

Modern Adatbázis Rendszerek MSc

ExistDB és XQuery

Készítette:

Vadon Enikő

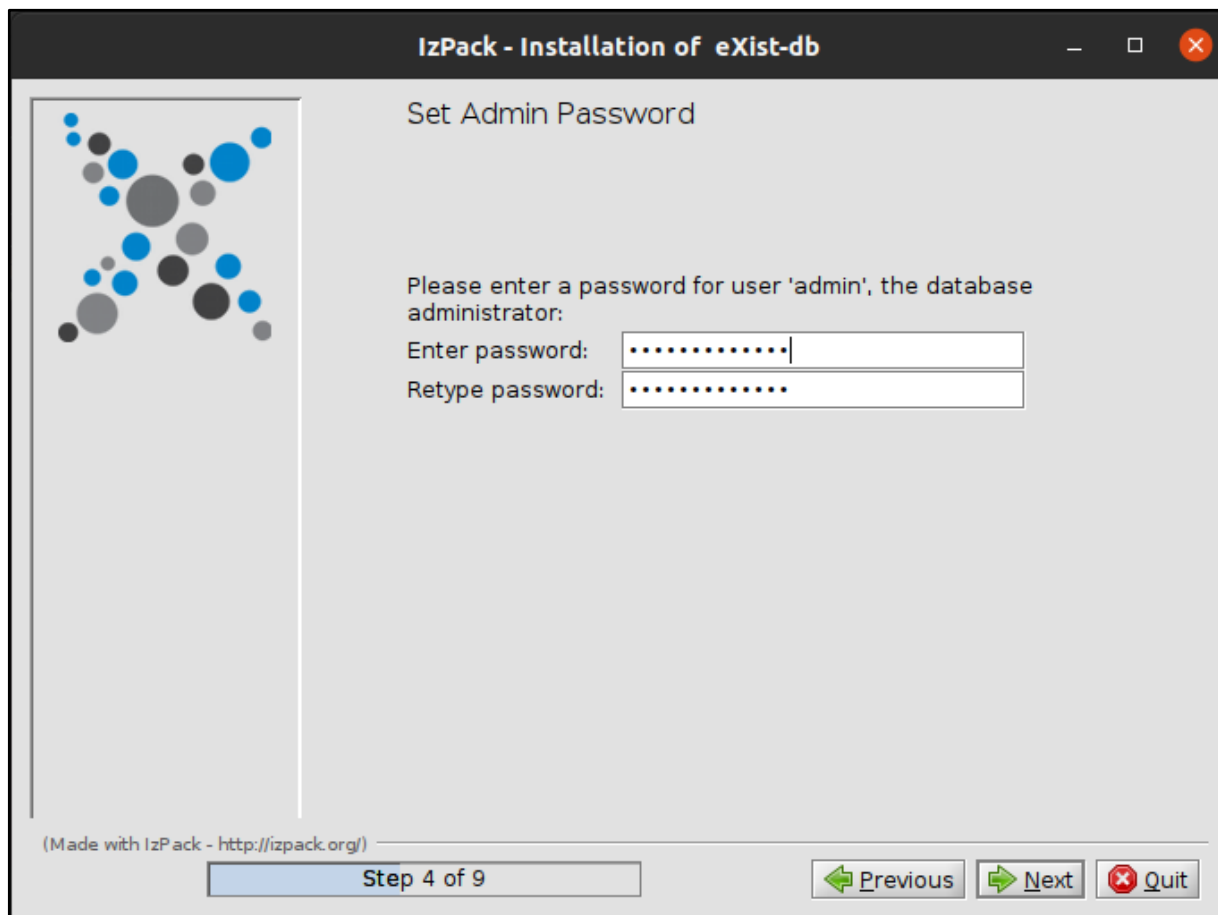
DW9YG6

2022.03.16

1, Feladat

Először egy környezet szükséges ahhoz, hogy XQuery-t tudjunk használni. Erre én két opciót ajánlok:

- 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. Ezen kívül jelen kell lennie egy .xqy kiterjesztésű XQuery fájlnak (vagy egy Stringként definiált XQuery parancsnak), valamint a feldolgozandó XML fájlnak.

```
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));
    }
}
```

