

# **Jegyzőkönyv**

Adatkezelés XML környezetben  
Féléves feladat

Készítette: **Ruzsin Péter**  
Neptunkód: **QOHYCR**

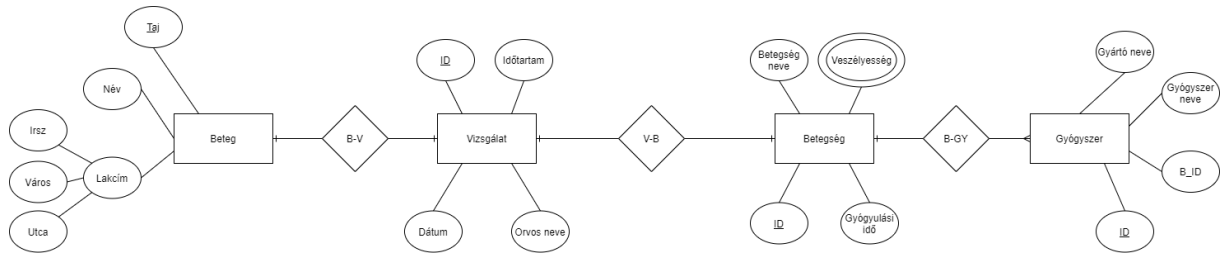
## **A feladat leírása:**

A feladatban egy rendelőintézet rendeléseihez kapcsolódó adatbázis elkészítése. Ehhez használunk 4 egyedet, melyeknél meg kell adni a következőket:

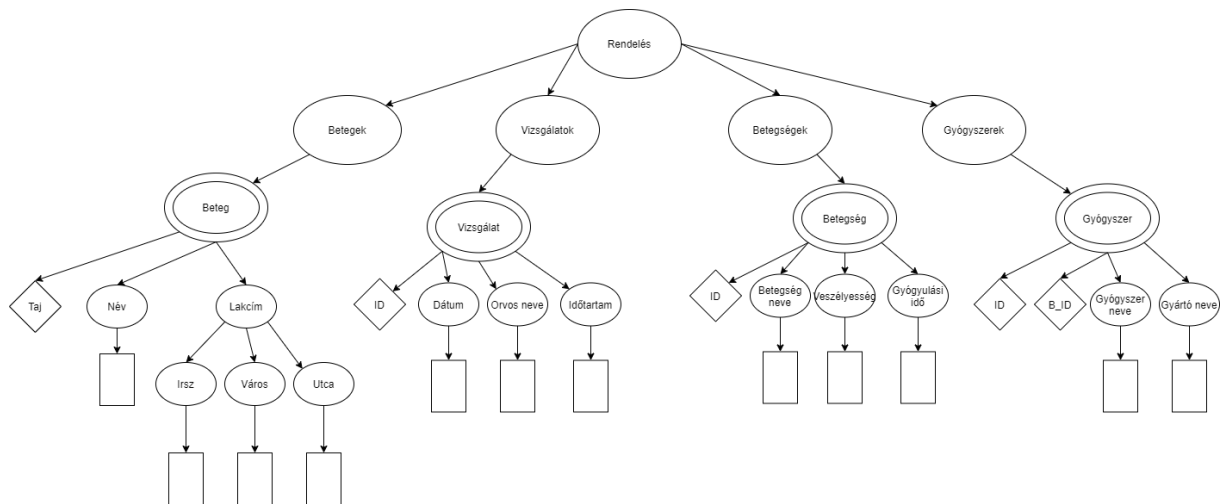
1. Beteg
  - a. Taj (Mint id)
  - b. Név
  - c. Lakcím
    - i. Irsz
    - ii. Város
    - iii. Utca
2. Vizsgálat
  - a. ID
  - b. Dátum
  - c. Orvos neve
  - d. Időtartam
3. Betegség
  - a. ID
  - b. Gyógyulási idő (Hétben)
  - c. Betegség neve
  - d. Veszélyesség (származtatott)
4. Gyógyszer
  - a.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

