

# **JEGYZŐKÖNYV**

Adatkezelés XML környezetben

Féléves feladat

Készítette: Radócz Nikolett

NeptunKód:WSI08I

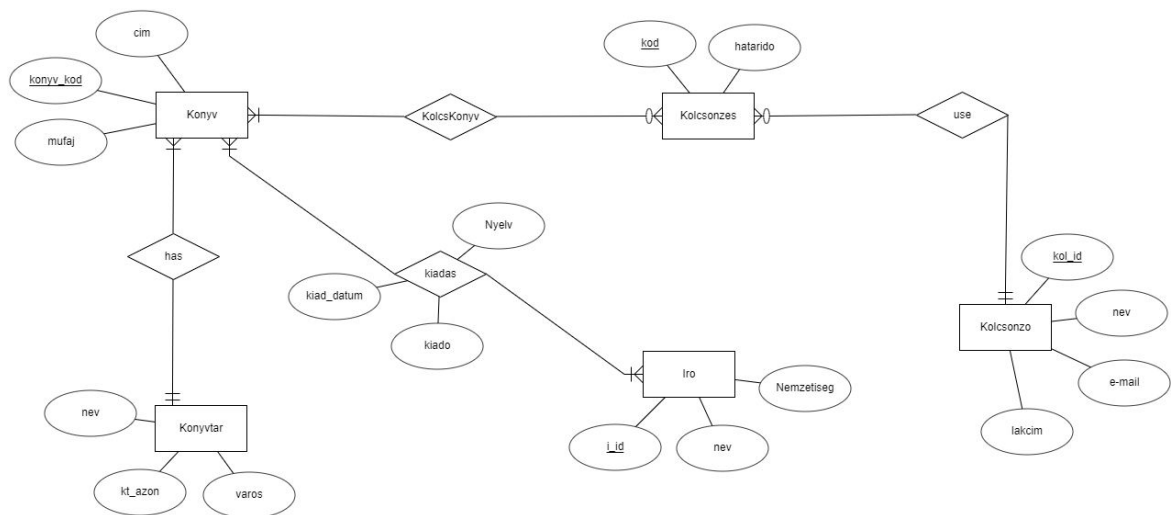
# A feladat leírása:

A feladatom az Adatbázis rendszerek I. tárgy beadandó témája alapján készült el. A Könyvtárak, Könyv, Író, Kölcsönzés és Kölcsönző közötti kapcsolatokat mutatja be.

