

Wykład XII

Cechy języka XQUERY

Język funkcyjny

- wszystkie konstrukcje języka są wyrażeniami zwracającymi wartość
- można je zagnieżdżać
- wartości zwracane przez wyrażenia są tzw. sekwencjami (w sensie XPath 2.0)

Charakter przetwarzanej informacji

- na wejściu dokument lub fragment względnie kolekcja dokumentów lub fragmentów
- na wyjściu dokument lub fragment

Format języka

- wielkość liter ma znaczenie
- słowa kluczowe pisze się małymi literami

Sekwencje

- uporządkowany ciąg jednostek węzłów lub wartości atomowych
- pojedyncza jednostka jest tożsama z jednoelementową sekwencją
 - porządek w sekwencji zgodny z porządkiem dokumentu (jeśli nie użyto klauzuli order by)
 - jawny zapis sekwencji

Źródła danych

ścieżka **XPath** użyta w **XQuery** musi zaczynać się od źródła danych
funkcje wbudowane :

doc(adres pliku xml) udostępnia dane zewnętrzne- repozytorium XML DB
ora:view(" nazwa tabeli relacyjnej ") w Oracle

Wyrażenia FLWOR

- akronim słów: **FOR**, **LET**, **WHERE**, **ORDER BY**, **RETURN**

- pozwala przejść po elementach zadanej sekwencji (klauzula **for**)
- zdefiniować zmienne pomocnicze (klauzula **let**)
- odfiltrować niepotrzebne elementy (klauzula **where**)
- uporządkować to, co pozostanie (klauzula **order by**)
- przekazać wynik w odpowiedniej postaci (klauzula **return**)

przykłady:

```
for $k in doc("plik.xml")//ksiazka
let $a:=$k/autorzy/autor
where count($a)=1
order by $k/cena
return $k/tytul
KLAUZULA for
```

Wykład XII

for \$i in (<opis><miejsce>Warszawa</miejsce></opis>,<kon/>)
return \$i

Odp:

<opis><miejsce>Warszawa</miejsce></opis>
<kon/>

for \$i in (1,2) , \$j in (3,4)
return <wynik> i = {\$i} , j = {\$j} </wynik>

Odp:

<wynik> i =1 , j =3 </wynik>
<wynik> i =1 , j =4 </wynik>
<wynik> i =2 , j =3 </wynik>
<wynik> i =2 , j =4 </wynik>

KLAUZULA let

let \$i :=(3,5,7)
return <out> {\$i} </out>

Odp:

<out> 3 5 7 </out>

let \$i :=(1,2) , \$j:=(3,4)
return <wynik> i = {\$i} , j = {\$j} </wynik>

Odp:

<wynik> i = 1 2 j = 3 4 </wynik>

*obie klauzule **for** i **let** służą do wiązania zmiennych , sposób w jaki to robią jest zupełnie różny*

for \$w in (<alfa/>,<beta/>,<gama/>)
return <wynik>{\$w}</wynik>

Odp:

<wynik><alfa/></wynik>
<wynik><beta/></wynik>
<wynik><gama/></wynik>

let \$w:=(<alfa/>,<beta/>,<gama/>)
return <wynik>{\$w}</wynik>

Odp:

<wynik><alfa></alfa><beta></beta><gama></gama></wynik>

Wykład XII

Przykłady: z wykorzystaniem tabel relacyjnych PRACOWNICY, ZESPOLY

```
for $i in ora:view("PRACOWNICY")  
return $i
```

Odp:

```
<ROW><NUMER>1000</NUMER><NAZWISKO>Lech</NAZWISKO><ETAT>Dyrektor</ETAT>  
<ZATRUDNIENIE>2001-01-01</ZATRUDNIENIE>  
<PLACA>9240</PLACA><DODATEK>570</DODATEK><ID_ZESP>10</ID_ZESP></ROW>  
<ROW>  
...
```

```
for $i in ora:view("PRACOWNICY")//NAZWISKO  
return $i
```

Odp:

```
<NAZWISKO>Lech</NAZWISKO>  
...
```

```
<ODPOWIEDZ>  
{  
for $i in ora:view("PRACOWNICY")//NAZWISKO  
return $i  
}  
</ODPOWIEDZ>
```

Odp:

```
<ODPOWIEDZ>  
<NAZWISKO>Lech</NAZWISKO>  
...  
<NAZWISKO>Kolski</NAZWISKO>  
</ODPOWIEDZ>
```

Łączenie danych

```
for $z in ora:view("ZESPOLY")//ID_ZESP  
let $p:= ora:view("PRACOWNICY")//ROW[ID_ZESP=$z]  
where count($p) > 3  
order by avg($p/PLACA) descending  
return  
<DUZY>  
{  
  $z,<LICZBA> {count($p) } </LICZBA>,  
  <SREDNIA> {avg($p/PLACA) } </SREDNIA>  
}  
</DUZY>
```

