

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Készítette: Magyar Márk

Neptun-kód: CWLKE4

Dátum: 2020. november 30.

A feladat leírása:

Egy gyógyszertárat és a hozzá tartozó entitásokat (gyógyszer, gyógyszerész, beszállító) modellezem. A beszállító és a gyógyszertár között lesz N:M kapcsolat, a többinél pedig 1:N kapcsolat.

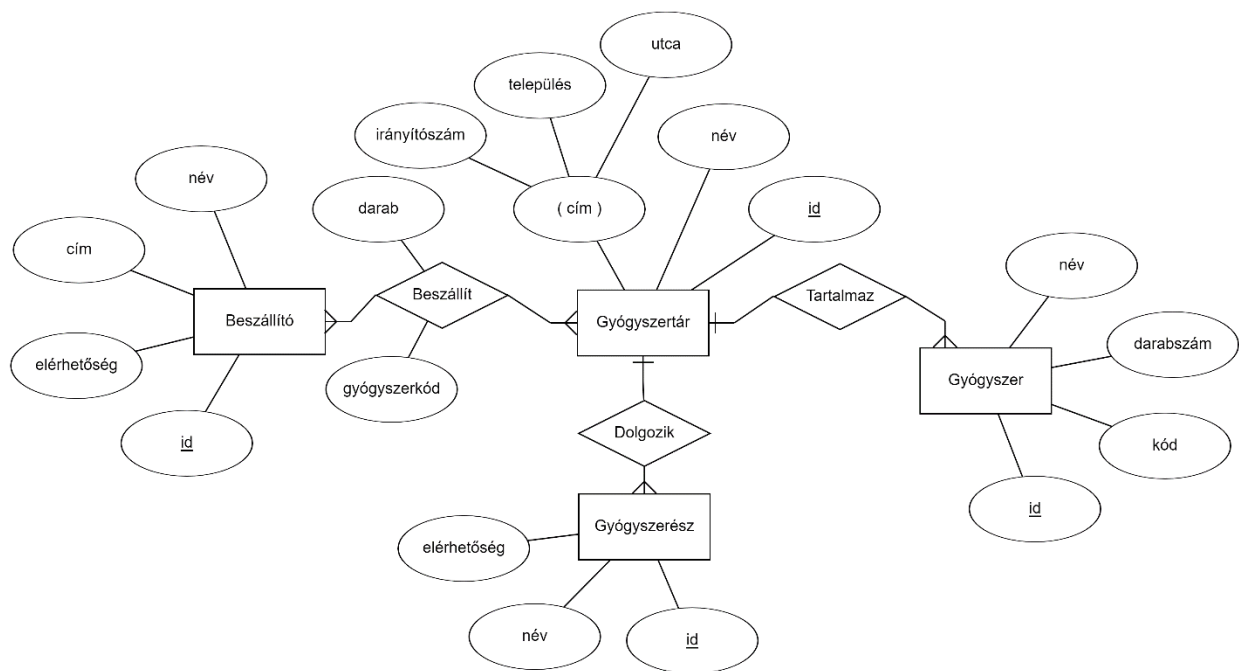
Minden entitás megkapta az attribútumát, saját egyedi kulcs, normál attribútum, valamint címe a gyógyszertárnak összetett.

Az első feladatban az ER modell látható, valamint az XDM modell, majd az XML dokumentum és a hozzá tartozó séma.

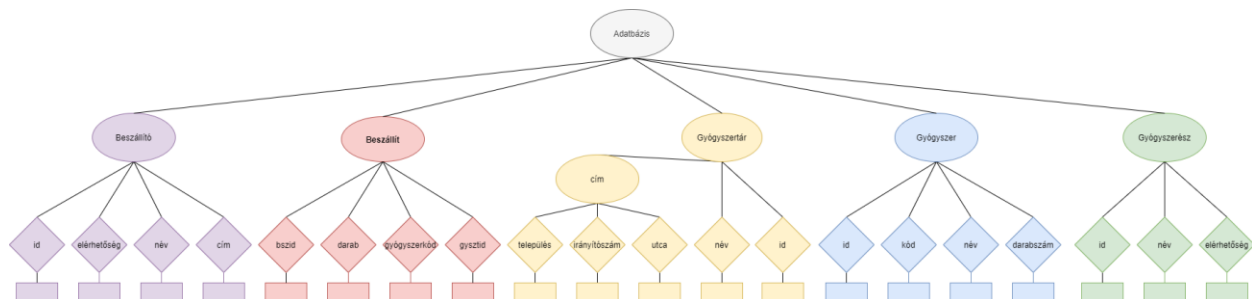
A második feladatban pedig az XML fájl összes entitását kiírja, majd a gyógyszertár nevét módosítani tudjuk, ez egy új fájlt fog létrehozni a már meglévő XML dokumentumból a belevitt módosítással.

