

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Éttermek kiszállításának adatbázisa

Készítette: **Palkó Patrik Dávid**

Neptunkód: **ZW7DOR**

A feladat leírása:

A beadandó témája egy olyan adatbázis, amely egyszerre több kiszállítós éttermet kezel. Rákereshetünk benne az éttermekben dolgozó futárookra, a beszállítókra, Étteek mennyiségére, árára, vagy a vevő adataira

Az Adatbázisrendszerek I. című tárgy beadandó feladatomat helyeztem át XML-be

Az egyedek közötti kapcsolat

- **Étterem és Futár:**

Az Étterem és a Futár egyedek között egy a többhöz kapcsolat van, mivel egy étterem alkalmazhat több futárt, de egy futár csak egy étteremnél dolgozik.

- **Étterem és Beszállító:**

Az Étterem és a Beszállító egyedek között több a többhöz kapcsolat van, mivel egy étterem rendelhet több beszállítótól, valamint egy beszállító beszállíthat több étteremnek is. A kapcsolat paraméterei: a Hozzávalók, amely a beszállító által beszállított hozzávalókat jelenti, valamint a Dátum, azaz a beszállítás dátuma.

- **Étterem és Étel:**

Az Étterem és az Étel egyedek között egy a többhöz kapcsolat van, mivel egy étteremnek lehet több étele is, de egy étel csak egy étteremhez tartozhat.

- **Étel és Vevő:**

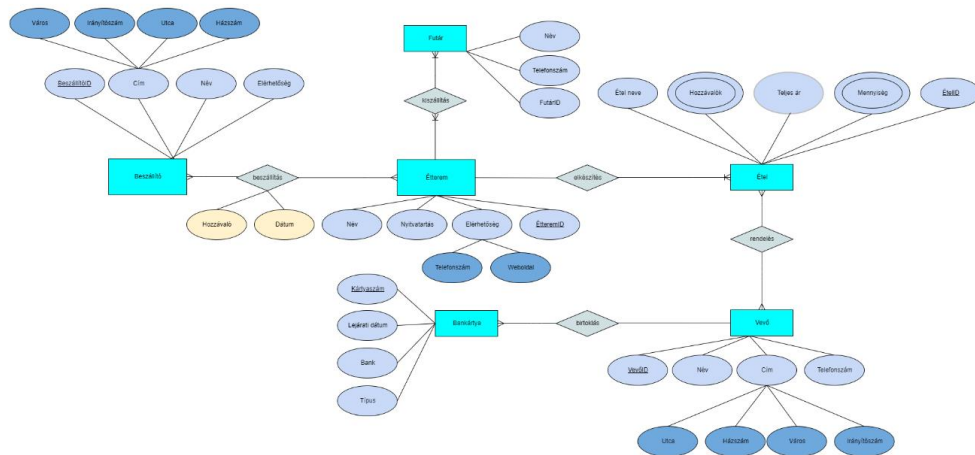
Az Étel és a Vevő egyedek között több a többhöz kapcsolat van, mivel egy vevő rendelhet többfajta ételt, és egy ételből rendelhet több különböző vevő is.

- **Vevő és Bankkártya:**

A Vevő és a Bankkártya egyedek között egy a többhöz kapcsolat van, mivel egy vevőnek lehet több bankkártyája is, de egy bankkártyának nem lehet több tulajdonosa

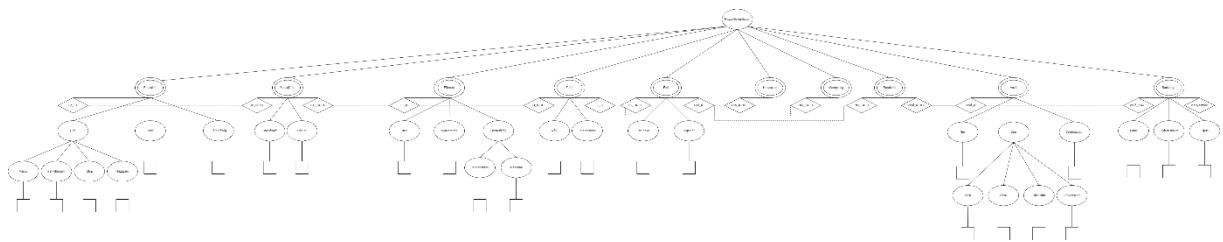
1.Feladat

1a/ Az adatbázis ER modellje:



