

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Könyvtári nyilvántartás létrehozása

Készítette: **Firtkó Bence**

Neptunkód: **TP0M8Y**

Dátum: **2022.11.29**

### **A feladat leírása:**

A feladat egy hiteles adatbázis létrehozása, különböző lekérdezések kialakítása XML nyelven keresztül. A beadandó feladat témája a profi angol bajnokságokban szereplőcsapatok és játékosok, valamint meccseiknek tárolására alkalmas adatbázis. Továbbá tartalmaz a mérkőzések helyszínéül szolgáló stadionokról információkat, valamint részletes játékos statisztikákat. Ezután az adatbázis szerkezetét leíró ER, valamint XDM modelljének kialakítása a következő lépés. Amint a megfelelő konvertálások megtörténnek a feladat folytatódik az xml fájl létrehozásával és feltöltésével. Az XML dokumentum feltöltése után egy XML Schema kell létre hozni.

### **Az ER modell egyedei és tulajdonságai:**

**Egyedek:** Csapat, Stadion, Merkozes, Labdarugo, Statisztika

#### **Egyedek tulajdonságai:**

##### **Csapat:**

- CsapatID: A csapat egyed elsődleges kulcsa
- Edzo: A csapat edzője
- CsNev: A csapat neve
- Liga: A bajnokság neve, ahol a csapat játszik, több értékű tulajdonság

##### **Stadion:**

- StadionID: A stadion egyed elsődleges kulcsa
- Nev: A stadion neve
- Cim: A stadion pontos címe
- Ferohely: A maximális nézőszám a stadionban

##### **Merkozes:**

- MerkozesID: A mérkőzés egyed elsődleges kulcsa
- HazaiGolk: A hazai csapat által szerzett gólok mennyisége
- VendegGolk: A vendégcsapat által szerzett gólok mennyisége
- Idopont: A mérkőzés időpontja

### **Labdarugo:**

- **LabdarugoID:** A labdarúgó egyed elsődleges kulcsa
- **LNev:** A játékos neve
- **Kor:** A labdarúgó kora
- **Poszt:** A labdarúgó posztja

### **Statisztika:**

- **StatisztikaID:** A statisztika egyed elsődleges kulcsa
- **JatekPerc:** A labdarúgó pályán töltött perceinek mennyisége
- **SargaLap:** A labdarúgó sárga lapjainak mennyisége
- **PirosLap:** A labdarúgó piros lapjainak mennyisége
- **Gol:** Góljainak száma
- **Golpassz:** Gólpasszainak száma

### **Egyedek közötti kapcsolat:**

A **Labdarúgó** és a **Statisztika** egyedek között 1:1 kapcsolat van, ugyanis

minden játékos rendelkezik a saját statisztikájával, a statisztikához pedig mindig egy adott játékos tartozik.

A **Labdarúgó** és a **Csapat** egyedek között 1:N kapcsolat van, ugyanis a

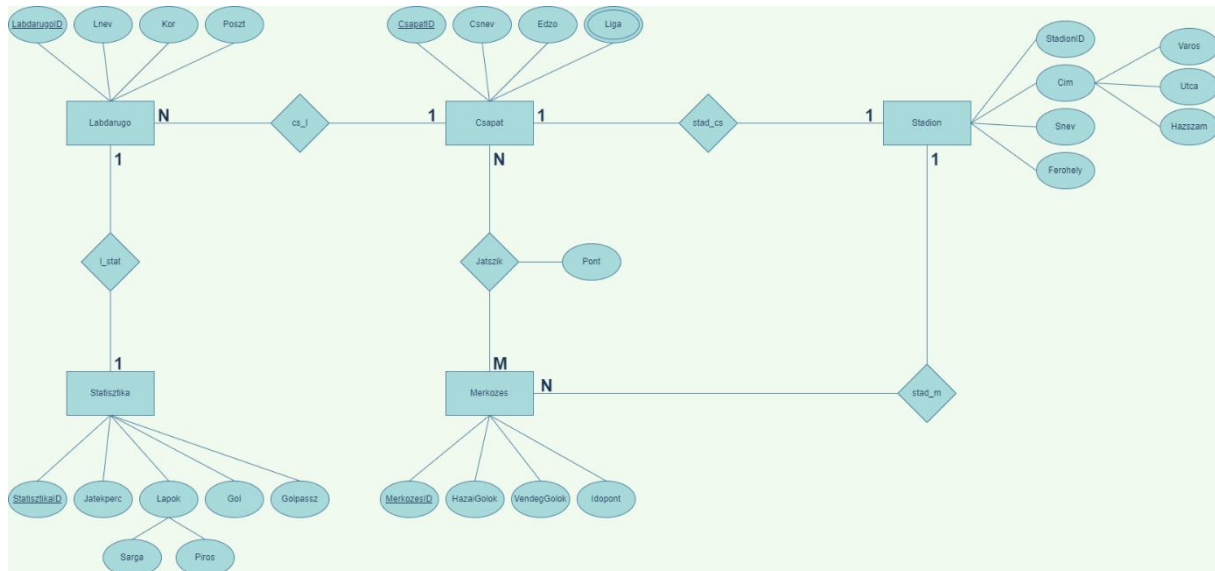
csapatok több játékosal rendelkeznek, de minden játékos egy csapathoz van leszerződve.

A **Csapat** és a **Stadion** egyedek között 1:1 kapcsolat van, mert minden csapat rendelkezik egy stadionnal, és minden stadionnak is pontosan egy csapata van, akinek ez a hazai pályája.

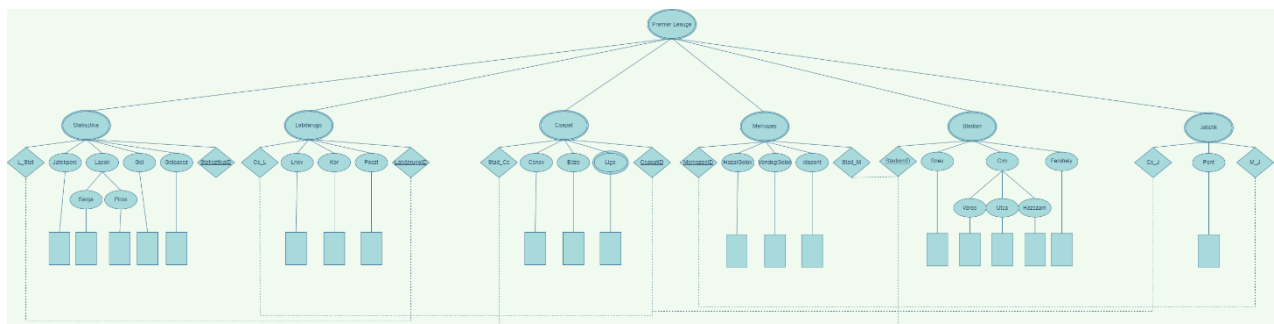
A **Stadion** és a **Mérkőzés** egyedek között 1:N kapcsolat van, mert egy mérkőzés nyilvánvalóan egy bizonyos stadionban játszódik, azonban

egy stadionban számos mérkőzés lejátszható.

