

Jegyzőkönyv

Modern Adatbázisrendszerek MSc

Féléves feladat

2022. tavasz

XQuery

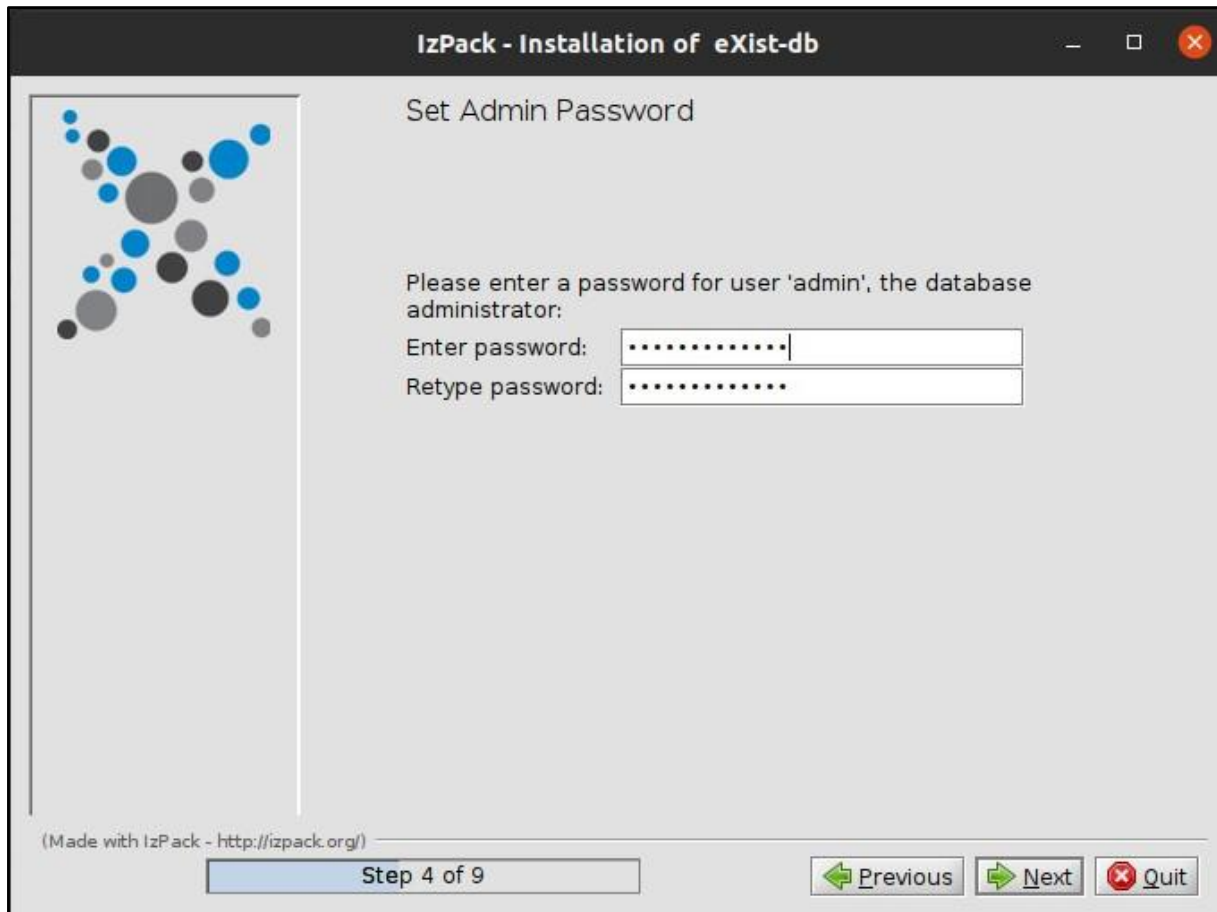
Készítette: Vadon Enikő

Neptunkód: DW9YG6

0. Feladat

Először egy környezet szükséges ahhoz, hogy XQuery-t tudjunk használni. Erre én két főbb opció lehetőség:

- A) Töltsük le az ExistDB egy futtatható .jar telepítőfájlját az alábbi helyről (<https://sourceforge.net/projects/exist/>), majd indítsuk el. Ez telepíti az ExistDB-t. A telepítés során meg kell adnunk az adminisztrátor felhasználóhoz tartozó jelszót is.



- B) Java környezetben nincs szükség az ExistDB-re, hanem helyette a Buildpath-nek tartalmaznia kell a: saxon9he.jar, saxon9-xqj.jar fájlokat.

