

Lab-8


Xquery yapısı ve örnekleri.

Xquery Nedir?


- ❖ W3C (World Wide Web Consortium) standardı.
- ❖ Bir sorgulama dili (Q.L.)
- ❖ XML dosyaları içerisinde arama yapabilmek için geliştirilmiştir yapısal bir dil.
- ❖ **tag** (<book></book>, <fname></lname>) üzerinde arama gerçekleştiren bir yapı.
- ❖ Veritabanı için SQL ne ise, XML için de Xquery o.
- ❖ SQL ifadelerinin tamamını Xquery'de de gerçekleyebiliyoruz.
- ❖ Web servislerinden bilgi çıkarımında çok daha kullanışlı.
- ❖ Relational Algebra **yok** !
- ❖ Relation **yok** !


BaseX Kurulumu

← → ↻ ⓘ Güvenli değil basex.org/download/


 [DOWNLOAD](#) [BASEX ▾](#) [ABOUT ▾](#) [DOCUMENTATION](#)


Current Version: 9.2.4




BaseX924.jar
Core Package



XML DBMS
XQuery Processor
Client/Server
BaseX GUI




BaseX924.zip
ZIP Package



XML DBMS
XQuery Processor
Client/Server
BaseX GUI
RESTXQ Web Apps



BaseX924.exe
Windows Installer


XML DBMS
XQuery Processor
Client/Server
BaseX GUI
RESTXQ Web Apps



BaseX924.war
Web Archive


XML DBMS
XQuery Processor
Client/Server
RESTXQ Web Apps
Libraries

Root Element: bookstore

Element: book, title, author,
year, price.
book -> parent
title, author, year, price -> child

Attribute: category

```
▼ <bookstore>
  ▼ <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  ▼ <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  ▼ <book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  ▼ <book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Company.xml Tanıtımı

Root Element: company

Element: employee,
department, dname, dnum

Attribute: dnumber, ssn, dep_no

```
<company>
  <employee ssn="888665555">
    <fname>James</fname>
    <minit>E</minit>
    <lname>Borg</lname>
    <bdate>10-NOV-27</bdate>
    <address>450 Stone, Houston, TX</address>
    <sex>M</sex>
    <salary>55000</salary>
    <superssn>null</superssn>
    <dno>1</dno>
  </employee>
  <department dnumber="5">
    <dname>Research</dname>
    <mgrssn>333445555</mgrssn>
    <mgrstartdate>22-MAY-78</mgrstartdate>
  </department>
  <dept_locations dnumber="1" dlocation="Houston" />
  <project pnumber="1">
    <pname>ProductX</pname>
    <plocation>Bellaire</plocation>
    <dnum>5</dnum>
  </project>
  <dependent dep_no="1">
    <essn>333445555</essn>
    <dependent_name>Alice</dependent_name>
    <sex>F</sex>
    <bdate>05-APR-76</bdate>
    <relationship>Daughter</relationship>
  </dependent>
  <works_on essn="123456789" pno="1">
    <hours>32.5</hours>
  </works_on>
</company>
```

Table 4-1. Simple path expressions

Example	Return value
<code>doc("catalog.xml")/catalog</code>	The <code>catalog</code> element that is the outermost element of the document
<code>doc("catalog.xml")//product</code>	All <code>product</code> elements anywhere in the document
<code>doc("catalog.xml")//product/@dept</code>	All <code>dept</code> attributes of <code>product</code> elements in the document
<code>doc("catalog.xml")/catalog/*</code>	All child elements of <code>catalog</code>
<code>doc("catalog.xml")/catalog/*/number</code>	All <code>number</code> elements that are grandchildren of <code>catalog</code>

XQUERY **FLWOR** ifadeleri

- **For :** Bir düğüm dizisini seçer.
- **Let:** Bir değişkene diziyi atar.
- **Where:** Düğümleri filtreler.
- **Order by:** Düğümleri sıralar
- **Return:** Sonuçları döndürür.

Xquery Örnekleri

Örnek 1:

Tüm çalışanların bilgilerini Xpath ve Xquery ile bulunuz.

SELECT * FROM employee;

Xpath: **doc("company.xml")/company/employee**

Xquery: **for \$x in doc("company.xml")/company/employee
return \$x**

Satırları görünmez yapmak istediğimizde;

(: Comment :)

Örnek 2:

Şirkette olan departmanların isimlerini bulan Xquery sorgusunu yazınız.

```
SELECT dname FROM department;  
for $x in doc("company.xml")/company/department  
    return $x/dname
```

Örnek 3:

‘Seattle’ da oturan işçinin/işçilerin adını ve doğum tarihini listeleyen Xquery sorgusunu yazınız.