A **Csapat** és a **Mérkőzés** egyedek között N:M kapcsolat van, mert egy mérkőzést több csapat játszik és egy csapat értelemszerűen sok mérkőzést játszik egy szezon alatt is.



Az XDM modellben a háromfajta jelölést használunk. Az ER modellben szereplő elemeket az XDM modellben ellipszissel, az elemek tulajdonságait, azaz az attribútumokat rombusszal, a szöveges tartalmakat pedig téglalappal jelöljük. A kulcstulajdonságok az XDM modellben is aláhúzásra kerülnek, valamint külön elemet kell létrehozni a több-több kapcsolat szemléltetésére.



## 1c) Az XDM modell alapján XML dokumentum készítése:

Az XDM modell alapján létrehoztam az XML dokumentumot.

Minden egyedhez előállítottam legalább három példát. Nagyjából helyes de nem a valóságot tükröző adatokat tartalmaz a dokumentum.

```
<?xml version="1.0" encoding="UTF-8"?>

<bajnoksag xmlns:xsi="">

  <!-- Statisztikák -->
  <statisztika statisztikaID="stat1" l_stat="l1">
    <jatekperc>300</jatekperc>
    <lapok>
      <sarga>4</sarga>
      <piros>0</piros>
    </lapok>
    <gol>4</gol>
    <golpassz>5</golpassz>
  </statisztika>

  <statisztika statisztikaID="stat2" l_stat="l2">
    <jatekperc>1000</jatekperc>
    <lapok>
      <sarga>2</sarga>
      <piros>0</piros>
    </lapok>
    <gol>10</gol>
    <golpassz>13</golpassz>
  </statisztika>

  <statisztika statisztikaID="stat3" l_stat="l3">
    <jatekperc>600</jatekperc>
    <lapok>
      <sarga>10</sarga>
      <piros>1</piros>
    </lapok>
    <gol>1</gol>
    <golpassz>0</golpassz>
  </statisztika>

  <!-- Labdarúgók -->
  <labdarugo labdarugoID="l1" cs_l="cs1">
    <lnev>James Maddison</lnev>
    <kor>22</kor>
    <poszt>CAM</poszt>
  </labdarugo>

  <labdarugo labdarugoID="l2" cs_l="cs2">
```

```
<lnev>Gabriel Jesus</lnev>
<kor>24</kor>
<poszt>ST</poszt>
</labdarugo>

<labdarugo labdarugoID="13" cs_l="cs3">
  <lnev>Ruben Dias</lnev>
  <kor>26</kor>
  <poszt>CB</poszt>
</labdarugo>

<!-- Csapatok -->
<csapat csapatID="cs1" stad_cs="s1">
  <csnev>Leicester City</csnev>
  <edzo>Jose Mourinho</edzo>
  <liga>Premier Leauge</liga>
  <liga>Europe Leauge</liga>
</csapat>

<csapat csapatID="cs2" stad_cs="s2">
  <csnev>Arsenal</csnev>
  <edzo>Mikel Arteta</edzo>
  <liga>Premier Leauge</liga>
  <liga>Europe Leauge</liga>
</csapat>

<csapat csapatID="cs3" stad_cs="s3">
  <csnev>Manchester City</csnev>
  <edzo>Pep Guardiola</edzo>
  <liga>Premier Leauge</liga>
  <liga>Champions Leauge</liga>
</csapat>

<!-- Mérkőzések -->
<merkozes merkozesID="m1" stad_m="s1">
  <hazaigolok>2</hazaigolok>
  <vendeggolok>0</vendeggolok>
  <idopont>2022/10/20 18:00</idopont>
</merkozes>

<merkozes merkozesID="m2" stad_m="s2">
  <hazaigolok>0</hazaigolok>
  <vendeggolok>0</vendeggolok>
  <idopont>2022/09/12 20:00</idopont>
</merkozes>

<merkozes merkozesID="m3" stad_m="s2">
  <hazaigolok>1</hazaigolok>
  <vendeggolok>2</vendeggolok>
```

```
<idopont>2022/09/30 21:00</idopont>  
</merkozes>
```

```
<!-- Stadionok -->
```

```
<stadion stadionID="s1">  
  <snev>King Power</snev>  
  <cim>  
    <varos>Leicester</varos>  
    <utca>Eper utca</utca>  
    <hazszam>12</hazszam>  
  </cim>  
  <ferohely>40000</ferohely>  
</stadion>
```

```
<stadion stadionID="s2">  
  <snev>Emirates Stadium</snev>  
  <cim>  
    <varos>London</varos>  
    <utca>Kakas utca</utca>  
    <hazszam>30</hazszam>  
  </cim>  
  <ferohely>80000</ferohely>  
</stadion>
```

```
<stadion stadionID="s3">  
  <snev>Etihad</snev>  
  <cim>  
    <varos>Manchester</varos>  
    <utca>Kossuth utca</utca>  
    <hazszam>16</hazszam>  
  </cim>  
  <ferohely>60000</ferohely>  
</stadion>
```

```
<!-- Mérkőzésen játszó csapatok -->
```

```
<jatszik cs_j="cs1" m_j="m1">  
  <pont>0</pont>  
</jatszik>
```

```
<jatszik cs_j="cs2" m_j="m1">  
  <pont>3</pont>  
</jatszik>
```

```
<jatszik cs_j="cs3" m_j="m2">  
  <pont>1</pont>  
</jatszik>
```

```
<jatszik cs_j="cs1" m_j="m2">  
  <pont>1</pont>
```

```

</jatszik>

<jatszik cs_j="cs2" m_j="m3">
  <pont>0</pont>
</jatszik>

<jatszik cs_j="cs3" m_j="m3">
  <pont>3</pont>
</jatszik>
</bajnoksag>

```

## 1d) Az XML dokumentum alapján XML Schema készítése:

```

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

  <!-- Saját típusok -->
  <xs:element name="kor">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="100"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>

  <xs:simpleType name="poszt">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CAM" />
      <xs:enumeration value="CB" />
      <xs:enumeration value="ST" />
      <xs:enumeration value="CDM" />
      <xs:enumeration value="GK" />
      <xs:enumeration value="CM" />
      <xs:enumeration value="RM" />
      <xs:enumeration value="LM" />
      <xs:enumeration value="RW" />
      <xs:enumeration value="LW" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="pont">
    <xs:restriction base="xs:integer">
      <xs:enumeration value="0" />
      <xs:enumeration value="1" />
      <xs:enumeration value="3" />
    </xs:restriction>
  </xs:simpleType>