1b/ Adatbázis

konvertálása XDM modellre:



1c/ XDM alapján egy XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<kiszallito_rendszer xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNamespaceSchemaLocation="XMLSchemaZW7DOR.xsd">

  <!-- Éterem -->

  <etterem ET_ID="1">
    <Nyitvatartas>11től 22:30ig</Nyitvatartas>
    <Név>Kalipszo</Név>
    <Elérhetőség>
      <Telefonszám>209876543</Telefonszám>
      <Weboldal>kalipszo.hu</Weboldal>
    </Elérhetőség>
  </etterem>

  <etterem ET_ID="2">
    <Nyitvatartas>9től 21ig</Nyitvatartas>
    <Név>Vendelin</Név>
    <Elérhetőség>
      <Telefonszám>208765432</Telefonszám>
      <Weboldal>vendelin.hu</Weboldal>
    </Elérhetőség>
  </etterem>

  <etterem ET_ID="3">
    <Nyitvatartas>8től 22ig</Nyitvatartas>
    <Név>Molnár Csárda</Név>
    <Elérhetőség>
      <Telefonszám>306876546</Telefonszám>
      <Weboldal>molnarcsarada.hu</Weboldal>
    </Elérhetőség>
  </etterem>

</kiszallito_rendszer>
```

```
<!--Futár-->
```

```
<futar F_ID="1" ET_IDREF="1">
```

```
  <Nev>Nagy Tíbor</Nev>
```

```
  <Telefonszam>202354342</Telefonszam>
```

```
</futar>
```

```
<futar F_ID="2" ET_IDREF="2">
```

```
  <Nev>Horváth Lajos</Nev>
```

```
  <Telefonszam>203521958</Telefonszam>
```

```
</futar>
```

```
<futar F_ID="3" ET_IDREF="3">
```

```
  <Nev>Palkó Patrik</Nev>
```

```
  <Telefonszam>202572553</Telefonszam>
```

```
</futar>
```

```
<!--Beszallito-->

<beszallito Be_ID="1">
  <Nev>Kis Piac</Nev>
  <Elerhetoseg>kis.piac@gmail.com</Elerhetoseg>
  <Cim>
    <Varos>Kazincbarcika</Varos>
    <Iranyitoszam>3700</Iranyitoszam>
    <Utca>Fő utca</Utca>
    <Hatszam>35</Hatszam>
  </Cim>
</beszallito>

<beszallito Be_ID="2">
  <Nev>Nagy piac</Nev>
  <Elerhetoseg>nagy.piac@gmail.com</Elerhetoseg>
  <Cim>
    <Varos>Encs</Varos>
    <Iranyitoszam>3896</Iranyitoszam>
    <Utca>Fő utca</Utca>
    <Hatszam>25</Hatszam>
  </Cim>
</beszallito>

<beszallito Be_ID="3">
  <Nev>Közepes piac</Nev>
  <Elerhetoseg>közepes.piac@gmail.com</Elerhetoseg>
  <Cim>
    <Varos>Miskolc</Varos>
    <Iranyitoszam>3885</Iranyitoszam>
    <Utca>Fő utca</Utca>
    <Hatszam>15</Hatszam>
  </Cim>
</beszallito>
```

```
<!--Beszállítás-->

<beszallitas Be_IDREF="1" ET_IDREF="1">
  <Hozzavalo>banan</Hozzavalo>
  <Datum>2022.08.06</Datum>
</beszallitas>