1. feladat

a. ER modell:



b. XDM modell:



c. XML dokumentum:

```
<?xml version="1.0" encoding="UTF-8"?>
<adatbazis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="XMLSchemaCwlke4.xsd">
  <gyogyszertar id="1">
    <cim>
      <telepules>Miskolc</telepules>
      <iranyitoszam>3530</iranyitoszam>
      <utca>Városháztér 2</utca>
    </cim>
  </gyogyszertar>

  <beszallito id="1">
    <elerhetoseg>06307070700</elerhetoseg>
    <nev>A beszállító</nev>
    <cim>3610 Gyöngyös, Patakpart 1</cim>
  </beszallito>

  <beszallito id="2">
    <elerhetoseg>06308080800</elerhetoseg>
    <nev>B beszállító</nev>
    <cim>3410 Határújfalu, István u. 7</cim>
  </beszallito>

  <gyogyszer id="1">
    <kod>124145244</kod>
    <nev>Antineptunoxid</nev>
    <darabszam>12</darabszam>
  </gyogyszer>

  <gyogyszer id="2">
    <kod>54565845</kod>
    <nev>Vanaláírástonocid</nev>
    <darabszam>21</darabszam>
  </gyogyszer>

  <gyogyszeresz id="1">
    <nev>Dr. Gyógyszerész Béla</nev>
    <elerhetoseg>06205556667</elerhetoseg>
  </gyogyszeresz>

  <gyogyszeresz id="2">
    <nev>Dr. Gyógyszerész András</nev>
    <elerhetoseg>06205556654</elerhetoseg>
  </gyogyszeresz>

  <beszallit bszid="1" gyszid="1">
```

```
        <darab>12</darab>
        <gyogyszerkod>823852856</gyogyszerkod>
    </beszallit>

    <beszallit bszid="2" gysztid="1">
        <darab>4</darab>
        <gyogyszerkod>464354861</gyogyszerkod>
    </beszallit>
</adatbazis>
```

d. XMLSchema dokumentum:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="adatbazis">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="gyogyszertar">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="telepules" type="xs:string"/></xs:element>
              <xs:element name="cim">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="telepules" type="xs:string"/></xs:element>
                    <xs:element name="iranyitoszam" type="xs:integer"/></xs:element>
                    <xs:element name="utca" type="xs:string"/></xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:integer"/></xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:element name="beszallito">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="elerhetoseg" type="xs:integer"/></xs:element>
              <xs:element name="nev" type="xs:string"/></xs:element>
              <xs:element name="cim" type="xs:string"/></xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:integer"/></xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszer">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="kod" type="xs:integer"/></xs:element>
              <xs:element name="nev" type="xs:string"/></xs:element>
              <xs:element name="darabszam" type="xs:integer"/></xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:integer"/></xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszeresz">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nev" type="xs:string"/></xs:element>
              <xs:element name="elerhetoseg" type="xs:integer"/></xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
        </xs:sequence>
        <xs:attribute name="id" type="xs:integer"></xs:attribute>
    </xs:complexType>
</xs:element>
<xs:element name="beszallit">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="darab" type="xs:integer"></xs:element>
            <xs:element name="gyogyszerkod" type="xs:integer"></xs:element>
        </xs:sequence>
        <xs:attribute name="bszid" type="xs:integer"></xs:attribute>
        <xs:attribute name="gysztid" type="xs:integer"></xs:attribute>
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>feladat
```

2. feladat

A feladat egy DOM program készítése az XML dokumentum adatainak adminisztrálása alapján:

a) adatolvasás

```
package hu.domparse.cwlke4;

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 javax.xml.transform.TransformerException;

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

public class DOMReadCWLKE4 {

    public static void main(String args[]) throws ParserConfigurationException,
    IOException, SAXException, TransformerException {
        try {
            File inputFile = new File("XMLCwlke4.xml");
            DocumentBuilderFactory dbFactory =
            DocumentBuilderFactory.newInstance();

            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            Document doc = dBuilder.parse(inputFile);
            doc.getDocumentElement().normalize();
            System.out.println("Gyökér elem : " +
            doc.getDocumentElement().getNodeName());
            System.out.println("-----");

