Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

NBA felépítése

Készítette: Hegyesi Kristóf

Neptunkód: JPTJY2

Dátum: 2022.11.27

Tartalomjegyzék:

1.feladat

A feladat leírása:

Témának az egyik kedvencem sportom, legerősebb bajnokságának a szerkezetét készítettem el. Az NBA világszinten különleges bajnokság gazdasági, szerkezeti, és üzleti szemszögből. Minden szezon más és más, elemzők próbálják megjósolni mi fog történni az elkövetkezendő alapszakaszban és rájátszásban, mondanom sem kell, hogy sokan elvéreznek. 30 csapat játszik egymással egy alapszakasz 82 meccsből áll és a rájátszásban 4 nyert meccsig tartó párharc veszi kezdetét a legjobb 16 csapat között. Egy csapatban 14 +-2 db játékos lehet. (itt lehet szó különböző nem „regular" szerződésekről, ezt most nem részletezem) A 2010-es évektől az NBA bevétele ugrásszerűen megemelkedett, elég csak abba belegondolni, hogy egy karrierét kezdő játékos évi 578K$-9M$ kereshet, amit, ha törik-ha szakad megkap. Jelenleg a legjobban kereső játékos Stephen Curry (2021/22-es szezontól a 2025/2026-os szezonig 262M$ fog keresni, ez átlagban 52,4M$ évente). Itt játszik a kosárlabda világ krémje, nap mint elképesztőbbnél elképesztőbb pillanatok történnek. Sokszor itt az emberfeletti átlagosnak tűnik. Remélem ezzel a rövid téma leírással és ismertetéssel elértem a célom hogy felhívjam a kíváncsiságot a beadandómra és terjesszem ezt a remek sportot és ligát Magyarországon.

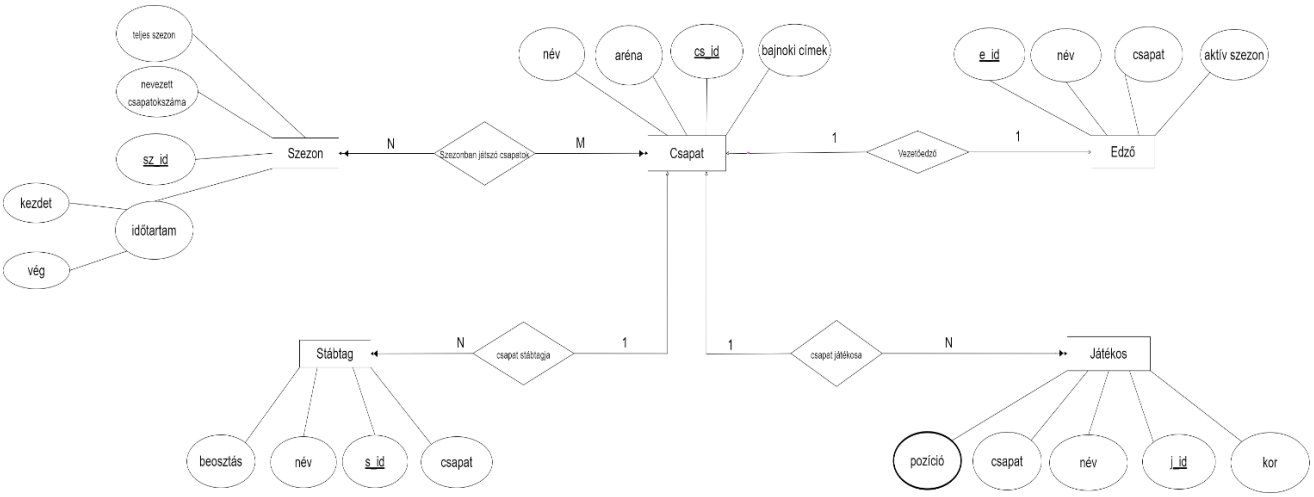
1.a ER modell:

Szezon, Csapat, Játékos, Edző, Stábtag. Ezek azok az egyedek, amik az ER modellt alkotják. A lentebb látható ábrán szépen leolvashatók a kapcsolatok, illetve, ha az ember kicsit belegondol logikusan következnek ezek a kapcsolatok:

-egy szezonban több csapat játszhat és egy csapat is több szezonban játszhat

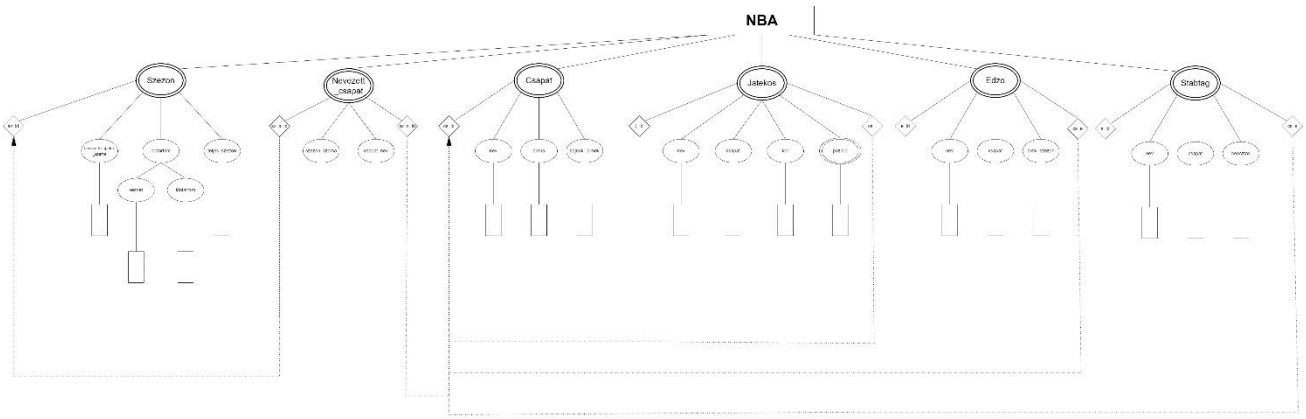
-egy csapatban több játékos játszhat, de egy játékos csak egy csapatban

-egy csapatban egy edző (head coach) lehet és egy edző is csak egy csapatot edzhet



1.b XDM modell:

Az ER modell alapján egy XDM modellt hoztam létre, ahol fa struktúrával mutatom meg az XML fájlom felépítését.



1.c XML

Az XDM és ER modell alapján létrehoztam az XML fájlt amelyben minden egyedből 3-at hoztam létre.

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

<!--Az nba-ben játszó/dolgozó személyek -->

<nba xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaJPTJY2.xsd">

    <!--Az nba-ben lejegyzett idények -->

    <szezon sz_id="1">
        <nevezettcsapatok_szama>30</nevezettcsapatok_szama>
        <teljes_szezon>Igen</teljes_szezon>
        <idotartam>
            <kezdet>2021-08-03</kezdet>
            <veg>2022-06-14</veg>
        </idotartam>
    </szezon>

    <szezon sz_id="2">
        <nevezettcsapatok_szama>30</nevezettcsapatok_szama>
        <teljes_szezon>Nem</teljes_szezon>
        <idotartam>
            <kezdet>2019-08-07</kezdet>
            <veg>2020-10-13</veg>
        </idotartam>
    </szezon>

    <szezon sz_id="3">
        <nevezettcsapatok_szama>28</nevezettcsapatok_szama>
        <teljes_szezon>Nem</teljes_szezon>
        <idotartam>
            <kezdet>2010-11-04</kezdet>
            <veg>2011-06-24</veg>
        </idotartam>
    </szezon>
```

```xml
<!--Az nba-ben lejegyzett csapatok -->

<csapat cs_id="1">
    <nev>Cleveland Cavaliers</nev>
    <arena>Rocket Morgtage Fieldhouse</arena>
    <bajnoki_cimek>1</bajnoki_cimek>
</csapat>

<csapat cs_id="2">
    <nev>Miami Heat</nev>
    <arena>FTX Arena</arena>
    <bajnoki_cimek>3</bajnoki_cimek>
</csapat>

<csapat cs_id="3">
    <nev>Dallas Mavericks</nev>
    <arena>American Airlines Center</arena>
    <bajnoki_cimek>1</bajnoki_cimek>
</csapat>

<!--Az nba-ben egyes csapatok vezetodezoje -->

<edzo e_id="1" cs_e="1">
    <nev>J.B Bickerstaff</nev>
    <csapat>Clevleand Cavaliers</csapat>
    <aktiv_szezon>3</aktiv_szezon>
</edzo>

<edzo e_id="2" cs_e="3">
    <nev>Jason Kidd</nev>
    <csapat>Dallas Mavericks</csapat>
    <aktiv_szezon>2</aktiv_szezon>
</edzo>

