

# JEGYZŐKÖNYV

Adatkezelés XML  
környezetben  
Féléves feladat

Labdarúgó csapatok nyílvántartása

Készítette: Takács Bálint  
Neptunkód: P2GNFT  
Dátum: 2022. 11. 28.

## **A feladat leírása:**

Az adatbázisban egy labdarúgó bajnokság egyik fordulójának az adatait tároljuk. Ez azt jelenti, hogy maga az adatbázis összesen 6 osztályból áll, ahhoz hogy le tudjuk tárolni a fordulóban résztvevő csapatok és a csapatok játékosainak valamint edzőjének az adatait, ezek mellett nyilvántartjuk még a mérkőzéseken vezető játékvezetők és partjelzők adatait, végezetül pedig az maguknak a mérkőzéseknek az adatait, mint például a mérkőzés helyszínét és időpontját. Mivel ez csak egy tervezett forduló ezért ezek a mérkőzések még nem kerültek lejátszásra, így a mérkőzéseken történekről nem tudtam adatot rögzíteni az adatbázisba.

Minden csapatnak van egy egyedi kulcsa valamint egy idegen kulcsa, ez az idegen kulcs jelöli, hogy az adott csapat mely azonosítójú mérkőzésen vesz részt. Ezek mellett eltárolásra kerültek még a csapatok nevei, az alapításuk éve, a székhelyük valamint a csaptok színei.

A játékos egyedek szintén egy egyedi és egy idegen kulccsal rendelkeznek azonban itt az idegen kulcs annak a csapatnak az azonosítójára mutat, amelyben az adott játékos játszik. Nyilván vannak még tartva a játékosok nevei, a posztjuk, a koruk valamint a nemzetiségük.

Az edzők hasonló tulajdonságokkal rendelkeznek, mint a játékosok, az egyedi kulcsuk mellett az idegen kulcsuk annak a csapatnak az azonosítójára mutatnak, amelyiknél dolgoznak. A nevük és a koruk mellett nyilvántartjuk még a nemzetiségüket is.

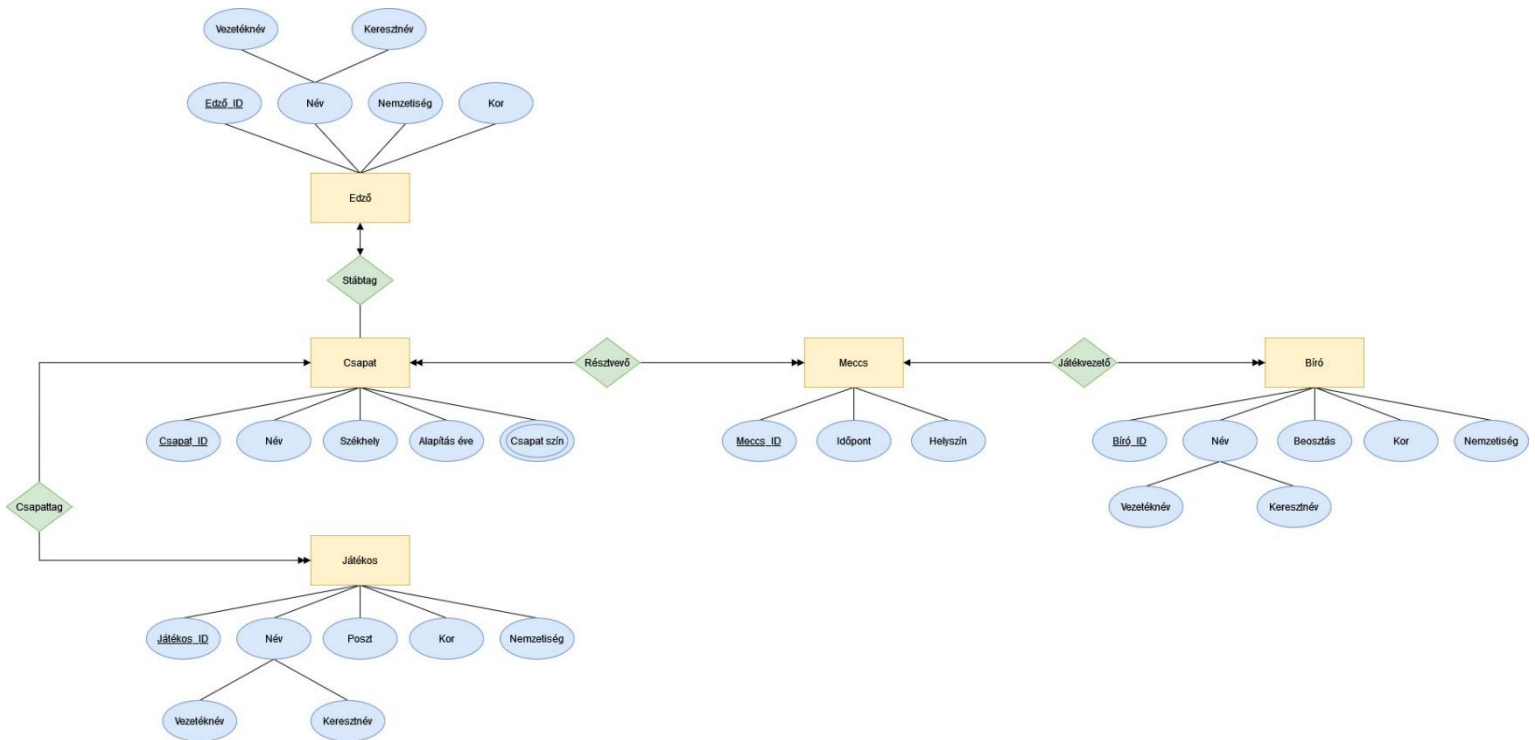
A bírók osztálya szintén két kulccsal, egy egyedivel valamint egy idegennel rendelkezik. Az idegen kulcs itt is a „csapat” osztályhoz hasonlóan annak a mérkőzésnek az azonosítójára mutat, amelyet a bírók vezetnek. Ezek mellett letárolásra kerültek még a bírók nevei, a mérkőzésen való beosztásuk, a koruk és a nemzetiségük.