            NodeList nList = doc.getElementsByTagName("gyogyszertar");
            for (int i = 0; i < nList.getLength(); i++) {
                Node nNode = nList.item(i);
                System.out.println("\nEntitás: " + nNode.getNodeName());
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    System.out.println("Gyógyszertár id: " +
                    eElement.getAttribute("id"));
                    System.out.println("Gyógyszertár neve: " +
                    eElement.getElementsByTagName("nev").item(0).getTextContent());
                    System.out.println("Település: " +
                    eElement.getElementsByTagName("telepules").item(0).getTextContent());
                    System.out.println("Írányítószám: " +
                    eElement.getElementsByTagName("iranyitoszam").item(0).getTextContent());
                    System.out.println("Utca: " +
                    eElement.getElementsByTagName("utca").item(0).getTextContent());
                }
            }
        }
    }
}
```

```

        nList = doc.getElementsByTagName("gyogyszer");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            System.out.println("\nEntitás: " + nNode.getNodeName());
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element eElement = (Element) nNode;
                System.out.println("Gyógyszer id: " +
eElement.getAttribute("id"));
                System.out.println("Gyógyszer kód: " +
eElement.getAttribute("kod"));
                System.out.println("Gyógyszer neve: " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
            }
        }

        nList = doc.getElementsByTagName("gyogyszeresz");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            System.out.println("\nEntitás: " + nNode.getNodeName());
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element eElement = (Element) nNode;
                System.out.println("Gyógyszerész id: " +
eElement.getAttribute("id"));
                System.out.println("Gyógyszerész neve: " +
eElement.getAttribute("nev"));
                System.out.println("Elérhetőség: " +
eElement.getAttribute("elerhetoseg"));
            }
        }

        nList = doc.getElementsByTagName("beszallito");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            System.out.println("\nEntitás: " + nNode.getNodeName());
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element eElement = (Element) nNode;
                System.out.println("Beszállító Id : " +
eElement.getAttribute("id"));
                System.out.println("Beszállító neve: " +
eElement.getElementsByTagName("nev").item(0).getTextContent());
                System.out.println("Elérhetőség: " +
eElement.getElementsByTagName("elerhetoseg").item(0).getTextContent());
                System.out.println("Cím: " +
eElement.getElementsByTagName("cim").item(0).getTextContent());
            }
        }

        nList = doc.getElementsByTagName("beszallit");
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            System.out.println("\nEntitás:" + nNode.getNodeName());
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element eElement = (Element) nNode;

```



```

        System.out.println("Beszállító id : " +
eElement.getAttribute("bszid"));
        System.out.println("Gyógyszertár id : " +
eElement.getAttribute("gysztid"));
        System.out.println("Darabszám: " +
eElement.getElementsByTagName("darab").item(0).getTextContent());
        System.out.println("Gyógyszerkód: " +
eElement.getElementsByTagName("gyogyszerkod").item(0).getTextContent());
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

b) adatmódosítás

```
package hu.domparse.cwlke4;

import java.io.File;
import java.io.IOException;
import java.util.Scanner;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

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 DOMModifyCWLKE4 {
    public static void main(String args[]) throws ParserConfigurationException,
        IOException, SAXException, TransformerException {

        String filePath = "XMLCwlke4.xml";
        File xmlFile = new File(filePath);

        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder;

        try {
            dBuilder = dbFactory.newDocumentBuilder();

            Document doc = dBuilder.parse(xmlFile);

            doc.getDocumentElement().normalize();

            gyogyszertarNevmodositas(doc);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static String ReadId() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Gyogyszertar Id: ");
        String id = sc.nextLine();
        return id;
    }

    public static void xmlFileIras(Document doc) throws TransformerException {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
```

```

        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(new
File("src/hu/domparse/CWLKE4/XMLCwlke4.updated.xml"));
        transformer.transform(source, result);
    }

    public static void gyogyszertarNevmodositas(Document doc) throws
TransformerException {

        System.out.println("Melyik gyógyszerár nevét írjuk át?");
        String idread = ReadId();

        Scanner sc = new Scanner(System.in);
        System.out.print("Új név: ");
        String gysztnev = sc.nextLine();

        NodeList nList = doc.getElementsByTagName("gyogyszertar");
        for (int i = 0; i < nList.getLength(); i++) {

            Node nNode = nList.item(i);

            if (nNode.getNodeType() == Node.ELEMENT_NODE) {

                Element element = (Element) nNode;

                String gysztid = element.getAttribute("id");

                if (gysztid.equals(idread)) {

                    Node node1 =
element.getElementsByTagName("nev").item(0);
                    node1.setTextContent(gysztnev);

                    System.out.println("Sikeres Modosítás");
                    xmlFileIras(doc);
                }
            }
        }
    }
}

```