```
SELECT fname, bdate
```

```
FROM employee
```

```
WHERE address LIKE '%Seattle%';
```

```
for $x in doc("company.xml")/company/employee[matches(address,'Seattle')]
```

```
return $x/fname | $x/bdate
```

```
<fname>Jon</fname>  
<bdate>22-AUG-1964</bdate>  
<fname>Ray</fname>  
<bdate>16-AUG-1949</bdate>|
```

Tag'siz listelemek istersek : return data(\$x/fname | \$x/bdate)

```
Jon  
22-AUG-1964  
Ray  
16-AUG-1949|
```

Örnek 4:

İsmi 'Franklin Wong' olan çalışanın çalıştığı projelerin numaralarını listeleyen Xquery sorgusunu yazınız.

SELECT pno **FROM** employee, works_on

WHERE fname = 'Franklin' **AND** lname = 'Wong' **AND** essn = ssn;

let \$com := doc("company.xml")

for \$emp **in** \$com//employee

let \$ad := \$emp/fname

let \$soyad := \$emp/lname

let \$tc := \$emp/@ssn

for \$wo **in** \$com//works_on

let \$etc := \$wo/@essn

where

\$ad = 'Franklin' **and** \$soyad = 'Wong' **and** \$tc = \$etc

return \$wo/@pno

NOT: attribute'ler @ ile belirtilir.

Örnek 5:

Şirketin satış departmanının ("Sales") hangi şehir(ler)de ofisi olduğunu bulan Xquery sorguyu yazınız. Şehirleri alfabetik olarak sıralayın.

```
SELECT dlocation FROM department d, dept_locations dl  
WHERE dname = 'Sales' AND d.dnumber=dl.dnumber;
```

```
let $com := doc("company.xml")  
for $dep in $com//department  
    let $disim := $dep/dname  
    let $dn1 := $dep/@dnumber  
for $loc in $com//dept_locations  
    let $dn2 := $loc/@dnumber  
    let $dloc := $loc/@dlocation  
where  
    $disim = 'Sales' and $dn1 = $dn2  
order by $dloc  
return $dloc
```

Örnek 6:

'Elizabeth' isminde akrabası olan çalışanın yöneticisinin (supervisor) adını ve soyadını bulan Xquery sorgusunu yazınız.

SELECT e2.fname, e2.lname **FROM** employee e1, employee e2, dependent d

WHERE d.dependent_name = 'Elizabeth' **AND** d.essn = e1.ssn **AND** e1.superssn = e2.ssn;

let \$com := doc("company.xml")

for \$emp **in** \$com//employee

let \$etc := \$emp/@ssn

let \$e_stc := \$emp/superssn

for \$dep **in** \$com//dependent

let \$disim := \$dep/dependent_name

let \$d_tc := \$dep/essn

for \$sup **in** \$com//employee

let \$stc := \$sup/@ssn

let \$sad := \$sup/fname

let \$ssoyad := \$sup/lname

where

\$disim = 'Elizabeth' **and** \$d_tc = \$etc **and** \$e_stc = \$stc

return \$sad | \$ssoyad

Örnek 7:

«OperatingSystems» isimli projede çalışanların ve «Software» departmanında çalışanların ad, soyad bilgilerini bulan Xquery sorgusunu yazınız.

```
SELECT fname, lname  
FROM employee e, project p, works_on wo  
WHERE pname = 'OperatingSystems'  
AND p.pnumber = wo.pno AND wo.essn = e.ssn
```

INTERSECT

```
SELECT fname, lname  
FROM employee e, department d  
WHERE dname = 'Software' AND e.dno = d.dnumber;
```

```
let $com := doc("company.xml")  
for $proje in $com//project  
  let $pro_no := $proje/@pnumber  
  let $pro_isim := $proje/pname  
for $wo in $com//works_on  
  let $wo_pno := $wo/@pno  
  let $wo_tc := $wo/@essn  
for $semp in $com//employee  
  let $se_ssn := $semp/@ssn  
  let $se_fname := $semp/fname  
  let $se_lname := $semp/lname  
  let $se_dno := $semp/dno  
for $d in $com//department  
  let $d_no := $d/@dnumber  
  let $d_isim := $d/dname  
where  
  $pro_isim = 'OperatingSystems' and  
  $pro_no = $wo_pno and  
  $se_ssn = $wo_tc and  
  $d_isim = 'Software' and  
  $d_no = $se_dno  
return $se_fname | $se_lname
```

SON