Az utolsó osztály a mérkőzések osztálya. Ez az osztály egy egyedi kulccsal rendelkezik, amelyre a csapatok és a bírók osztályának idegen kulcsa is mutat, valamint rendelkezik még két tulajdonsággal, amiket már korábban is említettem, mégpedig a meccsek helyszínével és az időpontjával.

### **1. feladat**

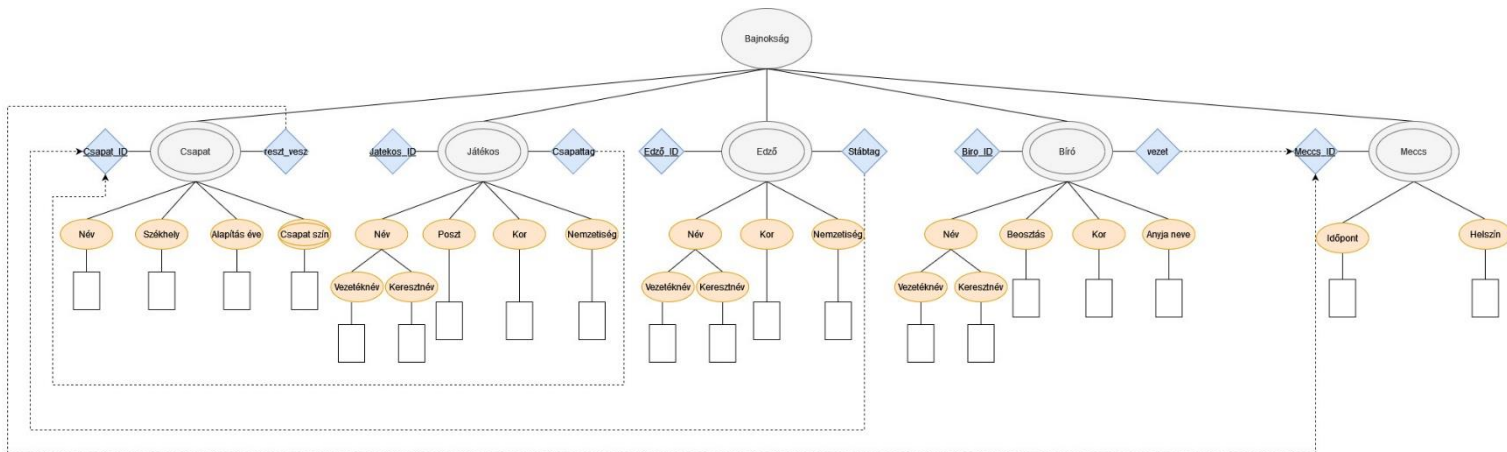
#### **1.a) feladat - Az adatbázis ER modell**

Az adatbázis felépítésének szemléltetés ER Modell formájában.



## 1.b) feladat - Az adatbázis konvertálása XDM modellre

Az előbbi ER modell konvertáltam át XDM modellre. Ezen már megjelenítésre kerültek az idegen kulcsok és a kapcsolataik.



### 1.c) feladat - Az XDM modell alapján XML dokumentum készítése

Az adatbázisban lévő egyedek és azok tulajdonságainak a rögzítése egy XML dokumentumba.

