

Jegyzőkönyv

Egészségügyi nyilvántartó rendszer

Adatkezelés XML környezetben

féléves feladat

A feladat leírása:

Ez egy olyan project adatbázismodellje, aminek célja az orvosi adminisztráció könnyítése. A megcélzott probléma az egészségügyben lévő kihasználatlanságok, mint például a hosszas várakozás, a beteg transfer, az orvosok leterheltsége, vagy az elavult időpont foglalás és sorszám vétel. A beteg tud például időpontot foglalni, látja mikor rendelnek az orvosok, diagnosztizálhatja a betegségét mesterséges intelligencia segítségével. Az orvos megnézheti a beteggel kapcsolatos fontos információkat, például allergiák, előző vagy tartós betegségeit.

Páciens: A PK-e egy páciens_id. A betegek nevét, születési idejét születési helyét, lakcímét: város, utca és házszám, irányítószám, valamint TAJ számát tároljuk, valamint a regisztrációs adatokat: Email cím és jelszó.

P - B: Kapcsolótábla a Páciens és a Betegség tábla között. A két tábla PK-ére mutató FK van benne valamint egy saját PK id és egy időtartam amiből meg lehet állapítani, hogy "aktív" vagy múltbeli betegségről van-e szó. Ezt onnan lehet tudni, hogy ha még nincs "lezáró" dátum írva akkor aktív.

Betegség: Van PK-e: betegség_id. Ezen kívül a betegség nevét és egy leírást tárolunk róla.

P - GY: Kapcsolótábla a Páciens és a Gyógyszer tábla között. Ennek is van id-je (PK) és a két táblára mutató FK-ei. Valamint az adott gyógyszer felírásának dátuma.

Gyógyszer: Gyógyszer_id az elsődleges kulcsa. A gyógyszerek neveit, árakat és szintén a leírásukat tároljuk.

Időpont foglalás: A Páciens és az Orvos összekötő tábla ami a foglalt időpont adatait tárolja egy táblában. Ennek is van id-ja PK-ként (időpont_id). 2 dátum mező is van benne, egyik a dátum ami év/hónap/nap formában mondja meg mikorra szól az időpont, a másik pedig a pontos idő ami óra:perc-re pontosan rögzíti. Van még időtartam ami percben mondja meg kb mennyit lesz bent az orvosnál a páciens. Ezen kívül az összekötést biztosító két FK.

Orvos: PK: orvos_id. Az orvos nevén kívül azt tároljuk még, hogy az adott pillanatban elérhető-e. + Regisztrációs adatok

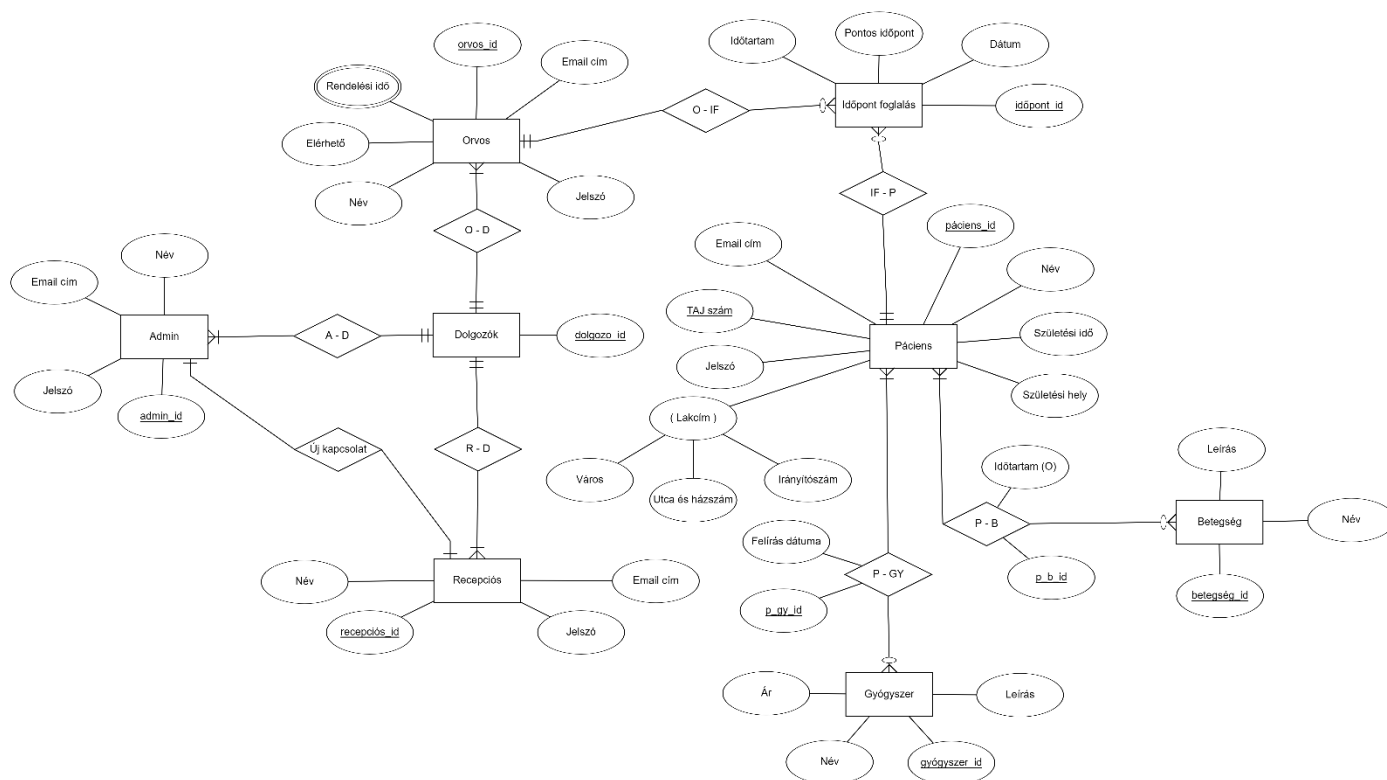
Orvos rendelési idő: Külön táblában van tárolva az orvosok rendelési ideje.