## 1 Feladat:

### 1a.) ER model:



### 1b.) XDM model:



### 1c.) XML dokumentum:

```
<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:sequence>
              <xs:element type="xs:string" name="nev"/>
              <xs:element name="lakcim">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:integer"
name="irsz"/>
                    <xs:element type="xs:string"
name="varos"/>
                    <xs:element type="xs:string"
name="utca"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:byte" name="taj"
use="required"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="vizsgalat">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="id"/>
              <xs:element type="xs:string" name="datum"/>
              <xs:element type="xs:string" name="orvos_neve"/>
              <xs:element type="xs:string" name="ido_tartam"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="betegseg">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="id"/>
              <xs:element type="xs:string" name="betegseg_neve"/>
              <xs:element type="xs:string" name="veszelyesség"/>
              <xs:element type="xs:string" name="gyogyulasi_ido"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="gyogyszer">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="id"/>
              <xs:element type="xs:string" name="b_id"/>
              <xs:element type="xs:string" name="gyogyszer_neve"/>
              <xs:element type="xs:string" name="gyarto_neve"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:complexType>
    </xs:element>

    <xs:element name="vizsgalat" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
            <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:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
        </xs:complexType>
    </xs:element>

    <xs:element name="betegseg" maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
            <xs:sequence>
                <xs:element type="xs:string"
name="betegseg_neve"></xs:element>
                <xs:element type="Veszelyesseg_tipus" name="veszelyesseg" />
                <xs:element type="xs:byte"
name="gyogyszer_ido"></xs:element>
            </xs:sequence>
            <xs:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
        </xs:complexType>
    </xs:element>

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

<xs:simpleType name="Veszelyesseg_tipus" final="restriction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Gyenge_lefolyasu" />
        <xs:enumeration value="Normal_lefolyasu" />
        <xs:enumeration value="Eros_lefolyasu" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

## 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:sequence>
                        <xs:element type="xs:string"
name="nev"></xs:element>
                        <xs:element name="lakcim">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element type="xs:integer"
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:attribute type="xs:byte" name="taj"
use="required"></xs:attribute>
                </xs:complexType>
            </xs:element>

            <xs:element name="vizsgalat" maxOccurs="unbounded"
minOccurs="0">
                <xs:complexType>
                    <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:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
                </xs:complexType>
            </xs:element>

            <xs:element name="betegseg" maxOccurs="unbounded"
minOccurs="0">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:string"
name="betegseg_neve"></xs:element>
                        <xs:element type="Veszelyesseg_tipus"
name="veszelyesseg" />
                        <xs:element type="xs:byte"
name="gyogylasi_ido"></xs:element>
                    </xs:sequence>
                    <xs:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
                </xs:complexType>
            </xs:element>

            <xs:element name="gyogyszer" maxOccurs="unbounded"
minOccurs="0">

```

```

        <xs:complexType>
            <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:attribute type="xs:byte" name="id"
use="required"></xs:attribute>
            <xs:attribute type="xs:byte" name="b_id"
use="required"></xs:attribute>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:simpleType name="Veszelyesseg_tipus" final="restriction">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Gyenge_lefolyasu" />
        <xs:enumeration value="Normal_lefolyasu" />
        <xs:enumeration value="Eros_lefolyasu" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

## 2 Feladat

### 2a.)

```
package hu.domparse.qohycr;

import javax.xml.parsers.DocumentBuilder;

public class DOMReadQOHYCR {

    private static void printNode(Node n) {
        //Végig megyünk az elemeken
        NodeList nl = n.getChildNodes();
        for (int i = 0; i < nl.getLength(); i++) {
            if(nl.item(i).getNodeName() != "#text") {
                //Gyerek elemeket is kilistázzuk
                Node no = nl.item(i);
                System.out.println(no.getNodeName());
                //Ha még a gyereknek is van gvereke abba is bele lépünk
                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();
            //XML dokumentum beolvasása
            Document document = db.parse("D:\\Project\\Eclipse\\DOMParseQOHYCR\\XMLqohycr.xml");
            Element root = document.getDocumentElement();
            //DOM fa kiírása
            printNode(root);
            //DOM fa módosítása
            DOMModifyQOHYCR.modifyDuration(root);
            //DOM fa kiírása változtatás után
            printNode(root);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

2b.)

```
package hu.domparse.qohycr;

import org.w3c.dom.Node;

public class DOMModifyQOHYCR {
    public static void modifyDuration(Node root) {
        try {
            System.out.println(root.getNodeType());
            //Lekérjük a node listát
            NodeList nl = root.getChildNodes();
            for(int i = 0; i < nl.getLength(); i++){
                //Megnézzük, hogy a vizsgalat node-nál vagyunk-e
                if (nl.item(i).getNodeName() == "vizsgalat") {
                    for(int k = 0; k < nl.item(i).getChildNodes().getLength(); k++) {
                        //A vizsgalat node idotartam ertekeket változtatjuk
                        if(nl.item(i).getChildNodes().item(k).getNodeName() == "idotartam") {
                            int duration = Integer.parseInt(nl.item(i).getChildNodes().item(k).getTextContent());
                            nl.item(i).getChildNodes().item(k).setTextContent(String.valueOf(duration + 15));
                        }
                    }
                }
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```