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

<bajnoksag xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaP2GNFT.xsd">

  <!--Csapatok-->

  <csapat csapat_ID = "001" rv = "01">
    <nev>DVTK</nev>
    <szekhely>Miskolc</szekhely>
    <alapitas_eve>1910</alapitas_eve>
    <szin>piros</szin>
  </csapat>

  <csapat csapat_ID = "002" rv = "01">
    <nev>FTC</nev>
    <szekhely>Budapest</szekhely>
    <alapitas_eve>1899</alapitas_eve>
    <szin>zöld</szin>
  </csapat>

  <csapat csapat_ID = "003" rv = "02">
    <nev>DVSC</nev>
    <szekhely>Debrecen</szekhely>
    <alapitas_eve>1902</alapitas_eve>
    <szin>fehér</szin>
  </csapat>

  <csapat csapat_ID = "004" rv = "02">
    <nev>Újpest FC</nev>
    <szekhely>Budapest</szekhely>
    <alapitas_eve>1885</alapitas_eve>
    <szin>lila</szin>
  </csapat>

  <csapat csapat_ID = "005" rv = "03">
    <nev>MTK</nev>
    <szekhely>Budapest</szekhely>
    <alapitas_eve>1888</alapitas_eve>
    <szin>kék</szin>
  </csapat>
```

```
<csapat csapat_ID = "006" rv = "03">
  <nev>Mezőkövesd Zsóry FC</nev>
  <szekhely>Mezőkövesd</szekhely>
  <alapitas_eve>1975</alapitas_eve>
  <szin>sárga</szin>
</csapat>

<!--Játékosok-->

<jatekos jatekos_ID = "101" csapattag = "001">
  <nev>
    <vezeteknev>Könyves</vezeteknev>
    <keresztnev>Norbert</keresztnev>
  </nev>
  <poszt>Csatár</poszt>
  <kor>35</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "102" csapattag = "001">
  <nev>
    <vezeteknev>Danilovic</vezeteknev>
    <keresztnev>Branislav</keresztnev>
  </nev>
  <poszt>Kapus</poszt>
  <kor>34</kor>
  <nemzetiseg>Szerb</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "201" csapattag = "002">
  <nev>
    <vezeteknev>Dibusz</vezeteknev>
    <keresztnev>Dénes</keresztnev>
  </nev>
  <poszt>Kapus</poszt>
  <kor>31</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "202" csapattag = "002">
  <nev>
    <vezeteknev>Nguen</vezeteknev>
    <keresztnev>Tokmac</keresztnev>
  </nev>
  <poszt>Csatár</poszt>
  <kor>28</kor>
```

```
        <nemzetiseg>Norvég</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "301" csapattag = "003">
    <nev>
        <vezeteknev>Dzsudzsák</vezeteknev>
        <keresztnev>Balázs</keresztnev>
    </nev>
    <poszt>Középpályás</poszt>
    <kor>35</kor>
    <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "302" csapattag = "003">
    <nev>
        <vezeteknev>Gróf</vezeteknev>
        <keresztnev>Dávid</keresztnev>
    </nev>
    <poszt>Kapus</poszt>
    <kor>30</kor>
    <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "401" csapattag = "004">
    <nev>
        <vezeteknev>Csongvai</vezeteknev>
        <keresztnev>Áron</keresztnev>
    </nev>
    <poszt>Középpályás</poszt>
    <kor>22</kor>
    <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "402" csapattag = "004">
    <nev>
        <vezeteknev>Csoboth</vezeteknev>
        <keresztnev>Kevin</keresztnev>
    </nev>
    <poszt>Csatár</poszt>
    <kor>21</kor>
    <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "501" csapattag = "005">
    <nev>
        <vezeteknev>Stiber</vezeteknev>
        <keresztnev>Zoltán</keresztnev>
```

```
        </nev>
        <poszt>Középpályás</poszt>
        <kor>35</kor>
        <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "502" csapattag = "005">
  <nev>
    <vezeteknev>Futács</vezeteknev>
    <keresztnev>Márkó</keresztnev>
  </nev>
  <poszt>Csatár</poszt>
  <kor>33</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "601" csapattag = "006">
  <nev>
    <vezeteknev>Bobál</vezeteknev>
    <keresztnev>Dávid</keresztnev>
  </nev>
  <poszt>Védő</poszt>
  <kor>24</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</jatekos>

<jatekos jatekos_ID = "602" csapattag = "006">
  <nev>
    <vezeteknev>Drazic</vezeteknev>
    <keresztnev>Stefan</keresztnev>
  </nev>
  <poszt>Csatár</poszt>
  <kor>31</kor>
  <nemzetiseg>Szerb</nemzetiseg>
</jatekos>

<!--Edzők-->

<edzo edzo_ID = "100" stabtag = "001">
  <nev>
    <vezeteknev>Kusznycov</vezeteknev>
    <keresztnev>Szergej</keresztnev>
  </nev>
  <kor>40</kor>
  <nemzetiseg>orosz</nemzetiseg>
</edzo>
```

