

JEGYZŐKÖNYV

Adatkezelés XML környezetben
Féléves feladat

Készítette: **Bajusz István**

Neptunkód: **OBYBZK**

A feladat leírása:

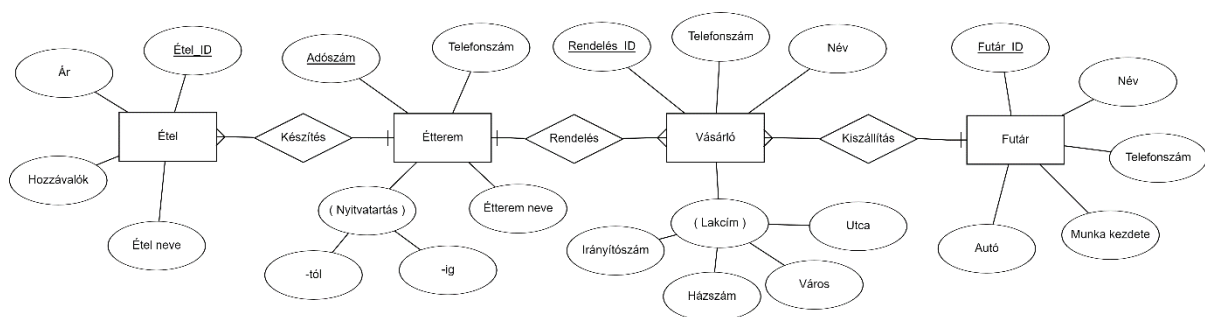
A feladatom bemutatja az éttermek fontosabb adatainak tárolását.

Az ER modell mutatja az étteremben lévő adatok egymáshoz való kapcsolódását. Az egyedeket egy-egy egyedi tulajdonságokkal láttam el (ID). Az első kapcsolat az étterem és az étel között van. Ez egy N:1 kapcsolat, mivel egy étteremhez tartozhat többféle étel is, viszont egy ételt egyszerre csak egy étteremhez köthetünk. Az ételek az „Étel_ID” nevű egyedi tulajdonsággal, illetve az „Ár”, „Hozzávalók” és az „Étel neve” tulajdonságokkal rendelkeznek. Az ételeket az étterem készíti, amelynek van neve, telefonszáma, nyitvatartási ideje (-tól, -ig), illetve adószáma (ID). Az étteremből lehetőség van ételrendelésre, így a következő kapcsolat a „Vásárló” és az „Éterem” egyed között jött létre, ami a „Rendelés”. A megrendelőnek van lakcíme (irányítószám, város, utca, házszám), telefonszáma, neve és rendelés ID-je. Az utolsó egyed a „Futár”. A futár és a vásárló közötti kapcsolat a kiszállítás. Egy futárnak van Futár ID-je, neve, telefonszáma, autója, illetve egy adat arról, hogy mikor kezdett dolgozni (saját típus).

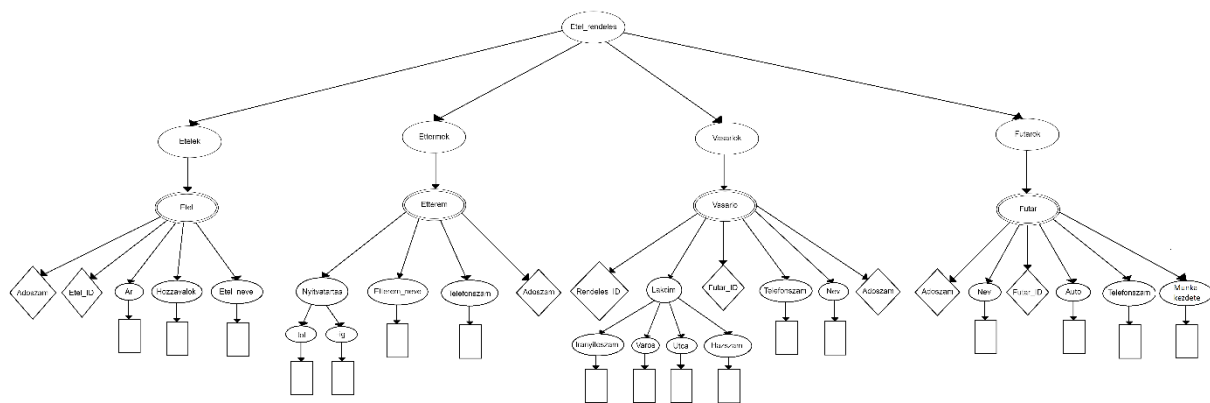
A rendeléseket a Vásárló tábla fogja rögzíteni, az elérhető összes ételt az Étel tábla, az elérhető éttermeket az Étterem tábla, illetve a dolgozó futárokat a Futár tábla. A vásárló idegen kulcsa fogja tárolni, hogy melyik étteremből rendelt. A futár szintén idegen kulccsal köthető egy étteremhez.