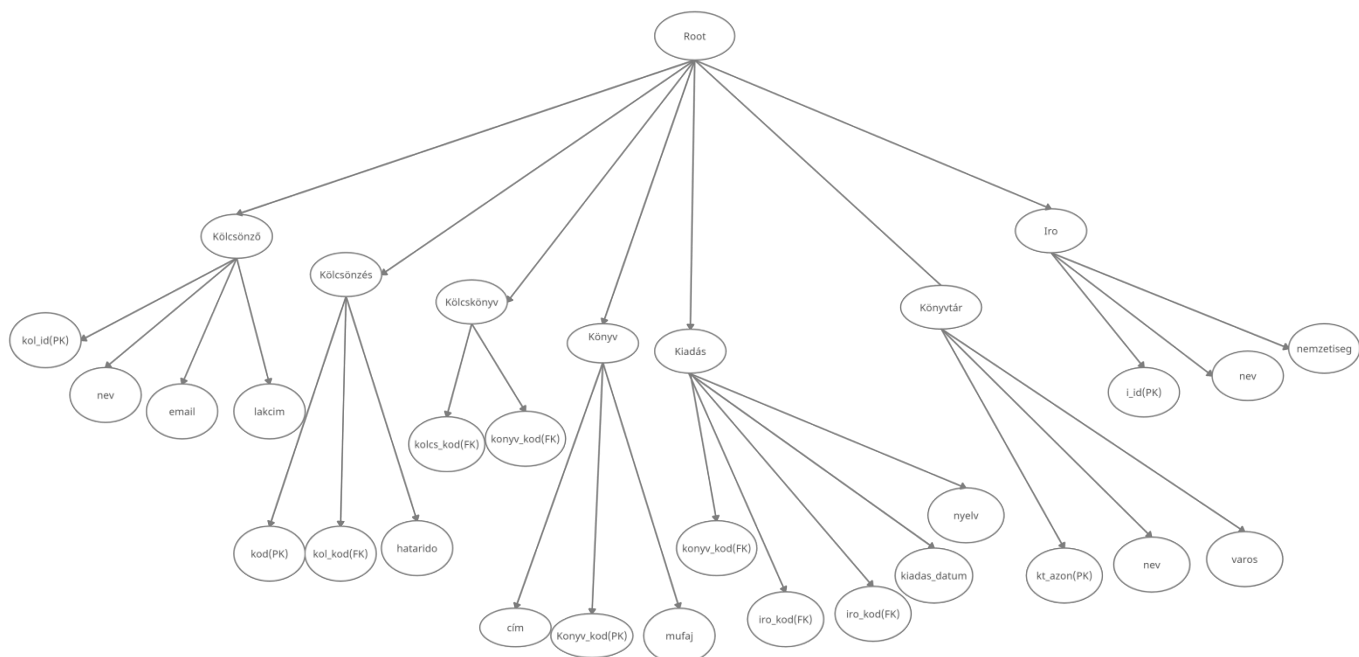
## 1. feladat

A feladat első részében az ER modellt, az XDM modellt az XML és az XMLSchema dokumentumot szemléltetem.

### 1.1 ER modell



## 1.2 XDM modell



## 1.2 XML kód

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="XMLSchemaSI08I.xsd">
3
4 <kolcsonzo kol_id="01">
5   <nev>Bene Anna</nev>
6   <email>banna@gmail.com</email>
7   <lakcim>Miskolc</lakcim>
8 </kolcsonzo>
9 <kolcsonzo kol_id="01">
10   <nev>Kiss Jozsef</nev>
11   <email>jozsi@gmail.com</email>
12   <lakcim>Budapest</lakcim>
13 </kolcsonzo>
14
15
16 <kolcsonzes kod="01" kol_id="01">
17   <hatarido>2020.12.20</hatarido>
18 </kolcsonzes>
19 <kolcsonzes kod="02" kol_id="02">
20   <hatarido>2021.01.10</hatarido>
21 </kolcsonzes>
22
23
24 <iro i_id="01">
25   <nev>Vavyan Fable</nev>
26   <nemzetiseg>Magyar</nemzetiseg>
27 </iro>
28 <iro i_id="02">
29   <nev>Jennifer A. Nelsen</nev>
30   <nemzetiseg>Angol</nemzetiseg>
31 </iro>
32 <iro i_id="03">
33   <nev>Tasmina Perry</nev>
34   <nemzetiseg>Angol</nemzetiseg>
35 </iro>
36
```

```

37
38=<konyvtar kt_azon="01">
39  <nev>II. Rákóczi Ferenc Megyei és Városi Könyvtár</nev>
40  <varos>Miskolc</varos>
41</konyvtar>
42
43
44=<konyv konyv_kod="01" kt_azon="01">
45  <cim>Vészbejárt</cim>
46  <műfaj>krimi</műfaj>
47</konyv>
48=<konyv konyv_kod="02" kt_azon="01">
49  <cim>Sztárok ügyvédje</cim>
50  <műfaj>romantikus</műfaj>
51</konyv>
52=<konyv konyv_kod="03" kt_azon="01">
53  <cim>The Shadow Throne</cim>
54  <műfaj>misztikus</műfaj>
55</konyv>
56
57
58=<kiadas i_id="01" konyv_kod="01">
59  <kiadas_datum>2000</kiadas_datum>
60  <nyelv>magyar</nyelv>
61</kiadas>
62=<kiadas i_id="02" konyv_kod="3">
63  <kiadas_datum>2015</kiadas_datum>
64  <nyelv>angol</nyelv>
65</kiadas>
66=<kiadas i_id="02" konyv_kod="02">
67  <kiadas_datum>2012</kiadas_datum>
68  <nyelv>német</nyelv>
69</kiadas>
70
71
72 <kolcskonyv kod="01" konyv_kod="03"/>
73 <kolcskonyv kod="01" konyv_kod="02"/>
74 <kolcskonyv kod="02" konyv_kod="01"/>
75</root>

```

## 1.3 XML Schema