```
<edzo edzo_ID = "200" stabtag = "002">
  <nev>
    <vezeteknev>Cserecseszov</vezeteknev>
    <keresztnev>Sztanyiszlav</keresztnev>
  </nev>
  <kor>62</kor>
  <nemzetiseg>orosz</nemzetiseg>
</edzo>
```

```
<edzo edzo_ID = "300" stabtag = "003">
  <nev>
    <vezeteknev>Blagojevic</vezeteknev>
    <keresztnev>Srdjan</keresztnev>
  </nev>
  <kor>46</kor>
  <nemzetiseg>Szerb</nemzetiseg>
</edzo>
```

```
<edzo edzo_ID = "400" stabtag = "004">
  <nev>
    <vezeteknev>Krusic</vezeteknev>
    <keresztnev>Milos</keresztnev>
  </nev>
  <kor>44</kor>
  <nemzetiseg>Szerb</nemzetiseg>
</edzo>
```

```
<edzo edzo_ID = "500" stabtag = "005">
  <nev>
    <vezeteknev>Bognár</vezeteknev>
    <keresztnev>György</keresztnev>
  </nev>
  <kor>60</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</edzo>
```

```
<edzo edzo_ID = "600" stabtag = "006">
  <nev>
    <vezeteknev>Kuttor</vezeteknev>
    <keresztnev>Attila</keresztnev>
  </nev>
  <kor>51</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</edzo>
```

```
<!--Bírók-->
```



```
<biro biro_ID = "011" vezet = "01">
  <nev>
    <vezeteknev>Kassai</vezeteknev>
    <keresztnev>Viktor</keresztnev>
  </nev>
  <beosztas>Játékvezető</beosztas>
  <kor>39</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</biro>

<biro biro_ID = "012" vezet = "01">
  <nev>
    <vezeteknev>Gulyás</vezeteknev>
    <keresztnev>András</keresztnev>
  </nev>
  <beosztas>Partjelző</beosztas>
  <kor>34</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</biro>

<biro biro_ID = "021" vezet = "02">
  <nev>
    <vezeteknev>Bognár</vezeteknev>
    <keresztnev>Tamás</keresztnev>
  </nev>
  <beosztas>Játékvezető</beosztas>
  <kor>35</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</biro>

<biro biro_ID = "022" vezet = "02">
  <nev>
    <vezeteknev>Kertész</vezeteknev>
    <keresztnev>Viktor</keresztnev>
  </nev>
  <beosztas>Partjelző</beosztas>
  <kor>29</kor>
  <nemzetiseg>Magyar</nemzetiseg>
</biro>

<biro biro_ID = "031" vezet = "03">
  <nev>
    <vezeteknev>Farkas</vezeteknev>
    <keresztnev>Ádám</keresztnev>
  </nev>
  <beosztas>Játékvezető</beosztas>
  <kor>44</kor>
```

```

    <nemzetiseg>Magyar</nemzetiseg>
</biro>

<biro biro_ID = "032" vezet = "03">
    <nev>
        <vezeteknev>Nagy</vezeteknev>
        <keresztnev>Mihály</keresztnev>
    </nev>
    <beosztas>Partjelző</beosztas>
    <kor>38</kor>
    <nemzetiseg>Magyar</nemzetiseg>
</biro>

<!--Meccsek-->

<meccs meccs_ID = "01">
    <idopont>2022.07.23.</idopont>
    <helyszin>Miskolc</helyszin>
</meccs>

<meccs meccs_ID = "02">
    <idopont>2022.07.24.</idopont>
    <helyszin>Debrecen</helyszin>
</meccs>

<meccs meccs_ID = "03">
    <idopont>2022.07.25.</idopont>
    <helyszin>Budapest</helyszin>
</meccs>

</bajnoksag>

```

### 1.d) feladat - Az XML dokumentum alapján

#### XMLSchema készítése

XMLSchema készítése az XML fájlhoz az adatok validálásának érdekében.

```

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

    <xs:element name="bajnoksag">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="csapat" type="csapatTipus"
maxOccurs="unbounded" />

```

```

        <xs:element name="jatekos" type="jatekosTipus"
maxOccurs="unbounded" />
        <xs:element name="edzo" type="edzoTipus"
maxOccurs="unbounded" />
        <xs:element name="biro" type="biroTipus"
maxOccurs="unbounded" />
        <xs:element name="meccs" type="meccsTipus"
maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>