```



```

        </xs:restriction>
    </xs:simpleType>

    <!-- Felépítés -->
    <xs:element name="bajnoksag">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="statisztika">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element type="xs:integer" name="jatekperc" />
                            <xs:element name="lapok">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element type="xs:integer"
name="sarga"/>
                                        <xs:element type="xs:integer"
name="piros"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element type="xs:integer" name="gol" />
                            <xs:element type="xs:integer" name="golpassz" />
                        </xs:sequence>
                        <xs:attribute type="xs:string" name="l_stat"
use="required" />
                        <xs:attribute type="xs:string" name="statisztikaID"
use="required" />
                    </xs:complexType>
                </xs:element>
                <xs:element name="labdarugo">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element type="xs:string" name="lnev" />
                            <xs:element type="kor" name="kor" />
                            <xs:element type="poszt" name="poszt" />
                        </xs:sequence>
                        <xs:attribute type="xs:string" name="cs_l"
use="required" />
                        <xs:attribute type="xs:string" name="labdarugoID"
use="required" />
                    </xs:complexType>
                </xs:element>
                <xs:element name="csapat">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element type="xs:string" name="csnev" />
                            <xs:element type="xs:string" name="edzo" />
                            <xs:element type="xs:string" name="liga" />

```

```

        </xs:sequence>
        <xs:attribute type="xs:string" name="stad_cs"
use="required" />
        <xs:attribute type="xs:string" name="csapatID"
use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="merkozés">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:integer" name="hazaigolok" />
            <xs:element type="xs:integer" name="vendeggolok"
/>
            <xs:element type="xs:date" name="idopont" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="stad_m"
use="required" />
        <xs:attribute type="xs:string" name="merkozésID"
use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="stadion">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:string" name="snev" />
            <xs:element name="cim">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:string"
name="varos"/>
                        <xs:element type="xs:string"
name="utca"/>
                        <xs:element type="xs:integer"
name="hazszam"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element type="xs:integer" name="ferohely" />
        </xs:sequence>
        <xs:attribute type="xs:string" name="stadionID"
use="required" />
    </xs:complexType>
</xs:element>
<xs:element name="jatszík">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="pont" name="pont" />
        </xs:sequence>

```

```

        <xs:attribute type="xs:string" name="cs_j"
use="required" />
        <xs:attribute type="xs:string" name="m_j"
use="required" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<!-- Kulcsok -->

<xs:key name="statisztika_kulcs">
    <xs:selector xpath="statisztika"/>
    <xs:field xpath="@statisztikaID" />
</xs:key>

<xs:key name="labdarugo_kulcs">
    <xs:selector xpath="labdarugo"/>
    <xs:field xpath="@labdarugoID" />
</xs:key>

<xs:key name="csapat_kulcs">
    <xs:selector xpath="csapat"/>
    <xs:field xpath="@csapatID" />
</xs:key>

<xs:key name="merkozes_kulcs">
    <xs:selector xpath="merkozes"/>
    <xs:field xpath="@merkozesID" />
</xs:key>

<xs:key name="stadion_kulcs">
    <xs:selector xpath="stadion"/>
    <xs:field xpath="@stadionID" />
</xs:key>

<!-- Idegen kulcsok -->

<xs:keyref refer="labdarugo_kulcs" name="labdarugo_idegen_kulcs">
    <xs:selector xpath="statisztika" />
    <xs:field xpath="@l_stat" />
</xs:keyref>

<xs:keyref refer="csapat_kulcs" name="csapat_idegen_kulcs">
    <xs:selector xpath="labdarugo" />
    <xs:field xpath="@cs_l" />
</xs:keyref>

```

```

        <xs:keyref refer = "stadion_kulcs" name
="stadion_csapat_idegen_kulcs">
            <xs:selector xpath ="csapat" />
            <xs:field xpath ="@stad_cs" />
        </xs:keyref>

        <xs:keyref refer = "stadion_kulcs" name
="stadion_csapat_idegen_kulcs">
            <xs:selector xpath ="merkozes" />
            <xs:field xpath ="@stad_m" />
        </xs:keyref>

        <xs:keyref refer = "csapat_kulcs" name ="csapat_jatszik_idegen_kulcs">
            <xs:selector xpath ="jatszik" />
            <xs:field xpath ="@cs_j" />
        </xs:keyref>

        <xs:keyref refer = "merkozes_kulcs" name
="merkozes_jatszik_idegen_kulcs">
            <xs:selector xpath ="jatszik" />
            <xs:field xpath ="@m_j" />
        </xs:keyref>
    </xs:element>

</xs:schema>

```

## 2a) adatolvasás és írás – DomReadTP0M8Y.java

Először a megfelelő importok használata, fájlok megnyitása, létrehozása, majd a builderek felépítése. Dokumentum, és a gyökérelem kezelése. Majd a megfelelő kimenet megalkotása és kiírása. XML fájlba való mentés.

A kód:

```

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.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.io.IOException;

public class DomReadTP0M8Y {
    public static void main(String[] args) throws SAXException,
        IOException, ParserConfigurationException, TransformerException

```

```

{
    //File megnyitás
    File xmlFile = new File("XMLTP0M8Y.xml");

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

    //Új és meglévő dokumnetum kibálasztás
    Document doc = dBuilder.parse(xmlFile);
    Document newdoc = dBuilder.newDocument();

    doc.getDocumentElement().normalize();

    //Gyökérelem megadás
    Element root = newdoc.createElementNS("XMLTP0M8Y", "bajnoksag");
    newdoc.appendChild(root);

    //Új XML elemeinek a feltöltése
    root.appendChild(createStatisztika(newdoc, "stat1", "300", "10",
"0", "4", "10", "11"));
    root.appendChild(createStatisztika(newdoc, "stat2", "200", "11",
"1", "0", "2", "12"));
    root.appendChild(createStatisztika(newdoc, "stat3", "400", "0", "0",
"0", "2", "13"));
    root.appendChild(createLabdarugo(newdoc, "11", "Callum Styles",
"25", "CM", "cs1"));
    root.appendChild(createLabdarugo(newdoc, "12", "Sallai Roland",
"25", "ST", "cs2"));
    root.appendChild(createLabdarugo(newdoc, "13", "Szoboszlai Dominik",
"21", "CAM", "cs3"));
    root.appendChild(createCsapat(newdoc, "cs1", "Millwall", "Kiss
Pista", new String[]{"Premier Leauge", "Europe Leauge"}, "s1"));
    root.appendChild(createCsapat(newdoc, "cs2", "Freiburg", "Nagy
Janos", new String[]{"Premier Leauge", "Europe Leauge"}, "s2"));
    root.appendChild(createCsapat(newdoc, "cs3", "Leipzig", "Kerek
Elemer", new String[]{"Premier Leauge", "Europe Leauge"}, "s3"));
    root.appendChild(createMerkozoes(newdoc, "m1", "0", "2", "17:00",
"s1"));
    root.appendChild(createMerkozoes(newdoc, "m2", "1", "1", "18:00",
"s2"));
    root.appendChild(createMerkozoes(newdoc, "m3", "3", "2", "20:00",
"s3"));
    root.appendChild(createStadion(newdoc, "s1", "Emirates Stadium",
"London", "Eper", "12", "60000"));
    root.appendChild(createStadion(newdoc, "s2", "London Stadium",
"London", "Malna", "12", "60000"));
    root.appendChild(createStadion(newdoc, "s3", "Etihad Stadium",
"Manchester", "Szilva", "12", "60000"));
    root.appendChild(createJatszik(newdoc, "cs1", "m1", "0"));
    root.appendChild(createJatszik(newdoc, "cs2", "m2", "1"));
    root.appendChild(createJatszik(newdoc, "cs3", "m3", "3"));

    //Transzformáció
    TransformerFactory transformerFactory =
TransformerFactory.newInstance();
    Transformer transf = transformerFactory.newTransformer();

