

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Póczos Péter**  
Neptunkód: **LHS2S6**

## A feladat leírása:

A feladatom egy XML alapú adatbázis létrehozása volt melynek először az Entity Relationship modelljét kellett elkészítenem majd ezt leképezni egy XDM-modellje majd ezen modellek alapján létrehozni az XML és a hozzá tartozó XML sémát, valamint a feladat második részében az XML adatait egy DOM program segítségével beolvasni majd módosítani.

Adatszerkezetnek egy hozzám közel álló témát választottam. Formula 1 es versenyek, valamint csapatok és versenyzőik tárolására alkalmas adatszerkezetet hoztam létre. A szerkezetet több logikailag elkülöníthető egyedekre bontottam mely segít a tárolás optimalizálásában.

Az első egyed a Versenyző egyed mely eltárolja az adott pilóta rajtszámát, nevét, a szezonban elért pontjait, valamint hivatkozik a csapatra amelyiknek a tagja az adott versenyző.

A második egyed a Csapat egyed melyben eltárolom a csapat sorszámát, nevét és a csapat konstruktóri pontjait.

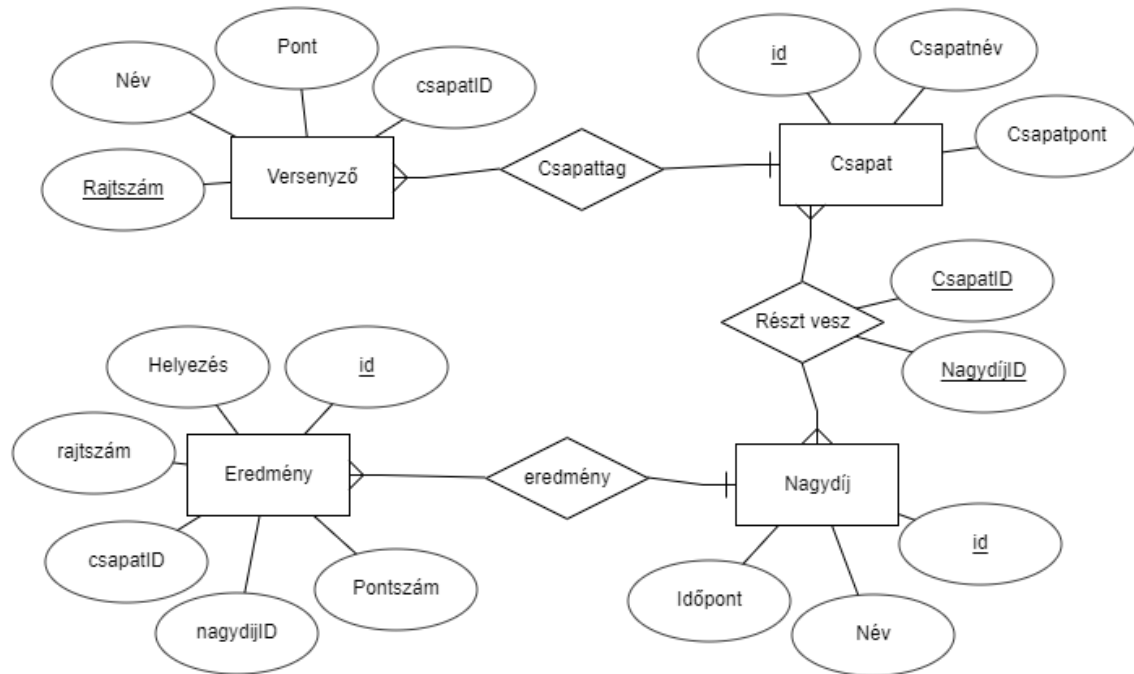
A harmadik egyed a nagydíj egyed, ahol eltárolom az adott nagydíj naptári sorszámát, a nagydíj nevét, valamint időpontját.

A negyedik és egyben utolsó egyed az Eredmény egyed, ami eltárolja a szezonban elért összes helyezést. Az adattagjai javarészt hivatkozásokból állnak másik egyedekre. Az egyed adattagjai eltárolják az eredmény sorszámát, az elért helyezést, a pilóta rajtszámát, a csapatának az azonosítóját, a nagydíj sorszámát, valamint az elért pontszámot.

A csapat és a nagydíj egyed között több-több kapcsolat áll ezért ott új egyedet kell létrehozni konvertálásnál, mely a két egyed közti kapcsolatot biztosító hivatkozásokat tartalmazza.

