

# Jegyzőkönyv

Adatkezelés XML környezetben

*Féléves feladat*

Bíró Erik

ACI3X3

## Feladat

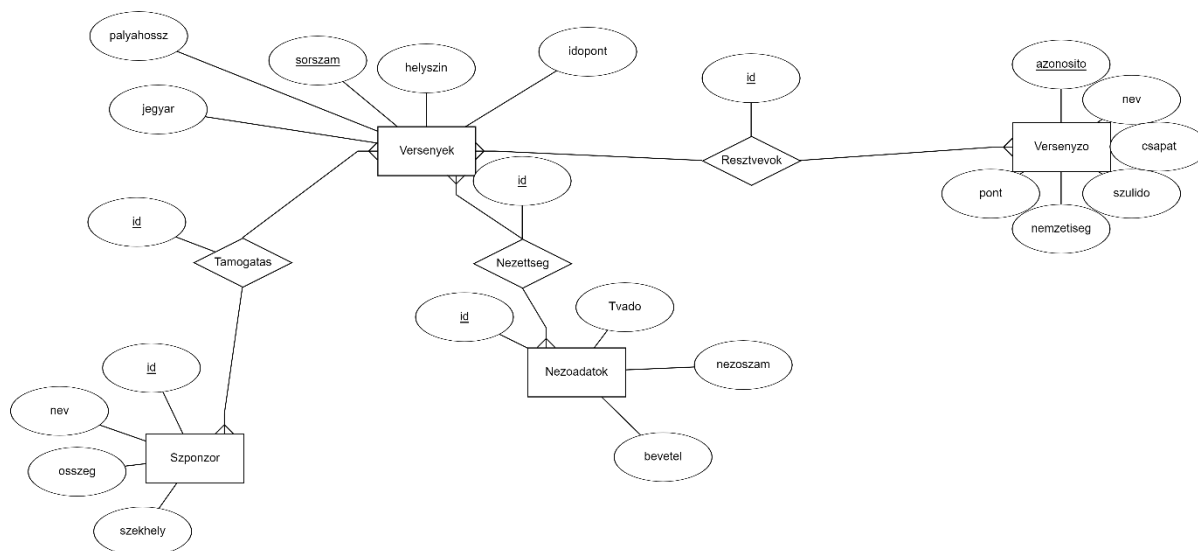
Feladatom egy F1 versenyeket tartalmazó adatbázis modell létrehozása volt .

XDM modellel valamint olvasható és szerkeszthető XML létrehozásával.

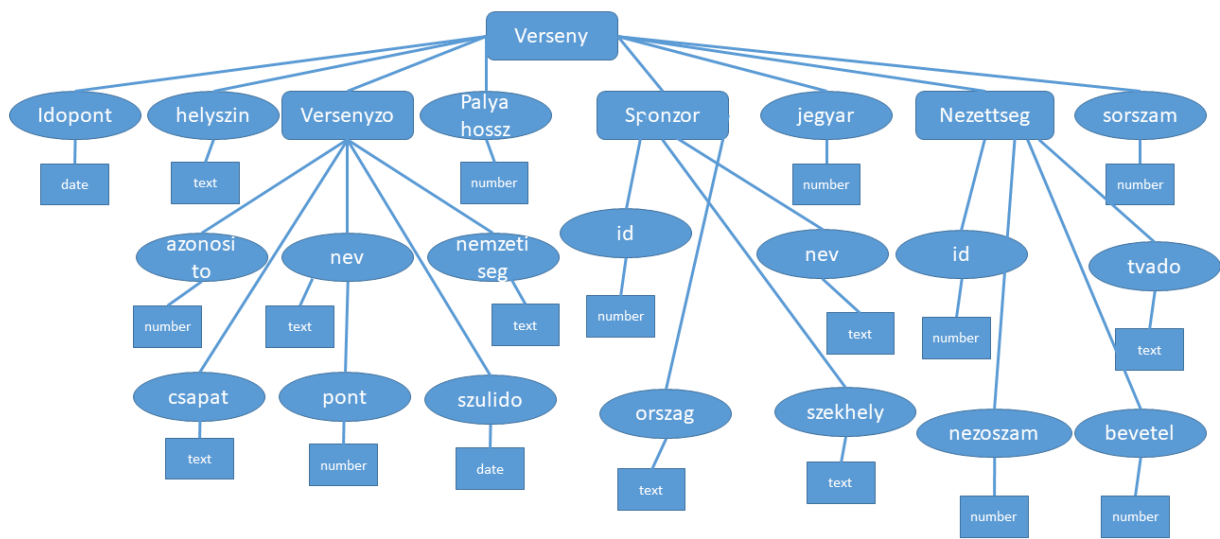
Tárolja a versenypálya adatait, versenyzők pontszámát és személyes adatokat.

És a versenyhez tartozó nézettségi adatokat és a versenyt szponzoráló cégekről információkat.