<edzo e_id="3" cs_e="2">
    <nev>Erik Spoelstra</nev>
    <csapat>Miami Heat</csapat>
    <aktiv_szezon>15</aktiv_szezon>
```

1.d XML Schema

Az XML fájl alapján egy sémát hoztam létre, ahol definiálom az elemeket. Saját típusokat hoztam létre, saját típusokat definiáltam.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="nba" type="nbaType">
  </xs:element>

  <xs:complexType name="idotartamType">
    <xs:sequence>
      <xs:element type="xs:date" name="kezdet"/>
      <xs:element type="xs:date" name="veg"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="szezonType">
    <xs:sequence>
      <xs:element type="xs:int" name="nevezettcsapatok_szama"/>
      <xs:element type="xs:string" name="teljes_szezon"/>
      <xs:element type="idotartamType" name="idotartam"/>
    </xs:sequence>
    <xs:attribute type="xs:int" name="sz_id" use="required"/>
  </xs:complexType>

  <xs:complexType name="csapatType">
    <xs:sequence>
      <xs:element type="xs:string" name="nev"/>
      <xs:element type="xs:string" name="arena"/>
      <xs:element type="xs:int" name="bajnoki_cimek"/>
    </xs:sequence>
    <xs:attribute type="xs:int" name="cs_id" use="required"/>
  </xs:complexType>

  <xs:complexType name="edzoType">
    <xs:sequence>
      <xs:element type="xs:string" name="nev"/>
      <xs:element type="xs:string" name="csapat"/>
      <xs:element type="xs:int" name="aktiv_szezon"/>
    </xs:sequence>
    <xs:attribute type="xs:int" name="e_id" use="required"/>
    <xs:attribute type="xs:int" name="cs_e" use="required"/>
  </xs:complexType>
```

2.feladat

2.a DOM Read:

Az XML fájlból minden elemet kiolvas. Mindegyik elemet külön metódus olvas ki, ezeket összefogóan egy DomReader metódus foglal össze.

```
public class DomReadJPTJY2 {
    kmhegyesi
    public static void main(String[] args) { DomReader(); }

    1 usage  kmhegyesi
    public static void  DomReader(){
        readSzezon();
        readCsapat();
        readEdzo();
        readStabtag();
        readJatekos();
    }

    //kiolvasom az összes játékost
    1 usage  kmhegyesi
    private static void readJatekos(){
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

        try {

            dbf.setFeature(XMLConstants.FEATURE_SECURE_PROCESSING, true);

            DocumentBuilder db = dbf.newDocumentBuilder();

            Document doc = db.parse(new File("XMLJPTJY2.xml"));

            doc.getDocumentElement().normalize();

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

            for (int temp = 0; temp < jatekosList.getLength(); temp++) {

                Node node = jatekosList.item(temp);

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

                    Element element = (Element) node;
```

```
            Element element = (Element) node;

            String id1 = element.getAttribute("j_id");
            String id2 = element.getAttribute("cs_j");
            String nev = element.getElementsByTagName("nev").item(0).getTextContent();
            String csapat = element.getElementsByTagName("csapat").item(0).getTextContent();
            String kor = element.getElementsByTagName("kor").item(0).getTextContent();
            String pozicio1 = element.getElementsByTagName("pozicio").item(0).getTextContent();
            String pozicio2 = element.getElementsByTagName("pozicio").item(1).getTextContent();


            System.out.println("Jelenlegi elem :" + node.getNodeName());
            System.out.println("Játékos Id : " + id1);
            System.out.println("Csapat Id : " + id2);
            System.out.println("Név : " + nev);
            System.out.println("Csapat : " + csapat);
            System.out.println("Kor : " + kor);
            System.out.println("Pozíció : " + "Elsődleges : " + pozicio1 + " Másodlagos : " + pozicio2);
            System.out.println("------------------");

        }
    }

} catch (ParserConfigurationException | SAXException | IOException e) {
    e.printStackTrace();
}
```