<!--Kulcsok-->

<xs:key name="csapat_kulcs">
    <xs:selector xpath="csapat"></xs:selector>
    <xs:field xpath="@csapat_ID"></xs:field>
</xs:key>
<xs:key name="jatekos_kulcs">
    <xs:selector xpath="jatekos"></xs:selector>
    <xs:field xpath="@jatekos_ID"></xs:field>
</xs:key>
<xs:key name="edzo_kulcs">
    <xs:selector xpath="edzo"></xs:selector>
    <xs:field xpath="@edzo_ID"></xs:field>
</xs:key>
<xs:key name="biro_kulcs">
    <xs:selector xpath="biro"></xs:selector>
    <xs:field xpath="@biro_ID"></xs:field>
</xs:key>
<xs:key name="meccs_kulcs">
    <xs:selector xpath="meccs"></xs:selector>
    <xs:field xpath="@meccs_ID"></xs:field>
</xs:key>

<!--Idegen kulcsok-->

    <xs:keyref refer="csapat_kulcs"
name="jatekos_idegen_kulcs">
        <xs:selector xpath="jatekos" />
        <xs:field xpath="@csapattag" />
    </xs:keyref>

    <xs:keyref refer="meccs_kulcs" name="csapat_idegen_kulcs">
        <xs:selector xpath="csapat" />
        <xs:field xpath="@rv" />
    </xs:keyref>

```

```

    <xs:keyref refer="meccs_kulcs" name="biro_idegen_kulcs">
        <xs:selector xpath="biro" />
        <xs:field xpath="@vezet" />
    </xs:keyref>

    <!-- Az 1:1 kapcsolat megvalósítása -->
    <xs:unique name="unique_edzo">
        <xs:selector xpath="edzo"></xs:selector>
        <xs:field xpath="@stabtag"></xs:field>
    </xs:unique>

</xs:element>

<!--Összetett típusok-->

<xs:complexType name="nevTipus">
    <xs:sequence>
        <xs:element name="vezeteknev" type="xs:string" />
        <xs:element name="keresztnev" type="xs:string" />
    </xs:sequence>
</xs:complexType>

<xs:complexType name="csapatTipus">
    <xs:sequence>
        <xs:element name="nev" type="xs:string" />
        <xs:element name="szekhely" type="xs:string" />
        <xs:element name="alapitas_eve" type="xs:integer" />
        <xs:element name="szin" type="xs:string" />
    </xs:sequence>

    <xs:attribute name="csapat_ID" type="xs:integer"
use="required" />
    <xs:attribute name="rv" type="xs:integer" use="required" />

</xs:complexType>

<xs:complexType name="jatekosTipus">
    <xs:sequence>
        <xs:element name="nev" type="nevTipus" />
        <xs:element name="poszt" type="xs:string" />
        <xs:element name="kor" type="xs:integer" />
        <xs:element name="nemzetiseg" type="xs:string" />
    </xs:sequence>

    <xs:attribute name="jatekos_ID" type="xs:integer"
use="required" />

```

```

        <xs:attribute name="csapattag" type="xs:integer"
use="required" />

    </xs:complexType>

    <xs:complexType name="edzoTipus">
        <xs:sequence>
            <xs:element name="nev" type="nevTipus" />
            <xs:element name="kor" type="xs:integer" />
            <xs:element name="nemzetiseg" type="xs:string" />
        </xs:sequence>

        <xs:attribute name="edzo_ID" type="xs:integer"
use="required" />
        <xs:attribute name="stabtag" type="xs:integer"
use="required" />

    </xs:complexType>

    <xs:complexType name="biroTipus">
        <xs:sequence>
            <xs:element name="nev" type="nevTipus" />
            <xs:element name="beosztas" type="xs:string" />
            <xs:element name="kor" type="xs:integer" />
            <xs:element name="nemzetiseg" type="xs:string" />
        </xs:sequence>

        <xs:attribute name="biro_ID" type="xs:integer"
use="required" />
        <xs:attribute name="vezet" type="xs:integer" use="required"
/>

    </xs:complexType>

    <xs:complexType name="meccsTipus">
        <xs:sequence>
            <xs:element name="idopont" type="xs:string" />
            <xs:element name="helyszin" type="xs:string" />
        </xs:sequence>

        <xs:attribute name="meccs_ID" type="xs:integer"
use="required" />

    </xs:complexType>

</xs:schema>

```

## 2. feladat

### 2.a) feladat – adatolvasás

Az adatok beolvasása és kiírása a konzolra struktúráltan.

