

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №14

Тема: «Мова виразів XPath»

Виконав:

Ст. Лук'янчук Денис

Група ПМІ-33

2025

Тема: «Мова виразів XPath».

Мета роботи: Вивчення мови виразів XPath та створення виразів XPath для ефективнішого використання XSLT та XQuery.

Завдання лабораторної роботи:

- Опрацювати теоретичний матеріал.
- Написати декілька виразів XPath для власного XML-документа.
- Використати онлайн XPath-тестер для перевірки коректності виразів.
- Оформити звіт із результатами перевірки.

Теоретичний матеріал:

XPath — це мова запитів для навігації та вибору вузлів у структурі XML-документа. Вона дозволяє отримувати дані за допомогою шляхів, предикатів, осей та функцій.

Основні можливості XPath:

- вибір вузлів елементів та атрибутів (//element, @attribute);
- фільтрація за умовами ([condition]);
- виконання обчислень і статистичних функцій (count(), sum(), avg(), min(), max()).

Хід роботи

Для виконання роботи використано файл xml файл:

```
<shop>
  <customer>
    <id>1</id>
    <name>Customer_1</name>
    <email>user1@example.com</email>
    <orders>
      <order>
        <order_id>16</order_id>
        <date>2025-02-17T23:49:18.683291</date>
        <status>New</status>
        <quantity>4</quantity>
        <part>
          <serial>SN000064</serial>
          <part_name>Part_64</part_name>
          <price>306.80</price>
        </part>
      </order>
    </orders>
  </customer>
  <customer>
    <id>2</id>
    <name>Customer_2</name>
    <email>user2@example.com</email>
    <orders>
      <order>
        <order_id>17</order_id>
        <date>2025-02-17T23:49:18.683291</date>
        <status>New</status>
        <quantity>2</quantity>
        <part>
          <serial>SN000065</serial>
          <part_name>Part_65</part_name>
          <price>200.00</price>
        </part>
      </order>
    </orders>
  </customer>
  <customer>
    <id>3</id>
    <name>Customer_3</name>
    <email>user3@example.com</email>
    <orders>
      <order>
        <order_id>18</order_id>
        <date>2025-02-17T23:49:18.683291</date>
        <status>New</status>
        <quantity>1</quantity>
        <part>
          <serial>SN000066</serial>
          <part_name>Part_66</part_name>
          <price>100.00</price>
        </part>
      </order>
    </orders>
  </customer>
  <customer>
    <id>4</id>
    <name>Customer_4</name>
    <email>user4@example.com</email>
    <orders>
      <order>
        <order_id>19</order_id>
        <date>2025-02-17T23:49:18.683291</date>
        <status>New</status>
        <quantity>3</quantity>
        <part>
          <serial>SN000067</serial>
          <part_name>Part_67</part_name>
          <price>250.00</price>
        </part>
      </order>
    </orders>
  </customer>
</shop>
```

Створені XPath-вирази:

1. //order/status - вибір усіх елементів <status> із документа

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

//order/status

Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

```
Element='<status>Completed</status>'  
Element='<status>Processing</status>'  
Element='<status>Processing</status>'  
Element='<status>Completed</status>'  
Element='<status>Cancelled</status>'  
Element='<status>Cancelled</status>'  
Element='<status>New</status>'  
Element='<status>New</status>'  
Element='<status>Cancelled</status>'  
Element='<status>New</status>'  
Element='<status>Cancelled</status>'  
Element='<status>Cancelled</status>'  
Element='<status>New</status>'  
Element='<status>Completed</status>'  
Element='<status>Processing</status>'  
Element='<status>New</status>'  
Element='<status>New</status>'  
Element='<status>Processing</status>'  
Element='<status>Completed</status>'
```

Copy Save

2. //order[status='Completed']/part/price - вибір цін усіх деталей замовень, які мають статус Completed

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

//order[status='Completed']/part/price

Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

```
Element='<price>182.23</price>'  
Element='<price>166.18</price>'  
Element='<price>661.17</price>'  
Element='<price>253.75</price>'  
Element='<price>673.47</price>'  
Element='<price>453.44</price>'  
Element='<price>225.82</price>'  
Element='<price>710.65</price>'  
Element='<price>914.26</price>'  
Element='<price>101.61</price>'  
Element='<price>675.98</price>'  
Element='<price>851.39</price>'  
Element='<price>186.79</price>'  
Element='<price>826.98</price>'  
Element='<price>640.53</price>'  
Element='<price>110.01</price>'  
Element='<price>509.85</price>'  
Element='<price>757.81</price>'  
Element='<price>406.79</price>'
```

Copy Save

3. **count(//customer)** - підрахунок кількості клієнтів у базі

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

count(//customer) Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

Int64='100'

Copy Save

4. **count(//order)** - кількість усіх замовлень у документі

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

count(//order) Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

Int64='100'

Copy Save

5. **sum(//order/part/price)** - сума всіх цін деталей

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

sum(//order/part/price) Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

Double='49774.1999999999'

Copy Save

6. $\text{avg}(\text{//order/part/price})$ - середня ціна деталей

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

`avg(//order/part/price)`

Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

Double='497.7419999999999'

Copy Save

7. $\text{//order}[part/price}>800\text{]/part/part_name}$ - назви деталей, дорожчих за 800

