

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра програмування

Лабораторна робота №8
“XQuery - мова запитів XML-документів”

Підготував:
студент групи ПМІ-31
Процьків Назарій

Львів 2023

Тема: Вивчення XQuery - мови запитів XML-документів.

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

Хід роботи

1. Опрацював теоретичний матеріал.
2. Написав кілька виразів шляху XQuery у відповідності до свого XML документа:

XML документ:

```
Formatted Data

"
<!-- Created with Liquid Technologies Online Tools
<socialnetwork>
  <person>
    <id>2</id>
    <firstname>Oleksandr</firstname>
    <lastname>Zhenchenko</lastname>
    <birthdate>2003-12-28</birthdate>
    <address>Lviv, Hnata Hotkevycha 1</address>
    <workplace>
      <company>Amazon</company>
      <startdate>2016-12-12</startdate>
      <enddate>2016-12-27</enddate>
    </workplace>
    <workplace>
      <company>Facebook</company>
      <startdate>2016-12-28</startdate>
      <enddate>2017-02-21</enddate>
    </workplace>
    <workplace>
      <company>Starlink</company>
      <startdate>2017-02-12</startdate>
      <enddate>2017-07-17</enddate>
    </workplace>
  </person>
</socialnetwork>
```

1. Запит, що виводить ім'я і прізвище всіх людей, які працювали в Google, якщо вони обоє існують (конкатенує їх через space і виводить), якщо ні, то повертає порожню конструкцію:

Pattern matching / XPath 3.1 / XQuery 3.1 / CSS 3 Selector Online Tester

(You can find the documentation below)

HTML/XML/JSON-Input file:

☐ Pattern matching ☐ XPath 3.1 ☒ XQuery 3.1 ☐ CSS 3.0 selectors ☐ Autodetect

```
1 <socialnetwork>
2   <person>
3     <id>2</id>
4     <firstname>Oleksandr</firstname>
5     <lastname>Zhenchenko</lastname>
6     <birthdate>2003-12-28</birthdate>
7     <address>Lviv, Hnata Hotkevycha 1</address>
8     <workplace>
9       <company>Amazon</company>
10      <startdate>2016-12-12</startdate>
11      <enddate>2016-12-27</enddate>
12    </workplace>
13    <workplace>
14      <company>Facebook</company>
15      <startdate>2016-12-28</startdate>
16      <enddate>2017-02-21</enddate>
17    </workplace>
18    <workplace>
19      <company>Starlink</company>
20      <startdate>2017-02-12</startdate>
21      <enddate>2017-07-17</enddate>
22    </workplace>
23  </person>
24 </socialnetwork>
```

```
1 for $person in /socialnetwork/person
2 where $person/workplace/company = 'Google'
3 return if (exists($person/firstname) and exists($person/lastname))
4         then $person/firstname || ' ' || $person/lastname
5         else ()
```

☐ disable auto refresh ☐ disable syntax highlighting
Output Options: Node format: Output format: ☐ Show types ☐ Hide variable names
Compatibility: ☐ Enable all extensions
Old languages: ☐ XPath 2.0 ☐ XPath 3.0 ☐ XQuery 1.0 ☐ XQuery 3.0

New languages: ☐ XPath 4.0 ☐ XQuery 4.0

Result of the above expression applied to the above input file:

```
1 Olena Hatala
2 Sofia Hoshko
3 QWERTY Tymchyshyn
4
```

for \$person in /socialnetwork/person
where \$person/workplace/company = 'Google'
return if (exists(\$person/firstname) and exists(\$person/lastname))
 then \$person/firstname || ' ' || \$person/lastname

2. Запит, який знаходить середній вік усіх людей в документі:

Pattern matching / XPath 3.1 / XQuery 3.1 / CSS 3 Selector Online Tester

(You can find the documentation below)

HTML/XML/JSON-Input file:
☐ Pattern matching
☐ XPath 3.1
☒ XQuery 3.1
☐ CSS 3.0 selectors
☐ Autodetect

```

1 <socialnetwork>
2   <person>
3     <id>2</id>
4     <firstname>Oleksandr</firstname>
5     <lastname>Zhenchenko</lastname>
6     <birthdate>2003-12-28</birthdate>
7     <address>Lviv, Hnata Hotkevycha 1</address>
8     <workplace>
9       <company>Amazon</company>
10      <startdate>2016-12-12</startdate>
11      <enddate>2016-12-27</enddate>
12    </workplace>
13    <workplace>
14      <company>Facebook</company>
15      <startdate>2016-12-28</startdate>
16      <enddate>2017-02-21</enddate>
17    </workplace>
18    <workplace>
19      <company>P&G</company>
20    </workplace>

```

```

1 let $currentYear := xs:integer(format-dateTime(current-dateTime(), "[Y]"))
2 return avg(for $person in /socialnetwork/person
3           return if (exists($person/birthdate))
4                 then $currentYear - xs:integer(substring($person/birthdate, 1, 4))
5                 else ())

```

☐ disable auto refresh
☐ disable syntax highlighting

Output Options: Node format:
Output format:
☐ Show types
☐ Hide variable names

Compatibility:

