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

Ключове слово **at** використовується для підрахунку ітерацій:

```
for $x at $i in //video/title  
return <title>{$i}. {data($x)}</title>
```

Результат:

```
<title>1. The Fugitive</title>  
<title>2. Enemy of the State</title>  
<title>3. Clerks</title>
```

for i let операнди можуть з'являтися в будь-якому порядку, і будь-яку кількість разів. Вираз FLOWR є набагато більше, ніж те, що висвітлено - для отримання додаткової інформації ознайомтеся з [навчальним посібником XQuery FLWOR](#).

Посилання для ознайомлення з теоретичним матеріалом теми «XQuery - мова запитів XML-документів»:

- [Beginning XML, 5-edition © 2012 Joe Fawcett, Liam R.E. Quin, Danny Ayers](#)
- [Develop with XQuery: A better programming language for the database programmer](#)
- [XQuery Tutorial W3C](#)

Перелік розділів та понять, з якими необхідно ознайомитись для виконання завдання лабораторної роботи:

1. Основні поняття XQuery
 - 1.1. Синтаксис
 - 1.2. XQuery FLWOR вирази

Хід роботи

1. Опрацювати теоретичний матеріал.
2. У відповідності до свого XML документа, написати декілька запитів XQuery застосувавши FLWOR вирази з обов'язковим використати функцій.
Для отримання максимального балу застосувати якісь з виразів [QuantifiedExpr](#), [SwitchExpr](#), [TypeswitchExpr](#), [IfExpr](#), [TryCatchExpr](#) або [OrExpr](#).
3. Використати довільний XQuery processor, наприклад [XQuery 3.0 Online Tester](#), для перевірки коректності роботи створених запитів XQuery.
4. Оформити звіт про виконання лабораторної роботи, який має містити:
 - титульну сторінку;
 - тему, мету та завдання лабораторної роботи;
 - короткий перелік та опис створеного шаблону перетворення;
 - навести скріни екрану з кодом запитів XQuery та отриманими результатами.
5. Завантажити в канал "БД. Лабораторна робота" своєї команди в Teams.