Dolgozók: Csak egy kapcsolótábla, benne 3 FK: orvos_id, recepció_id, admin_id és a saját id-ja.

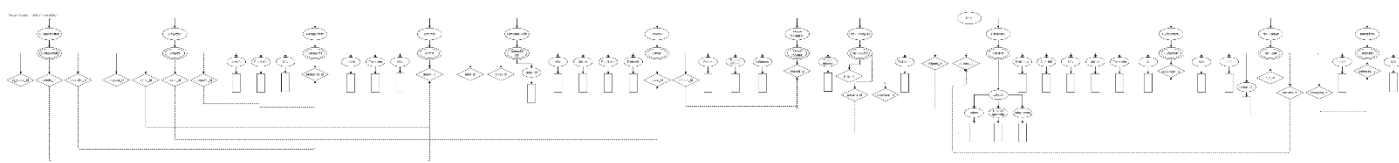
Recepció: Csak a nevét tároljuk és a saját id-jét (PK). + Regisztrációs adatok

Admin: lásd Recepció. + Regisztrációs adatok

Az ER modell:



Az XDM modell:



Az XML dokumentum:

```

<?xml version="1.0" encoding="utf-8"?>

<?xml-model href="XMLSchemaF6MYAZ.xsd"?>
<root>
  <dolgozok>
    <dolgozo admin_id="a1" dolgozo_id="d1"></dolgozo>
    <dolgozo orvos_id="o1" dolgozo_id="d2"></dolgozo>
    <dolgozo recepcios_id="rec1" dolgozo_id="d3"></dolgozo>
    <dolgozo recepcios_id="rec2" dolgozo_id="d4"></dolgozo>
    <dolgozo recepcios_id="rec3" dolgozo_id="d5"></dolgozo>
  </dolgozok>
  <recepciosok>
    <recepcios recepcios_id="rec1">
      <nev>Soós Péter</nev>
      <email_cim>soospeter@hospital.hu</email_cim>
      <jelszo>ot4hvpih35</jelszo>
    </recepcios>
    <recepcios recepcios_id="rec2">