Ekkor a java fájl mellett jelen kell lennie egy .xqy kiterjesztésű XQuery fájlnek (vagy egy Stringként definiált XQuery parancsnak), valamint a feldolgozandó XML fájlnek.

```
private static void execute() throws Exception {
    InputStream inputStream = new FileInputStream(new File("etterem.xqy"));
    XQDataSource ds = new SaxonXQDataSource();
    XQConnection conn = ds.getConnection();
    XQPreparedExpression exp = conn.prepareExpression(inputStream);
    XQResultSequence result = exp.executeQuery();
    while (result.next()) {
        System.out.println(result.getItemAsString(null));
    }
}
```

1. feladat

Készítsünk egy XQuery kódot az alábbi XML adatok feldolgozására:

Cars: <https://raw.githubusercontent.com/altova/XPath-XQuery-SQL-Tutorial/master/data/Cars.xml>

Manufacturers: https://raw.githubusercontent.com/altova/XPath-XQuery-SQL-Tutorial/master/data/Cars_Manufacturer_Countries.xml

A)

Feladat:

A Cars adatok alapján állapítsuk meg, átlagosan hány évig van egy autómодell piacon!

Megoldás:

```
1 xquery version "3.1";
2
3 let $cars := doc('https://raw.githubusercontent.com/altova/XPath-XQuery-SQL-Tutorial/master/data/Cars
4 .xml')/CarsDB/Car
5
6 let $val:=
7   for $c in $cars
8   let $a:=xs:integer($c/YearTill)-xs:integer($c/YearFrom)
9   return
10    $a
11 return <Atlagos-piacon-toltott-evek-szama> {avg($val)} </Atlagos-piacon-toltott-evek-szama>
```

Eredmény:

```
1 <Atlagos-piacon-toltott-evek-szama>1.484712704269899842</Atlagos-piacon-toltott-evek-szama>
```

B)

Feladat:

Adott a fenti 2 XML internetes forrásfájl. Listázzuk ki azt az 5, legtöbb olyan autómoddelt gyártó országot, ami dízelt használ üzemanyagként, valamint hozzájuk tartozó darabszámokat, XML-ként formázva.

Megoldás:

```
1 xquery version "3.1";
2
3 let $cars := doc('https://raw.githubusercontent.com/altova/XPath-XQuery-SQL-Tutorial/master/data/Cars.xml')/CarsDB/Car
4 let $man := doc('https://raw.githubusercontent.com/altova/XPath-XQuery-SQL-Tutorial/master/data/Cars_Manufacturer_Countries.xml')/CCDB/CarCountry
5
6 for $res in
7   (for $c in $cars[Fuel eq 'Diesel'], $m in $man
8    where distinct-values($c/Manufacturer) eq $m/Manufacturer
9    group by $country := $m/Country
10   order by count($c/Model) descending
11   return
12     <country name="{ $country }">
13       {count($c/Model)}
14     </country> [position() = 1 to 5]
15 return $res
```

Eredmény:

```
1 <country name="Germany">398</country>
2 <country name="France">218</country>
3 <country name="Japan">108</country>
4 <country name="Sweden">78</country>
5 <country name="Italy">72</country>
```

C)

Feladat:

Adott a Cars.xml adathalmaz. Készítsünk egy függvényt, ami kiszámolja, hogy mekkora kedvezményt kapnánk az autómodellekre, ha minden év, amit a modell forgalomban töltött 2% kedvezményt ad. Listázzuk ki a modelleket és a hozzájuk tartozó kedvezményt.

Megoldás:

```
1 xquery version "3.1";
2
3 declare function local:calcDiscount($year)
4 as xs:decimal? {
5   let $currentYear:= fn:year-from-date(fn:current-date())
6   let $discount := ( (xs:integer($currentYear)-xs:integer($year)) * 0.02)
7   return $discount
8 };
9 let $cars := doc('https://raw.githubusercontent.com/altova/XPath-XQuery-SQL-Tutorial/master/data/Cars.xml')/CarsDB/Car
10
11 for $c in $cars
12 return (
13   <Car>
14     <Model> {$c/Model/text()} </Model>
15     <Discount> {local:calcDiscount($c/YearFrom)*100}% </Discount>
16   </Car>)
```

Eredmény:

```
1 <Car>
   <Model>550i xDrive Gran Turismo</Model>
   <Discount>18% </Discount>
</Car>
2 <Car>
   <Model>550i xDrive GranTurismo</Model>
   <Discount>24% </Discount>
</Car>
3 <Car>
   <Model>640i Convertible</Model>
   <Discount>20% </Discount>
</Car>
4 <Car>
```

2. feladat

Vegyük a korábban elkészített saját Étterem nyilvántartáshoz tartozó XML fájlt! Az adott workspacebe importáljuk be a fájlt (ha módosítani kell a fájlt, lekérdezések során a webes forrás is ugyanúgy viselkedik), és ezzel dolgozzunk a továbbiakban.