## ER modell



## XDM modell



## XML

```
<?xml version="1.0" encoding="UTF-8"?>
<verseny sorszam="1">
  <idopont>2004.09.22</idopont>
  <helyszin>Sydney</helyszin>
  <palyahossz>2342</palyahossz>
  <jegyar>1516</jegyar>
  <versenyzo azonosito="4">
    <nev>John Lock</nev>
    <nemzetiseg>Ausztrál</nemzetiseg>
    <pont>4</pont>
    <szulido>1978.08.15</szulido>
    <csapat>Ferrari</csapat>
  </versenyzo>
  <sponzor id="1">
    <nev>MacCutcheon</nev>
    <szehely>Los Angeles</szehely>
    <orszag>Amerika</orszag>
  </sponzor>
  <nezettseg id="1">
    <nezoszam>108</nezoszam>
    <bevetel>4815162342</bevetel>
    <nezoszam>1623</nezoszam>
    <tvado>ABC</tvado>
  </nezettseg>
</verseny>
```

## XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="verseny" type="versenyType"/>
  <xs:complexType name="versenyzoType">
    <xs:sequence>
      <xs:element type="xs:string" name="nev"/>
      <xs:element type="xs:string" name="nemzetiseg"/>
      <xs:element type="xs:byte" name="pont"/>
      <xs:element type="xs:string" name="szulido"/>
      <xs:element type="xs:string" name="csapat"/>
    </xs:sequence>
    <xs:attribute type="xs:byte" name="azonosito"/>
  </xs:complexType>
  <xs:complexType name="sponzorType">
    <xs:sequence>
      <xs:element type="xs:string" name="nev"/>
      <xs:element type="xs:string" name="szehely"/>
      <xs:element type="xs:string" name="ország"/>
    </xs:sequence>
    <xs:attribute type="xs:byte" name="id"/>
  </xs:complexType>
  <xs:complexType name="nezettsegType">
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:element type="xs:byte" name="nezoszam"/>
      <xs:element name="bevetel">
        <xs:simpleType>
          <xs:restriction base="xs:long">
            <xs:enumeration value="4815162342"/>
            <xs:enumeration value="1623"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element type="xs:string" name="tvado"/>
    </xs:choice>
    <xs:attribute type="xs:byte" name="id"/>
  </xs:complexType>
  <xs:complexType name="versenyType">
    <xs:sequence>
      <xs:element type="xs:string" name="idopont"/>
      <xs:element type="xs:string" name="helyszin"/>
      <xs:element type="xs:short" name="palyahossz"/>
      <xs:element type="xs:short" name="jegyar"/>
      <xs:element type="versenyzoType" name="versenyzo"/>
      <xs:element type="sponzorType" name="sponzor"/>
      <xs:element type="nezettsegType" name="nezettseg"/>
    </xs:sequence>
    <xs:attribute type="xs:byte" name="sorszam"/>
  </xs:complexType>
</xs:schema>
```

## DOM Read

```
12 public class DomRead {
13
14     public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException {
15         File inputFile = new File("D:\\XMLaci3x3.xml"); //file helyének megadása
16
17         Document doc = DocumentBuilderFactory
18             .newInstance()
19             .newDocumentBuilder()
20             .parse(inputFile);
21
22         Element root = doc.getDocumentElement();
23         root.normalize(); //olvasható egyed létrehozása
24
25         printDocument(root, ""); //alábbi metódus meghívása
26     }
27
28     public static void printDocument(Node root, String separator) {
29         String nodename = root.getNodeName(); //egyedek listázása
30         if (!nodename.contains("text")) {
31             System.out.println(separator + nodename);
32         }
33         separator += " ";
34
35         NodeList children = root.getChildNodes();
36         //egyedek gyerekelemeinek feldolgozása
37         for (int i = 0; i < children.getLength(); i++) {
38             Node child = children.item(i);
39             boolean isComplex = child.getTextContent().contains("\n");
40
41             if (isComplex) {
42                 printDocument(child, separator);
43             } else {
44                 System.out.print(separator + child.getNodeName()); //kiírás
45                 System.out.println(": " + child.getTextContent());
46             }
47         }
48     }
49 }
```

<

Problems @ Javadoc Declaration Console Call Hierarchy

<terminated> DomRead [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (2020. dec. 1. 15:13:49 – 15:13:50)

```
versenyzo
nev: John Lock
nemzetiseg: modify
pont: 4
szulido: 1978.08.15
csapat: Ferrari
sponzor
nev: MacCutcheon
szehely: Los Angeles
orszag: Amerika
nezettseg
nezoszam: 108
bevetel: 4815162342
nezoszam: 1623
tvado: ABC
```

## DOM Modify

```
import org.w3c.dom.Document;

public class DomModify {

    public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException, TransformerException {
        File inputFile = new File("D:\\XML\\Laci3x3.xml"); //hely megadása

        Document doc = DocumentBuilderFactory
            .newInstance()
            .newDocumentBuilder()
            .parse(inputFile);

        Element root = doc.getDocumentElement();
        root.normalize(); //olvasható egyed létrehozása

        Node beosztas = doc.getElementsByTagName("nemzetiseg").item(0); //módos tándó adattag kiválsztása
        beosztas.setTextContent("modify"); //adott adatttag módos tása
        Transformer transformer = TransformerFactory
            .newInstance()
            .newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(inputFile);
        transformer.transform(source, result); //mododítások mentése

        System.out.println("kesz");
    }
}
```