```

```

        <nev>Vincze Erik</nev>
        <email_cim>vinczeerik@hospital.hu</email_cim>
        <jelszo>rj56uh645w</jelszo>
    </receprios>
    <receprios recepcios_id="rec3">
        <nev>Kis Kitti</nev>
        <email_cim>kiskitti@hospital.hu</email_cim>
        <jelszo>5nbe5junwzqvaere68</jelszo>
    </receprios>
</recepriosok>
<adminok>
    <admin admin_id="a1">
        <nev>Bodnár Gábor</nev>
        <email_cim>bodnargabor@hospital.hu</email_cim>
        <jelszo>i5vuhngioq5vhg</jelszo>
    </admin>
</adminok>
<rendeleesi_idok>
    <rendeleesi_ido rend_id="ren1" orvos_id="o1">
        <rendeleesi_ido>8:00-14:00</rendeleesi_ido>
    </rendeleesi_ido>
    <rendeleesi_ido rend_id="ren2" orvos_id="o1">
        <rendeleesi_ido>9:00-17:00</rendeleesi_ido>
    </rendeleesi_ido>
</rendeleesi_idok>
<orvosok>
    <orvos orvos_id="o1" idopont_id="i1">
        <nev>Major Júlia</nev>
        <email_cim>majorjulia@hospital.hu</email_cim>
        <jelszo>jkáádkiopbh49</jelszo>
        <elerheto>true</elerheto>
    </orvos>
</orvosok>
<idopont_foglalasok>
    <idopont_foglalas idopont_id="i1">
        <datum>2021-12-01</datum>
        <pontos_idopont>14:00</pontos_idopont>
        <idotartam>30</idotartam>
    </idopont_foglalas>
    <idopont_foglalas idopont_id="i2">
        <datum>2021-12-06</datum>
        <pontos_idopont>08:00</pontos_idopont>
        <idotartam>60</idotartam>
    </idopont_foglalas>
    <idopont_foglalas idopont_id="i3">
        <datum>2021-12-18</datum>
        <pontos_idopont>12:00</pontos_idopont>
        <idotartam>15</idotartam>
    </idopont_foglalas>
</idopont_foglalasok>
<pac-gyogy-ek>
    <pac-gyogy p_gy_id="p-gy1">

```

```

        <feliras_datuma>2021-11-14</feliras_datuma>
    </pac-gyogy>
</pac-gyogy-ek>
<paciensek>
    <paciens taj_szam="532689741" paciens_id="p1" idopont_id="i1">
        <nev>Balázs Dorottya</nev>
        <email_cim>balazsdorottya@gmail.com</email_cim>
        <jelszo>892bz549o2t67gg8gwr</jelszo>
        <szuletesi_ido>1986-01-08</szuletesi_ido>
        <szuletesi_hely>Budapest</szuletesi_hely>
        <lakcim>
            <varos>Budapest</varos>
            <utca_es_hazszam>Erzsébet krt. 11.</utca_es_hazszam>
            <iranyitoszam>1073</iranyitoszam>
        </lakcim>
    </paciens>
    <paciens taj_szam="112334456" paciens_id="p2" idopont_id="i2">
        <nev>Biró Benedek</nev>
        <email_cim>birobenedek@gmail.com</email_cim>
        <jelszo>hbsth</jelszo>
        <szuletesi_ido>1999-12-28</szuletesi_ido>
        <szuletesi_hely>Miskolc</szuletesi_hely>
        <lakcim>
            <varos>Miskolc</varos>
            <utca_es_hazszam>Arany János utca 28.</utca_es_hazszam>
            <iranyitoszam>3530</iranyitoszam>
        </lakcim>
    </paciens>
    <paciens taj_szam="998776554" paciens_id="p3" idopont_id="i3">
        <nev>Hajdú Gyula</nev>
        <email_cim>hajdugyula@gmail.com</email_cim>
        <jelszo>wgrg</jelszo>
        <szuletesi_ido>1970-08-11</szuletesi_ido>
        <szuletesi_hely>Nyíregyháza</szuletesi_hely>
        <lakcim>
            <varos>Tiszaújváros</varos>
            <utca_es_hazszam>Szent István út 47.</utca_es_hazszam>
            <iranyitoszam>3550</iranyitoszam>
        </lakcim>
    </paciens>
</paciensek>
<gyogyszerek>
    <gyogyszer gyogyszer_id="gy1">
        <ar>16000</ar>
        <nev>Nasivin</nev>
        <leiras>Orrcsepp</leiras>
    </gyogyszer>
</gyogyszerek>
<pac-bet-ek>
    <pac-bet p_b_id="pb1">
        <idotartam>3 nap</idotartam>
    </pac-bet>

```

```

</pac-bet-ek>
<betegsegek>
  <betegseg betegseg_id="b1">
    <nev>Megfázás</nev>
    <leiras>Megfázás leírása</leiras>
  </betegseg>
</betegsegek>
</root>

