Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Ruzsin Péter**

Neptunkód: **QOHYCR**

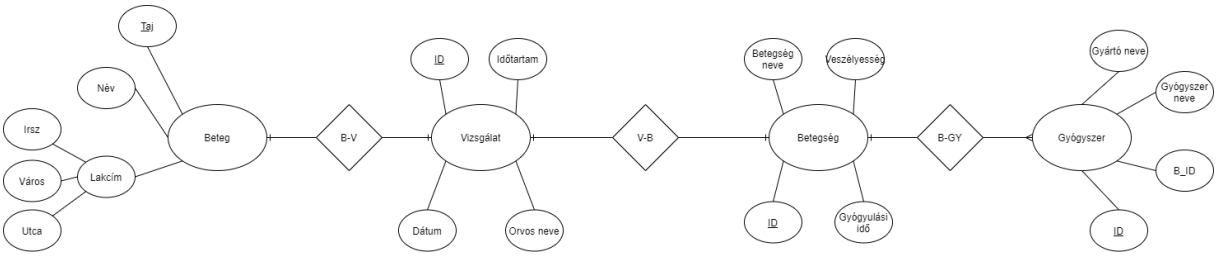
A feladat leírása:

A feladatban egy rendelőintézet rendeléseihez kacsolódó adatbázis elkészítése. Ehhez használunk 4 egyedet, melyeknél megkell adni a következőket:

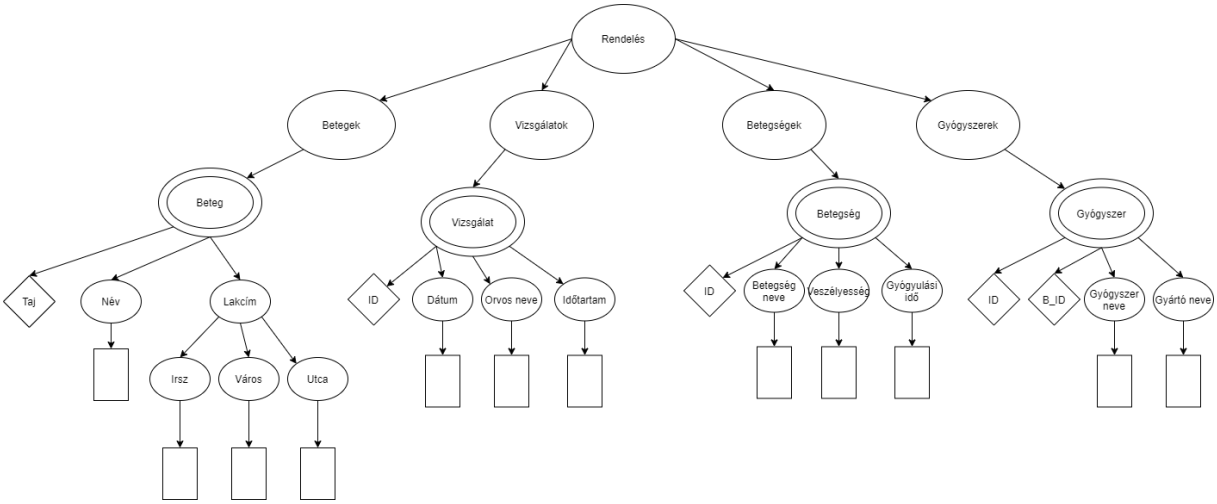
1. Beteg
    a. Taj (Mint id)
    b. Név
    c. Lakcím
        i. Irsz
        ii. Város
        iii. Utca
2. Vizsgálat
    a. ID
    b. Dátum
    c. Orvos neve
3. Betegség
    a. ID
    b. Gyógyulási idő (Hétben)
    c. Betegség neve
4. Gyógyszer
    a. ID
    b. Gyógyszer neve
    c. Gyártó neve

1 Feladat:

1a.) ER model:



1b.) XDM model:



1c.) XML dokumentum:

```
<?xml version="1.0" encoding="UTF-8"?>
<rendeles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaQOHYCR.xsd">
  <betegek>
      <beteg taj="001">
        <nev>Kiss József</nev>
        <lakcím>
            <irsz>3532</irsz>
            <varos>Miskolc</varos>
            <utca>Herman Ottó utca 6</utca>
```

```
        </lakcím>
      </beteg>
      <beteg taj="002">
        <nev>Nagy Béla</nev>
        <lakcím>
            <irsz>3532</irsz>
            <varos>Miskolc</varos>
            <utca>Thököly utca 16</utca>
        </lakcím>
      </beteg>
      <beteg taj="003">
        <nev>Kovács Jenő</nev>
        <lakcím>
            <irsz>3532</irsz>
            <varos>Miskolc</varos>
            <utca>Aranyásó utca 35</utca>
        </lakcím>
      </beteg>
</betegek>
<vizsgalatok>
      <vizsgalat ID="1">
        <datum>2020.11.29.</datum>
        <orvos_neve>DR. Surányi Eszter</orvos_neve>
        <idotartam>20</idotartam>
```

```
    </vizsgalat>
    <vizsgalat ID="2">
      <datum>2020.11.29.</datum>
      <orvos_neve>DR. Surányi Eszter</orvos_neve>
      <idotartam>40</idotartam>
    </vizsgalat>
    <vizsgalat ID="3">
      <datum>2020.11.29.</datum>
      <orvos_neve>DR. Surányi Eszter</orvos_neve>
      <idotartam>52</idotartam>
    </vizsgalat>
    <vizsgalat ID="4">
      <datum>2020.11.30.</datum>
      <orvos_neve>DR. Surányi Eszter</orvos_neve>
      <idotartam>95</idotartam>
    </vizsgalat>
    <vizsgalat ID="5">
      <datum>2020.11.30.</datum>
      <orvos_neve>DR. Surányi Eszter</orvos_neve>
      <idotartam>34</idotartam>
    </vizsgalat>
</vizsgalatok>
<betegsegek>
    <betegseg ID="1">
```

```
        <betegseg_neve>Torok gyulladás</betegseg_neve>
        <veszelyesseg>4</veszelyesseg>
        <gyogyulasi_ido>2</gyogyulasi_ido>
      </betegseg>
      <betegseg ID="2">
        <betegseg_neve>Innhüvely gyulladás</betegseg_neve>
        <veszelyesseg>6</veszelyesseg>
        <gyogyulasi_ido>3</gyogyulasi_ido>
      </betegseg>
  </betegsegek>
  <gyogyszerek>
      <gyogyszer ID="1" B_ID="1">
        <gyogyszer_neve>Neocitran</gyogyszer_neve>
        <gyarto_neve>GlaxoSmithKline-Consumer Kft</gyarto_neve>
      </gyogyszer>
      <gyogyszer ID="2" B_ID="1">
        <gyogyszer_neve>ACC Long</gyogyszer_neve>
        <gyarto_neve>Sandoz Hungária Kft</gyarto_neve>
      </gyogyszer>
  </gyogyszerek>
</rendeles>
```

