

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: Lehoczky
András
Neptunkód: B0WCKK

A feladat leírása:

A feladatban filmek adatbázisát készítettem el. Gyökérelemnek a filmeket adtam meg, ennek vannak a gyerekei amelyek maga a film, és az ahhoz kapcsolódó adatai. A filmek két attribute-ot adtam, a címét és a genre-t. A címe unique, ami a valóságban nem teljesen helyt álló, mivel előfordul átfedés, de jelen adatbázis szempontjából szerintem megfelelő, mivel a filmeket a címük alapján lehet legjobban elkülöníteni, és az átfedés nem okoz gondot.

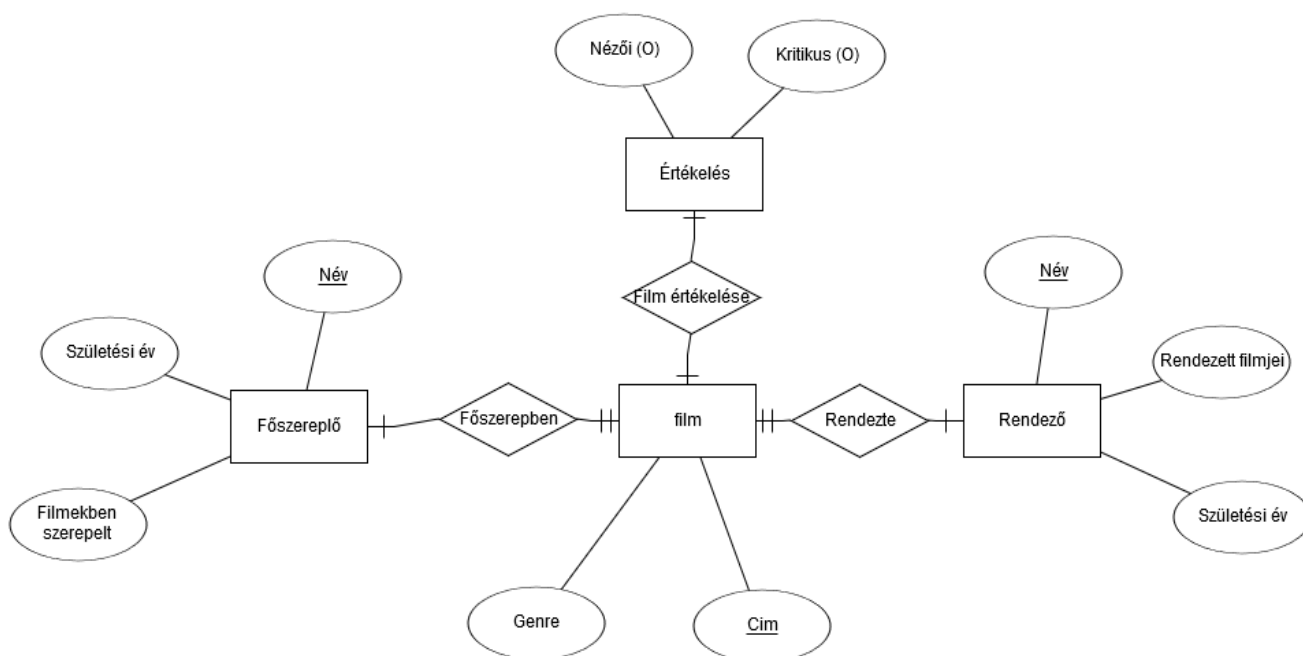
Három másik elementem van, a rendező, a főszereplő és az értékelés. Egy filmhez csak egy rendező rendelhető, és a rendezőnek vannak további attribútumai. Neve, ami megint csak egyedi, születési éve, és hogy hány filmet rendezett életében. Ezek forrása a wikipédia oldaluk. A főszereplőnek hasonló adatai vannak.

Az értékeléseket az alapján különböztetem meg, hogy kritikusoktól milyen értékelést kapott, illetve a nézőktől milyen. Ezeknek a forrása rottentomatoes.com.

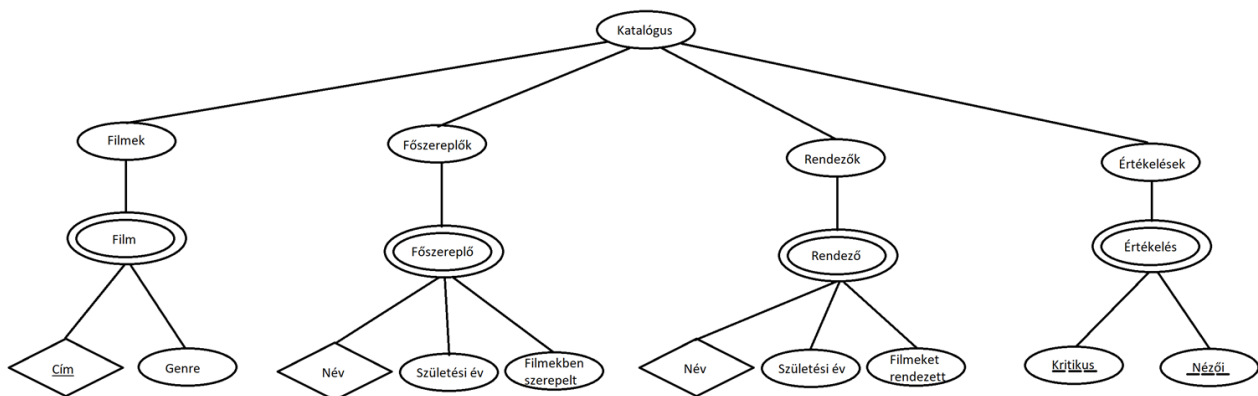
Az ER model egyszerűen tükrözi mindezt, a kapcsolatok számosságát is tekintve, ahol szükséges.