Ha még nincs egy vendéghez több rendelés rendelve, akkor vegyünk fel még plusz rendelés(ek)e)t.

A)

Feladat:

Készítsünk az étterem nyilvántartás alapján egy tartozás nyilvántartást a vendégekhez, ahol feltüntetjük a vendégek nevét és az összeget, amivel az éteknek összesen tartoznak.

Megoldás:

```
1 xquery version "3.1";
2
3 let $sett := doc("https://raw.githubusercontent.com/Sevilda/DW9YG6MDbGyak/main/03.02/VE_XML.xml"
4 )/DW9YG6
5 for $r in $sett/rendeles, $v in $sett/vendeg
6 where $r/@fk_vkod eq $v/@vkod and $r/@fk_ekod
7 group by $vendeg := $r/@fk_vkod
8 return
9   <Tartozas>
10     <Vendeg>    {$v/nev/text()}
11   </Vendeg>
12   <Fizetendo-osszeg> {sum($r/osszeg)} </Fizetendo-osszeg>
13 </Tartozas>
```

Eredmény:

```
1 <Tartozas>
   <Vendeg>Toth Lenke</Vendeg>
   <Fizetendo-osszeg>64000</Fizetendo-osszeg>
</Tartozas>
2 <Tartozas>
   <Vendeg>Vadon E</Vendeg>
   <Fizetendo-osszeg>2250</Fizetendo-osszeg>
</Tartozas>
3 <Tartozas>
   <Vendeg>Kiss Mari</Vendeg>
   <Fizetendo-osszeg>2500</Fizetendo-osszeg>
</Tartozas>
4 <Tartozas>
   <Vendeg>Nagy Medve</Vendeg>
   <Fizetendo-osszeg>2500</Fizetendo-osszeg>
</Tartozas>
```

B)

Feladat:

Készítsünk olyan számla elemeket, amit az eredeti XML fájlhoz hozzá tudunk adni és az egy valid XML marad.

Szerepeljen benne egy saját kulcs a számla azonosításához, valamint a vendégre és az étteremre mutató idegen kulcsok. A számla tartalmazza elemként a fizetendő összeget, valamint az aktuális időt!

Megoldás:

```
1 xquery version "3.1";
2
3 let $ett := doc('etterem.xml')/DW9YG6
4 let $result :=
5 for $r in $ett/rendeles, $v in $ett/vendeg, $e in $ett/etterem
6 where $r/@fk_vkod eq $v/@vkod and $r/@fk_ekod eq $e/@ekod
7 return
8 <res>
9   <ett>{$e} </ett>
10  <vendeg>{$v} </vendeg>
11  <rendeles> {$r} </rendeles>
12 </res>
13
14 let $szamla:=
15 for $n in 1 to count($result)
16 let $date:= current-dateTime()
17 return
18 <Szamla szkod="{ $n}" fk_ekod="{ $result[position()=$n]/ett/etterem/@ekod}" fk_vkod
  ="{ $result[position()=$n]/vendeg/vendeg/@vkod}">
19 <Fizetendo-osszeg> { $result[position()=$n]/rendeles/rendeles/osszeg/text() } </Fizetendo
   -osszeg>
20 <Dátum> { $date} </Dátum>
21 </Szamla>
22
23 return $szamla
```

Eredmény:

```
1 <Szamla szkod="1" fk_ekod="e1" fk_vkod="v1">
   <Fizetendo-osszeg>64000</Fizetendo-osszeg>
   <Dátum>2022-05-03T21:10:09.307+02:00</Dátum>
</Szamla>
2 <Szamla szkod="2" fk_ekod="e1" fk_vkod="v2">
   <Fizetendo-osszeg>500</Fizetendo-osszeg>
   <Dátum>2022-05-03T21:10:09.307+02:00</Dátum>
</Szamla>
3 <Szamla szkod="3" fk_ekod="e1" fk_vkod="v2">
   <Fizetendo-osszeg>750</Fizetendo-osszeg>
   <Dátum>2022-05-03T21:10:09.307+02:00</Dátum>
</Szamla>
4 <Szamla szkod="4" fk_ekod="e2" fk_vkod="v2">
   <Fizetendo-osszeg>1000</Fizetendo-osszeg>
   <Dátum>2022-05-03T21:10:09.307+02:00</Dátum>
</Szamla>
5 <Szamla szkod="5" fk_ekod="e1" fk_vkod="v4">
   <Fizetendo-osszeg>1000</Fizetendo-osszeg>
   <Dátum>2022-05-03T21:10:09.307+02:00</Dátum>
</Szamla>
6 <Szamla szkod="6" fk_ekod="e2" fk_vkod="v4">
```

C)

Feladat:

Az ilyen módon elkészített számlákat adjuk hozzá az XML fájlunkhoz!

