

## **Бази даних та інформаційні системи**

### **ЛАБОРАТОРНА РОБОТА №7**

#### **XSLT – мова перетворення XML-документів**

Виконав:

Ст. Заречанський

Олексій

Група ПМІ-33

Оцінка

Прийняв:

ас. Жировецький В.В.

# Тема

Вивчення мови XSLT - мови перетворення XML-документів.

## Мета роботи

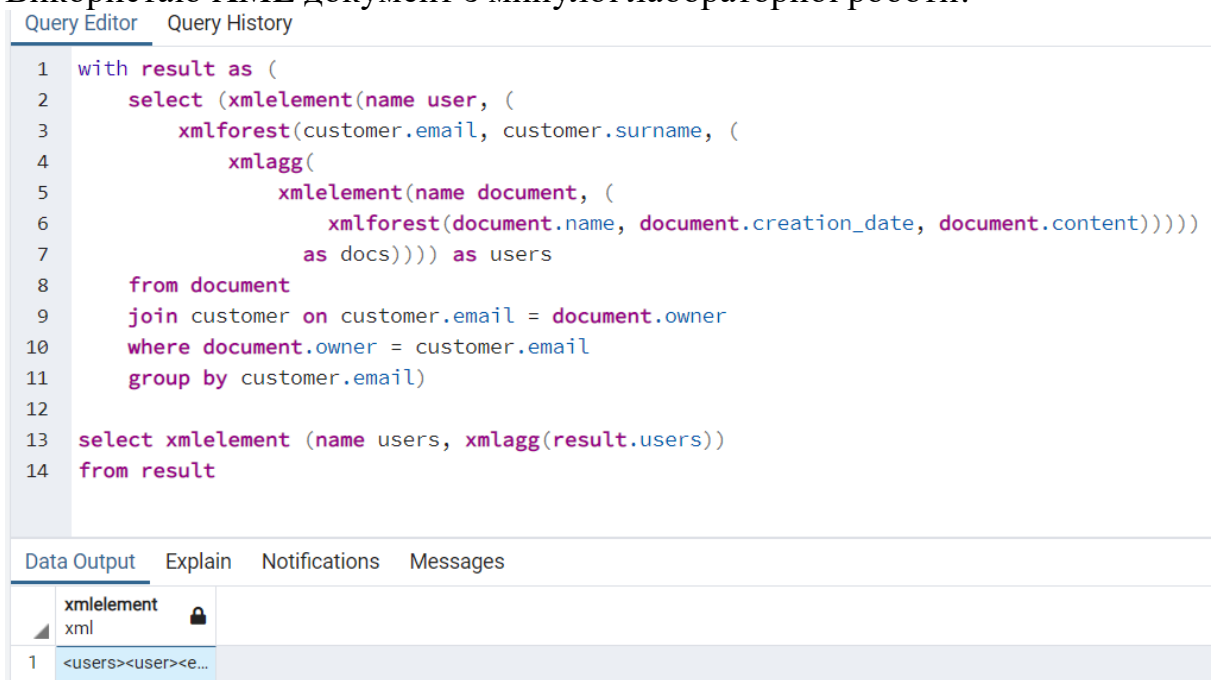
Ознайомлення з синтаксисом мови XSLT та отримання практичних навичок перетворенням фрагмента XML документа з використанням XSLT.

## Завдання

Розробити базу даних для web представлення системи управління документами. Система підтримує інформацію про документи, які проходять через неї. Крім власне документа важливими є інформація про його створення, історія змін, користувачі, які вносили зміни, та інформація про оффлайнове представлення документа (паперовий, CD, flash). Користувачі можуть створювати і змінювати документи і асоціювати їх з оффлайновим представленням, а також шукати документи за користувачами, які змінювали їх, станом чи представленням.

## Звіт

1. Використаю XML документ з минулої лабораторної роботи:



The screenshot shows a SQL query editor with a query that uses XQuery functions like `xmlelement`, `xmlforest`, and `xmlagg` to generate an XML document. The query is as follows:

```
1 with result as (  
2     select (xmlelement(name user, (  
3         xmlforest(customer.email, customer.surname, (  
4             xmlagg(  
5                 xmlelement(name document, (  
6                     xmlforest(document.name, document.creation_date, document.content))))  
7             as docs)))) as users  
8     from document  
9     join customer on customer.email = document.owner  
10    where document.owner = customer.email  
11    group by customer.email)  
12  
13 select xmlelement (name users, xmlagg(result.users))  
14 from result
```

Below the query editor, the 'Data Output' tab is active, showing the XML result of the query. The output is an XML document with a root element `<users>` containing a list of `<user>` elements, each with a list of associated `<document>` elements.

З цієї квері з минулої лабораторної роботи ми отримали XML документ, який містить всіх користувачів, які є власниками документів в базі даних. Формується XML документ де вказані дані користувача та список всіх документів власниками яких вони є, з даними про ці документи.