Magában az XML adatrendszerben négy példát adtam meg.

1.) ER model



2.) XDM model



3.) XML kód

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="autokB0WCKK4.xsl" ?>
<filmek>
  <film>
    <cím>Egy új remény</cím>
    <genre>sci-fi</genre>
    <rendező név="George Lucas">
      <rendezett_filmjei>28</rendezett_filmjei> <!--Darabszám-->
      <születési_év>1944</születési_év>
    </rendező>
    <főszereplő név="Mark Hamill">
      <filmekben_szerpelt>22</filmekben_szerpelt>
      <születési_év>1951</születési_év>
    </főszereplő>
    <értékelés>
      <kritikus>8.6</kritikus>
      <nézői>9.0</nézői>
    </értékelés>
  </film>

  <film>
    <cím>Bosszúállók</cím>
    <genre>akciófilm</genre>
    <rendező név="Joss Whedon">
      <rendezett_filmjei>9</rendezett_filmjei> <!--Darabszám-->
      <születési_év>1964</születési_év>
    </rendező>
    <főszereplő név="Robert Downey Jr.">
      <filmekben_szerpelt>41</filmekben_szerpelt>
      <születési_év>1965</születési_év>
    </főszereplő>
    <értékelés>
      <kritikus>92</kritikus>
      <nézői>91</nézői>
    </értékelés>
  </film>
</filmek>
```

```
</értékelés>
</film>
```

```
<film>
  <cím>Titanic</cím>
  <genre>katasztrófafilm</genre>
  <rendező név="James Cameron">
    <rendezett_filmjei>30</rendezett_filmjei> <!--Darabszám-->
    <születési_év>1954</születési_év>
  </rendező>
  <főszereplő név="Leonardo DiCaprio">
    <filmekben_szerelt>20</filmekben_szerelt>
    <születési_év>1974</születési_év>
  </főszereplő>
  <értékelés>
    <kritikus>89</kritikus>
    <nézői>69</nézői>
  </értékelés>
</film>
```

```
<film>
  <cím>Csupasz pisztoly</cím>
  <genre>vígjáték</genre>
  <rendező név="David Zucker">
    <rendezett_filmjei>25</rendezett_filmjei> <!--Darabszám-->
    <születési_év>1947</születési_év>
  </rendező>
  <főszereplő név="Leslie Nielsen">
    <filmekben_szerelt>26</filmekben_szerelt>
    <születési_év>1926</születési_év>
  </főszereplő>
  <értékelés>
    <kritikus>88</kritikus>
    <nézői>84</nézői>
  </értékelés>
</film>
```

```
</filmek>
```

4.) XDM kód

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="filmek">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="film">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="cím" type="xs:string" />
              <xs:element name="genre" type="xs:string" />
              <xs:element name="rendező">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="rendezett_filmjei" type="xs:integer" />
                    <xs:element name="születési_év" type="xs:integer" />
                  </xs:sequence>
                  <xs:attribute name="név" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
              <xs:element name="főszereplő">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="filmekben_szerelt" type="xs:integer" />
                    <xs:element name="születési_év" type="xs:integer" />
                  </xs:sequence>
                  <xs:attribute name="név" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
              <xs:element name="értékelés">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="kritikus" type="xs:decimal" />
                    <xs:element name="nézői" type="xs:decimal" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

5.) DOM kód

```
1 package parseB0WCKK;
2
3 import java.nio.file.Path;
4 import java.nio.file.Paths;
5
6 import java.io.File;
7 import java.io.IOException;
8
9 import org.w3c.dom.Document;
10 import org.w3c.dom.Element;
11 import org.w3c.dom.Node;
12 import org.w3c.dom.NodeList;
13 import org.xml.sax.SAXException;
14
15 import javax.xml.parsers.DocumentBuilder;
16 import javax.xml.parsers.DocumentBuilderFactory;
17 import javax.xml.parsers.ParserConfigurationException;
18
19 public class DOMreadB0WCKK {
20
21     public static void main(String[] args) {
22
23         //java file jelenlegi hely keresése, troubleshootinghoz
24         Path currentRelativePath = Paths.get("");
25         String s = currentRelativePath.toAbsolutePath().toString();
26         System.out.println("Current relative path is: " + s);
27
28         try {
29
30             //xml file elérése. Nem találja a file-t sehogy sem.
31             String filepath = "C:/Users/Intel/eclipse-workspace/parseB0WCKK/XMLB0WCKK.xml";
32             File fXmlFile = new File(filepath);
33             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
34             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
35             Document doc = dBuilder.parse(fXmlFile);
36
37             //struktúra normalizálás
38             doc.getDocumentElement().normalize();
39
40             System.out.println("Root element : " + doc.getDocumentElement().getNodeName());
41
42             NodeList nList = doc.getElementsByTagName("filmek");
43
44             System.out.println("-----");
45
46             for (int temp = 0; temp < nList.getLength(); temp++) {
47
48                 Node nNode = nList.item(temp);
49
50                 System.out.println("\nCurrent Element : " + nNode.getNodeName());
51
52                 if (nNode.getNodeType() == Node.ELEMENT_NODE) {
53
54                     Element eElement = (Element) nNode;
55
56
57                     System.out.println("Film címe : " + eElement.getElementsByTagName("cím").item(0).getTextContent());
58
59                     System.out.println("rendező neve : " + eElement.getAttribute("név"));
60                     System.out.println("Genre : " + eElement.getElementsByTagName("genre").item(0).getTextContent());
61
62                 }
63             }
64         } catch (Exception e) {
65             e.printStackTrace();
66         }
67     }
68
69 }
70
71
72 }
```