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

`//order[part/price>800]/part/part_name`

Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

Element='<part_name>Part_51</part_name>'
Element='<part_name>Part_7</part_name>'
Element='<part_name>Part_71</part_name>'
Element='<part_name>Part_68</part_name>'
Element='<part_name>Part_44</part_name>'
Element='<part_name>Part_51</part_name>'
Element='<part_name>Part_94</part_name>'
Element='<part_name>Part_7</part_name>'
Element='<part_name>Part_59</part_name>'
Element='<part_name>Part_71</part_name>'
Element='<part_name>Part_2</part_name>'
Element='<part_name>Part_23</part_name>'
Element='<part_name>Part_44</part_name>'

Copy Save

8. $\text{//order}[status='Cancelled']/count(part)$ - кількість відмінених замовлень

Option 2: Or upload your XML file File encoding

Огляд... Файл не вибрано. UTF-8

XPath expression

`count(//order[status='Cancelled'])`

Include 'XML Item Type' in output

Evaluate XPath Evaluate XPath to new window

-XPath Result-

Int64='25'

Copy Save

9. //customer[orders/order/status='Processing']/name - імена клієнтів, чиї замовлення обробляються

The screenshot shows an online tool for evaluating XPath expressions against XML files. At the top, there's a file upload section with a placeholder 'Option 2: Or upload your XML file' and a note 'Файл не вибрано.' Below it, the 'File encoding' is set to 'UTF-8'. The 'XPath expression' field contains the query `//customer[orders/order/status='Processing']/name`. To the right of the expression is a toggle switch labeled 'Include 'XML Item Type' in output', which is turned off. Below the expression are two buttons: 'Evaluate XPath' (highlighted in blue) and 'Evaluate XPath to new window'. The main result area is titled '-XPath Result-' and displays a list of XML elements. The list consists of 24 entries, each starting with 'Element=' followed by an XML element tag for a customer name, such as 'Customer_5', 'Customer_8', etc., up to 'Customer_82'. To the right of the result list are two buttons: 'Copy' and 'Save'.

```
Element='<name>Customer_5</name>
Element='<name>Customer_8</name>
Element='<name>Customer_12</name>
Element='<name>Customer_14</name>
Element='<name>Customer_15</name>
Element='<name>Customer_17</name>
Element='<name>Customer_19</name>
Element='<name>Customer_34</name>
Element='<name>Customer_38</name>
Element='<name>Customer_41</name>
Element='<name>Customer_48</name>
Element='<name>Customer_52</name>
Element='<name>Customer_53</name>
Element='<name>Customer_54</name>
Element='<name>Customer_60</name>
Element='<name>Customer_63</name>
Element='<name>Customer_66</name>
Element='<name>Customer_67</name>
Element='<name>Customer_82</name>'
```

Висновок:

Під час виконання цієї лабораторної роботи було опрацьовано принципи побудови XPath-виразів, використано основні функції для вибірки та аналізу даних із XML-документа. Створені запити дозволили отримати підсумкову статистику, відібрати потрібні елементи та здійснити навігацію в ієрархії вузлів. Робота далі мені практичні навички застосування XPath у поєднанні з XML для подальшої роботи з XSLT та XQuery.

Відповіді на контрольні питання

1. Що означає назва мови XPath?

XPath (XML Path Language) — це мова для навігації (пошуку та вибору) елементів і атрибутів у XML-документі. Назва буквально означає “мова шляхів для XML”.

2. Що таке порядок документа і яким обмеженням він має задовільнити?

Порядок документа — це послідовність розташування вузлів у XML-файлі від початку до кінця.

Обмеження:

- кореневий елемент йде першим;
- усі дочірні елементи розташовані після свого батьківського;

- порядок збережений так, як у вихідному документі (XPath опрацьовує вузли саме в цьому порядку).

3. Який взаємозв'язок вузлів XML-документа?

Вузли XML утворюють ієрархічне дерево, де:

- кореневий вузол → найвищий рівень;
- елементи → батьки та нащадки (діти);
- вузли одного рівня → сусіди (брати/сестри);
- атрибути належать до певного елемента, але не є його дітьми.

4. Які вирази шляху у короткій формі використовуються для вибору вузлів?

Приклади коротких форм XPath:

- / — корінь документа;
- // — вибір усіх вузлів незалежно від рівня вкладеності;
- . — поточний вузол;
- .. — батьківський вузол;
- @ — атрибут.

5. Що таке абсолютний і відносний вирази шляху?

- Абсолютний шлях починається з / і описує повний шлях від кореня.
Напр.: /catalog/book/title
- Відносний шлях починається без / і шукає вузли від поточного місця.
Напр.: book/title

6. Наведіть приклад предикатів для пошуку конкретного вузла або вузла, який містить конкретне значення.

- //book[@id="b2"] — вибирає елемент <book> з атрибутом id="b2".
- //book[price>100] — вибирає книги, де ціна більша за 100.
- //book[author="King"] — вибирає книги автора King.

7. Наведіть приклад вибору невідомих XML-вузлів.

- /* — вибирає усі вузли документа.
- //@* — вибирає усі атрибути у документі.
- //book/* — усі дочірні елементи вузла <book>.

8. Як у виразі XPath можна вибрати кілька шляхів?

Використовується оператор об'єднання — вертикальна риска |.
Приклад:

```
//title | //author
```

Цей вираз вибирає всі вузли <title> і <author> у документі.