<beszallitas Be_IDREF="2" ET_IDREF="2">
  <Hozzavalo>csoki</Hozzavalo>
  <Datum>2022.03.01</Datum>
</beszallitas>

<beszallitas Be_IDREF="3" ET_IDREF="3">
  <Hozzavalo>alma</Hozzavalo>
  <Datum>2022.10.21</Datum>
</beszallitas>

<!--Étel-->

<etel Etel_ID="1" ET_IDREF="1">
  <etel_neve>Banános palacsinta</etel_neve>
  <teljes_ar>2500</teljes_ar>
</etel>

<etel Etel_ID="2" ET_IDREF="2">
  <etel_neve>Csokis zserbó</etel_neve>
  <teljes_ar>3400</teljes_ar>
</etel>

<etel Etel_ID="3" ET_IDREF="3">
  <etel_neve>Disznó almával a szájában</etel_neve>
  <teljes_ar>25000</teljes_ar>
</etel>
```

```
<!--Vevő-->

<vevo vevo_ID="1">
  <Nev>Palkó Zsolt</Nev>
  <Telefonszam>202566311</Telefonszam>
  <Cim>
    <Utca>Károlyi Gáspár</Utca>
    <Varos>Szerencs</Varos>
    <Hazzsam>5</Hazzsam>
    <Iranyitoszam>3880</Iranyitoszam>
  </Cim>
</vevo>

<vevo vevo_ID="2">
  <Nev>Lakatos Valentin</Nev>
  <Telefonszam>202599210</Telefonszam>
  <Cim>
    <Utca>Kálmán</Utca>
    <Varos>Budapest</Varos>
    <Hazzsam>31</Hazzsam>
    <Iranyitoszam>3550</Iranyitoszam>
  </Cim>
</vevo>

<vevo vevo_ID="3">
  <Nev>Raul Patricio</Nev>
  <Telefonszam>202599901</Telefonszam>
  <Cim>
    <Utca>Nagy Lajos</Utca>
    <Varos>Pálháza</Varos>
    <Hazzsam>2</Hazzsam>
    <Iranyitoszam>3770</Iranyitoszam>
  </Cim>
</vevo>
```

```

<!--Bankártya-->

<bankartya kartyaszam="7629746286234247" vevo_IDREF="1">
  <Bank>OTP</Bank>
  <Lejarati_datum>2025.10.28</Lejarati_datum>
  <Tipus>betéti kártya</Tipus>
</bankartya>

<bankartya kartyaszam="(0543345349664954" vevo_IDREF="1">
  <Bank>MKB</Bank>
  <Lejarati_datum>2025.10.29</Lejarati_datum>
  <Tipus>hitelkártya</Tipus>
</bankartya>

<bankartya kartyaszam="5432467233423432" vevo_IDREF="1">
  <Bank>CIB Bank</Bank>
  <Lejarati_datum>2025.10.30</Lejarati_datum>
  <Tipus>többfunkciós kártya</Tipus>
</bankartya>

/kiszallito_rendszer>

```

1d/ XML alapján XMLSchema készítése:


