ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Факультет прикладної математики та інформатики

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №16

Тема: «XQuery - мова запитів XML-документів»

Виконав:

Ст. *Лук'янчук Денис*

Група ПМІ-33

2025

Тема: «XQuery - мова запитів XML-документів».

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

Завдання лабораторної роботи:

- Опрацювати теоретичний матеріал.
- Створити декілька XQuery-запитів до свого XML-документа з використанням FLWOR.
- Обов’язково використати функції (count, sum, max, data, contains, тощо).
- Для отримання максимального балу застосувати деякі з виразів
- QuantifiedExpr, SwitchExpr, TypeswitchExpr, IfExpr, TryCatchExpr або OrExpr.
- Перевірити запити у будь-якому XQuery-процесорі.

Теоретичний матеріал:

XQuery — мова запитів до XML-документів, рекомендована W3C. Працює поверх XPath, розширюючи його можливості.

FLWOR

Назва від:

- **For** → перебір вузлів
- **Let** → визначення змінних
- **Where** → фільтрація
- **Order by** → сортування
- **Return** → формування результату

Типи виразів в XQuery:

- **IfExpr —** умовні конструкції
- **OrExpr —** логічні вирази
- **QuantifiedExpr —** some, every
- **SwitchExpr —** багатоваріантний вибір
- **TypeswitchExpr —** подібність до pattern matching
- **TryCatchExpr —** обробка помилок

Функції XQuery:

- **count()**
- **distinct-values()**
- **sum()**
- **max()**
- **data()**
- **contains()**
- **format-number()**
- **xs:dateTime()**

Хід роботи

Для виконання роботи використано мій xml файл:

Створені XQuery-запити:

1. Перелік запчастин до певного механізму

```
for $p in //part[part_name = "Part_35"]
return
<partInfo>
    <serial>{data($p/serial)}</serial>
    <name>{data($p/part_name)}</name>
    <price>{data($p/price)}</price>
</partInfo>

```

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

```

    SN000035
    Part_35
    119.65


    SN000035
    Part_35
    119.65


    SN000035
    Part_35
    119.65

```

2. Перелік усіх клієнтів та їх email

```
for $c in //customer
return
<customer>
    <name>{data($c/name)}</name>
    <email>{data($c/email)}</email>
</customer>

```

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

```

    Customer_1
    user1@example.com


    Customer_2
    user2@example.com


    Customer_3
    user3@example.com


    Customer_4
    user4@example.com


    Customer_5
    user5@example.com


    Customer_6
    user6@example.com


    Customer_7
    user7@example.com
```

3. Перелік механізмів та кількість запчастин до кожного

```
for $m in distinct-values(//part/part_name)
let $count := count(//part[part_name = $m])
order by $count descending
return
<mechanism>
    <name>{$m}</name>
    <parts_count>{$count}</parts_count>
</mechanism>

```

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

```

    Part_9
    5


    Part_24
    4


    Part_13
    3


    Part_18
    3


    Part_37
    3
```

4. Механізм з максимальною кількістю запчастин

```
let $all :=
    for $m in distinct-values(//part/part_name)
    let $count := count(//part[part_name = $m])
    return <item name="{$m}" count="{$count}"/>

let $max := max($all/@count)

for $i in $all[@count = $max]
return
<maxMechanism>
    <name>{data($i/@name)}</name>
    <count>{data($i/@count)}</count>
</maxMechanism>

```

Result of the above expression applied t

```

    Part_9
    5

```

5. IfExpr — визначення типу замовлення

(якщо ціна > 700 → дороге)

```
for $o in //order
return
<orderCheck>
  <customer>{data($o/../name)}</customer>
  <price>{data($o/part/price)}</price>
  {
    if ($o/part/price > 700)
    then <note>Дуже дороге замовлення</note>
    else <note>Звичайне замовлення</note>
  }
</orderCheck>