```
package hu.domparse.p2gnft;

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

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 DOMReadP2GNFT {
    public static void main(String[] args) throws
ParserConfigurationException, IOException, SAXException {

        File xmlFile = new File("XMLP2GNFT.xml");

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

        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        System.out.println("Root element: " +
doc.getDocumentElement().getNodeName());

        NodeList nList1 = doc.getElementsByTagName("csapat");

        System.out.println("\n--Csapatok--\n");
    }
}
```

```

        for (int i = 0; i < nList1.getLength(); i++) {

            Node node = nList1.item(i);

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

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

                Element elem = (Element) node;

                String uid = elem.getAttribute("csapat_ID");

                Node node1 =
elem.getElementsByTagName("nev").item(0);
                String name = node1.getTextContent();

                Node node2 =
elem.getElementsByTagName("szekhely").item(0);
                String place = node2.getTextContent();

                Node node3 =
elem.getElementsByTagName("alapitas_eve").item(0);
                String found = node3.getTextContent();

                Node node4 =
elem.getElementsByTagName("szin").item(0);
                String color = node4.getTextContent();

                System.out.printf("Csapat ID = %s\n", uid);
                System.out.printf("Nev = %s\n", name);
                System.out.printf("Szekhely = %s\n", place);
                System.out.printf("Alapitas eve = %s\n", found);
                System.out.printf("Szin = %s\n", color);

            }
        }

        NodeList nList2 = doc.getElementsByTagName("jatekos");

        System.out.println("\n--Jatekosok--\n");

        for (int i = 0; i < nList2.getLength(); i++) {

            Node node = nList2.item(i);

```

```

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

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

            Element elem = (Element) node;

            String uid = elem.getAttribute("jatekos_ID");

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

            Node node2 =
elem.getElementsByTagName("keresztnev").item(0);
            String fname = node2.getTextContent();

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

            Node node4 =
elem.getElementsByTagName("kor").item(0);
            String age = node4.getTextContent();

            Node node5 =
elem.getElementsByTagName("nemzetiseg").item(0);
            String nation = node5.getTextContent();

            System.out.printf("Jatekos ID = %s\n", uid);
            System.out.printf("Vezeteknev = %s\n", fname);
            System.out.printf("Keresztnev = %s\n", lname);
            System.out.printf("Poszt = %s\n", position);
            System.out.printf("Kor = %s\n", age);
            System.out.printf("Nemzetiseg = %s\n", nation);

        }
    }

    NodeList nList3 = doc.getElementsByTagName("edzo");

    System.out.println("\n--Edzok--\n");

    for (int i = 0; i < nList3.getLength(); i++) {

        Node node = nList3.item(i);

```



```

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

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

            Element elem = (Element) node;

            String uid = elem.getAttribute("edzo_ID");

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

            Node node2 =
elem.getElementsByTagName("keresztnev").item(0);
            String fname = node2.getTextContent();

            Node node3 =
elem.getElementsByTagName("kor").item(0);
            String age = node3.getTextContent();

            Node node4 =
elem.getElementsByTagName("nemzetiseg").item(0);
            String nation = node4.getTextContent();

            System.out.printf("Edzo ID = %s\n", uid);
            System.out.printf("Vezeteknev = %s\n", fname);
            System.out.printf("Keresztnev = %s\n", lname);
            System.out.printf("Kor = %s\n", age);
            System.out.printf("Nemzetiseg = %s\n", nation);

        }
    }

    NodeList nList4 = doc.getElementsByTagName("biro");

    System.out.println("\n--Birok--\n");

    for (int i = 0; i < nList4.getLength(); i++) {

        Node node = nList4.item(i);

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

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

```

```

        Element elem = (Element) node;

        String uid = elem.getAttribute("biro_ID");

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

        Node node2 =
elem.getElementsByTagName("keresztnev").item(0);
        String fname = node2.getTextContent();

        Node node3 =
elem.getElementsByTagName("beosztas").item(0);
        String role = node3.getTextContent();

        Node node4 =
elem.getElementsByTagName("kor").item(0);
        String age = node4.getTextContent();

        Node node5 =
elem.getElementsByTagName("nemzetiseg").item(0);
        String nation = node5.getTextContent();

        System.out.printf("Biro ID = %s\n", uid);
        System.out.printf("Vezeteknev = %s\n", fname);
        System.out.printf("Keresztnev = %s\n", lname);
        System.out.printf("Beosztas = %s\n", role);
        System.out.printf("Kor = %s\n", age);
        System.out.printf("Nemzetiseg = %s\n", nation);

    }
}

NodeList nList5 = doc.getElementsByTagName("meccs");

System.out.println("\n--Meccsek--\n");

for (int i = 0; i < nList5.getLength(); i++) {

    Node node = nList5.item(i);

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

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

```

```

        Element elem = (Element) node;

        String uid = elem.getAttribute("meccs_ID");

        Node node1 =
elem.getElementsByTagName("idopont").item(0);
        String date = node1.getTextContent();

        Node node2 =
elem.getElementsByTagName("helyszin").item(0);
        String place = node2.getTextContent();

        System.out.printf("Meccs ID = %s\n", uid);
        System.out.printf("Idopont = %s\n", date);
        System.out.printf("Helyszin = %s\n", place);