2, 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) A Cars adatok alapján állapítsuk meg, átlagosan hány évig van egy autómодell piacon?

```
<Atlagos-piacon-toltott-evok-szama>1.484712704269899842</Atlagos-piacon-toltott-evok-szama>
```

- B) Adott a fenti 2 XML internetes forrásfájl. Feladat az, hogy listázzuk ki azt az 5, legtöbb olyan autómодellt gyártó országot, ami dízelt használ üzemanyagként, valamint hozzájuk tartozó darabszámokat, XML-ként formázva.

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

```
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>
```

3, Feladat

Vegyük a korábban elkészített saját Étterem nyilvántartáshoz tartozó XML fájlt! Az adott workspace-be importáljuk be a fájlt, és ezzel dolgozzunk a továbbiakban.

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

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

```
1 <Tartozas>
    <Vendeg>Toth Lenke</Vendeg>
    <Fizetendo-osszeg>64000</Fizetendo-osszeg>
</Tartozas>
2 <Tartozas>
    <Vendeg>Vadon E</Vendeg>
    <Fizetendo-osszeg>2250</Fizetendo-osszeg>
</Tartozas>
```

- B) 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!

- C) Ezek után adjuk hozzá a számla elemeket az XML fájlhoz!

```
111 </rendeles>
112 <rendeles fk_vkod="v2" fk_ekod="e2">
113 <osszeg>1000</osszeg>
114 <etel>Kókuszkecska</etel>
115 </rendeles>
116
117
118 <Szamla szkod="1" fk_ekod="e1" fk_vkod="v1">
119 <Fizetendo-osszeg>64000</Fizetendo-osszeg>
120 <Datum>2022-03-15T14:03:00.631+01:00</Datum>
121 </Szamla>
122 <Szamla szkod="2" fk_ekod="e1" fk_vkod="v2">
123 <Fizetendo-osszeg>500</Fizetendo-osszeg>
124 <Datum>2022-03-15T14:03:00.631+01:00</Datum>
125 </Szamla>
126 <Szamla szkod="3" fk_ekod="e1" fk_vkod="v2">
127 <Fizetendo-osszeg>750</Fizetendo-osszeg>
128 <Datum>2022-03-15T14:03:00.631+01:00</Datum>
129 </Szamla>
130 <Szamla szkod="4" fk_ekod="e2" fk_vkod="v2">
131 <Fizetendo-osszeg>1000</Fizetendo-osszeg>
132 <Datum>2022-03-15T14:03:00.631+01:00</Datum>
133 </Szamla>
134
135 </DW9YG6>
```

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

Példa megoldások:

2, A)

```
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
5
6 let $val:=
7 for $c in $cars
8 let $a:=xs:integer($c/YearTill)-xs:integer($c/YearFrom)
9 return
10     $a
11
12 return <Atlagos-piacon-toltott-evek-szama> {avg($val)} </Atlagos-piacon-toltott-evek-szama
  >
```

2, B)

```
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
```

2, C)

```
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
12 for $c in $cars
13 return (
14     <Car>
15     <Model> { $c/Model/text() } </Model>
16     <Discount> {local:calcDiscount($c/YearFrom)*100}% </Discount>
17     </Car>)
```

3, A)

```
1 xquery version "3.1";
2
3
4 let $ett := doc("https://raw.githubusercontent.com/Sevilda/DW9YG6MdbGyak/main/03.02/VE_XML.xml")/DW9YG6
5
6 for $r in $ett/rendeles, $v in $ett/vendeg
7 where $r/@fk_vkod eq $v/@vkod and $r/@fk_ekod
8 group by $vendeg := $r/@fk_vkod
9 return
10     <Tartozas>
11     <Vendeg>      {$v/nev/text()}
12     </Vendeg>
13     <Fizetendo-osszeg> {sum($r/osszeg)} </Fizetendo-osszeg>
14     </Tartozas>
```

3, B) és 3, C)

```
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
```

(1 sor különbség, return \$szamla helyett update insert-et adunk rá.)

3, D)

```
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
```