```

```

transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
transf.setOutputProperty(OutputKeys.INDENT, "yes");
transf.setOutputProperty("{https://xml.apache.org/xslt}indent-
amount", "2");

//Fájl létrehozása
DOMSource source = new DOMSource(newdoc);
File myFile = new File("XMLTP0M8Y1.xml");

//Kiíratás
StreamResult console = new StreamResult(System.out);
StreamResult file = new StreamResult(myFile);

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

//Gyökér elem kiírás
System.out.println("Root element: " +
doc.getDocumentElement().getNodeName());

//Adatok kigyűjtése az XML-ből
NodeList statList = doc.getElementsByTagName("statisztika");
for(int i = 0; i < statList.getLength(); i++) {
Node nNode = statList.item(i);

System.out.println("\nCurrent Element: " + nNode.getNodeName());

if( nNode.getNodeType() == Node.ELEMENT_NODE) {
Element elem = (Element) nNode;

String sid = elem.getAttribute("statisztikaID");
String lstatid = elem.getAttribute("l_stat");

Node node1 = elem.getElementsByTagName("jatekperc").item(0);
String mins = node1.getTextContent();

Node node2 = elem.getElementsByTagName("sarga").item(0);
Node node3 = elem.getElementsByTagName("piros").item(0);
String yellow = node2.getTextContent();
String red = node3.getTextContent();

Node node4 = elem.getElementsByTagName("gol").item(0);
String goals = node4.getTextContent();

Node node5 = elem.getElementsByTagName("golpassz").item(0);
String assists = node5.getTextContent();

System.out.println("Statistics id: " + sid);
System.out.println("Played minutes: " + mins);
System.out.println("Yellow cards: " + yellow);
System.out.println("Red cards: " + red);
System.out.println("Goals: " + goals);
System.out.println("Assists: " + assists);
System.out.println("Foreign key to football player: " +
lstatid);
}
}
}

```

```

NodeList lList = doc.getElementsByTagName("labdarugo");

for(int i = 0; i < lList.getLength(); i++) {
Node nNode = lList.item(i);

System.out.println("\nCurrent Element: " + nNode.getNodeName());

if( nNode.getNodeType() == Node.ELEMENT_NODE) {
    Element elem = (Element) nNode;

    String lid = elem.getAttribute("labdarugoID");
    String cslid = elem.getAttribute("cs_1");

    Node node1 = elem.getElementsByTagName("lnev").item(0);
    String lname = node1.getTextContent();

    Node node2 = elem.getElementsByTagName("kor").item(0);
    String age =node2.getTextContent();

    Node node3 = elem.getElementsByTagName("poszt").item(0);
    String pos = node3.getTextContent();

    System.out.println("Football player id: " + lid);
    System.out.println("Name: " + lname);
    System.out.println("Age: " + age);
    System.out.println("Position: " + pos);
    System.out.println("Foreign key to team: " + cslid);
}
}

NodeList csList = doc.getElementsByTagName("csapat");

for(int i = 0; i < csList.getLength(); i++) {
Node nNode = csList.item(i);

System.out.println("\nCurrent Element: " + nNode.getNodeName());

if( nNode.getNodeType() == Node.ELEMENT_NODE) {
    Element elem = (Element) nNode;

    String csid = elem.getAttribute("csapatID");
    String stadcsid = elem.getAttribute("stad_cs");

    Node node1 = elem.getElementsByTagName("csnev").item(0);
    String csname = node1.getTextContent();

    Node node2 = elem.getElementsByTagName("edzo").item(0);
    String manager =node2.getTextContent();

    System.out.println("Team id: " + csid);
    System.out.println("Name: " + csname);
    System.out.println("Manager: " + manager);

    for (int j = 0; j <
elem.getElementsByTagName("liga").getLength(); j++) {
        Node node3 = elem.getElementsByTagName("liga").item(j);
        System.out.println("League: " + node3.getTextContent());
    }
}
}

```

```

        System.out.println("Foreign key to stadium: " + stadcsid);
    }
}

NodeList mList = doc.getElementsByTagName("merkozés");

for(int i = 0; i < mList.getLength(); i++) {
    Node nNode = mList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

    if( nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        String mid = elem.getAttribute("merkozésID");
        String stadmid = elem.getAttribute("stad_m");

        Node node1 = elem.getElementsByTagName("házaigolók").item(0);
        String hgoals = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("vendéggolók").item(0);
        String agoals = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("időpont").item(0);
        String time = node3.getTextContent();

        System.out.println("Match id: " + mid);
        System.out.println("Home goals: " + hgoals);
        System.out.println("Away goals: " + agoals);
        System.out.println("Start time: " + time);
        System.out.println("Foreign key to stadium: " + stadmid);
    }
}

NodeList stadList = doc.getElementsByTagName("stadion");

for(int i = 0; i < stadList.getLength(); i++) {
    Node nNode = stadList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

    if( nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        String sid = elem.getAttribute("stadionID");

        Node node1 = elem.getElementsByTagName("snev").item(0);
        String sname = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("város").item(0);
        String city = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("utca").item(0);
        String street = node3.getTextContent();

        Node node4 = elem.getElementsByTagName("házszám").item(0);
        String hnumber = node4.getTextContent();

        Node node5 = elem.getElementsByTagName("főutca").item(0);
        String cap = node5.getTextContent();
    }
}

```



```

        System.out.println("Game id: " + sid);
        System.out.println("Name: " + sname);
        System.out.println("City: " + city);
        System.out.println("Street: " + street);
        System.out.println("House Number: " + hnumber);
        System.out.println("Capacity: " + cap);
    }
}

NodeList jList = doc.getElementsByTagName("jatszik");

for(int i = 0; i < jList.getLength(); i++) {
    Node nNode = jList.item(i);

    System.out.println("\nCurrent Element: " + nNode.getNodeName());

    if( nNode.getNodeType() == Node.ELEMENT_NODE) {
        Element elem = (Element) nNode;

        String csjid = elem.getAttribute("cs_j");
        String mjid = elem.getAttribute("m_j");

        Node node1 = elem.getElementsByTagName("pont").item(0);
        String point = node1.getTextContent();

        System.out.println("Team id: " + csjid);
        System.out.println("Game id: " + mjid);
        System.out.println("Earned point(s): " + point);
    }
}

//Segéd függvények az XML fájl íráshoz
private static Node createStatisztika(Document newdoc, String id,
String jatekperc, String sarga, String piros, String gol, String golpassz,
String fid) {

    Element node = newdoc.createElement("statisztika");

    node.setAttribute("statisztikaID", id);
    node.setAttribute("l_stat", fid);
    node.appendChild(createElement(newdoc, "jatekperc", jatekperc));

    Node node1 = createElement(newdoc, "sarga", sarga);
    Node node2 = createElement(newdoc, "piros", piros);

    Element cards = newdoc.createElement("lapok");
    cards.appendChild(node1);
    cards.appendChild(node2);
    node.appendChild(cards);

    node.appendChild(createElement(newdoc, "gol", gol));
    node.appendChild(createElement(newdoc, "golpassz", golpassz));

    return node;
}

private static Node createLabdarugo(Document newdoc, String id, String
lnev, String kor, String poszt, String fid) {