1. feladat

1a) Az adatbázis ER modell:



1b) Az adatbázis konvertálása XDM modellre:



1c) Az XDM modell alapján XML dokumentum készítése:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?><etel_rendeles
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:XMLSchemaObybzk.xsd">
  <etelek>
    <etel Adoszam="123456789101" Etel_ID="01">
      <etel_neve>hamburger</etel_neve>
      <ar>1590</ar>
      <hozzavalok>buci, huspogacs, paradicsom, szoszok, salata</hozzavalok>
    </etel>
    <etel Adoszam="123456789102" Etel_ID="02">
      <etel_neve>pizza</etel_neve>
      <ar>2390</ar>
      <hozzavalok>paradicsomos alap, sonka, gomba, kukorica, sajt,
csirkemell</hozzavalok>
    </etel>
    <etel Adoszam="123456789103" Etel_ID="03">
      <etel_neve>gyros</etel_neve>
      <ar>2190</ar>
      <hozzavalok>gyros hus, zoldsegek, hasaburgonya, szosz</hozzavalok>
    </etel>
  </etelek>
  <ettermek>
    <etterem Adoszam="123456789101">
      <etterem_neve>Elso etterem</etterem_neve>
      <telefonszam>06301234123</telefonszam>
    </etterem>
  </ettermek>
</etel_rendeles>
  
```

```
<nyitvatartas>
  <tol>8</tol>
  <ig>18</ig>
</nyitvatartas>
</etterem>
<etterem Adoszam="123456789102">
  <etterem_neve>Masodik etterem</etterem_neve>
  <telefonszam>06201234123</telefonszam>
  <nyitvatartas>
    <tol>8</tol>
    <ig>22</ig>
  </nyitvatartas>
</etterem>
<etterem Adoszam="123456789103">
  <etterem_neve>Harmadik etterem</etterem_neve>
  <telefonszam>06701234123</telefonszam>
  <nyitvatartas>
    <tol>10</tol>
    <ig>18</ig>
  </nyitvatartas>
</etterem>
</ettermek>
<vasarlok>
  <vasarlo Adoszam="123456789101" Futar_ID="11" Rendeles_ID="1">
    <nev>Nagy Bela</nev>
    <telefonszam>06301234567</telefonszam>
    <lakcim>
      <iranyitoszam>3600</iranyitoszam>
      <varos>Ozd</varos>
      <utca>Kossuth</utca>
      <hazszam>12</hazszam>
    </lakcim>
  </vasarlo>
  <vasarlo Adoszam="123456789102" Futar_ID="12" Rendeles_ID="2">
    <nev>Kovacs Andrea</nev>
    <telefonszam>06701234567</telefonszam>
    <lakcim>
      <iranyitoszam>2220</iranyitoszam>
      <varos>Vecses</varos>
      <utca>Fo</utca>
      <hazszam>3</hazszam>
    </lakcim>
  </vasarlo>
  <vasarlo Adoszam="123456789103" Futar_ID="13" Rendeles_ID="3">
    <nev>Kiss Sandor</nev>
    <telefonszam>06201234567</telefonszam>
    <lakcim>
```

```

        <iranyitoszam>1098</iranyitoszam>
        <varos>Budapest</varos>
        <utca>Petofi</utca>
        <hazszam>42</hazszam>
    </lakcim>
</vasarlo>
</vasarlok>
<futarok>
    <futar Adoszam="123456789101" Futar_ID="11">
        <nev>Kiss Bela</nev>
        <auto>suzuki</auto>
        <telefonszam>06309876543</telefonszam>
        <munkaKezdete>2005</munkaKezdete>
    </futar>
    <futar Adoszam="123456789102" Futar_ID="12">
        <nev>Kovacs Ferenc</nev>
        <auto>toyota</auto>
        <telefonszam>06309876543</telefonszam>
        <munkaKezdete>2019</munkaKezdete>
    </futar>
    <futar Adoszam="123456789103" Futar_ID="13">
        <nev>Nemeth Sandor</nev>
        <auto>opel</auto>
        <telefonszam>06309876543</telefonszam>
        <munkaKezdete>2021</munkaKezdete>
    </futar>
</futarok>
</etel_rendeles>

```

1d) Az XML dokumentum alapján XML Schema készítése (saját típus):

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="etel_rendeles">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="etelek">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="etel" maxOccurs="unbounded" minOccurs="0">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element type="xs:string" name="etel_neve"/>