```

1  <?xml version="1.0" encoding="UTF-8"?>
2=<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3=<xs:element name="root">
4  <xs:complexType>
5    <xs:sequence>
6=<xs:element name="kolcsonzo" maxOccurs="unbounded" >
7  <xs:complexType>
8    <xs:sequence>
9      <xs:element name="nev" type="xs:string"/>
10     <xs:element name="email" type="xs:string"/>
11     <xs:element name="lakcim" type="xs:string"/>
12   </xs:sequence>
13   <xs:attribute name="kol_id" type="xs:int"/>
14 </xs:complexType>
15</xs:element>
16
17=<xs:element name="kolcsonzes" maxOccurs="unbounded">
18  <xs:complexType>
19    <xs:sequence>
20      <xs:element name="hatarido" type="xs:date"/>
21    </xs:sequence>
22    <xs:attribute name="kod" type="xs:int"/>
23    <xs:attribute name="kol_id" type="xs:int"/>
24  </xs:complexType>
25</xs:element>
26
27=<xs:element name="iro" maxOccurs="unbounded">
28  <xs:complexType>
29    <xs:sequence>
30      <xs:element name="nev" type="xs:string"/>
31      <xs:element name="nemzetiseg" type="xs:string"/>
32    </xs:sequence>
33    <xs:attribute name="i_id" type="xs:int"/>
34  </xs:complexType>
35</xs:element>
36

```

```

37<xs:element name="konyvtar" maxOccurs="unbounded">
38  <xs:complexType>
39    <xs:sequence>
40      <xs:element name="nev" type="xs:string"/>
41      <xs:element name="varos" type="xs:string"/>
42    </xs:sequence>
43    <xs:attribute name="kt_azon" type="xs:int"/>
44  </xs:complexType>
45</xs:element>
46
47<xs:element name="konyv" maxOccurs="unbounded">
48  <xs:complexType>
49    <xs:sequence>
50      <xs:element name="cim" type="xs:string"/&;
51      <xs:element name="mufaj" type="xs:string"/>
52    </xs:sequence>
53    <xs:attribute name="konyv_kod" type="xs:int"/>
54  </xs:complexType>
55</xs:element>
56
57<xs:element name="kiadas" maxOccurs="unbounded">
58  <xs:complexType>
59    <xs:sequence>
60      <xs:element name="kiadas_datum" type="xs:int"/>
61      <xs:element name="nyelv" type="xs:string"/>
62    </xs:sequence>
63    <xs:attribute name="i_id" type="xs:int"/>
64    <xs:attribute name="konyv_kod" type="xs:int"/>
65  </xs:complexType>
66</xs:element>
67
68
69
70<xs:element name="kolcskonyv">
71  <xs:complexType>
72    <xs:attribute name="kol_id" type="xs:int"/>
73    <xs:attribute name="konyv_kod" type="xs:int"/>
74  </xs:complexType>
75</xs:element>
76
77  </xs:sequence>
78</xs:complexType>
79
80<xs:key name="key1">
81  <xs:selector xpath="kolcsonzo"></xs:selector>
82  <xs:field xpath="@kol_id"></xs:field>
83</xs:key>
84
85<xs:key name="key2">
86  <xs:selector xpath="iro"></xs:selector>
87  <xs:field xpath="@i_id"></xs:field>
88</xs:key>
89
90<xs:key name="key3">
91  <xs:selector xpath="kolcsonzes"></xs:selector>
92  <xs:field xpath="@kod"></xs:field>
93</xs:key>
94
95<xs:key name="key4">
96  <xs:selector xpath="konyv"></xs:selector>
97  <xs:field xpath="@konyv_kod"></xs:field>
98</xs:key>
99
100<xs:key name="key5">
101  <xs:selector xpath="konyvvtar"></xs:selector>
102  <xs:field xpath="@kt_azon"></xs:field>
103</xs:key>
104
105<xs:keyref refer="key1" name="ref1">
106  <xs:selector xpath="kolcsonzes" />
107  <xs:field xpath="@kol_id" />
108</xs:keyref>
109
110<xs:keyref refer="key2" name="ref2">
111  <xs:selector xpath="kiadas" />
112  <xs:field xpath="@i_id" />
113</xs:keyref>
114
115<xs:keyref refer="key4" name="ref3">
116  <xs:selector xpath="kiadas" />
117  <xs:field xpath="@konyv_kod" />
118</xs:keyref>
119
120<xs:keyref refer="key3" name="ref4">
121  <xs:selector xpath="kolcskonyv" />
122  <xs:field xpath="@kod" />
123</xs:keyref>
124
125<xs:keyref refer="key4" name="ref5">
126  <xs:selector xpath="kolcskonyv" />
127  <xs:field xpath="@konyv_kod" />
128</xs:keyref>
129<xs:keyref refer="key5" name="ref6">
130  <xs:selector xpath="konyv" />
131  <xs:field xpath="@kt_azon" />
132</xs:keyref>
133
134</xs:element>
135
136
137</xs:schema>

```

## 2. feladat

## 2.1 DOM adatolvasás