```

Az XML Schema:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="dolgozok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="dolgozo" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="admin_id" />
                  <xs:attribute name="orvos_id" />
                  <xs:attribute name="recepacios_id" />
                  <xs:attribute name="dolgozo_id" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="recepaciosok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="recepacios" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="nev" type="xs:string" />
                    <xs:element name="email_cim" type="xs:string" />
                    <xs:element name="jelszo" type="xs:string" />
                  </xs:sequence>
                  <xs:attribute name="recepacios_id" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="adminok">
          <xs:complexType>

```

```

        <xs:sequence>
            <xs:element name="admin" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="nev" type="xs:string" />
                        <xs:element name="email_cim" type="xs:string" />
                        <xs:element name="jelszo" type="xs:string" />
                    </xs:sequence>
                    <xs:attribute name="admin_id" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="rendelési_idok">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="rendelési_ido" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="rendelési_ido" type="xs:string"
/>
                    </xs:sequence>
                    <xs:attribute name="rend_id" use="required" />
                    <xs:attribute name="orvos_id" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="orvosok">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="orvos" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="nev" type="xs:string" />
                        <xs:element name="email_cim" type="xs:string" />
                        <xs:element name="jelszo" type="xs:string" />
                        <xs:element name="elerheto" type="xs:boolean" />
                    </xs:sequence>
                    <xs:attribute name="orvos_id" use="required" />
                    <xs:attribute name="idopont_id" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="idopont_foglalasok">
    <xs:complexType>
        <xs:sequence>

```

```

        <xs:element name="idopont_foglalas" minOccurs="0"
maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="datum" type="xs:date" />
                    <xs:element name="pontos_idopont" type="xs:string"
/>
                    <xs:element name="idotartam" type="xs:decimal" />
                </xs:sequence>
                <xs:attribute name="idopont_id" use="required" />
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="pac-gyogy-ek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="pac-gyogy" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="feliras_datuma" type="xs:date"
/>
                        </xs:sequence>
                        <xs:attribute name="p_gy_id" use="required" />
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
<xs:element name="paciensek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="paciens" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="nev" type="xs:string" />
                        <xs:element name="email_cim" type="xs:string" />
                        <xs:element name="jelszo" type="xs:string" />
                        <xs:element name="szuletesi_ido" type="xs:date" />
                        <xs:element name="szuletesi_hely" type="xs:string"
/>
                        <xs:element name="lakcim">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="varos"
type="xs:string" />
                                    <xs:element name="utca_es_hazszam"
type="xs:string" />
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```



```

type="iranyitoszam" />
        <xs:element name="iranyitoszam"
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:sequence>
<xs:attribute name="taj_szam" type="xs:string"
use="required"></xs:attribute>
    <xs:attribute name="paciens_id" use="required" />
    <xs:attribute name="idopont_id" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="gyogyszerek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="gyogyszer" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="ar" type="xs:decimal" />
                        <xs:element name="nev" type="xs:string" />
                        <xs:element name="leiras" type="xs:string" />
                    </xs:sequence>
                    <xs:attribute name="gyogyszer_id" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="pac-bet-ek">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="pac-bet" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="idotartam"
type="xs:string" />
                    </xs:sequence>
                    <xs:attribute name="p_b_id" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="betegsegek">
    <xs:complexType>
        <xs:sequence>