```

```

        <xs:element type="xs:short" name="ar"/>
        <xs:element type="xs:string" name="hozzavalok"/>
    </xs:sequence>
    <xs:attribute type="xs:integer" name="Etel_ID" use="required"/>
    <xs:attribute type="xs:integer" name="Adoszam" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ettermek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="etterem" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:string" name="etterem_neve"/>
                        <xs:element type="xs:string" name="telefonszam"/>
                        <xs:element name="nyitvatartas">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element type="xs:short" name="tol"/>
                                    <xs:element type="xs:short" name="ig"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:attribute type="xs:integer" name="Adoszam" use="required"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="vasarlok">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="vasarlo" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:string" name="nev"/>
                        <xs:element type="xs:string" name="telefonszam"/>
                        <xs:element name="lakcim">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element type="xs:string" name="iranyitoszam"/>
                                    <xs:element type="xs:string" name="varos"/>
                                    <xs:element type="xs:string" name="utca"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```
<xs:element type="xs:string" name="hazszam"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>  
</xs:sequence>  
<xs:attribute type="xs:boolean" name="Rendeles_ID" use="required"/>  
<xs:attribute type="xs:integer" name="Adoszam" use="required"/>  
<xs:attribute type="xs:boolean" name="Futar_ID" use="required"/>  
</xs:complexType>  
</xs:element>  
</xs:sequence>  
</xs:complexType>  
</xs:element>  
<xs:element name="futarok">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="futar" maxOccurs="unbounded" minOccurs="0">  
        <xs:complexType>  
          <xs:sequence>  
            <xs:element type="xs:string" name="nev"/>  
            <xs:element type="xs:string" name="auto"/>  
            <xs:element type="xs:string" name="telefonszam"/>  
            <xs:element name="munkaKezdete">  
              <xs:simpleType>  
                <xs:restriction base="xs:gYear">  
                  <xs:minInclusive value="2000"/>  
                  <xs:maxInclusive value="2021"/>  
                </xs:restriction>  
              </xs:simpleType>  
            </xs:element>  
          </xs:sequence>  
          <xs:attribute type="xs:boolean" name="Futar_ID" use="required"/>  
          <xs:attribute type="xs:integer" name="Adoszam" use="required"/>  
        </xs:complexType>  
      </xs:element>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>  
</xs:sequence>  
</xs:complexType>  
</xs:element>  
</xs:sequence>  
</xs:complexType>
```

2. feladat

2a) Adatolvasás - DOMReadObybzk.java

```
1 package hu.domparse.obybkz;
2
3 import java.io.File;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21 public class DomReadObybzk {
22
23     private static void printNode(Node r) {
24
25         // Node nevének kiírása
26         if (r.getNodeName() != "#text") {
27             System.out.println(r.getNodeName());
28         }
29
30         // Gyerekek node listába helyezése
31         NodeList children = r.getChildNodes();
32         for (int i = 0; i < children.getLength(); i++) {
33             Node child = children.item(i);
34             boolean isComplex = child.getTextContent().contains("\n");
35
36             // Attribute kiírása
37             if (child.hasAttributes()) {
38                 NamedNodeMap attributes = child.getAttributes();
39                 int numAttrs = attributes.getLength();
40                 for (int j = 0; j < numAttrs; j++) {
41                     Attr attr = (Attr) attributes.item(j);
42                     String attrName = attr.getNodeName();
43                     String attrValue = attr.getNodeValue();
44                     System.out.println(" " + attrName + " : " + attrValue);
45                 }
46             }
47
48             // New es tartalom kiírása
49             if (isComplex) {
50                 printNode(child);
51             } else {
52                 System.out.print(" " + child.getNodeName());
53                 System.out.println(": " + child.getTextContent());
54             }
55         }
56     }
57
58     public static void main(String[] args) {
59         try {
60
61             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
62             DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
```



```

63     Document doc = dbBuilder.parse("src/hu/domparse/obybzk/XMLObybzk.xml");
64     String filepath = "src/hu/domparse/obybzk/XMLObybzk.xml";
65     doc.getDocumentElement().normalize();
66
67     // XPath
68     XPath xPath = XPathFactory.newInstance().newXPath();
69
70     Element root = doc.getDocumentElement();
71
72     System.out.println("\n ----- \n Modositas elott: \n ----- \n");
73
74     printNode(root);
75     DOMModifyObybzk.modifyPrices(root);
76
77     System.out.println("\n ----- \n Modositas utan: \n ----- \n");
78
79     printNode(root);
80
81     // Modositott XML mentese
82     TransformerFactory transformerFactory = TransformerFactory.newInstance();
83     Transformer transformer = transformerFactory.newTransformer();
84     DOMSource source = new DOMSource(doc);
85     StreamResult result = new StreamResult(new File(filepath));
86     transformer.transform(source, result);
87
88     } catch (Exception e) {
89         e.printStackTrace();
90     }
91 }
92
93 }
94