    }
}
}
}

```

## 2.b) feladat – adatlekérdezés

Ebben a feladatban az alábbi lekérdezéseket hajtom végre:

- Az adatbázisban rögzített csatárok adatainak kiírása
- Az adatbázisban rögzített idegenlégiósok adatainak kiírása
- Azon játékosok adatainak a kiírása, akik még 30 évnél fiatalabbak
- Azon csapatok adatainak a kiírása, amelyeket 1900 után alapítottak
- A budapesti székhelyű csapatok adatainak a kiírása

```

package hu.domparse.p2gnft;

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

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 DOMQueryP2GNFT {

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

        File xmlFile = new File("XMLP2GNFT.xml");

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

        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        System.out.println("Root element: " +
doc.getDocumentElement().getNodeName());

        NodeList nList = doc.getElementsByTagName("jatekos");

        // Csatárok kiírása

        System.out.println("\n--Csatárok kiírása--\n");

        for (int i = 0; i < nList.getLength(); i++) {

            Node node = nList.item(i);

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

                Element elem = (Element) node;

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

                if (position.equals("Csatár")) {

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

                    String uid = elem.getAttribute("jatekos_ID");

```

```

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

        Node node2 =
elem.getElementsByTagName("keresztnev").item(0);
        String fname = node2.getTextContent();

        Node node4 =
elem.getElementsByTagName("kor").item(0);
        String age = node4.getTextContent();

        Node node5 =
elem.getElementsByTagName("nemzetiseg").item(0);
        String nation = node5.getTextContent();

        System.out.printf("Jatekos ID = %s\n", uid);
        System.out.printf("Vezeteknev = %s\n", fname);
        System.out.printf("Keresztnev = %s\n", lname);
        System.out.printf("Poszt = %s\n", position);
        System.out.printf("Kor = %s\n", age);
        System.out.printf("Nemzetiseg = %s\n", nation);

    }

}

// Idegen légiósok kiírása

System.out.println("\n--Idegen légiósok kiírása--\n");

for (int i = 0; i < nList.getLength(); i++) {

    Node node = nList.item(i);

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

        Element elem = (Element) node;

        Node node5 =
elem.getElementsByTagName("nemzetiseg").item(0);
        String nation = node5.getTextContent();

        if (!nation.equals("Magyar")) {

```

```

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

        String uid = elem.getAttribute("jatekos_ID");

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

        Node node2 =
elem.getElementsByTagName("keresztnev").item(0);
        String fname = node2.getTextContent();

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

        Node node4 =
elem.getElementsByTagName("kor").item(0);
        String age = node4.getTextContent();

        System.out.printf("Jatekos ID = %s\n", uid);
        System.out.printf("Vezeteknev = %s\n", fname);
        System.out.printf("Keresztnev = %s\n", lname);
        System.out.printf("Poszt = %s\n", position);
        System.out.printf("Kor = %s\n", age);
        System.out.printf("Nemzetiseg = %s\n", nation);

    }

}

// 30 év alattiak kiíratása

System.out.println("\n--30 év alattiak kiíratása--\n");

for (int i = 0; i < nList.getLength(); i++) {

    Node node = nList.item(i);

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

        Element elem = (Element) node;

        Node node4 =
elem.getElementsByTagName("kor").item(0);

```

```

        int age = Integer.parseInt(node4.getTextContent());

        if (age < 30) {

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

            String uid = elem.getAttribute("jatekos_ID");

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

            Node node2 =
elem.getElementsByTagName("keresztnev").item(0);
            String fname = node2.getTextContent();

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

            Node node5 =
elem.getElementsByTagName("nemzetiseg").item(0);
            String nation = node5.getTextContent();

            System.out.printf("Jatekos ID = %s\n", uid);
            System.out.printf("Vezeteknev = %s\n", fname);
            System.out.printf("Keresztnev = %s\n", lname);
            System.out.printf("Poszt = %s\n", position);
            System.out.printf("Kor = %d\n", age);
            System.out.printf("Nemzetiseg = %s\n", nation);

        }

    }

    NodeList nList2 = doc.getElementsByTagName("csapat");

    // 1900 után alapított csapatok kiírása

    System.out.println("\n--1900 után alapított csapatok
kiírása--\n");

    for (int i = 0; i < nList2.getLength(); i++) {

        Node node = nList2.item(i);

```

```

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

            Element elem = (Element) node;

            Node node4 =
elem.getElementsByTagName("alapitas_eve").item(0);
            int year =
Integer.parseInt(node4.getTextContent());

            if (year > 1900) {

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

                String uid = elem.getAttribute("csapat_ID");

                Node node1 =
elem.getElementsByTagName("nev").item(0);
                String name = node1.getTextContent();