```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:element name="kiszallito_rendszer">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="etterem" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Nyitvatartas" type="xs:string"/></xs:element>
              <xs:element name="Nev" type="xs:string"/></xs:element>
              <xs:element name="Elerhetoseg">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Telefonszam" type="xs:int"/></xs:element>
                    <xs:element name="Weboldal" type="xs:string"/></xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="ET_ID" type="xs:int"/></xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:element name="futar" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Nev" type="xs:string"/></xs:element>
              <xs:element name="Telefonszam" type="xs:int"/></xs:element>
            </xs:sequence>
            <xs:attribute name="F_ID" type="xs:int"/></xs:attribute>
            <xs:attribute name="ET_IDREF" type="xs:int"/></xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:element name="beszallito" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Nev" type="xs:string"/></xs:element>
              <xs:element name="Elerhetoseg" type="xs:string"/></xs:element>
              <xs:element name="Cim">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Varos" type="xs:string"/></xs:element>
                    <xs:element name="Iranditoszam" type="xs:int"/></xs:element>
                    <xs:element name="Utca" type="xs:string"/></xs:element>
                    <xs:element name="Hatszszam" type="xs:int"/></xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Be_ID" type="xs:int"></xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="beszallitas" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Hozzavalo" type="xs:string"></xs:element>
      <xs:element name="Datum" type="xs:date"></xs:element>
    </xs:sequence>
    <xs:attribute name="Be_IDREF" type="xs:int"></xs:attribute>
    <xs:attribute name="ET_IDREF" type="xs:int"></xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="etel" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="etel_neve" type="xs:string"></xs:element>
      <xs:element name="teljes_ar" type="xs:int"></xs:element>
    </xs:sequence>
    <xs:attribute name="Etel_ID" type="xs:int"></xs:attribute>
    <xs:attribute name="ET_IDREF" type="xs:int"></xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="vevo" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Nev" type="xs:string"></xs:element>
      <xs:element name="Telefonszam" type="xs:int"></xs:element>
      <xs:element name="Cim">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Utca" type="xs:string"></xs:element>
            <xs:element name="Varos" type="xs:string"></xs:element>
            <xs:element name="Hazszam" type="xs:int"></xs:element>
            <xs:element name="Iranditoszam" type="xs:int"></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="vevo_ID" type="xs:int"></xs:attribute>
  </xs:complexType>

```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="bankartya" maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Bank" type="xs:string"></xs:element>
        <xs:element name="Lejarati_datum" type="xs:date"></xs:element>
        <xs:element name="Tipus" type="xs:string"></xs:element>
      </xs:sequence>
      <xs:attribute name="kartyaszam" type="xs:int"></xs:attribute>
      <xs:attribute name="vevo_IDREF" type="xs:int"></xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name="xmlns:xs" type="xs:string"></xs:attribute>
<xs:attribute name="xs:noNamespaceSchemaLocation" type="xs:string"></xs:attribute>
</xs:complexType>
</xs:element>
</xs:schema>

```

2.Feladat

2a/ Adatolvasás

```

package hu.dompars.ZW7DOR;

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.w3c.dom.Text;
import org.xml.sax.SAXException;

public class DomReadZW7DOR {

    public static void main(String[] args) {

        File xmlFile = new File("XMLZW7DOR.xml");
        Document doc = introduceFile(xmlFile);

        if (doc != null) {
            doc.getDocumentElement().normalize();
            System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        } else {
            System.out.println("Document is null");
            System.exit(-1);
        }

        NodeList nodeList = doc.getDocumentElement().getChildNodes();
        String indent = "";
        listData(nodeList, indent);

    }

    public static Document introduceFile (File xmlFile){
        Document doc = null;

        try{
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            doc = dbBuilder.parse(xmlFile);
        } catch (ParserConfigurationException | SAXException | IOException e) {
            e.printStackTrace();
        }

        return doc;
    }
}

```

```

public static void listData(NodeList nodeList, String indent){
    indent += "\t";

    if(nodeList != null) {
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE && !node.getTextContent().trim().isEmpty()) {
                System.out.println(indent + "{" + node.getNodeName() + "}:"");
                NodeList nodeList_new = node.getChildNodes();
                listData(nodeList_new, indent);
            } else if (node instanceof Text){
                String value = node.getNodeValue().trim();
                if (value.isEmpty()){
                    continue;
                }
                System.out.println(indent + node.getTextContent());
            }
        }
    }
}