```

2b) Adatmódosítás – DOMModifyObybzk.java

```

1  package hu.domparse.obybzk;
2
3  import org.w3c.dom.Node;
4
5
6  public class DOMModifyObybzk {
7
8      public static void modifyPrices(Node root) {
9          try {
10              System.out.println(root.getNodeType());
11              NodeList els = root.getChildNodes();
12              for (int j = 0; j < els.getLength(); j++) {
13
14                  // Vegyem meg a root gyerekait
15                  if (els.item(j).getNodeName() == "etelek") {
16                      for (int i = 0; i < els.item(j).getChildNodes().getLength(); i++) {
17
18                          // Nostalalja az etelek-et, most annak gyerekait megy megis
19                          if (els.item(j).getChildNodes().item(i).getNodeName() == "etel") {
20                              for (int k = 0; k < els.item(j).getChildNodes().item(i).getChildNodes().getLength(); k++) {
21
22                                  //Az etelek arat megkeresi, majd megnovelji 200-al
23                                  if (els.item(j).getChildNodes().item(i).getChildNodes().item(k).getNodeName() == "ar") {
24                                      int ar = Integer.parseInt(els.item(j).getChildNodes().item(i).getChildNodes().
25                                          .item(k).getTextContent());
26                                      els.item(j).getChildNodes().item(i).getChildNodes().item(k)
27                                          .setTextContent(String.valueOf(ar + 200));
28                                  }
29                              }
30                          }
31                      }
32                  }
33              }
34          } catch (Exception e) {
35              e.printStackTrace();
36          }
37      }
38
39  }
40
41

```

2c) Adatlekérdezés – DOMQueryObybzk.java

```
1 package hu.domparse.obybkz;
2
3 import java.io.File;
4
5
6
7
8
9
10
11
12
13
14
15
16 public class DOMQueryObybzk {
17
18     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
19
20         File xmlFile = new File("src/hu/domparse/obybkz/XMLObybzk.xml");
21
22         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
23         DocumentBuilder dBuilder = factory.newDocumentBuilder();
24
25         Document doc = dBuilder.parse(xmlFile);
26
27         doc.getDocumentElement().normalize();
28
29         System.out.println("Root: " + doc.getDocumentElement().getNodeName() + "\n");
30
31         // Kiinja azokat a futarokat, akiknek suzuki autoiuk van
32         System.out.println("A suzuki-t hasznalo futarok neve: \n");
33
34         NodeList nodeList = doc.getElementsByTagName("futar");
35
36         boolean foundAny = false;
37
38         for(int i = 0; i < nodeList.getLength(); i++) {
39
40             Node node = nodeList.item(i);
41
42             if(node.getNodeType() == Node.ELEMENT_NODE) {
43                 Element elem = (Element) node;
44
45                 Node node2;
46
47                 // Megvizsgalom az autok tipusat
48                 node2 = elem.getElementsByTagName("auto").item(0);
49                 String carType = node2.getTextContent();
50
51                 // Ha suzuki, akkor kiinom a futar nevét
52                 if("suzuki".equals(carType)) {
53                     Node nodeName = elem.getElementsByTagName("nev").item(0);
54                     String name = nodeName.getTextContent();
55
56                     System.out.println(name + "\n");
57
58                     foundAny = true;
59                 }
60             }
61         }
62
63         if(foundAny == false) {
64
65         }
```

```

65
66     System.out.println("\n Nem talalhato olyan futar, aki suzuki-t vezet. \n");
67 }
68 else {
69     foundAny = false;
70 }
71
72 // Kiinria azoknak az ettermek a nevét, ahol kaphato gyros
73 System.out.println("\n Gyrost arulo ettermek: \n");
74
75 nodeList = doc.getElementsByTagName("etel");
76
77 foundAny = false;
78
79 for(int i = 0; i < nodeList.getLength(); i++) {
80     Node node = nodeList.item(i);
81
82     if(node.getNodeType() == Node.ELEMENT_NODE) {
83         Element elem = (Element) node;
84
85         Node node2;
86
87         // Megvizsgálja az eteleket
88         node2 = elem.getElementsByTagName("etel_neve").item(0);
89         String foodName = node2.getTextContent();
90
91         // Ha gyros, akkor kiinria az etelt arulo ettermek adoszamát
92         if("gyros".equals(foodName)) {
93
94             String taxNumber = elem.getAttribute("Adoszam");
95
96             System.out.println(taxNumber + "\n");
97
98             foundAny = true;
99         }
100     }
101 }
102 }
103
104 if(foundAny == false) {
105     System.out.println("\n Nem talalhato olyan etterem az adatbazisban, amelyik arul gyrost.\n");
106 }
107
108 }
109
110 }
111

```