```

```

        <xs:element name="betegseg" minOccurs="0"
maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="nev" type="xs:string" />
                    <xs:element name="leiras" type="xs:string" />
                </xs:sequence>
                <xs:attribute name="betegseg_id" use="required" />
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="PK_dolgozo">
    <xs:selector xpath="dolgozok/dolgozo"></xs:selector>
    <xs:field xpath="@dolgozo_id"></xs:field>
</xs:key>
<xs:key name="PK_recepcios">
    <xs:selector xpath="recepciosok/recepcios"></xs:selector>
    <xs:field xpath="@recepcios_id"></xs:field>
</xs:key>
<xs:key name="PK_admin">
    <xs:selector xpath="adminok/admin"></xs:selector>
    <xs:field xpath="@admin_id"></xs:field>
</xs:key>
<xs:key name="PK_rendelesi_ido">
    <xs:selector xpath="rendelesi_idok/rendelesi_ido"></xs:selector>
    <xs:field xpath="@rend_id"></xs:field>
</xs:key>
<xs:key name="PK_orvos">
    <xs:selector xpath="orvosok/orvos"></xs:selector>
    <xs:field xpath="@orvos_id"></xs:field>
</xs:key>
<xs:key name="PK_idopont_foglalas">
    <xs:selector xpath="idopont_foglalasok/idopont_foglalas"></xs:selector>
    <xs:field xpath="@idopont_id"></xs:field>
</xs:key>
<xs:key name="PK_pac-gyogy">
    <xs:selector xpath="pac-gyogy-ek/pac-gyogy"></xs:selector>
    <xs:field xpath="@p_gy_id"></xs:field>
</xs:key>
<xs:key name="PK_paciens">
    <xs:selector xpath="paciensek/paciens"></xs:selector>
    <xs:field xpath="@paciens_id"></xs:field>
</xs:key>
<xs:key name="PK_gyogyszer">
    <xs:selector xpath="gyogyszerek/gyogyszer"></xs:selector>
    <xs:field xpath="@gyogyszer_id"></xs:field>
</xs:key>
<xs:key name="PK_pac-bet">

```

```

        <xs:selector xpath="pac-bet-ek/pac-bet"></xs:selector>
        <xs:field xpath="@p_b_id"></xs:field>
    </xs:key>
    <xs:key name="PK_betegseg">
        <xs:selector xpath="betegsegek/betegseg"></xs:selector>
        <xs:field xpath="@betegseg_id"></xs:field>
    </xs:key>
    <xs:keyref name="FK_dolgozo-admin" refer="PK_admin">
        <xs:selector xpath="dolgozok/dolgozo"></xs:selector>
        <xs:field xpath="@admin_id"></xs:field>
    </xs:keyref>
    <xs:keyref name="FK_dolgozo-orvos" refer="PK_orvos">
        <xs:selector xpath="dolgozok/dolgozo"></xs:selector>
        <xs:field xpath="@orvos_id"></xs:field>
    </xs:keyref>
    <xs:keyref name="FK_dolgozo-recepcios" refer="PK_recepcios">
        <xs:selector xpath="dolgozok/dolgozo"></xs:selector>
        <xs:field xpath="@recepcios_id"></xs:field>
    </xs:keyref>
    <xs:keyref name="FK_rend_ido-orvos" refer="PK_orvos">
        <xs:selector xpath="rendeleesi_idok/rendeleesi_ido"></xs:selector>
        <xs:field xpath="@orvos_id"></xs:field>
    </xs:keyref>
    <xs:keyref name="FK_orvos-idopont" refer="PK_idopont_foglalas">
        <xs:selector xpath="orvosok/orvos"></xs:selector>
        <xs:field xpath="@idopont_id"></xs:field>
    </xs:keyref>
    <xs:keyref name="FK_paciens-idopont" refer="PK_idopont_foglalas">
        <xs:selector xpath="paciensek/paciens"></xs:selector>
        <xs:field xpath="@idopont_id"></xs:field>
    </xs:keyref>
</xs:element>
<xs:simpleType name="iranyitoszam">
    <xs:restriction base="xs:positiveInteger">
        <xs:minInclusive value="1000"></xs:minInclusive>
        <xs:maxInclusive value="9999"></xs:maxInclusive>
    </xs:restriction>
</xs:simpleType>
</xs:schema>

```

DOM Read:

Kírja konzolra az egész XML dokumentumot, felsorolva.

```

package XMLTaskF6MYAZ.DOMParseF6MYAZ.src.hu.domparse.F6MYAZ;

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

import javax.xml.parsers.DocumentBuilder;

```

```

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

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 DOMReadF6MYAZ {
    public static void main(String arg[])
        throws SAXException, IOException, ParserConfigurationException,
XPathExpressionException {
        // Get file
        File xmlFile = new File("XMLTaskF6MYAZ/XMLF6MYAZ.xml");

        // Get DocumentBuilder
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        // Parse document and normalize
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        // Build XPath tree
        XPath xPath = XPathFactory.newInstance().newXPath();

        // Get all
        String expression = "//*";

        // NodeList declaration
        NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);

        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);

            if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("dolgozo")) {
                Element element = (Element) node;

                System.out.println("Admin ID: " + element.getAttribute("admin_id"));
                System.out.println("Orvos ID: " + element.getAttribute("orvos_id"));
                System.out.println("Recepció ID: " +
element.getAttribute("receprios_id"));
                System.out.println("Dolgozó ID: " + element.getAttribute("dolgozo_id"));
            }
        }
    }
}