                Node node2 =
elem.getElementsByTagName("szekhely").item(0);
                String place = node2.getTextContent();

                Node node3 =
elem.getElementsByTagName("szin").item(0);
                String color = node3.getTextContent();

                System.out.printf("Csapat ID = %s\n", uid);
                System.out.printf("Nev = %s\n", name);
                System.out.printf("Szekhely = %s\n", place);
                System.out.printf("Alapitas eve = %d\n", year);
                System.out.printf("Szin = %s\n", color);

            }

        }
    }

    // Budapesti csapatok kiírása

    System.out.println("\n--Budapesti csapatok--\n");

    for (int i = 0; i < nList2.getLength(); i++) {

        Node node = nList2.item(i);

```



```

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

            Element elem = (Element) node;

            Node node2 =
elem.getElementsByTagName("szekhely").item(0);
            String place = node2.getTextContent();

            if (place.equals("Budapest")) {

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

                String uid = elem.getAttribute("csapat_ID");

                Node node1 =
elem.getElementsByTagName("nev").item(0);
                String name = node1.getTextContent();

                Node node3 =
elem.getElementsByTagName("alapitas_eve").item(0);
                int year =
Integer.parseInt(node3.getTextContent());

                Node node4 =
elem.getElementsByTagName("szin").item(0);
                String color = node4.getTextContent();

                System.out.printf("Csapat ID = %s\n", uid);
                System.out.printf("Nev = %s\n", name);
                System.out.printf("Szekhely = %s\n", place);
                System.out.printf("Alapitas eve = %s\n", year);
                System.out.printf("Szin = %s\n", color);

            }

        }

    }

}
}

```

## 2.c) feladat – adatmódosítás

Az alábbi adatok módosítása, majd mentése egy új XML fájlba:

- A DVTK azonosítójának megváltoztatása 01-ről 00-ra
- A DVTK nevének megváltoztatása "Diósgyőri VTK"-ra
- Könyves Norbert korának korrigálása 35-ről 33-ra
- Könyves Norbert posztjának módosítása középpályásra
- Meccsek törlése

```
package hu.domparse.p2gnft;

import java.io.File;

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;

public class DOMModifyP2GNFT {

    public static void main(String[] args) {

        try {

            // Forrás fájl beolvasása

            File inputFile = new File("XMLP2GNFT_Copy.xml");

            DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
```

```

        DocumentBuilder docBuilder =
docBuilderFactory.newDocumentBuilder();

        Document doc = docBuilder.parse(inputFile);

        // Egy csapat és egy játékos adatainak a mentése

        Node csapat1 =
doc.getElementsByTagName("csapat").item(0);
        Node jatekos1 =
doc.getElementsByTagName("jatekos").item(0);

        // Gyökérelem

        Node bajnoksag = doc.getFirstChild();

        // A DVTK azonosítójának megváltoztatása 01-ről 00-ra

        NamedNodeMap attr = csapat1.getAttributes();
        Node nodeAttr = attr.getNamedItem("csapat_ID");
        nodeAttr.setTextContent("00");

        // A DVTK nevének megváltoztatása "Diósgyőri VTK"-ra

        NodeList csapatnev = csapat1.getChildNodes();

        for (int temp = 0; temp < csapatnev.getLength();
temp++) {
            Node node = csapatnev.item(temp);

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

                if ("nev".equals(elem.getNodeName())) {

                    elem.setTextContent("Diósgyőri VTK");

                }
            }
        }

        // Könyves Norbert korának korrigálása 35-ről 33-ra

        NodeList list = jatekos1.getChildNodes();

        for (int temp = 0; temp < list.getLength(); temp++) {
            Node node = list.item(temp);

```

```

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

            if ("kor".equals(elem.getNodeName())) {

                elem.setTextContent("33");

            }
        }
    }

    // Könyves Norbert posztjának módosítása középpályásra

    NodeList list2 = jatekos1.getChildNodes();

    for (int temp = 0; temp < list2.getLength(); temp++) {
        Node node = list2.item(temp);

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

            if ("poszt".equals(elem.getNodeName())) {

                elem.setTextContent("Középpályás");

            }
        }
    }

    // Meccsek törlése

    NodeList childNodes = bajnoksag.getChildNodes();

    for (int count = 0; count < childNodes.getLength();
count++) {
        Node node = childNodes.item(count);

        if ("meccs".equals(node.getNodeName())) {
            bajnoksag.removeChild(node);
        }
    }

    // Kiíratás

    TransformerFactory transformerFactory =
TransformerFactory.newInstance();

```

```
        Transformer transformer =
transformerFactory.newTransformer();

        DOMSource source = new DOMSource(doc);

        System.out.println("-----Módosított fájl-----");
        StreamResult consoleResult = new
StreamResult(System.out);
        transformer.transform(source, consoleResult);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```