Wykład XII

Odp:

```
<DUZY><ID_ZESP>20</ID_ZESP><LICZBA>4</LICZBA><SREDNIA>3855</SREDNIA></DUZY>
<DUZY><ID_ZESP>30</ID_ZESP><LICZBA>4</LICZBA><SREDNIA>3615</SREDNIA></DUZY>
```

Grupowanie

```
for $p in distinct-values ( ora:view("PRACOWNICY")//ETAT/text())
let $l:=ora:view("PRACOWNICY")//ROW[ETAT=$p]
return
<ETAT Nazwa="{ $p}"> {count($l)} </ETAT>
```

Odp:

```
<ETAT Nazwa="Dyrektor">1</ETAT>
<ETAT Nazwa="Sekretarka">1</ETAT>
<ETAT Nazwa="Adiunkt">4</ETAT>
<ETAT Nazwa="Profesor">2</ETAT>
<ETAT Nazwa="Stażysta">3</ETAT>
<ETAT Nazwa="Asystent">3</ETAT>
```

Porządkowanie

```
for $p in ora:view("PRACOWNICY")/ROW
order by number($p/PLACA) descending
return <WYNIK> { $p/NAZWISKO } { $p/PLACA } </WYNIK>
```

Odp:

```
<WYNIK><NAZWISKO>Lech</NAZWISKO><PLACA>9240</PLACA></WYNIK>
<WYNIK><NAZWISKO>Delecki</NAZWISKO><PLACA>5820</PLACA></WYNIK>
...
<WYNIK><NAZWISKO>Bulski</NAZWISKO><PLACA>2040</PLACA></WYNIK>
```

Podzapytania

- w zapytaniach Xquery można zagnieżdżać wyrażenia FLWOR
- najczęściej zagnieżdżenia występują
- w klauzuli RETURN
- jako argumenty funkcji
- w klauzuli WHERE pod postacią wyrażeń ilościowych
- zagnieżdżenia mogą wykorzystywać wcześniej przypisane zmienne

```
<WYNIK>
{ for $z in ora:view("ETATY")//NAZWA
  return $z }
</WYNIK>
```

Wykład XII

Odp:

```
<WYNIK>
<NAZWA>Referent</NAZWA>
<NAZWA>Stażysta</NAZWA>
<NAZWA>Sekretarka</NAZWA>
<NAZWA>Asystent</NAZWA>
...
</WYNIK>
```

Wyrażenia warunkowe

*Xquery wspiera wyrażenia warunkowe oparte na konstrukcji **IF THEN ELSE***

```
for $z in ora:view("ZESPOLY")//ID_ZESP/text()
return <DZIAL> {$z/../../NAZWA}
{ if ($z=10) then
for $p at $i in ora:view("PRACOWNICY")/ROW
where $p/ID_ZESP=$z
return $p/NAZWISKO
else () }
</DZIAL>
```

Odp:

```
<DZIAL><NAZWA>NAZWA</NAZWA></DZIAL>
<DZIAL><NAZWA>Administracja</NAZWA><NAZWISKO>Lech</NAZWISKO>
<NAZWISKO>Koliberek</NAZWISKO></DZIAL>
<DZIAL><NAZWA>Bazy danych</NAZWA></DZIAL>
...
<DZIAL><NAZWA>Multimedia</NAZWA></DZIAL>
```

```
<SZELOWIE>
{
for $szefu in ora:view("PRACOWNICY")
let $leszcze := ora:view("PRACOWNICY")/ROW[SZEF=$szefu/ROW/NUMER]/NAZWISKO
where count($leszcze) >= 1
return
    <SZEF NUMER="{ $szefu/ROW/NUMER}">
        { $szefu/ROW/NAZWISKO }
        <PODWLADNI>
        { $leszcze }
        </PODWLADNI>
    </SZEF>
}
</SZELOWIE>
```

Odp:

```
<SZELOWIE>
<SZEF NUMER="1000"><NAZWISKO>Lech</NAZWISKO>
<PODWLADNI><NAZWISKO>Koliberek</NAZWISKO><NAZWISKO>Podgajny</NAZWISKO>
<NAZWISKO>Delecki</NAZWISKO><NAZWISKO>Lubicz</NAZWISKO></PODWLADNI></SZEF>
...
</SZELOWIE>
```