

## Лабораторна робота № 5

### Мова запитів XQuery в СУБД BaseX

#### Мета роботи

вивчити склад та основні терміни мови запитів до XML-баз даних XQuery, порядок створення та виконання запитів на мові XQuery в СУБД BaseX

#### Короткі теоретичні відомості

##### 1. Основні терміни мови запитів до XML-баз даних XQuery

Ще однією з важливих функцій діалогового інтерфейсу СУБД BaseX є підтримання мови запитів до XML-даних XQuery. XQuery – це мова запитів до баз даних XML, яка за призначенням є подібною до мови SQL для реляційних баз даних. Склад та структуру мови XQuery регламентують рекомендації W3C (*World Wide Web Consortium*, W3C – головна міжнародна організація, що розробляє й впроваджує технологічні стандарти для роботи в мережі Інтернет). Сьогодні актуальним є стандарт XQuery 1.0. Детальний опис основних положень цього документу подано на сайті W3C ([https://www.w3schools.com/xml/xquery\\_intro.asp](https://www.w3schools.com/xml/xquery_intro.asp)). Застосування XQuery як засобу для побудови та виконання запитів до XML-даних ґрунтується на таких засадах, визначених W3C:

- XQuery визначають як мову запитів до баз даних XML (на відміну від XPath),
- XQuery виконує щодо баз даних XML функції, аналогічні мові SQL щодо реляційних баз даних,
- XQuery базується на виразах та способах опису даних, які визначено для мови XPath,
- XQuery підтримують не тільки XML СУБД, а й більшість інших СУБД (в тому числі SQL-сумісних),
- XQuery є засобом, який рекомендовано W3C.

Мови XQuery 1.0 і XPath 3.0 використовують однакову модель даних, підтримують однаковий набір функцій і операторів. В XQuery оперують аналогічними XPath поняттями вершини (Nodes), атомарні величини (Atomic values), одиниці даних (Items), шлях (Path) тощо. Тому для розуміння основних принципів XQuery, необхідно володіти знаннями XPath.

Основними завданнями мови XQuery є такі:

- видобування даних у Web-сервісах,
- генерування підсумкових звітів,
- перетворення даних з формату XML до формату HTML,
- пошук релевантних даних у Web-ресурсах,

Мова XQuery є сумісною зі стандартами W3C, такими як XML, Namespaces, XSLT, XPath і XML-схеми.

Синтаксис мови XQuery ґрунтується на таких принципах:

- вирази XQuery є чутливими до регістру,
- в XQuery елементи, атрибути і змінні мають мати дійсні і підтверджені XML-імена,
- символічні значення подають в одинарних або подвійних лапках,
- імена змінних в XQuery обов'язково починають символом \$, за яким слідує сама ім'я,
- коментарі в XQuery обмежують парою дужок і двокрапок (: та :).

В XQuery використовують п'ять основних операторів – For, Let, Where, Order by, Return, які прийнято позначати акронімом FLWOR (читають як "flower"). Призначенням цих операторів є

- For – задає операцію вибору послідовності вершин,
- Let – пов'язує змінну з послідовністю вершин або іншим значенням (константою),
- Where – описує фільтр для вершин,
- Order by – визначає порядок сортування вершин,
- Return – формує і повертає результат виконання запиту.

За допомогою цих операторів створюють XQuery-запити для пошуку і вибору даних з XML-документів та формування кінцевого результату у визначеному форматі.

##### 2. Створення і виконання запитів на мові XQuery в СУБД BaseX

Запити XQuery в СУБД BaseX формують і виконують в діалоговому режимі за допомогою вбудованого редактора (того самого, до якого за допомогою якого редагують XML-дані). Для виконання запиту використовують кнопку із зеленою стрілкою. Результати виконання запитів відображають у вікні Result. Побудований XQuery-запит може бути збережено для повторного виконання.

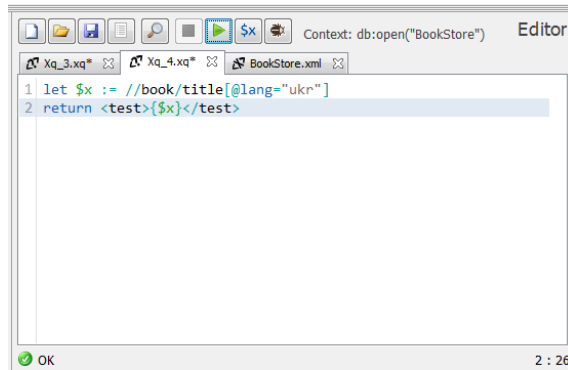


Рис 5.1. Вікно редактора запитів XQuery в СУБД BaseX

Призначення, формат запису та порядок застосування операторів XQuery є таким.

