

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Baranyai Zoltán**

Neptunkód: **RV4TJX**

## A feladat leírása:

A feladatban egy pizzéria adatbázisát terveztem meg, az alábbi jellemzőkkel:

A rendelők tudnak rendelést leadni, ami ételeket tartalmaz, valamint van hozzá rendelve egy futár is aki azt a rendelést kiszállítja. A rendelés és étel kapcsolata között van egy tulajdonság, ami az étel elkészítőjét tartalmazza.

A rendelő tulajdonságai: név, telefonszám, lakcím(irányítószám, város, utca, házszám), rendelő az.

A rendelés tulajdonságai: ár, kiszállítási idő, rendelés ideje, rendelés azonosító.

A futár tulajdonságai: név, fizetés, telefonszám, futár azonosító.

Étel tulajdonságai: név, elkészítési idő, feltétek, sorszám.

Ezek közötti kapcsolatok a következők:

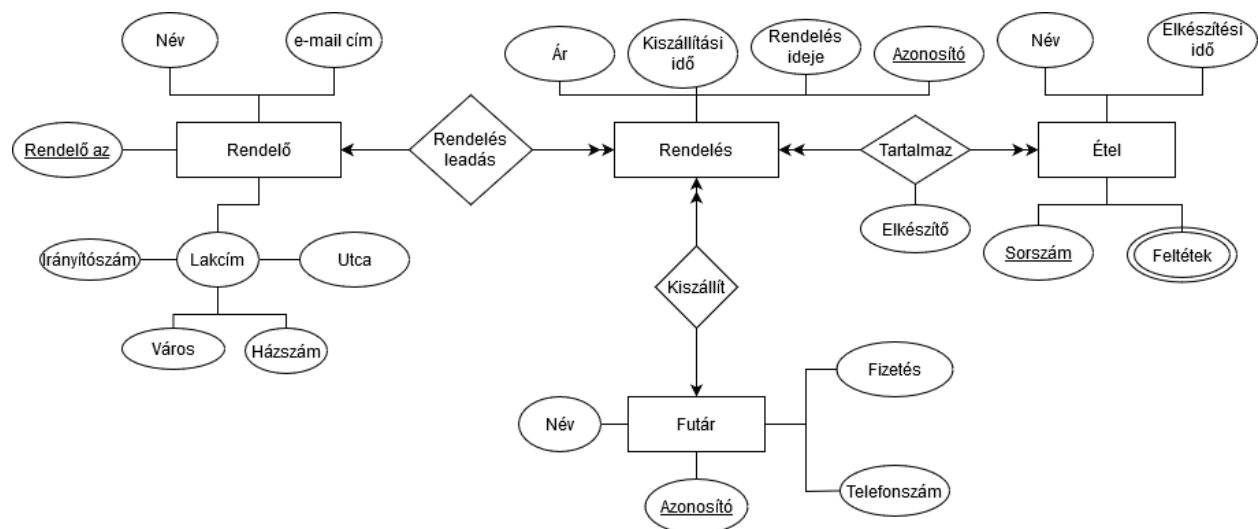
Rendelés leadás: Rendelő és Rendelés között 1:N kapcsolat.

Tartalmaz: Rendelés és Étél között N:M kapcsolat elkészítő tulajdonsággal.

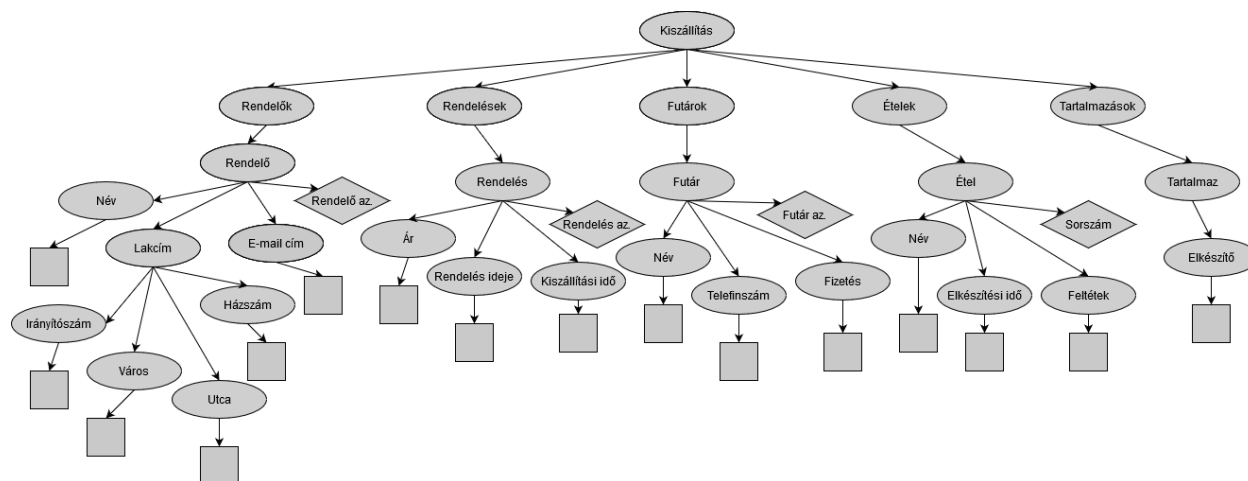
Kiszállít: Futár és a Rendelés között 1:N kapcsolat.