```

2b/ Adatlekérdezés

```

package hu.domparsa.ZW7DOR;

import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.Text;
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 DomQueryZW7DOR {

    public static void main(String[] args) {

        File xmlFile = new File("XMLZW7DOR.xml");
        Document doc = introduceFile(xmlFile);

        if (doc != null) {
            doc.getDocumentElement().normalize();
        } else {
            System.out.println("Document is null");
            System.exit(-1);
        }

        NodeList queryList = doc.getDocumentElement().getElementsByTagName("etel");

        for (int i = 0; i < queryList.getLength(); i++) {
            NodeList query = queryList.item(i).getChildNodes();
            for (int j = 0; j < query.getLength(); j++) {
                if (query.item(j).getNodeName().equals("teljes_ar") && Integer.parseInt(query.item(j).getTextContent()) > 3000){
                    System.out.println("{etel}");
                    listData(queryList.item(i).getChildNodes(), "");
                }
            }
        }

        public static Document introduceFile (File xmlFile){
            Document doc = null;

```

```

public static Document introduceFile (File xmlFile){
    Document doc = null;

    try {
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
        doc = dbBuilder.parse(xmlFile);
    } catch (ParserConfigurationException | SAXException | IOException e) {
        e.printStackTrace();
    }
    return doc;
}

public static void listData(NodeList nodeList, String indent){
    indent += "\t";

    if (nodeList != null) {
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE && !node.getTextContent().trim().isEmpty()) {
                System.out.println(indent + "{" + node.getNodeName() + "}:");
                NodeList nodeList_new = node.getChildNodes();
                listData(nodeList_new, indent);
            } else if (node instanceof Text){
                String value = node.getNodeValue().trim();
                if (value.isEmpty()){
                    continue;
                }
                System.out.println(indent + node.getTextContent());
            }
        }
    }
}

```

2a/ Adatmódosítás

```

package hu.domp.parse.ZW7DOR;

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.*;
import org.xml.sax.SAXException;

public class DomModifyZW7DOR {

    public static void main(String[] args) {

        File xmlFile = new File("XMLZW7DOR.xml");
        Document doc = introduceFile(xmlFile);

        if (doc != null) {
            doc.getDocumentElement().normalize();
            System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
        } else {
            System.out.println("Document is null");
            System.exit(-1);
        }

        NodeList modifyList = doc.getDocumentElement().getElementsByTagName("teljes_ar");
        modifyData(modifyList);

        NodeList queryList = doc.getDocumentElement().getElementsByTagName("etel");
        for (int i = 0; i < queryList.getLength(); i++) {
            NodeList query = queryList.item(i).getChildNodes();
            for (int j = 0; j < query.getLength(); j++) {
                if (query.item(j).getNodeName().equals("teljes_ar")){
                    System.out.println("{etel}");
                    listData(queryList.item(i).getChildNodes(), "");
                }
            }
        }
    }

    public static Document introduceFile (File xmlFile){
        Document doc = null;
    }

```

```

public static Document introduceFile (File xmlFile){
    Document doc = null;

    try{
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
        doc = dbBuilder.parse(xmlFile);
    } catch (ParserConfigurationException | SAXException | IOException e) {
        e.printStackTrace();
    }
    return doc;
}

public static void listData(NodeList nodeList, String indent){
    indent += "\t";

    if(nodeList != null) {
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE && !node.getTextContent().trim().isEmpty()) {
                System.out.println(indent + "{" + node.getNodeName() + "}:");
                NodeList nodeList_new = node.getChildNodes();
                listData(nodeList_new, indent);
            } else if (node instanceof Text){
                String value = node.getNodeValue().trim();
                if (value.isEmpty()){
                    continue;
                }
                System.out.println(indent + node.getTextContent());
            }
        }
    }
}

public static void modifyData(NodeList nodeList){
    if (nodeList != null){
        for (int i = 0; i < nodeList.getLength(); i++) {
            int valuechange = Integer.parseInt(nodeList.item(i).getTextContent());
            valuechange -= 4500;
            nodeList.item(i).setTextContent(Integer.toString(valuechange));
        }
    }
}

```