**Оператор return.** Цей оператор є необхідним елементом у кожному запиті, за його допомогою формують і повертають результат виконання запиту. Формат оператора є таким:

*return вираз*

де виразом є ім'я змінної, або ім'я змінної обмежене тегами, або ланцюжок символів, обмежених тегами, або послідовність змінних та символічних ланцюжків, обмежених тегами. Якщо вираз утворюють з використанням тегів та символічних ланцюжків, то імена змінних подають у фігурних дужках. Наприклад, якщо змінній *\$s* присвоєне значення "Тест оператора return", то результати будуть наступними:

1. оператор виду

*return \$s*

поверне ланцюжок символів

Тест оператора return

2. оператор виду

*return <Результат>{\$s}</Результат>*

поверне результат

*<Результат>Тест оператора return</Результат>*

3. оператор виду

*return <Результат>*

*<Частина\_1>Це тестовий результат*

*{ \$s }</Частина\_1>*

*<Частина\_2>Кінець тесту</Частина\_2>*

*</Результат>*

виведе в результаті

*<Результат>*

*<Частина\_1> Це тестовий результат*

Тест оператора return</Частина\_1>

*<Частина\_2>Кінець тесту</Частина\_2>*

*</Результат>*

Якщо змінній *\$h* присвоєно значення набору вершин, наприклад */Catalog/book/title* то оператор

*return \$h*

поверне в результаті набір вершин з тегом *, які містять відомості про назви книг –*

*<title lang="ukr">Організація баз даних</title>*

*<title lang="ukr">Системи управління базами даних</title>*

*<title lang="en">XQuery Kick Start</title>*

*<title lang="ukr">Нереляційні бази даних</title>*

**Оператор let.** За допомогою цього оператора присвоюють значення змінній. Формат оператора –

*let \$змінна:=значення,*

де *\$змінна* – ім'я змінної, якій присвоюють значення сформоване згідно правил синтаксису XQuery,

*:=* – символ присвоєння,

*значення*, яке присвоюють змінній може бути

1. символічною константою – ланцюжком символів, обмеженим лапками (" або ' '), наприклад

*let \$s:="Ланцюжок символів"*

2. набором символьних констант, обмеженим дужками ( та ), який подають у форматі  
`let $s:=("aaaa","bbbb","cccc","dddd")`

3. набором чисел, обмеженим дужками ( та ), який подають у форматі  
`let $s:=(1,2,3,4,5)`

4. діапазоном чисел, обмеженим дужками ( та ) який подають у форматі  
`let $s:=(0 to 9)`

5. посиланням на множину вершин у форматі  
`let $h:=/Catalog/book/title`

6. посиланням на множину вершин з використанням фільтру відбору у форматі  
`let $h:=/Catalog/book/title[@lang="ukr"]`

**Оператор for.** За допомогою оператора `for` деяку змінну пов'язують з набором вершин або набором значень. Формат оператора

*for \$змінна in набір*

де *\$змінна* – ім'я змінної, якій присвоюють значення сформоване згідно правил синтаксису XQuery, *набір* – шлях до набору вершин XML-документу, набір або діапазон числових значень або набір символьних констант, обмежені круглими дужками. Одночасно в операторі `for` можна виконати операцію пов'язування наборів з однією або декількома змінними. В цьому випадку, результат утворює декартів добуток наборів значень, що відповідають певним змінним. Приклади правильного написання оператора `for`

`for $x in /Catalog/book`

`for $x in (1,2,3)`

`for $x in (5 to 8)`

`for $x in ('aaaa','bbbb','cccc')`

`for $x in /Catalog/book, $y in (5 to 8)`

Для виконання запиту і отримання результату до його складу має бути включено оператор `return`. Приклади виконання запитів для розробленої тестової бази даних XML.