```

```

        Element node = newdoc.createElement("labdarugo");

        node.setAttribute("labdarugoID", id);
        node.setAttribute("l_stat", fid);
        node.appendChild(createElement(newdoc, "lnev", lnev));
        node.appendChild(createElement(newdoc, "kor", kor));

        node.appendChild(createElement(newdoc, "poszt", poszt));

        return node;
    }

    private static Node createCsapat(Document newdoc, String id, String
csnev, String edzo, String[] ligak, String fid) {

        Element node = newdoc.createElement("csapat");

        node.setAttribute("csapatID", id);
        node.setAttribute("l_stat", fid);
        node.appendChild(createElement(newdoc, "csnev", csnev));
        node.appendChild(createElement(newdoc, "edzo", edzo));
        for (String liga:ligak) {
            node.appendChild(createElement(newdoc, "liga", liga));
        }

        return node;
    }

    private static Node createMerkozoes(Document newdoc, String id, String
hazaigolok, String vendeggolok, String idopont, String fid) {

        Element node = newdoc.createElement("merkozoes");

        node.setAttribute("merkozoesID", id);
        node.setAttribute("stad_m", fid);
        node.appendChild(createElement(newdoc, "hazaigolok", hazaigolok));
        node.appendChild(createElement(newdoc, "vendeggolok", vendeggolok));
        node.appendChild(createElement(newdoc, "idopont", idopont));

        return node;
    }

    private static Node createStadion(Document newdoc, String id, String
snev, String varos, String utca, String hazszam, String ferohely) {

        Element node = newdoc.createElement("stadion");

        node.setAttribute("stadionID", id);
        node.appendChild(createElement(newdoc, "snev", snev));

        Node node1 = createElement(newdoc, "varos", varos);
        Node node2 = createElement(newdoc, "utca", utca);
        Node node3 = createElement(newdoc, "hazszam", hazszam);

        Element address = newdoc.createElement("cim");
        address.appendChild(node1);
        address.appendChild(node2);
        address.appendChild(node3);
        node.appendChild(address);
    }

```

```

        node.appendChild(createElement(newdoc,"ferohely",ferohely));

        return node;
    }

    private static Node createJatszik(Document newdoc, String fid2, String
fid1, String pont) {

        Element node = newdoc.createElement("jatszik");

        node.setAttribute("cs_j", fid1);
        node.setAttribute("m_j", fid2);
        node.appendChild(createElement(newdoc,"pont", pont));

        return node;
    }

    private static Node createElement(Document newdoc, String name, String
value) {

        Element node = newdoc.createElement(name);
        node.appendChild(newdoc.createTextNode(value));

        return node;
    }

}

```

## Output:

Current Element: labdarugo Football player id: l1 Name: James Maddison Age: 22 Position: CAM Foreign key to team: cs1	Current Element: merkozoes Match id: m1 Home goals: 3 Away goals: 1 Start time: 2022/10/20 18:00 Foreign key to stadium: s1	Current Element: stadion Game id: s1 Name: King Power City: Leicester Street: Eper utca House Number: 12 Capacity: 39000	Current Element: csapat Team id: cs1 Name: Leicester City Manager: Jose Mourinho Leauge:Premier Leauge Leauge:Europe Leauge Foreign key to stadium: s1
Current Element: labdarugo Football player id: l2 Name: Gabriel Jesus Age: 24 Position: ST Foreign key to team: cs2	Current Element: merkozoes Match id: m2 Home goals: 0 Away goals: 0 Start time: 2022/09/12 20:00 Foreign key to stadium: s2	Current Element: stadion Game id: s2 Name: Emirates Stadium City: London Street: Kakas utca House Number: 30 Capacity: 80000	Current Element: csapat Team id: cs2 Name: Arsenal Manager: Mikel Arteta Leauge:Premier Leauge Leauge:Europe Leauge Foreign key to stadium: s2
Current Element: labdarugo Football player id: l3 Name: Ruben Dias Age: 26 Position: CB Foreign key to team: cs3	Current Element: merkozoes Match id: m3 Home goals: 1 Away goals: 2 Start time: 2022/09/30 21:00 Foreign key to stadium: s2	Current Element: stadion Game id: s3 Name: Etihad City: Manchester Street: Kossuth utca House Number: 16 Capacity: 60000	Current Element: csapat Team id: cs3 Name: Manchester City Manager: Pep Guardiola Leauge:Premier Leauge Leauge:Champions Leauge Foreign key to stadium: s3



```
Current Element: statisztika
Statistics id: stat1
Played minutes: 800
Yellow cards: 4
Red cards: 0
Goals: 4
Assits: 5
Foreign key to football player: l1
```

```
Current Element: statisztika
Statistics id: stat2
Played minutes: 1000
Yellow cards: 2
Red cards: 0
Goals: 10
Assits: 13
Foreign key to football player: l2
```

```
Current Element: statisztika
Statistics id: stat3
Played minutes: 600
Yellow cards: 10
Red cards: 1
```

```
Current Element: statisztika
Statistics id: stat1
Played minutes: 800
Yellow cards: 4
Red cards: 0
Goals: 4
Assits: 5
Foreign key to football player: l1
```

```
Current Element: statisztika
Statistics id: stat2
Played minutes: 1000
Yellow cards: 2
Red cards: 0
Goals: 10
Assits: 13
Foreign key to football player: l2
```

```
Current Element: statisztika
Statistics id: stat3
Played minutes: 600
Yellow cards: 10
Red cards: 1
Goals: 1
Assits: 0
```

```
Current Element: játszik
Team id: cs1
Game id: m1
Earned point(s): 0
```

```
Current Element: játszik
Team id: cs2
Game id: m1
Earned point(s): 3
```

```
Current Element: játszik
Team id: cs3
Game id: m2
Earned point(s): 1
```

```
Current Element: játszik
Team id: cs1
Game id: m2
Earned point(s): 1
```

```
Current Element: játszik
Team id: cs2
Game id: m3
Earned point(s): 0
```

```
Current Element: játszik
Team id: cs3
Game id: m3
Earned point(s): 3
```

## 2b) adatmódosítás – DomModifyTP0M8Y.java

Első lépés a megfelelő importok használata, fájlok megnyitása, létrehozása, majd a builderek felépítése. Dokumentum, és a gyökérelem kezelése. A szükséges módosítások véghez vitele. Minden módosítást külön ciklusba írba.