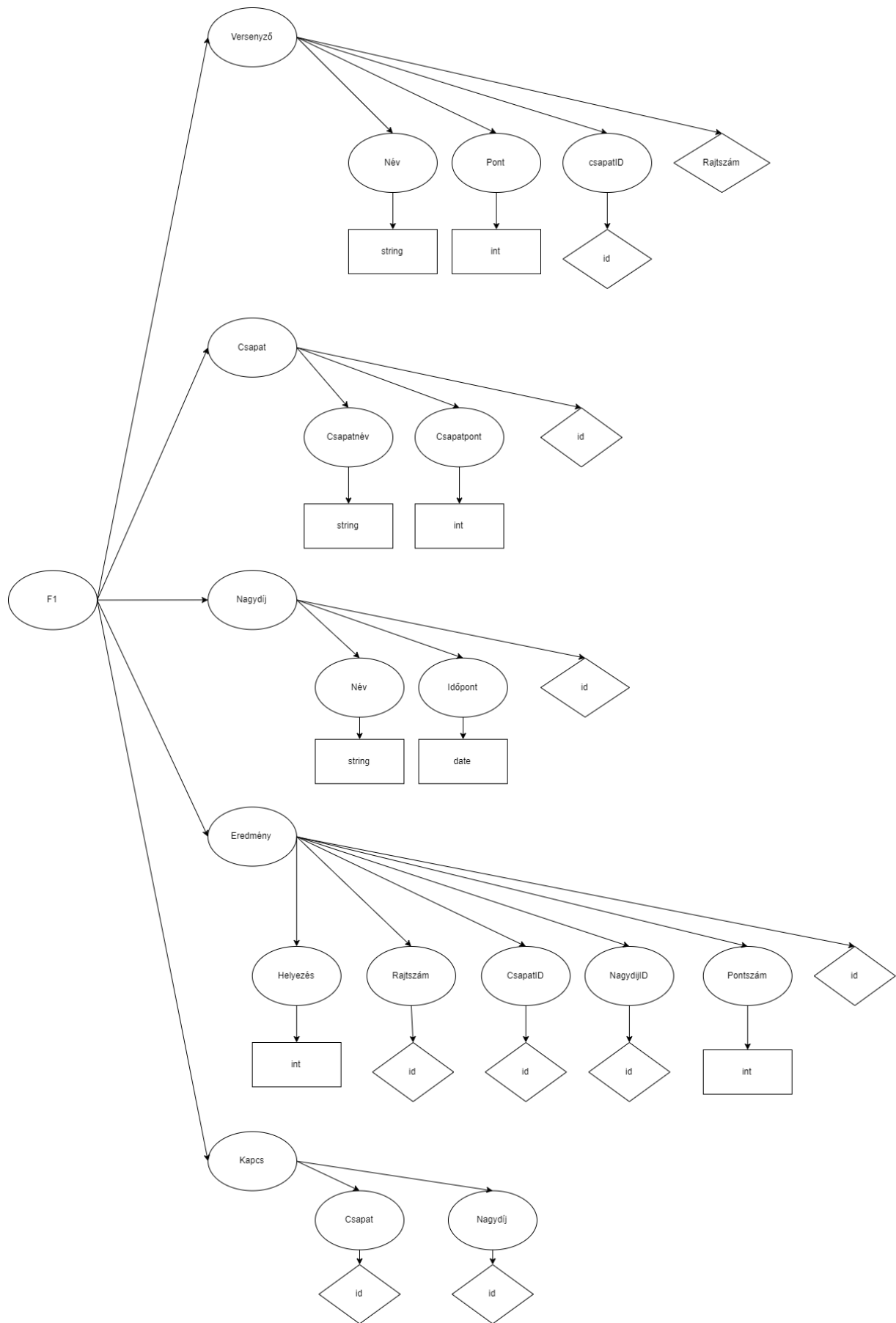
## 1. Feladat

### 1a) Az adatbázis ER-modell:



### **1b) Az adatbázis konvertálása XDM-modellre:**

Az ER modell XDM modellre alakítás során kisebb változtatásokat kellett végrehajtani. Be kellett vezetni egy gyökérelemet mely összekapcsolja az ER modell egyedeit, így egy szülőelemet létrehozva. Az ER modell egyedeiből elem lesz, a kulcs elemekből attribútum a tulajdonságokból pedig gyerekelem. Az elemeket ellipszis jelöli, a kulcs attribútumokat rombusz, valamint megjelenik a téglalap mely a gyerekelemek alatt jelzi, hogy az elemek adatot tartalmaznak. Így az ER modellből egy XDM hierarchikus modell jön létre.