Результат виконання запиту:

```
<?xml version="1.0" encoding="UTF-8"?>
<users>
  <user>
    <email>anotheremail228@mail.ua</email>
    <surname>Chungus</surname>
    <docs>
      <document>
        <name>Third document</name>
        <creation_date>2021-03-01</creation_date>
        <content>Another user</content>
      </document>
      <document>
        <name>Empty document</name>
        <creation_date>2021-03-01</creation_date>
        <content>empty</content>
      </document>
    </docs>
  </user>
  <user>
    <email>anotheremail@gmail.com</email>
    <surname>Floppa</surname>
    <docs>
      <document>
        <name>Last document</name>
        <creation_date>2021-03-01</creation_date>
        <content>Not empty</content>
      </document>
    </docs>
  </user>
  <user>
    <email>last@mail.ua</email>
    <surname>Shreker</surname>
    <docs>
      <document>
        <name>Second document</name>
        <creation_date>2021-03-01</creation_date>
        <content>Some more content</content>
      </document>
      <document>
        <name>First document</name>
        <creation_date>2021-03-01</creation_date>
        <content>Some content12</content>
      </document>
    </docs>
  </user>
</users>
```

## 2. Створюємо шаблон перетворення XSLT:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <xsl:stylesheet version="1.0"
4  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
5
6  <xsl:template match="/">
7  <html>
8  <body>
9  <h2>Users</h2>
10 <xsl:for-each select="users/user">
11   <h3>
12     <xsl:number format="1. " level="any" />
13     <xsl:value-of select="email" />
14   </h3>
15   <table border="1">
16     <tr bgcolor="#9acd32">
17       <th>Name</th>
18       <th>Creation date</th>
19       <th>Content</th>
20     </tr>
21     <xsl:for-each select="docs/document">
22       <xsl:sort select="name" />
23       <xsl:if test="not(contains(content, 'empty'))">
24         <tr>
25           <td><xsl:value-of select="name" /></td>
26           <td><xsl:value-of select="creation_date" /></td>
27           <td><xsl:value-of select="content" /></td>
28         </tr>
29       </xsl:if>
30     </xsl:for-each>
31   </table>
32 </xsl:for-each>
33 </body>
34 </html>
35 </xsl:template>
36
37 </xsl:stylesheet>
```

Цей шаблон має заголовок Users після якого за допомогою xsl:number та xsl:for-each пронумеровано кожного користувача в XML документі в форматі “1. Емейл користувача”. Для кожного користувача створений заголовок з його емейлом, та таблиця його документів. В таблиці документів знову завдяки xsl:for-each перечислені документи користувача. При цьому вони відфільтровуються так, щоб їх вміст не мав слова ‘empty’ за допомогою xsl:if. Також відфільтровані документи сортуються за назвою в алфавітному порядку за допомогою xsl:sort. Для отримання значень емейлу користувачів, імені, дати створення і вмісту документів використовується xsl:value-of.

### 3. Використаємо онлайн утиліту для перетворення за шаблоном:

Enter your XML code or it's URL:

```
<?xml version="1.0" encoding="UTF-8" ?>
<document>
  </docs>
  </user>
  <user>
    <email>anotheremail@gmail.com</email>
  </user>
</document>
```

Enter your XSLT or it's URL:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html" />
  <xsl:template match="/">
    <table border="1">
      <tr>
        <th>Name</th>
        <th>Creation date</th>
        <th>Content</th>
      </tr>
      <xsl:for-each select="docs/document">
        <xsl:sort select="name" />
        <xsl:if test="not(contains(content, 'empty'))">
          <tr>
            <td><xsl:value-of select="name" /></td>
            <td><xsl:value-of select="creation-date" /></td>
            <td><xsl:value-of select="content" /></td>
          </tr>
        </xsl:if>
      </xsl:for-each>
    </table>
  </template>
</xsl:stylesheet>
```

The result is:

```
<html>
<body>
  <h2>Users</h2>
  <h3>1. anotheremail228@mail.ua</h3>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Name</th>
      <th>Creation date</th>
      <th>Content</th>
    </tr>
    <tr>
      <td>Third document</td>
      <td>2021-03-01</td>
      <td>Another user</td>
    </tr>
  </table>
  <h3>2. anotheremail@gmail.com</h3>
  <table border="1">
    <tr>
      <td>First document</td>
      <td>2021-03-01</td>
      <td>Some content12</td>
    </tr>
    <tr>
      <td>Second document</td>
      <td>2021-03-01</td>
      <td>Some more content</td>
    </tr>
  </table>
  <h3>3. last@mail.ua</h3>
  <table border="1">
    <tr>
      <td>First document</td>
      <td>2021-03-01</td>
      <td>Some content12</td>
    </tr>
    <tr>
      <td>Second document</td>
      <td>2021-03-01</td>
      <td>Some more content</td>
    </tr>
  </table>

```

[Run Transformation](#) [Open Result](#)

Бачимо що результат успішний, спробуємо його запустити:

**Transformation Result:**

[Close](#) **Users**

**1. anotheremail228@mail.ua**

Name	Creation date	Content
Third document	2021-03-01	Another user

**2. anotheremail@gmail.com**

Name	Creation date	Content
------	---------------	---------

**3. last@mail.ua**

Name	Creation date	Content
First document	2021-03-01	Some content12
Second document	2021-03-01	Some more content

Все пройшло успішно:

- Всі користувачі пронумеровані.
- Документи посортовані по назві (таблиця 3).
- Документів зі словом “empty” немає в таблицях.
- Всі решта документів та користувачі присутні.