## 1. feladat

### 1a) Az adatbázis ER modellje:



## 1b) XDM modellre konvertálva:



## 1c) Az XDM alapján az elkészített XML dokumentum:

```
<?xml version="1.0" encoding="UTF-8"?>
<kiszallitas xmlns="beadando"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="beadando XMLSchemaRV4TJX.xsd">
  <rendelok>
    <rendelo rendeloid="0">
      <nev>Kiss Péter</nev>
      <email>kpeti@gmail.com</email>
      <lakcim>
        <iranyitoszam>1234</iranyitoszam>
        <varos>Kukutyin</varos>
        <utca>Kosztuh Lajos</utca>
        <hazzsam>123</hazzsam>
      </lakcim>
    </rendelo>
    <rendelo rendeloid="1">
      <nev>Nagy Sándor</nev>
      <email>sanyesz@gmail.com</email>
      <lakcim>
        <iranyitoszam>1234</iranyitoszam>
        <varos>Kukutyin</varos>
        <utca>Petőfi Sándor</utca>
        <hazzsam>321</hazzsam>
      </lakcim>
    </rendelo>
    <rendelo rendeloid="2">
      <nev>Példa Péter</nev>
      <email>petike@gmail.com</email>
      <lakcim>
        <iranyitoszam>1235</iranyitoszam>
```

```

        <varos>Alsókukutyin</varos>
        <utca>Főtér</utca>
        <hazszam>12</hazszam>
    </lakcim>
</rendelo>
<rendelo rendeloid="3">
    <nev>Jócska János</nev>
    <email>joja@gmail.com</email>
    <lakcim>
        <iranyitoszam>1234</iranyitoszam>
        <varos>Kukutyin</varos>
        <utca>Kossith Lajos</utca>
        <hazszam>12</hazszam>
    </lakcim>
</rendelo>
</rendelok>
<rendelesek>
    <rendeles rendeloid="2" futid="0">
        <ar>20000</ar>
        <kiszallido>30</kiszallido>
        <rendelesido>2020-11-10</rendelesido>
    </rendeles>
    <rendeles rendeloid="0" futid="1">
        <ar>3500</ar>
        <kiszallido>20</kiszallido>
        <rendelesido>2020-11-11</rendelesido>
    </rendeles>
    <rendeles rendeloid="1" futid="1">
        <ar>7500</ar>
        <kiszallido>20</kiszallido>
        <rendelesido>2020-11-11</rendelesido>
    </rendeles>
    <rendeles rendeloid="3" futid="2">
        <ar>5000</ar>
        <kiszallido>30</kiszallido>
        <rendelesido>2020-11-12</rendelesido>
    </rendeles>
    <rendeles rendeloid="0" futid="1">
        <ar>3500</ar>
        <kiszallido>15</kiszallido>
        <rendelesido>2020-11-15</rendelesido>
    </rendeles>
    <rendeles rendeloid="1" futid="0">
        <ar>9000</ar>
        <kiszallido>25</kiszallido>
        <rendelesido>2020-11-16</rendelesido>
    </rendeles>
</rendelesek>