```

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

```


  306.80
  Звичайне замовлення


  110.01
  Звичайне замовлення


  182.23
  Звичайне замовлення


  987.36
  Дуже дороге замовлення


  182.23
  Звичайне замовлення


  166.18
  Звичайне замовлення

```

6. SwitchExpr — категоризація статусів замовлення

```
for $s in //order/status
return
<statusCategory>{
    switch ($s)
        case "Completed" return "Завершено"
        case "Cancelled" return "Скасовано"
        case "Processing" return "У роботі"
        default return "Інше"
}</statusCategory>

```

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

```
Інше
У роботі
У роботі
Інше
Завершено
Завершено
Інше
У роботі
Завершено
Завершено
Завершено
Скасовано
У роботі
Скасовано
У роботі
Завершено
У роботі
У роботі
```

7. QuantifiedExpr — чи існує хоча б одне дуже дороге замовлення (>900/1000)

```
some $price in //part/price satisfies ($price > 900)
```

Result of the above expression applied to t

```
true
```

```
some $price in //part/price satisfies ($price > 1000)
```

Result of the above expression applied t

```
false
```

Висновок: У результаті виконання лабораторної роботи було створено та протестовано декілька XQuery-запитів для обробки XML-документа. У роботі застосовано FLWOR-вирази разом із вбудованими функціями (count(), distinct-values(), max(), data()), що дозволило виконати пошук, сортування й агрегування даних.

Також були використані додаткові конструкції XQuery — IfExpr, SwitchExpr та QuantifiedExpr, що продемонструвало розширені можливості мови для умовної логіки, вибору та логічної перевірки даних.

Відповіді на контрольні питання

1. Для чого застосовується мова XQuery?

Мова XQuery використовується для пошуку, обробки й перетворення даних у форматі XML. Вона дозволяє виконувати вибірку даних, фільтрацію, сортування, об’єднання, побудову нових XML-структур, а також реалізовувати складні запити до XML-документів і XML-баз даних.

2. Які особливості синтаксису мови XQuery?

Основні особливості синтаксису XQuery:

- **ґрунтується на принципах XPath, розширюючи їх можливостями змінних, циклів та умов.**
- **підтримує конструкції FLWOR, if-then-else, switch, quantified expressions.**
- **дозволяє створювати нові XML-вузли прямо в запиті.**
- **має декларативний стиль: описує *що вибрати*, а не *як виконувати*.**
- **підтримує роботу з функціями, модулями та просторами імен.**

3. Що означає вираз FLWOR?

FLWOR — це основна конструкція XQuery, назва якої утворена від перших літер етапів обробки даних:

- **FOR – перебір елементів у послідовності**
- **LET – оголошення змінних**
- **WHERE – фільтрація даних**
- **ORDER BY – сортування**
- **RETURN – формування результату**

Це аналог SQL-запиту, але застосований до XML.

4. Приклад використання QuantifiedExpr

QuantifiedExpr — це вирази some або every, які повертають true/false.

Приклад: знайти книги, де *хоча б один* автор має ім’я Denys:

```
for $b in /library/book
where some $a in $b/authors/author satisfies ($a = "Denys")
return $b/title
```

Приклад “every”:

every $p in /orders/order/price satisfies ($p > 0)

5. Приклад використання SwitchExpr

Вираз switch дозволяє виконати різні дії залежно від значення змінної.

```
let $code := "UA"
return
switch ($code)
   case "UA" return "Україна"
   case "PL" return "Польща"
   case "DE" return "Німеччина"
   default return "Невідома країна"
```

6. Як використовується TryCatchExpr?

try/catch дозволяє обробляти помилки під час виконання XQuery-запиту.

Приклад:

```
try {
    xs:integer("abc")       (: викличе помилку :)
}
catch * {
    "Помилка: неможливо перетворити у число"
}
```

Механізм працює аналогічно до блоків try/catch у звичайних мовах програмування — якщо в блоці try виникає помилка, виконується блок catch.