2.b DOM Query

Az XML fájlból kérdez le adatokat XPath segítségével. 1 lekérdezés az összes játékost lekérdezi. A maradék 4 pedig egy input paraméter után végzi a lekérdezéseket.

```
public static void DomQuery(){
    try {
        DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder;

        dBuilder = dbFactory.newDocumentBuilder();

        document = dBuilder.parse("XMLJPTJY2.xml");

        document.getDocumentElement().normalize();

        //az összes játékos lekérdezése
        getAllPlayers();

        // input, játékos lekérés megadott Id alapján
        System.out.println("Search a player by id");
        Scanner input = new Scanner(System.in);
        String id = input.nextLine();
        getPlayerById(id);

        // input, csapat lekérés megadott csapatnév alapján
        System.out.println("Search team by name");
        String team = input.nextLine();
        getTeamByName(team);

        // input, játékosok lekérése pozíció alapján
        System.out.println("Search player by position");
        String position = input.nextLine();
        getPlayersByPosition(position);

        // input, stábtagok lekérése beosztás alapján
        System.out.println("Search staff by role");
        String role = input.nextLine();
        getStaffByRole(role);

    } catch (ParserConfigurationException e) {
        e.printStackTrace();
    } catch (SAXException e) {
        e.printStackTrace();
    } catch (IOException e) {
```

```java
private static void getAllPlayers() {


    String expression = String.format("/nba/jatekos");
    try {

        System.out.printf("NBA játékosok lekerdezese: \n");

        NodeList playersList = (NodeList) xPath.compile(expression).evaluate(document, XPathConstants.NODESET);

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

            Node node = playersList.item(i);

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

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

                Element elem = (Element) node;

                String id1 = elem.getAttribute("j_id");
                String id2 = elem.getAttribute("cs_j");
                String jatekosNev = elem.getElementsByTagName("nev").item(0).getTextContent();
                String jatekosCsapata = elem.getElementsByTagName("csapat").item(0).getTextContent();
                String jatekosKora = elem.getElementsByTagName("kor").item(0).getTextContent();

                System.out.println("Játékos Id : " + id1);
                System.out.println("Csapat Id : " + id2);
                System.out.println("Név : " + jatekosNev);
                System.out.println("Csapat : " + jatekosCsapata);
                System.out.println("Kor : " + jatekosKora);
                System.out.println("------------------");

            }
        }

    } catch (XPathExpressionException e) {
        e.printStackTrace();
    }
```

3.c DOM Modify

Az XML fájl elemeinek tetszőleges vagy összes adatait módosítom, majd ezeket kiírom egy új XML fájlba.

```
public static void DomModify()  throws SAXException,
        IOException, ParserConfigurationException{

    File xmlFile = new File("XMLJPTJY2.xml");

    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = factory.newDocumentBuilder();
    doc = dBuilder.parse(xmlFile);

    modifyNo2SeasonsId();

    modifySeasonsToNotWhole();

    addChampionToNo2Season();

    modifyStaffMembersRoleToHeadCoach();

    deleteAgeElementFromEveryPlayer();

    try (FileOutputStream output = new FileOutputStream("ModifiedXMLJPTJY2.xml")) {
        writeXml(doc, output);
    } catch (TransformerException e) {
        e.printStackTrace();
    }
}
```

```java
// a 2. Id-val  rendelkező szezonhoz hozzáadunk egy bajnok elemet
1 usage  kmhegyesi
private static void addChampionToNo2Season(){

    NodeList szezonList = doc.getElementsByTagName("szezon");
    Node szezon = szezonList.item(2);

    Element bajnok = doc.createElement("bajnok");
    bajnok.appendChild(doc.createTextNode("LosAngeles Lakers"));
    szezon.appendChild(bajnok);
}

// a 2. Id val rendelkező szezont Id-ját 4 re változtatjuk
1 usage  kmhegyesi
private static void modifyNo2SeasonsId(){

    NodeList szezonokList = doc.getElementsByTagName("szezon");
    Node szezon = szezonokList.item(1);

    szezon.getAttributes().getNamedItem("sz_id").setTextContent("4");
}

// minden szezon teljesz_szezon elemének értékét "Nem"-re állítjuk
1 usage  kmhegyesi
private static void modifySeasonsToNotWhole() {
    NodeList szezonList = doc.getElementsByTagName("szezon");
    modifyElement(szezonList,  tagName: "teljes_szezon",  newValue: "Nem");
}

// minden stábtag beosztását "Heac Coach"-ra állítjuk
1 usage  kmhegyesi
private static void modifyStaffMembersRoleToHeadCoach(){
    NodeList staffList = doc.getElementsByTagName("stabtag");
    modifyElement(staffList, tagName: "beosztas", newValue: "Head Coach");

}
```