1. Запит виду

*for \$x in /Catalog/book*

*return \$x/title*

повертає результат

`<title lang="ukr">Організація баз даних</title>`

`<title lang="ukr">Системи управління базами даних</title>`

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

`<title lang="ukr">Нереляційні бази даних</title>`

2. При визначенні шляху до набору вершин додатково може бути використано фільтр, так запит

виду

*for \$x in /Catalog/book[title/@lang='en']*

*return \$x/title*

повертає результат

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

3. Запит з використанням наборів констант виду

*for \$x in (1,2,3)*

*return <Числове\_значення>{\$x}</Числове\_значення>*

дає результат

`<Числове_значення>1</Числове_значення>`

`<Числове_значення>2</Числове_значення>`

`<Числове_значення>3</Числове_значення>`

запит виду

*for \$x in ('aaaa','bbbb','cccc')*

*return <Символьне\_значення> {\$x} </Символьне\_значення>*

повертає в результаті

`<Символьне_значення>aaaa</Символьне_значення>`

`<Символьне_значення>bbbb</Символьне_значення>`

`<Символьне_значення>cccc</Символьне_значення>`

запит виду

```

for $x in (5 to 8)
return <Числове_значення>{$x}</Числове_значення>
відповідно, повертає
<Числове_значення>5</Числове_значення>
<Числове_значення>6</Числове_значення>
<Числове_значення>7</Числове_значення>
<Числове_значення>8</Числове_значення>

```

**Оператор where.** Оператор where у запитах XQuery застосовують разом з оператором for для формулювання критеріїв відбору вершин або значень. Критерій визначають за допомогою предикатів, які порівнюють значення XML-елемента або атрибута з деякою величиною на рівність, нерівність, співвідношення більше, не більше, менше або не менше. Декілька порівнянь в операторі where поєднують операторами кон'юнкції *and* або диз'юнкції *or*. Приклади виконання запитів з критеріями вибору.

1. Запит з вибором за значенням елемента XML-документу виду

```

for $x in /Catalog/book
where $x/price>=130
return $x/title

```

дає результат як перелік назв книг з ціною більшою або рівною значенню 130

```

<title lang="ukr">Організація баз даних</title>
<title lang="ukr">Нереляційні бази даних</title>

```

2. Запит з вибором за значенням атрибута

```

for $x in /Catalog/book
where $x/title/@lang="en"
return $x/title

```

утворює результат

```

<title lang="en">XQuery Kick Start</title>

```

3. Запит зі складеними критеріями виду

```

for $x in /Catalog/book
where $x/price>=100 and $x/publication/year >=2017
return $x/title

```

дає результат

```

<title lang="ukr">Системи управління базами даних</title>
<title lang="ukr">Нереляційні бази даних</title>

```

4. Критерії можна застосовувати не лише для вибору з наборів вершин, але і з наборів значень

```

for $x in (5 to 8)
where $x != 7
return <Числове_значення>{$x}</Числове_значення>

```

сформує набір чисел серед яких немає рівного 7

```

<Числове_значення>5</Числове_значення>
<Числове_значення>6</Числове_значення>
<Числове_значення>8</Числове_значення>

```

**Оператор order.** Оператор застосовують для впорядкування результатів виконання запитів за значенням визначеної вершини. порядок сортування задають додатково за допомогою параметра *ascending* – зростання або *descending* – спадання (коли порядок сортування не встановлено, за замовчуванням встановлюється зростання). Наприклад, запит виду

```

for $x in /Catalog/book
where $x/price>=40 and $x/price<=130
order by $x/publication/year ascending
return $x/title

```

дасть в результаті перелік назв книг в порядку зростання року видання

```

<title lang="en">XQuery Kick Start</title>
<title lang="ukr">Організація баз даних</title>
<title lang="ukr">Системи управління базами даних</title>

```

Аналогічний результат дає використання

```

order by $x/publication/year

```

Запит виду

```

for $x in /Catalog/book

```

*where \$x/price>=40 and \$x/price<=130*

*order by \$x/publication/year descending*

*return \$x/title*

впорядкує ті самі результати у зворотному порядку –

<title lang="ukr">Системи управління базами даних</title>

<title lang="ukr">Організація баз даних</title>

<title lang="en">XQuery Kick Start</title>

Використання операторів набору FLWOR забезпечує мову XQuery функціонально повний набір можливостей для повноцінного створення і виконання запитів до баз даних XML, зокрема в СУБД BaseX.

Більше про порядок формування і виконання запитів XQuery – на сайті W3C ([https://www.w3schools.com/xml/xquery\\_intro.asp](https://www.w3schools.com/xml/xquery_intro.asp)).

## **Хід роботи**

Для розробленої в попередній лабораторній роботі XML-бази даних за допомогою засобів діалогового середовища СУБД BaseX

1. розробити за зразками, поданими в даній інструкції і виконати запити на вибір даних з використанням операторів let – return,
2. розробити за зразками, поданими в даній інструкції і виконати запити на вибір даних з використанням операторів
  - for – return,
  - for – where – return,
  - for – where – order – return.
3. Порівняти результати виконання різних запитів.