Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Bíró Erik

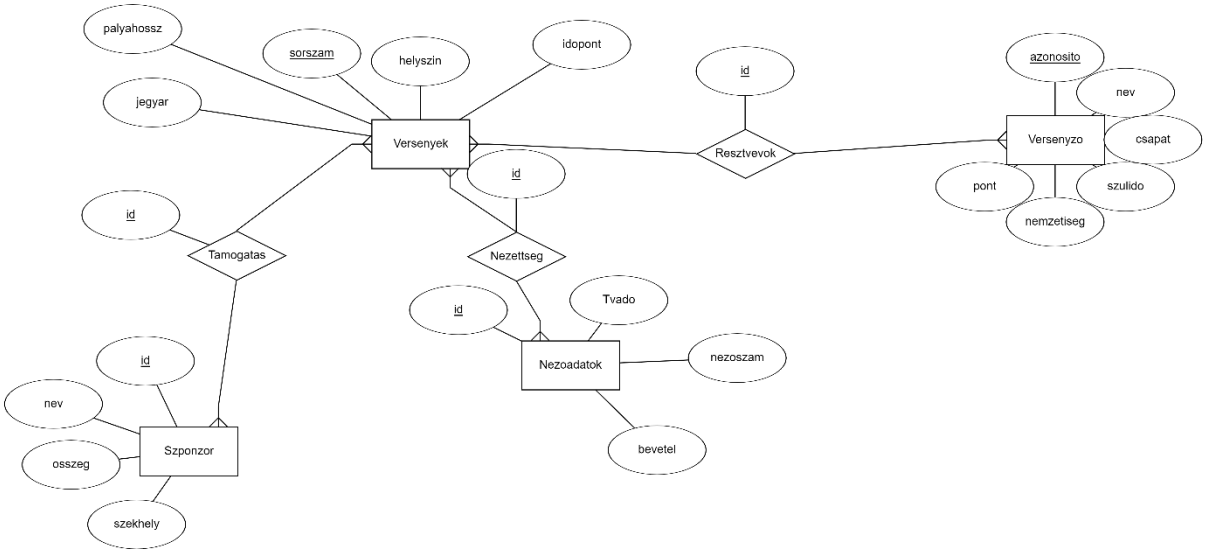
ACI3X3

Feladat

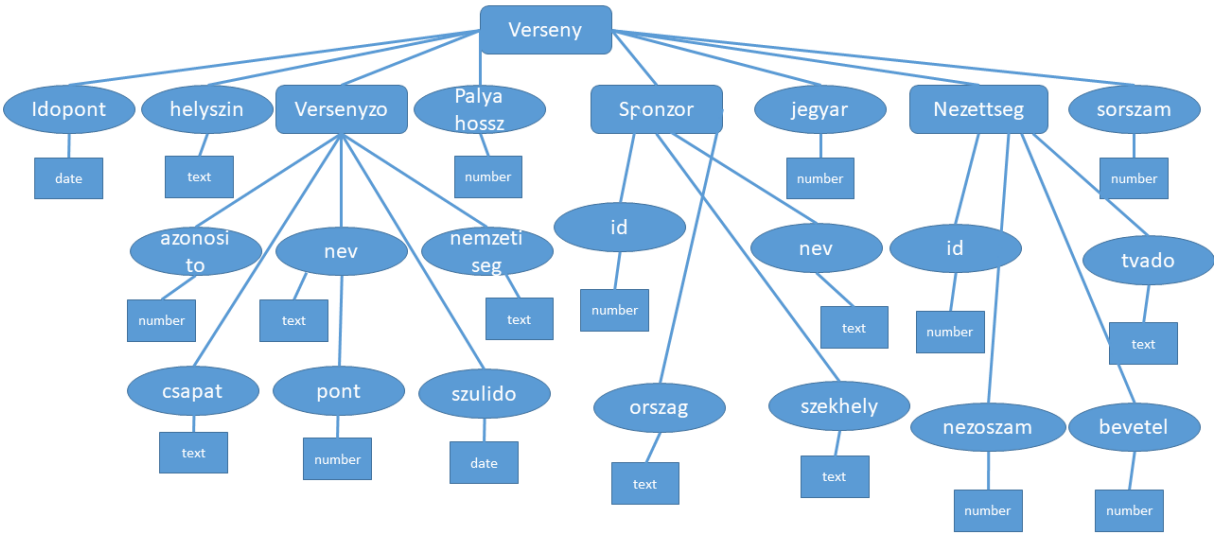
Feladatom egy F1 versenyeket tartalmazó adatbázis modell létrehozása volt .

XDM modellel valamint olvasható és szerkeszthető XML létrehozsával.

ER modell


palyahossz
sorszam
helyszin
idopont
jegyar
Versenyek
id
Resztvevok
azonosito
nev
Versenyzo
csapat
pont
szulido
nemzetiseg
id
Tamogatas
id
Nezettseg
id
Tvado
Nezoadatok
nezoszam
bevetel
id
nev
Szponzor
osszeg
szekhely

XDM modell


Verseny
Idopont
helyszin
Versenyzo
Palya hossz
Sponzor
jegyar
Nezettseg
sorszam
date
text
number
number
number
azonosito
nev
nemzetiseg
id
nev
id
tvado
number
text
text
number
text
number
text
csapat
pont
szulido
orszag
szekhely
nezoszam
bevetel
text
number
date
text
text
number
number

XML

```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="orszag"/>
    </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

```java
public class DomRead {

    public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException {
        File inputFile = new File("D:\\XMLaci3x3.xml");

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

        Element root = doc.getDocumentElement();
        root.normalize();

        printDocument(root, "");
    }

    public static void printDocument(Node root, String separator) {
        String nodename = root.getNodeName();
        if (!nodename.contains("text")) {
            System.out.println(separator + nodename);
        }
        separator += "  ";

        NodeList children = root.getChildNodes();

        for (int i = 0; i < children.getLength(); i++) {
            Node child = children.item(i);
            boolean isComplex = child.getTextContent().contains("\n");

            if (isComplex) {
                printDocument(child, separator);
            } else {
                System.out.print(separator + child.getNodeName());
                System.out.println(": " + child.getTextContent());
            }
        }
    }
}
```

DOM Modify

```java
public class DomModify {

    public static void main(String[] args) throws ParserConfigurationException, IOException, SAXException, TransformerException {
        File inputFile = new File("D:\\XMLaci3x3.xml");

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

        Element root = doc.getDocumentElement();
        root.normalize();

        Node beosztas = doc.getElementsByTagName("nemzetiseg").item(0);

        beosztas.setTextContent("Amerikai");

        Transformer transformer = TransformerFactory
                .newInstance()
                .newTransformer();
        DOMSource source = new DOMSource(doc);
        StreamResult result = new StreamResult(inputFile);
        transformer.transform(source, result);

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