1d.) XMLSchema:

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
<xs:element name="rendeles">
    <xs:complexType>
     <xs:sequence>
          <xs:element name="beteg" maxOccurs="unbounded"
minOccurs="0">
           <xs:complexType>
                <xs:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
                <xs:sequence>
                 <xs:element type="xs:string"
name="nev"></xs:element>
                 <xs:element name="lakcim">
                      <xs:complexType>
                       <xs:sequence>
                            <xs:element type="xs:byte"
name="irsz"></xs:element>
                            <xs:element type="xs:string"
name="varos"></xs:element>
                            <xs:element type="xs:string"
name="utca"></xs:element>
                       </xs:sequence>
                      </xs:complexType>
                 </xs:element>
                </xs:sequence>
           </xs:complexType>
          </xs:element>
```

```
            <xs:element name="vizsgalat" maxOccurs="unbounded"
minOccurs="0">
              <xs:complexType>
                <xs:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
                <xs:sequence>
                  <xs:element type="xs:date"
name="datum"></xs:element>
                  <xs:element type="xs:string"
name="orvos_neve"></xs:element>
                  <xs:element type="xs:byte"
name="idotartam"></xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="betegseg" maxOccurs="unbounded"
minOccurs="0">
              <xs:complexType>
                <xs:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
                <xs:sequence>
                  <xs:element type="xs:string"
name="betegseg_neve"></xs:element>
                  <xs:element type="xs:byte"
name="veszelyesseg"></xs:element>
                  <xs:element type="xs:byte"
name="gyogyulasi_ido"></xs:element>
```

```
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszer" maxOccurs="unbounded"
minOccurs="0">
            <xs:complexType>
                <xs:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
                <xs:attribute type="xs:byte" name="b_id"
use="required"></xs:attribute>
                <xs:sequence>
                    <xs:element type="xs:string"
name="gyogyszer_neve"></xs:element>
                    <xs:element type="xs:string"
name="gyarto_neve"></xs:element>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

2 Feladat

2a.)

```java
package hu.domparse.qohycr;

import javax.xml.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import hu.domparse.qohycr.DOMModifyQOHYCR;

public class DOMReadQOHYCR {
    private static void printNode(Node n) {
        NodeList nl = n.getChildNodes();
        for (int i = 0; i < nl.getLength(); i++) {
            if(nl.item(i).getNodeName) {
                Node no = nl.item(i);
                System.out.println(no.getNodeName());
                if(no.hasChildNodes()) {
                    for(int j = 0; j < no.getChildNodes().getLength(); j++) {
                        Node node = no.getChildNodes().item(j);
                        if(node.getNodeType() == Node.ELEMENT_NODE) {
                            System.out.println("\t" + node.getNodeName() + ": " + node.getTextContent());
                        }
                    }
                }
            }
        }
    }

    public static void main(String[] args) {
        try {
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document document = db.parse("XMLqohycr.xml");
            Element root = document.getDocumentElement();
            printNode(root);
            DOMModifyQOHYCR.modifyDuration(root);
            printNode(root);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

2b.)

```java
package hu.domparse.qohycr;

import javax.xml.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMModifyQOHYCR {
    public static void modifyDuration(Node root) {
        try {
            System.out.println(root.getNodeType());
            NodeList nl = root.getChildNodes();
            for(int i = 0; i< nl.getLength(); i++){
                if (nl.item(i).getNodeName() == "vizsgalatok") {
                    for(int j = 0; j < nl.item(i).getChildNodes().getLength(); j++) {
                        if (nl.item(i).getChildNodes().item(j).getNodeName() == "vizsgalat") {
                            for(int k = 0; k < nl.item(i).getChildNodes().item(j).getChildNodes().getLength(); k++) {
                                if(nl.item(i).getChildNodes().item(j).getChildNodes().item(k).getNodeName() == "idotartam") {
                                    int duration = Integer.parseInt(nl.item(i).getChildNodes().item(j).getChildNodes().item(k).getTextContent());
                                    nl.item(i).getChildNodes().item(j).getChildNodes().item(k).setTextContent(String.valueOf(duration + 15));
                                }
                            }
                        }
                    }
                }
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```