Megoldás:

Előzőhöz képest annyi a változás, hogy *return* helyett egy *return update insert* parancsot adunk ki.

```
1 xquery version "3.1";
2
3 let $ett := doc('etterem.xml')/DW9YG6
4 let $result :=
5 for $r in $ett/rendeles, $v in $ett/vendeg, $e in $ett/etterem
6 where $r/@fk_vkod eq $v/@vkod and $r/@fk_ekod eq $e/@ekod
7 return
8 <res>
9   <ett>{$e} </ett>
10   <vendeg>{$v} </vendeg>
11   <rendeles> {$r} </rendeles>
12 </res>
13
14 let $szamla:=
15 for $n in 1 to count($result)
16 let $date:= current-dateTime()
17 return
18 <Szamla szkod="{ $n}" fk_ekod="{ $result[position()=$n]/ett/etterem/@ekod}" fk_vkod
19  ="{ $result[position()=$n]/vendeg/vendeg/@vkod}">
20   <Fizetendo-osszeg> { $result[position()=$n]/rendeles/rendeles/osszeg/text() } </Fizetendo-
21   -osszeg>
22   <Dátum> { $date } </Dátum>
23   </Szamla>
24
25 return update insert $szamla into $ett
```

Eredmény:

Az XML fájl vége eddig a rendelések voltak, látható hogy a számlák most már megjelentek a fájl végén:

```
</rendeles>
  <rendeles fk_vkod="v3" fk_ekod="e2">
    <osszeg>2500</osszeg>
    <etel>Lángos</etel>
  </rendeles>

<Szamla szkod="1" fk_ekod="e1" fk_vkod="v1">
  <Fizetendo-osszeg>64000</Fizetendo-osszeg>
  <Dátum>2022-05-03T21:13:44.382+02:00</Dátum>
</Szamla>
<Szamla szkod="2" fk_ekod="e1" fk_vkod="v2">
  <Fizetendo-osszeg>500</Fizetendo-osszeg>
  <Dátum>2022-05-03T21:13:44.382+02:00</Dátum>
</Szamla>
<Szamla szkod="3" fk_ekod="e1" fk_vkod="v2">
```


D)

Feladat:

Töröljünk minden olyan számlát, ami a 2. étteremhez tartozik!

Megoldás:

```
1 xquery version "3.1";
2
3 let $ett := doc('etterem.xml')/DW9YG6
4
5 for $sz in $ett/Szamla[@fk_ekod eq "e2"]
6
7 return update delete $sz
8
```

Eredmény:

Parancs lefutása után látható, a számlák között már nincsen olyan, ami a 2. Étteremhez tartozna.

```
140 <rendeles fk_vkod="v3" fk_ekod="e2">
141   <osszeg>2500</osszeg>
142   <etel>Lángos</etel>
143 </rendeles>
144
145
146 <Szamla szkod="1" fk_ekod="e1" fk_vkod="v1">
147   <Fizetendo-osszeg>64000</Fizetendo-osszeg>
148   <Dátum>2022-05-03T21:13:44.382+02:00</Dátum>
149 </Szamla>
150 <Szamla szkod="2" fk_ekod="e1" fk_vkod="v2">
151   <Fizetendo-osszeg>500</Fizetendo-osszeg>
152   <Dátum>2022-05-03T21:13:44.382+02:00</Dátum>
153 </Szamla>
154 <Szamla szkod="3" fk_ekod="e1" fk_vkod="v2">
155   <Fizetendo-osszeg>750</Fizetendo-osszeg>
156   <Dátum>2022-05-03T21:13:44.382+02:00</Dátum>
157 </Szamla>
158 <Szamla szkod="5" fk_ekod="e1" fk_vkod="v4">
159   <Fizetendo-osszeg>1000</Fizetendo-osszeg>
160   <Dátum>2022-05-03T21:13:44.382+02:00</Dátum>
161 </Szamla>
162 </DW9YG6>
```