```

```
<futarok>
  <futar futid="0">
    <nev>Schmidt Henrik</nev>
    <telefonszam>06709453211</telefonszam>
    <fizetes>20000</fizetes>
  </futar>
  <futar futid="1">
    <nev>German Jakab</nev>
    <telefonszam>06303625510</telefonszam>
    <fizetes>30000</fizetes>
  </futar>
  <futar futid="2">
    <nev>Szabó Alen</nev>
    <telefonszam>0620</telefonszam>
    <fizetes>20000</fizetes>
  </futar>
</futarok>
<etelek>
  <etel etelid="0">
    <nev>Pepperoni pizza</nev>
    <elkeszitando>12</elkeszitando>
    <feltetek>Pepperoni</feltetek>
  </etel>
  <etel etelid="1">
    <nev>Sonkás pizza</nev>
    <elkeszitando>12</elkeszitando>
    <feltetek>Sonka</feltetek>
  </etel>
  <etel etelid="2">
    <nev>Hamburger</nev>
    <elkeszitando>8</elkeszitando>
    <feltetek>Húspogácsa</feltetek>
  </etel>
  <etel etelid="3">
    <nev>Rántott hús</nev>
    <elkeszitando>20</elkeszitando>
    <feltetek>Sültkrumpli</feltetek>
  </etel>
  <etel etelid="4">
    <nev>Májás rizotto</nev>
    <elkeszitando>35</elkeszitando>
    <feltetek>Rizs</feltetek>
  </etel>
</etelek>
<tartalmazas>
  <tartalmaz etelid="1" rendelesid="0">
    <elkeszito>Pista</elkeszito>
  </tartalmaz>
```

```

    <tartalmaz etelid="2" rendelesid="0">
      <elkeszito>Pista</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="3" rendelesid="0">
      <elkeszito>Pista</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="4" rendelesid="1">
      <elkeszito>János</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="0" rendelesid="1">
      <elkeszito>János</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="1" rendelesid="2">
      <elkeszito>János</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="3" rendelesid="3">
      <elkeszito>Pista</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="2" rendelesid="3">
      <elkeszito>Pista</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="0" rendelesid="4">
      <elkeszito>János</elkeszito>
    </tartalmaz>
    <tartalmaz etelid="0" rendelesid="5">
      <elkeszito>János</elkeszito>
    </tartalmaz>
  </tartalmazas>
</kiszallitas>

```

## 1d) XML dokumentum alapján XMLSchema:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema elementFormDefault="qualified" targetNamespace="beadando" xmlns="beadando"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="kiszallitas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="rendelok">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded"
name="rendelo">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="nev"
type="xs:string" />

```

```

        <xs:element name="email"
            type="xs:string" />
        <xs:element name="lakcim">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="iranyitoszam"
                        type="xs:unsignedLong" />
                    <xs:element name="varos"
                        type="xs:string" />
                    <xs:element name="utca"
                        type="xs:string" />
                    <xs:element name="hazszam"
                        type="xs:unsignedInt" />
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
    <xs:attribute name="rendeloid"
        type="xs:unsignedByte"
        use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rendelesek">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded"
                name="rendeles">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="ar"
                            type="xs:unsignedInt" />
                        <xs:element name="kiszallido"
                            type="xs:unsignedInt" />
                        <xs:element name="rendelesido"
                            type="xs:date" />
                    </xs:sequence>
                    <xs:attribute name="rendelesid"
                        type="xs:unsignedByte"
                        use="required" />
                    <xs:attribute name="rendeloid"
                        type="xs:unsignedByte"
                        use="required" />
                    <xs:attribute name="futid"
                        type="xs:unsignedByte"

```

```

        use="required" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="futarok">
<xs:complexType>
<xs:sequence>
    <xs:element maxOccurs="unbounded"
        name="futar">
    <xs:complexType>
    <xs:sequence>
        <xs:element name="nev"
            type="xs:string" />
        <xs:element name="telefonszam"
            type="xs:unsignedLong" />
        <xs:element name="fizetes"
            type="xs:unsignedInt" />
    </xs:sequence>
    <xs:attribute name="futid"
        type="xs:unsignedByte"
        use="required" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="etelek">
<xs:complexType>
<xs:sequence>
    <xs:element maxOccurs="unbounded"
        name="etel">
    <xs:complexType>
    <xs:sequence>
        <xs:element name="nev"
            type="xs:string" />
        <xs:element name="elkeszitido"
            type="xs:unsignedInt" />
        <xs:element name="feltetek"
            type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="etelid"
        type="xs:unsignedByte"
        use="required" />
    </xs:complexType>
</xs:element>

```



```

        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="tartalmazas">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded"
                name="tartalmaz">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="elkeszito"
                            type="xs:string" />
                    </xs:sequence>
                    <xs:attribute name="etelid"
                        type="xs:unsignedByte"
                        use="required" />
                    <xs:attribute name="rendelesid"
                        type="xs:unsignedByte"
                        use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

## 2. feladat

### 2a) adatolvasás

```

package hu.domparse.rv4tjx;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;

```

```

public class DOMReadRV4TJX {
    public static void main(String argv[])
    {
        try
        {
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document doc = db.parse(new File("XMLFeladatRV4TJX/XMLRV4TJX.xml")); //Hivatkozás a
            beolvasott file-ra
            doc.getDocumentElement().normalize();

            Element rootElement = doc.getDocumentElement(); //Gyökér elem kiválasztása
            System.out.println("Gyökér elem: " + rootElement.getNodeName());
            NodeList childNodes = rootElement.getChildNodes(); //Gyökér elem gyerekeinek eltárolása

            for(int i=0; i<childNodes.getLength(); i++) { //Gyerek elemek végigjárása
                Node node = childNodes.item(i);
                if(node.getNodeType() == Node.ELEMENT_NODE) {
                    System.out.println(node.getNodeName() + ": ");
                    Element element = (Element)node;
                    Node actualNode = element.getFirstChild();
                    while(actualNode != null) {
                        if(actualNode.getNodeType() == Node.ELEMENT_NODE) {
                            Element actualElement = (Element)actualNode;
                            System.out.println(" " + actualElement.getNodeName() + ": " +
                            actualElement.getTextContent()); //Kírfjuk az aktuális elem adatait
                        }
                        actualNode=actualNode.getNextSibling();
                    }
                }
            }
        }
        catch(ParserConfigurationException e){ //Futás esetében létrejövő kivételek kezelése
            e.printStackTrace();
        }
        catch (SAXException e) {
            e.printStackTrace();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

**2b)**