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
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.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import java.io.File;
```

```

public class DomModifyTPOM8Y {
    public static void main(String argv[]) {

        try {

            File inputFile = new File("XMLTPOM8Y.xml");

            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

            Document doc = documentBuilder.parse(inputFile);

            // statisztika módosítása
            Node stat = doc.getElementsByTagName("statisztika").item(0);
            NodeList statList = stat.getChildNodes();

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

                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) node;

                    if ("jatekperc".equals(eElement.getNodeName())) {

                        eElement.setTextContent("300");
                    }
                }
            }

            //stadion módosítása
            Node stad = doc.getElementsByTagName("stadion").item(0);
            NodeList stadlist = stad.getChildNodes();

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

                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) node;

                    if ("ferohely".equals(eElement.getNodeName())) {
                        if ("65000".equals(eElement.getTextContent())) {
                            eElement.setTextContent("60000");
                        }
                        if ("39000".equals(eElement.getTextContent())) {
                            eElement.setTextContent("40000");
                        }
                    }
                }
            }

            //merkozes módosítása
            Node merk = doc.getElementsByTagName("merkozes").item(0);
            NodeList merklist = merk.getChildNodes();

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

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

```

```

        Element eElement = (Element) node;

        if ("hazaigolok".equals(eElement.getNodeName())) {
            if ("3".equals(eElement.getTextContent())) {
                eElement.setTextContent("2");
            }
        }
        if ("vendeggolok".equals(eElement.getNodeName())) {
            if ("1".equals(eElement.getTextContent())) {
                eElement.setTextContent("0");
            }
        }
    }
}

// Tartalom konzolra és fájlba való írása

TransformerFactory transformerFactory =
TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

System.out.println("----Módosított fájl----");

StreamResult consoleResult = new StreamResult(System.out);
StreamResult file = new StreamResult(inputFile);

transformer.transform(source, consoleResult);
transformer.transform(source, file);

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

```

Output:

```

<?xml version="1.0" encoding="UTF-8"
standalone="no"?><bajnoksag>
  <!-- Statisztikák -->
  <statisztika l_stat="l1" statisztikaID="stat1">
    <jatekperc>300</jatekperc>
    <lapok>
      <sarga>4</sarga>
      <piros>0</piros>
    </lapok>
    <gol>4</gol>
    <golpassz>5</golpassz>
  </statisztika>

```

```
<statisztika l_stat="l2" statisztikaID="stat2">
  <jatekperc>1000</jatekperc>
  <lapok>
    <sarga>2</sarga>
    <piros>0</piros>
  </lapok>
  <gol>10</gol>
  <golpassz>13</golpassz>
</statisztika>
```

```
<statisztika l_stat="l3" statisztikaID="stat3">
  <jatekperc>600</jatekperc>
  <lapok>
    <sarga>10</sarga>
    <piros>1</piros>
  </lapok>
  <gol>1</gol>
  <golpassz>0</golpassz>
</statisztika>
```

```
<!-- Labdarúgók -->
<labdarugo cs_l="cs1" labdarugoID="l1">
  <lnev>James Maddison</lnev>
  <kor>22</kor>
  <poszt>CAM</poszt>
</labdarugo>
```

```
<labdarugo cs_l="cs2" labdarugoID="l2">
  <lnev>Gabriel Jesus</lnev>
  <kor>24</kor>
  <poszt>ST</poszt>
</labdarugo>
```

```
<labdarugo cs_l="cs3" labdarugoID="l3">
```



<lnev>Ruben Dias</lnev>  
<kor>26</kor>  
<poszt>CB</poszt>  
</labdarugo>

<!-- Csapatok -->  
<csapat csapatID="cs1" stad\_cs="s1">  
  <csnev>Leicester City</csnev>  
  <edzo>Jose Mourinho</edzo>  
  <liga>Premier Leauge</liga>  
  <liga>Europe Leauge</liga>  
</csapat>

<csapat csapatID="cs2" stad\_cs="s2">  
  <csnev>Arsenal</csnev>  
  <edzo>Mikel Arteta</edzo>  
  <liga>Premier Leauge</liga>  
  <liga>Europe Leauge</liga>  
</csapat>

<csapat csapatID="cs3" stad\_cs="s3">  
  <csnev>Manchester City</csnev>  
  <edzo>Pep Guardiola</edzo>  
  <liga>Premier Leauge</liga>  
  <liga>Champions Leauge</liga>  
</csapat>

<!-- Mérkőzések -->  
<merkozes merkozesID="m1" stad\_m="s1">  
  <hazaigolok>2</hazaigolok>  
  <vendeggolok>0</vendeggolok>  
  <idopont>2022/10/20 18:00</idopont>  
</merkozes>

<merkozes merkozesID="m2" stad\_m="s2">

```
<hazaigolok>0</hazaigolok>
<vendeggolok>0</vendeggolok>
<idopont>2022/09/12 20:00</idopont>
</merkozes>
```

```
<merkozes merkozesID="m3" stad_m="s2">
  <hazaigolok>1</hazaigolok>
  <vendeggolok>2</vendeggolok>
  <idopont>2022/09/30 21:00</idopont>
</merkozes>
```

```
<!-- Stadionok -->
<stadion stadionID="s1">
  <snev>King Power</snev>
  <cim>
    <varos>Leicester</varos>
    <utca>Eper utca</utca>
    <hazszam>12</hazszam>
  </cim>
  <ferohely>40000</ferohely>
</stadion>
```

```
<stadion stadionID="s2">
  <snev>Emirates Stadium</snev>
  <cim>
    <varos>London</varos>
    <utca>Kakas utca</utca>
    <hazszam>30</hazszam>
  </cim>
  <ferohely>80000</ferohely>
</stadion>
```

```
<stadion stadionID="s3">
  <snev>Etihad</snev>
  <cim>
```

```
<varos>Manchester</varos>
<utca>Kossuth utca</utca>
<hazszam>16</hazszam>
</cim>
<ferohely>60000</ferohely>
</stadion>
```

```
<!-- Mérkőzésen játszó csapatok -->
<jatszik cs_j="cs1" m_j="m1">
  <pont>0</pont>
</jatszik>
```