```

```

        if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("receprios")) {
            Element element = (Element) node;

            System.out.println("ID: " + element.getAttribute("receprios_id"));
            System.out.println("Név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("Email: " +
element.getElementsByTagName("email_cim").item(0).getTextContent());
            System.out.println("Jelszó: " +
element.getElementsByTagName("jelszo").item(0).getTextContent());
        }

        if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("admin")) {
            Element element = (Element) node;

            System.out.println("ID: " + element.getAttribute("admin_id"));
            System.out.println("Név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("Email: " +
element.getElementsByTagName("email_cim").item(0).getTextContent());
            System.out.println("Jelszó: " +
element.getElementsByTagName("jelszo").item(0).getTextContent());
        }

        if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("orvos")) {
            Element element = (Element) node;

            System.out.println("Orvos ID: " + element.getAttribute("orvos_id"));
            System.out.println("Időpont ID: " + element.getAttribute("idopont_id"));
            System.out.println("Név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
            System.out.println("Email: " +
element.getElementsByTagName("email_cim").item(0).getTextContent());
            System.out.println("Jelszó: " +
element.getElementsByTagName("jelszo").item(0).getTextContent());
            System.out.println("Elérhető: " +
element.getElementsByTagName("elerheto").item(0).getTextContent());
        }

        if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("idopont_foglalas")) {
            Element element = (Element) node;

            System.out.println("ID: " + element.getAttribute("idopont_id"));
            System.out.println("Dátum: " +
element.getElementsByTagName("datum").item(0).getTextContent());
            System.out.println(
                "Pontos időpont: " +
element.getElementsByTagName("pontos_idopont").item(0).getTextContent());

```

```

        System.out.println("Időtartam: " +
element.getElementsByTagName("idotartam").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE && node.getNodeName().equals("pac-
gyogy")) {
        Element element = (Element) node;

        System.out.println("ID: " + element.getAttribute("p_gy_id"));
        System.out.println(
            "Felírás dátuma: " +
element.getElementsByTagName("feliras_datuma").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("paciens")) {
        Element element = (Element) node;

        System.out.println("Tajszám: " + element.getAttribute("taj_szam"));
        System.out.println("Páciens ID: " + element.getAttribute("paciens_id"));
        System.out.println("Időpont ID: " + element.getAttribute("idopont_id"));
        System.out.println("Név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Email: " +
element.getElementsByTagName("email_cim").item(0).getTextContent());
        System.out.println("Jelszó: " +
element.getElementsByTagName("jelszo").item(0).getTextContent());
        System.out.println(
            "Születési Idő: " +
element.getElementsByTagName("szuletesi_ido").item(0).getTextContent());
        System.out.println(
            "Születési hely: " +
element.getElementsByTagName("szuletesi_hely").item(0).getTextContent());
        System.out.println(
            "Születési hely: " +
element.getElementsByTagName("szuletesi_ido").item(0).getTextContent());
        System.out.println("Lakcím:");
        System.out.println("Város: " +
element.getElementsByTagName("varos").item(0).getTextContent());
        System.out
            .println("Város: " +
element.getElementsByTagName("utca_es_hazszam").item(0).getTextContent());
        System.out.println("Város: " +
element.getElementsByTagName("iranyitoszam").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("gyogyszer")) {
        Element element = (Element) node;

        System.out.println("ID: " + element.getAttribute("gyogyszer_id"));
    }

```

```

        System.out.println("Ár: " +
element.getElementsByTagName("ar").item(0).getTextContent());
        System.out.println("Név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Leírás: " +
element.getElementsByTagName("leiras").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE && node.getNodeName().equals("pac-
bet")) {
        Element element = (Element) node;

        System.out.println("ID: " + element.getAttribute("p_b_id"));
        System.out.println("Időtartam: " +
element.getElementsByTagName("idotartam").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("betegseg")) {
        Element element = (Element) node;

        System.out.println("ID: " + element.getAttribute("betegseg_id"));
        System.out.println("Név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
        System.out.println("Leírás: " +
element.getElementsByTagName("leiras").item(0).getTextContent());
    }
}
}
}
}

```

DOM Query:

Kiírja a konzolra az XML dokumentumból azokat az elemeket, ahol a páciens egyed 1980 után született.

```

package XMLTaskF6MYAZ.DOMParseF6MYAZ.src.hu.domparsed.F6MYAZ;

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

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 DOMQueryF6MYAZ {
    public static void main(String arg[])
        throws SAXException, IOException, ParserConfigurationException,
XPathExpressionException {
        // Get file
        File xmlFile = new File("XMLTaskF6MYAZ/XMLF6MYAZ.xml");

        // Get DocumentBuilder
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        // Parse document and normalize
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        // Build XPath tree
        XPath xPath = XPathFactory.newInstance().newXPath();

        // Get all páciens born after 1980
        String expression = "//paciens[number(translate(születesi_ido, '-', '')) >
19800101]";

        // NodeList declaration
        NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);

        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);

            if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("paciens")) {
                Element element = (Element) node;

                System.out.println("Tajszám: " + element.getAttribute("taj_szam"));
                System.out.println("Páciens ID: " + element.getAttribute("paciens_id"));
                System.out.println("Időpont ID: " + element.getAttribute("idopont_id"));
                System.out.println("Név: " +
element.getElementsByTagName("nev").item(0).getTextContent());
                System.out.println("Email: " +
element.getElementsByTagName("email_cim").item(0).getTextContent());
                System.out.println("Jelszó: " +
element.getElementsByTagName("jelszo").item(0).getTextContent());
                System.out.println(
                    "Születési Idő: " +
element.getElementsByTagName("szuletesi_ido").item(0).getTextContent());
                System.out.println(
                    "Születési hely: " +
element.getElementsByTagName("szuletesi_hely").item(0).getTextContent());

```



```

        System.out.println(
            "Születési hely: " +
element.getElementsByTagName("szuletesi_ido").item(0).getTextContent());
        System.out.println("Lakcím:");
        System.out.println("Város: " +
element.getElementsByTagName("varos").item(0).getTextContent());
        System.out
            .println("Város: " +
element.getElementsByTagName("utca_es_hazszam").item(0).getTextContent());
        System.out.println("Város: " +
element.getElementsByTagName("iranyitoszam").item(0).getTextContent());
    }
}
}
}
}

```

DOM Modify:

Megkeresi az XML dokumentumban az „o1” ID-ve rendelkező egyedet és módosítja annak nevét „Új név”-re, úgy hogy egy új dokumentumot hoz létre, valamint kiírja ezt a konzolra is.

```
package XMLTaskF6MYAZ.DOMParseF6MYAZ.src.hu.domparse.F6MYAZ;
```

```

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.OutputKeys;
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 javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

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 DOMModifyF6MYAZ {
    public static void main(String arg[])

```

```

        throws SAXException, IOException, ParserConfigurationException,
XPathExpressionException,
        TransformerException {
    // Get file
    File inputXMLFile = new File("XMLTaskF6MYAZ/XMLF6MYAZ.xml");
    File outputXMLFile = new File("XMLTaskF6MYAZ/XMLF6MYAZ_output.xml");

    // Get DocumentBuilder
    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = factory.newDocumentBuilder();

    // Parse document and normalize
    Document doc = dBuilder.parse(inputXMLFile);
    doc.getDocumentElement().normalize();

    // Build XPath tree
    XPath xPath = XPathFactory.newInstance().newXPath();

    // Get all orvos
    String expression = "//orvos";

    // NodeList declaration
    NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(doc,
XPathConstants.NODESET);

    for (int i = 0; i < nodeList.getLength(); i++) {
        if (nodeList.item(i).getNodeType() == Node.ELEMENT_NODE) {
            Element element = (Element) nodeList.item(i);
            if (element.getAttribute("orvos_id").equals("o1")) {
                element.getElementsByTagName("nev").item(0).setTextContent("Új név");
            }
        }
    }

    // Write to file and console
    modify(doc, outputXMLFile);
}

private static void modify(Document doc, File outputFile)
    throws TransformerException, UnsupportedEncodingException {
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transf = transformerFactory.newTransformer();
    transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
    transf.setOutputProperty(OutputKeys.INDENT, "yes");
    transf.setOutputProperty("{http://xml.apache.org/xslt}indent-amunt", "2");

    DOMSource source = new DOMSource(doc);

    StreamResult console = new StreamResult(System.out);
    StreamResult file = new StreamResult(outputFile);

```

```
    transf.transform(source, console);  
    transf.transform(source, file);  
  }  
}
```