Old languages:
☐ XPath 2.0
☐ XPath 3.0
☐ XQuery 1.0
☐ XQuery 3.0

New languages:
☐ XPath 4.0
☐ XQuery 4.0

Result of the above expression applied to the above input file:

1	19.833333333333333
2	

```

let $currentYear := xs:integer(format-dateTime(current-dateTime(), "[Y]"))
return avg(for $person in /socialnetwork/person
  return if (exists($person/birthdate))
    then $currentYear - xs:integer(substring($person/birthdate, 1, 4))
    else ())

```

3. Запит, який виводить людей, які ніколи ніде не працювали:

Pattern matching / XPath 3.1 / XQuery 3.1 / CSS 3 Selector Online Tester

(You can find the documentation below)

HTML/XML/JSON-Input file:

☐ Pattern matching
☐ XPath 3.1
☒ XQuery 3.1
☐ CSS 3.0 selectors
☐ Autodetect

```

1 <socialnetwork>
2   <person>
3     <id>2</id>
4     <firstname>Oleksandr</firstname>
5     <lastname>Zherchenko</lastname>
6     <birthdate>2003-12-28</birthdate>
7     <address>Lviv, Hnata Hotkevycha 1</address>
8     <workplace>
9       <company>Amazon</company>
10      <startdate>2016-12-12</startdate>
11      <enddate>2016-12-27</enddate>
12    </workplace>
13    <workplace>
14      <company>Facebook</company>
15      <startdate>2016-12-28</startdate>
16      <enddate>2017-02-21</enddate>
17    </workplace>
18    <workplace>
19
20

```

```

1 for $person in /socialnetwork/person
2 where empty($person/workplace)
3 return typeswitch ($person)
4   case element(person) return $person
5   default return ()
6

```

☐ disable auto refresh
☐ disable syntax highlighting

Output Options:
Node format:
Output format:
☐ Show types
☐ Hide variable names

Compatibility:
☐ Enable all extensions

Old languages:
☐ XPath 2.0
☐ XPath 3.0
☐ XQuery 1.0
☐ XQuery 3.0

New languages:
☐ XPath 4.0
☐ XQuery 4.0

Result of the above expression applied to the above input file:

```

1      6
2      bad
3      user
4      2002-10-21
5      Lviv, Shevchenka 1
6
7
8
9      7
10     7
11     7
12     2001-08-01
13     address7

```

```
for $person in /socialnetwork/person
where empty($person/workplace)
return typeswitch ($person)
  case element(person) return $person
  default return ()
```

4. Запит, який повертає ім'я, прізвище та дату народження найстаршої людини:

Pattern matching / XPath 3.1 / XQuery 3.1 / CSS 3 Selector Online Tester

(You can find the documentation below)

HTML/XML/JSON-Input file:

☐ Pattern matching ☐ XPath 3.1 ☒ XQuery 3.1 ☐ CSS 3.0 selectors ☐ Autodetect

```
1 <socialnetwork>
2 <person>
3   <id>2</id>
4   <firstname>Oleksandr</firstname>
5   <lastname>Zhenchenko</lastname>
6   <birthdate>2003-12-28</birthdate>
7   <address>Lviv, Hnata Hotkevycha 1</address>
8   <workplace>
9     <company>Amazon</company>
10    <startdate>2016-12-12</startdate>
11    <enddate>2016-12-27</enddate>
12  </workplace>
13  <workplace>
14    <company>Facebook</company>
15    <startdate>2016-12-28</startdate>
16    <enddate>2017-02-21</enddate>
17  </workplace>
18 </person>
19 </socialnetwork>
```

```
1 let $oldest := /socialnetwork/person[not(empty(birthdate))]
2 | (., xs:date(birthdate))
3 order by xs:date(birthdate) descending
4 return if (exists($oldest[1]/firstname) and exists($oldest[1]/lastname))
5 then ($oldest[1]/firstname || ' ' || $oldest[1]/lastname, $oldest[1]/birthdate)
6 else ()
7
```

☐ disable auto refresh ☐ disable syntax highlighting

Output Options: Node format: Output format: ☐ Show types ☐ Hide variable names

Compatibility: ☐ Enable all extensions

Old languages: ☐ XPath 2.0 ☐ XPath 3.0 ☐ XQuery 1.0 ☐ XQuery 3.0

New languages: ☐ XPath 4.0 ☐ XQuery 4.0

Result of the above expression applied to the above input file:

```
1 Oleksandr Zhenchenko
2 2003-12-28
3
```

```
let $oldest := /socialnetwork/person[not(empty(birthdate))]
! (., xs:date(birthdate))
order by xs:date(birthdate) descending
return if (exists($oldest[1]/firstname) and exists($oldest[1]/lastname))
then ($oldest[1]/firstname || ' ' || $oldest[1]/lastname, $oldest[1]/birthdate)
else ()
```

Висновок: під час виконання цієї лабораторної роботи, я практично ознайомився з синтаксисом мови XQuery та конструюванням запитів даних XML документа.