```
1 package hu.domparsa.ws108i;
2
3 import java.io.File;
4
14 public class DOMReads108I {
15
17     public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException {
18         //XML file objektum létrehozása
19         File xmlFile = new File("xmlWS108I.xml");
20
21         // Document Builder létrehozása
22         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
23         DocumentBuilder dBuilder = factory.newDocumentBuilder();
24
25         Document doc = dBuilder.parse(xmlFile);
26
27         //normalizálás
28         doc.getDocumentElement().normalize();
29
30         //Gyoker elem kiiratas
31         Element root = doc.getDocumentElement();
32         System.out.println("Gyoker elem: " + root.getNodeName());
33
34
35         NodeList NL1 = doc.getElementsByTagName("kolcsonzo");
36
37         for (int i = 0; i < NL1.getLength(); i++) {
38             Node nNode = NL1.item(i);
39
40             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
41                 System.out.println();
42
43                 Element elem = (Element) nNode;
44
45                 System.out.println("Elem: " + nNode.getNodeName());
46                 String kol_id = elem.getAttribute("kol_id");
47
48                 Node n1 = elem.getElementsByTagName("nev").item(0);
49                 String nev = n1.getTextContent();
50
51                 Node n2 = elem.getElementsByTagName("email").item(0);
52                 String email = n2.getTextContent();
53
54                 Node n3 = elem.getElementsByTagName("lakcim").item(0);
55                 String lakcim = n3.getTextContent();
56
57                 System.out.println("id: " + kol_id);
58                 System.out.println("Neve: " + nev);
59                 System.out.println("Email: " + email);
60                 System.out.println("lakcim: " + lakcim);
61
62             }
63         }
64
65         NodeList NL2 = doc.getElementsByTagName("kolcsonzes");
66
67         for (int i = 0; i < NL2.getLength(); i++) {
68             Node nNode = NL2.item(i);
69
70             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
71                 System.out.println();
72
73                 Element elem = (Element) nNode;
74
75                 System.out.println("Elem: " + nNode.getNodeName());
76                 String kod = elem.getAttribute("kod");
77                 String kol_id = elem.getAttribute("kol_id");
78
79                 Node n1 = elem.getElementsByTagName("hatarido").item(0);
80                 String hat = n1.getTextContent();
81
82                 System.out.print("kod: " + kod + ", ");
83                 System.out.println("kolcsonzes id: " + kol_id);
84                 System.out.println("hatarido: " + hat);
85
86             }
87         }
88     }
89 }
90
91
92 }
```