```
package hu.domparse.rv4tjx;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DOMModifyRV4TJX {
    public static void main(String[] args) {
        try{
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document doc = db.parse(new File("XMLFeladatRV4TJX/XMLRV4TJX.xml")); //XML
            dokumentum betöltése

            //Futárok neveinek és fizetéseinek megváltoztatása
            Node futarok = doc.getElementsByTagName("futarok").item(0);
            NodeList flist = futarok.getChildNodes();
            System.out.println("Módosított elemek: ");
            for (int i = 0; i < flist.getLength(); i++) {
                Node node = flist.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) node;
                    System.out.println("Név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
                    if(eElement.getElementsByTagName("nev").item(0).getTextContent().equals("Schmidt
Henrik")) {
                        eElement.getElementsByTagName("nev").item(0).setTextContent("Kiss Csaba");
                        System.out.println("Módosított név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
                    }
                    if(eElement.getElementsByTagName("nev").item(0).getTextContent().equals("German
Jakab")) {
                        eElement.getElementsByTagName("nev").item(0).setTextContent("Nagy Béla");
```

```

        System.out.println("Módosított név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
    }
    if(eElement.getElementsByTagName("fizetes").item(0).getTextContent().equals("20000")) {
        eElement.getElementsByTagName("fizetes").item(0).setTextContent("25000");
        System.out.println("Módosított fizetés : " +
eElement.getElementsByTagName("fizetes").item(0).getTextContent()+", nála:
"+eElement.getElementsByTagName("nev").item(0).getTextContent());
    }
}
}
//Rendelők adatainak módosítása
Node rendelok = doc.getElementsByTagName("rendelok").item(0);
NodeList rlist = rendelok.getChildNodes();

for (int i = 0; i < rlist.getLength(); i++) {
    Node node = rlist.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) node;
        System.out.println("Név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
        if(eElement.getElementsByTagName("nev").item(0).getTextContent().equals("Kiss Péter"))
        {
            eElement.getElementsByTagName("nev").item(0).setTextContent("Példa János");
            System.out.println("Módosított név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
        }
    }

    if(eElement.getElementsByTagName("email").item(0).getTextContent().equals("sanyesz@gmail.com"))
    {

        eElement.getElementsByTagName("email").item(0).setTextContent("sanyeszka@gmail.com");
        System.out.println("Módosított email : " +
eElement.getElementsByTagName("email").item(0).getTextContent());
    }
}
//Ételek adatainak módosítása
Node etelek = doc.getElementsByTagName("etelek").item(0);
NodeList elist = etelek.getChildNodes();

for (int i = 0; i < elist.getLength(); i++) {
    Node node = elist.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) node;
        System.out.println("Név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());

```

```

        if(eElement.getElementsByTagName("nev").item(0).getTextContent().equals("Sonkás
pizza")) {
            eElement.getElementsByTagName("nev").item(0).setTextContent("Hawaii pizza");
            System.out.println("Módosított név : " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
        }
        if(eElement.getElementsByTagName("feltetek").item(0).getTextContent().equals("Sonka"))
        {
            eElement.getElementsByTagName("feltetek").item(0).setTextContent("Ananász");
            System.out.println("Módosított feltet : " +
eElement.getElementsByTagName("feltetek").item(0).getTextContent());
        }
    }
}

//Read Osztály alapján való kiíratás
Element rootElement = doc.getDocumentElement();
System.out.println("Gyökér elem: " + rootElement.getNodeName());
NodeList childNodes = rootElement.getChildNodes();
for(int i=0; i<childNodes.getLength(); i++) {
    Node node = childNodes.item(i);
    if(node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element)node;
        Node actualNode = element.getFirstChild();
        while(actualNode != null) {
            if(actualNode.getNodeType() == Node.ELEMENT_NODE) {
                Element actualElement = (Element)actualNode;
                System.out.println(" " + actualElement.getNodeName() + ": " +
actualElement.getTextContent());
            }
            actualNode=actualNode.getNextSibling();
        }
        System.out.println();
    }
}

} catch(ParserConfigurationException e){
    e.printStackTrace();
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

```