```
<jatszik cs_j="cs2" m_j="m1">
  <pont>3</pont>
</jatszik>
```

```
<jatszik cs_j="cs3" m_j="m2">
  <pont>1</pont>
</jatszik>
```

```
<jatszik cs_j="cs1" m_j="m2">
  <pont>1</pont>
</jatszik>
```

```
<jatszik cs_j="cs2" m_j="m3">
  <pont>0</pont>
</jatszik>
```

```
<jatszik cs_j="cs3" m_j="m3">
  <pont>3</pont>
</jatszik>
</bajnoksag>
```

Process finished with exit code 0

## 2c) adatlekérdezés – DomQueryTP0M8Y.java

Első lépés a megfelelő importok használata, fájlok megnyitása, létrehozása, majd a builderek felépítése. Dokumentum, és a gyökérelem kezelése. A kívánt lekérdezések megírása. Majd kiíráshoz szükséges ciklusok létrehozása.

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

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
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 DomQueryTP0M8Y {
    public static void main(String[] args) {

        try {
            File xmlFile = new File("studentTP0M8Y.xml");

            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

            Document document =documentBuilder.parse("XMLTP0M8Y.xml");

            document.getDocumentElement().normalize();

            XPath xPath = XPathFactory.newInstance().newXPath();

            //A statisztikák kiválasztása
            //String expression = "/bajnoksag/statisztika";

            //A csapat első elemének a kiválasztása
            //String expression = "/bajnoksag/csapat[1]";

            //Az össze elem kiválasztása a dokumentumból
            //String expression = "//*";

            //A mérkzések időpontjainak a kiválasztása
            //String expression = "//idopont";

            //A 40000-nél nagyobb férőhelyű stadionok kiválasztása
            //String expression = "//stadion[ferohely>40000]";

            // Hazai és vendég gólok számának lekérdezése
            String expression = "//hazaigolok | //vendeggolok";
```

```

        NodeList nodeList = (NodeList)
XPath.compile(expression).evaluate(document, XPathConstants.NODESET);

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

            System.out.println("\nAktuális elem: " +
node.getNodeName());

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

                System.out.println("Statisztika ID:" +
element.getAttribute("statisztikaID"));

                System.out.println("Idegen kulcs:" +
element.getAttribute("l_stat"));

                System.out.println("Játékperc: " +
element.getElementsByTagName("jatekperc").item(0).getTextContent());

                System.out.println("Sárga lapok: " +
element.getElementsByTagName("sarga").item(0).getTextContent());

                System.out.println("Piros lapok: " +
element.getElementsByTagName("piros").item(0).getTextContent());

                System.out.println("Gólok: " +
element.getElementsByTagName("gol").item(0).getTextContent());

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

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

                System.out.println("Labdarúgó ID:" +
element.getAttribute("labdarugoID"));

                System.out.println("Idegen kulcs:" +
element.getAttribute("cs_l"));

                System.out.println("Név: " +
element.getElementsByTagName("lnev").item(0).getTextContent());

                System.out.println("Kor: " +
element.getElementsByTagName("kor").item(0).getTextContent());

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

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

```

```

        System.out.println("Csapat ID:" +
element.getAttribute("csapatID"));

        System.out.println("Idegen kulcs:" +
element.getAttribute("stad_cs"));

        System.out.println("Név: " +
element.getElementsByTagName("csnev").item(0).getTextContent());

        System.out.println("Edző: " +
element.getElementsByTagName("edzo").item(0).getTextContent());

        if (nodeList.item(i).getChildNodes().getLength() > 2) {
            int db = 0;
            Node liga =
element.getElementsByTagName("liga").item(0);
            while (liga != null) {
                liga =
element.getElementsByTagName("liga").item(db);
                if (liga != null) {
                    String liganev = liga.getTextContent();
                    System.out.println("Liga amelyben játszik a
csapat: " + liganev);
                }
                db++;
            }
        }

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

            System.out.println("Mérkőzés ID:" +
element.getAttribute("merkozoesID"));

            System.out.println("Idegen kulcs:" +
element.getAttribute("stad_m"));

            System.out.println("Hazai gólok száma: " +
element.getElementsByTagName("hazaigolok").item(0).getTextContent());

            System.out.println("Vendég gólok száma: " +
element.getElementsByTagName("vendeggolok").item(0).getTextContent());

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

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

            System.out.println("Stadion ID:" +
element.getAttribute("stadionID"));

            System.out.println("Név: " +
element.getElementsByTagName("snev").item(0).getTextContent());

            System.out.println("Cím: " +

```

```

element.getElementsByTagName("varos").item(0).getTextContent()
    + ", " +
element.getElementsByTagName("utca").item(0).getTextContent()
    + " " +
element.getElementsByTagName("hazszam").item(0).getTextContent());

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

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

        System.out.println("Csapatra mutató idegen kulcs:" +
element.getAttribute("cs_j"));

        System.out.println("Mérkőzésre mutató idegen kulcs:" +
element.getAttribute("m_j"));

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

    // Mérkőzés időpontjának kiíratása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("idopont")) {

        Element element = (Element) node;

        System.out.println("Idopont: " +
element.getTextContent());

    }

    // Hazai gólok számának kiíratása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("hazaigolok")) {

        Element element = (Element) node;

        System.out.println("Hazai gólok száma: " +
element.getTextContent());

    }

    // Vendég gólok számának kiíratása
    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("vendeggolok")) {

        Element element = (Element) node;

        System.out.println("Vendég gólok száma: " +
element.getTextContent());

    }

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

```

```

    }
    catch (SAXException e)
    {
        e.printStackTrace();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
    catch (XPathExpressionException e)
    {
        e.printStackTrace();
    }
}
}

```

Output:

-----  
 Query output:  
 -----

Aktuális elem: statisztika

Statisztika ID:stat1

Idegen kulcs:l1

Játékperc: 300

Sárga lapok: 4

Piros lapok: 0

Gólok: 4

Gólpasszok: 5

Aktuális elem: statisztika

Statisztika ID:stat2

Idegen kulcs:l2

Játékperc: 1000

Sárga lapok: 2

Piros lapok: 0

Gólok: 10

Gólpasszok: 13

Aktuális elem: statisztika

Statisztika ID:stat3

Idegen kulcs:l3

Játékperc: 600

Sárga lapok: 10

Piros lapok: 1

Gólok: 1



Gólpaszok: 0

-----  
End of the query  
-----

-----  
Query output:  
-----

Aktuális elem: csapat

Csapat ID:cs1

Idegen kulcs:s1

Név: Leicester City

Edző: Jose Mourinho

Liga amelyben játszik a csapat: Premier League

Liga amelyben játszik a csapat: Europe League

-----  
End of the query  
-----

-----  
Query output:  
-----

Aktuális elem: bajnoksag

Aktuális elem: statisztika

Statisztika ID:stat1

Idegen kulcs:l1

Játékperc: 300

Sárga lapok: 4

Piros lapok: 0

Gólok: 4

Gólpaszok: 5

Aktuális elem: jatekperc

Aktuális elem: lapok

Aktuális elem: sarga

Aktuális elem: piros

Aktuális elem: gol

Aktuális elem: golpassz

Aktuális elem: statisztika

Statisztika ID:stat2

Idegen kulcs:l2

Játékperc: 1000

Sárga lapok: 2

Piros lapok: 0

Gólok: 10

Gólpasszok: 13

Aktuális elem: jatekperc

Aktuális elem: lapok

Aktuális elem: sarga

Aktuális elem: piros

Aktuális elem: gol

Aktuális elem: golpassz

Aktuális elem: statisztika

Statisztika ID:stat3

Idegen kulcs:l3

Játékperc: 600

Sárga lapok: 10

Piros lapok: 1

Gólok: 1

Gólpasszok: 0

Aktuális elem: jatekperc

Aktuális elem: lapok

Aktuális elem: sarga

Aktuális elem: piros

Aktuális elem: gol

Aktuális elem: golpassz

Aktuális elem: labdarugo

Labdarúgó ID:l1

Idegen kulcs:cs1

Név: James Maddison

Kor: 22

Poszt: CAM

Aktuális elem: lnev

Aktuális elem: kor

Aktuális elem: poszt

Aktuális elem: labdarugo

Labdarúgó ID:l2

Idegen kulcs:cs2

Név: Gabriel Jesus

Kor: 24

Poszt: ST

Aktuális elem: lnev

Aktuális elem: kor

Aktuális elem: poszt

Aktuális elem: labdarugo

Labdarúgó ID:l3

Idegen kulcs:cs3

Név: Ruben Dias

Kor: 26

Poszt: CB

Aktuális elem: lnev

Aktuális elem: kor

Aktuális elem: poszt

Aktuális elem: csapat  
Csapat ID:cs1  
Idegen kulcs:s1  
Név: Leicester City  
Edző: Jose Mourinho  
Liga amelyben játszik a csapat: Premier League  
Liga amelyben játszik a csapat: Europe League

Aktuális elem: csnev

Aktuális elem: edzo

Aktuális elem: liga

Aktuális elem: liga

Aktuális elem: csapat  
Csapat ID:cs2  
Idegen kulcs:s2  
Név: Arsenal  
Edző: Mikel Arteta  
Liga amelyben játszik a csapat: Premier League  
Liga amelyben játszik a csapat: Europe League

Aktuális elem: csnev

Aktuális elem: edzo

Aktuális elem: liga

Aktuális elem: liga

Aktuális elem: csapat  
Csapat ID:cs3  
Idegen kulcs:s3  
Név: Manchester City  
Edző: Pep Guardiola  
Liga amelyben játszik a csapat: Premier League  
Liga amelyben játszik a csapat: Champions League

Aktuális elem: csnev

Aktuális elem: edzo

Aktuális elem: liga

Aktuális elem: liga

Aktuális elem: merkozes

Mérkőzés ID:m1

Idegen kulcs:s1

Hazai gólok száma: 2

Vendég gólok száma: 0

Időpont: 2022/10/20 18:00

Aktuális elem: hazaigolok

Hazai gólok száma: 2

Aktuális elem: vendeggolok

Vendég gólok száma: 0

Aktuális elem: idopont

Idopont: 2022/10/20 18:00

Aktuális elem: merkozes

Mérkőzés ID:m2

Idegen kulcs:s2

Hazai gólok száma: 0

Vendég gólok száma: 0

Időpont: 2022/09/12 20:00

Aktuális elem: hazaigolok

Hazai gólok száma: 0

Aktuális elem: vendeggolok

Vendég gólok száma: 0

Aktuális elem: idopont

Idopont: 2022/09/12 20:00

Aktuális elem: merkozes

Mérkőzés ID:m3

Idegen kulcs:s2

Hazai gólok száma: 1

Vendég gólok száma: 2  
Időpont: 2022/09/30 21:00

Aktuális elem: hazaigolok  
Hazai gólok száma: 1

Aktuális elem: vendeggolok  
Vendég gólok száma: 2

Aktuális elem: idopont  
Idopont: 2022/09/30 21:00

Aktuális elem: stadion  
Stadion ID:s1  
Név: King Power  
Cím: Leicester, Eper utca 12  
Férőhely: 40000

Aktuális elem: snev

Aktuális elem: cim

Aktuális elem: varos

Aktuális elem: utca

Aktuális elem: hazszam

Aktuális elem: ferohely

Aktuális elem: stadion  
Stadion ID:s2  
Név: Emirates Stadium  
Cím: London, Kakas utca 30  
Férőhely: 80000

Aktuális elem: snev

Aktuális elem: cim

Aktuális elem: varos

Aktuális elem: utca

Aktuális elem: hazszam

Aktuális elem: ferohely

Aktuális elem: stadion

Stadion ID:s3

Név: Etihad

Cím: Manchester, Kossuth utca 16

Férőhely: 60000

Aktuális elem: snev

Aktuális elem: cim

Aktuális elem: varos

Aktuális elem: utca

Aktuális elem: hazszam

Aktuális elem: ferohely

Aktuális elem: játszik

Csapatra mutató idegen kulcs:cs1

Mérkőzésre mutató idegen kulcs:m1

Pont: 0

Aktuális elem: pont

Aktuális elem: játszik

Csapatra mutató idegen kulcs:cs2

Mérkőzésre mutató idegen kulcs:m1

Pont: 3

Aktuális elem: pont

Aktuális elem: játszik

Csapatra mutató idegen kulcs:cs3

Mérkőzésre mutató idegen kulcs:m2

Pont: 1

Aktuális elem: pont

Aktuális elem: játszik  
Csapatra mutató idegen kulcs:cs1  
Mérkőzésre mutató idegen kulcs:m2  
Pont: 1

Aktuális elem: pont

Aktuális elem: játszik  
Csapatra mutató idegen kulcs:cs2  
Mérkőzésre mutató idegen kulcs:m3  
Pont: 0

Aktuális elem: pont

Aktuális elem: játszik  
Csapatra mutató idegen kulcs:cs3  
Mérkőzésre mutató idegen kulcs:m3  
Pont: 3

Aktuális elem: pont

-----  
End of the query  
-----

-----  
Query output:  
-----

Aktuális elem: idopont  
Idopont: 2022/10/20 18:00

Aktuális elem: idopont  
Idopont: 2022/09/12 20:00

Aktuális elem: idopont  
Idopont: 2022/09/30 21:00

-----  
End of the query  
-----  
-----



Query output:

-----

Aktuális elem: stadion  
Stadion ID:s2  
Név: Emirates Stadium  
Cím: London, Kakas utca 30  
Férőhely: 80000

Aktuális elem: stadion  
Stadion ID:s3  
Név: Etihad  
Cím: Manchester, Kossuth utca 16  
Férőhely: 60000

-----

End of the query

-----

Query output:

-----

Aktuális elem: hazaigolok  
Hazai gólok száma: 2

Aktuális elem: vendeggolok  
Vendég gólok száma: 0

Aktuális elem: hazaigolok  
Hazai gólok száma: 0

Aktuális elem: vendeggolok  
Vendég gólok száma: 0

Aktuális elem: hazaigolok  
Hazai gólok száma: 1

Aktuális elem: vendeggolok  
Vendég gólok száma: 2

-----

End of the query

-----