```

93     }
94
95     NodeList NL3 = doc.getElementsByTagName("iro");
96
97     for (int i = 0; i < NL3.getLength(); i++) {
98         Node nNode = NL3.item(i);
99
100         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
101             Element elem = (Element) nNode;
102             System.out.println();
103
104             System.out.println("Elem: " + nNode.getNodeName());
105             String iid = elem.getAttribute("i_id");
106
107             Node n1 = elem.getElementsByTagName("nev").item(0);
108             String nev = n1.getTextContent();
109
110             Node n2 = elem.getElementsByTagName("nemzetiseg").item(0);
111             String nemz = n1.getTextContent();
112
113             System.out.println("Iro id: " + iid);
114             System.out.println("nev: " + nev);
115             System.out.println("nemzetiseg: " + nemz);
116         }
117     }
118 }
119
120 NodeList NL4 = doc.getElementsByTagName("konyvtar");
121
122 for (int i = 0; i < NL4.getLength(); i++) {
123     Node nNode = NL4.item(i);
124
125     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
126         Element elem = (Element) nNode;
127         System.out.println();
128
129         System.out.println("Elem: " + nNode.getNodeName());
130         String ktazon = elem.getAttribute("kt_azon");
131
132         Node n1 = elem.getElementsByTagName("nev").item(0);
133         String nev = n1.getTextContent();
134
135         Node n2 = elem.getElementsByTagName("varos").item(0);
136         String varos = n1.getTextContent();
137
138         System.out.println("Konyvtar azonosito: " + ktazon);
139         System.out.println("nev: " + nev);
140         System.out.println("varos: " + varos);
141     }
142 }
143
144 NodeList NL5 = doc.getElementsByTagName("konyv");
145
146 for (int i = 0; i < NL5.getLength(); i++) {
147     Node nNode = NL5.item(i);
148
149     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
150         Element elem = (Element) nNode;
151         System.out.println();
152
153         System.out.println("Elem: " + nNode.getNodeName());
154
155         String kkod = elem.getAttribute("konyv_kod");
156         String ktazon = elem.getAttribute("kt_azon");
157
158         Node n1 = elem.getElementsByTagName("cim").item(0);
159         String cim = n1.getTextContent();
160
161         Node n2 = elem.getElementsByTagName("mufaj").item(0);
162         String mufaj = n1.getTextContent();
163
164         System.out.println("konyv kod: " + kkod);
165         System.out.println("Konyvtar azonosito: " + ktazon);
166         System.out.println("cim: " + cim);
167         System.out.println("mufaj: " + mufaj);
168     }
169 }
170
171 NodeList NL6 = doc.getElementsByTagName("kiadas");
172
173 for (int i = 0; i < NL6.getLength(); i++) {
174     Node nNode = NL6.item(i);
175
176     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
177         Element elem = (Element) nNode;
178         System.out.println();
179
180         System.out.println("Elem: " + nNode.getNodeName());
181         String iid = elem.getAttribute("i_id");
182         String kkod = elem.getAttribute("konyv_kod");
183
184         Node n1 = elem.getElementsByTagName("kiadas_datum").item(0);
185         String kdatum = n1.getTextContent();
186
187         Node n2 = elem.getElementsByTagName("nyelv").item(0);
188         String nyelv = n1.getTextContent();
189
190         System.out.println("Iro id: " + iid);
191         System.out.println("Konyv kod: " + kkod);
192         System.out.println("kiadas datum: " + kdatum);
193         System.out.println("nyelv: " + nyelv);
194     }
195 }

```

```

195     }
196   }
197 }
198
199 NodeList NL7 = doc.getElementsByTagName("kolcskonyv");
200
201 for (int i = 0; i < NL7.getLength(); i++) {
202     Node nNode = NL7.item(i);
203
204     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
205         Element elem = (Element) nNode;
206         System.out.println();
207
208         System.out.println("Elem: " + nNode.getNodeName());
209         String kod = elem.getAttribute("kod");
210         String kkod = elem.getAttribute("konyv_kod");
211
212
213         System.out.println("Kolcsonzes kodja: " + kod);
214         System.out.println("Konyv kod: " + kkod);
215     }
216 }
217 }
218 }
219 }
220 }
221 }

```

## 2.2 DOM adatmódosítás

```

1 package hu.domparse.wsi081;
2
3 import java.io.File;
4 import java.io.IOException;
5 import javax.xml.parsers.DocumentBuilder;
6 import javax.xml.parsers.DocumentBuilderFactory;
7 import javax.xml.parsers.ParserConfigurationException;
8 import javax.xml.xpath.XPath;
9 import javax.xml.xpath.XPathConstants;
10 import javax.xml.xpath.XPathExpressionException;
11 import javax.xml.xpath.XPathFactory;
12 import org.w3c.dom.Document;
13 import org.w3c.dom.Element;
14 import org.w3c.dom.Node;
15 import org.w3c.dom.NodeList;
16 import org.xml.sax.SAXException;
17 public class DOMModifyWSI081 {
18
19     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException, XPathExpressionException {
20         // meghatározza az XML file helyét
21         File xmlFile = new File("xmlWSI081.xml");
22
23         // létrehozza a Document Buildert
24         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
25         DocumentBuilder builder = factory.newDocumentBuilder();
26
27         // DOM objektumra konvertál
28         Document doc = builder.parse(xmlFile);
29
30         // normalizál
31         doc.getDocumentElement().normalize();
32
33         // XPath segítségével módosítandó elem meghatározása
34         XPath xPath = XPathFactory.newInstance().newXPath();
35
36         // elem amit módosítunk
37         String expr1 = "/root/kolcsonzo[@kol_id = '01']/nev/text()";
38         NodeList nodeList1 = (NodeList) xPath.compile(expr1).evaluate(doc, XPathConstants.NODESET);
39
40         // Mire szeretnénk átírni
41         nodeList1.item(0).setNodeValue("Radocz Nikolett");
42
43         // Elérés megadása
44         String expr2 = "/root/kolcsonzo[@kol_id = '01']";
45         nodeList1 = (NodeList) xPath.compile(expr2).evaluate(doc, XPathConstants.NODESET);
46
47         // másik elemet i módosítunk
48         String expr3 = "/root/iro[@i_id = '02']/nev/text()";
49         nodeList2 = (NodeList) xPath.compile(expr3).evaluate(doc, XPathConstants.NODESET);
50
51         nodeList2.item(0).setNodeValue("temp");
52
53         String expr4 = "/root/iro[@i_id = '02']";
54         nodeList2 = (NodeList) xPath.compile(expr4).evaluate(doc, XPathConstants.NODESET);
55
56         for (int i = 0; i < nodeList1.getLength(); i++) {
57             Node nNode = nodeList1.item(i);
58
59             System.out.println("\nCurrent Element : " + nNode.getNodeName());
60
61             // Ciklussal kiíratjuk az értékeket
62             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
63                 Element eElement = (Element) nNode;
64
65                 System.out.println("Kolcsonzo id: " + eElement.getAttribute("kol_id"));
66
67                 System.out.println("Nev: " + eElement.getElementsByTagName("nev").item(0).getTextContent());
68
69                 System.out.println("E-mail: " + eElement.getElementsByTagName("email").item(0).getTextContent());
70
71                 System.out.println("Lakcim: " + eElement.getElementsByTagName("lakcim").item(0).getTextContent());
72             }
73         }
74
75         for (int i = 0; i < nodeList2.getLength(); i++) {
76             Node nNode = nodeList2.item(i);
77
78             System.out.println("\nCurrent Element : " + nNode.getNodeName());
79
80             if (nNode.getNodeType() == Node.ELEMENT_NODE) {
81                 Element eElement = (Element) nNode;

```



```
82         System.out.println("Iro id: " + eElement.getAttribute("i_id"));
83
84         System.out.println("Iro neve: " + eElement.getElementsByTagName("nev").item(0).getTextContent());
85
86         System.out.println("Iro nemzetiseg: " + eElement.getElementsByTagName("nemzetiseg").item(0).getTextContent());
87     }
88 }
89
90
91 }
92
93 }
94
```