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

```
<?xml version="1.0" encoding="UTF-8"?>
<f1
  xsi:noNamespaceSchemaLocation="file:///C:/Users/poczo/Desktop/xml%20beadando/f1.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <csapat>
    <id>1</id>
    <csapatnev>Mercedes AMG Petronas</csapatnev>
    <csapatpont>100</csapatpont>
  </csapat>
  <csapat>
    <id>2</id>
    <csapatnev>Scuderia Ferrari</csapatnev>
    <csapatpont>50</csapatpont>
  </csapat>
  <csapat>
    <id>3</id>
    <csapatnev>RedBull Racing Honda</csapatnev>
    <csapatpont>70</csapatpont>
  </csapat>
  <csapat>
    <id>4</id>
    <csapatnev>McLaren Renault</csapatnev>
    <csapatpont>40</csapatpont>
  </csapat>
  <eredmeny>
    <id>1</id>
    <nagydijid>1</nagydijid>
    <helyezes>1</helyezes>
    <rajtszam>44</rajtszam>
    <csapatid>1</csapatid>
    <pontszam>26</pontszam>
  </eredmeny>
  <eredmeny>
    <id>2</id>
    <nagydijid>1</nagydijid>
    <helyezes>2</helyezes>
    <rajtszam>77</rajtszam>
    <csapatid>1</csapatid>
    <pontszam>18</pontszam>
  </eredmeny>
```

```
<eredmeny>
  <id>3</id>
  <nagydijid>1</nagydijid>
  <helyezes>3</helyezes>
  <rajtszam>33</rajtszam>
  <csapatid>3</csapatid>
  <pontszam>15</pontszam>
</eredmeny>
<eredmeny>
  <id>4</id>
  <nagydijid>1</nagydijid>
  <helyezes>4</helyezes>
  <rajtszam>5</rajtszam>
  <csapatid>2</csapatid>
  <pontszam>12</pontszam>
</eredmeny>
<eredmeny>
  <id>5</id>
  <nagydijid>1</nagydijid>
  <helyezes>5</helyezes>
  <rajtszam>16</rajtszam>
  <csapatid>2</csapatid>
  <pontszam>10</pontszam>
</eredmeny>
<eredmeny>
  <id>6</id>
  <nagydijid>1</nagydijid>
  <helyezes>6</helyezes>
  <rajtszam>23</rajtszam>
  <csapatid>3</csapatid>
  <pontszam>8</pontszam>
</eredmeny>
<eredmeny>
  <id>7</id>
  <nagydijid>2</nagydijid>
  <helyezes>1</helyezes>
  <rajtszam>77</rajtszam>
  <csapatid>1</csapatid>
  <pontszam>25</pontszam>
</eredmeny>
<eredmeny>
  <id>8</id>
  <nagydijid>2</nagydijid>
  <helyezes>2</helyezes>
  <rajtszam>33</rajtszam>
  <csapatid>3</csapatid>
  <pontszam>19</pontszam>
</eredmeny>
```

```
<eredmeny>
  <id>9</id>
  <nagydijid>2</nagydijid>
  <helyezes>3</helyezes>
  <rajtszam>44</rajtszam>
  <csapatid>1</csapatid>
  <pontszam>15</pontszam>
</eredmeny>
<eredmeny>
  <id>10</id>
  <nagydijid>2</nagydijid>
  <helyezes>4</helyezes>
  <rajtszam>16</rajtszam>
  <csapatid>2</csapatid>
  <pontszam>12</pontszam>
</eredmeny>
<eredmeny>
  <id>11</id>
  <nagydijid>2</nagydijid>
  <helyezes>5</helyezes>
  <rajtszam>23</rajtszam>
  <csapatid>3</csapatid>
  <pontszam>10</pontszam>
</eredmeny>
<eredmeny>
  <id>12</id>
  <nagydijid>2</nagydijid>
  <helyezes>6</helyezes>
  <rajtszam>5</rajtszam>
  <csapatid>2</csapatid>
  <pontszam>8</pontszam>
</eredmeny>
<kapcs>
  <csapat>1</csapat>
  <nagydiij>1</nagydiij>
</kapcs>
<kapcs>
  <csapat>2</csapat>
  <nagydiij>1</nagydiij>
</kapcs>
<kapcs>
  <csapat>3</csapat>
  <nagydiij>1</nagydiij>
</kapcs>
<kapcs>
  <csapat>1</csapat>
  <nagydiij>2</nagydiij>
</kapcs>
```



```
<kapcs>
  <csapat>2</csapat>
  <nagydi>2</nagydi>
</kapcs>
<kapcs>
  <csapat>3</csapat>
  <nagydi>2</nagydi>
</kapcs>
<nagydi>
  <id>1</id>
  <nev>Ausztrál nagydíj</nev>
  <idopont>2019-03-17</idopont>
</nagydi>
<nagydi>
  <id>2</id>
  <nev>Bahrain nagydíj</nev>
  <idopont>2019-03-31</idopont>
</nagydi>
<versenyz>
  <rajtszam>4</rajtszam>
  <nev>Lando Norris</nev>
  <pont>20</pont>
  <csapatid>4</csapatid>
</versenyz>
<versenyz>
  <rajtszam>5</rajtszam>
  <nev>Sebastian Vettel</nev>
  <pont>20</pont>
  <csapatid>2</csapatid>
</versenyz>
<versenyz>
  <rajtszam>16</rajtszam>
  <nev>Charles Leclerc</nev>
  <pont>30</pont>
  <csapatid>2</csapatid>
</versenyz>
<versenyz>
  <rajtszam>23</rajtszam>
  <nev>Alexander Albon</nev>
  <pont>20</pont>
  <csapatid>3</csapatid>
</versenyz>
<versenyz>
  <rajtszam>33</rajtszam>
  <nev>Max Verstappen</nev>
  <pont>50</pont>
  <csapatid>3</csapatid>
</versenyz>
```

```
<versenyzo>
  <rajtszam>44</rajtszam>
  <nev>Lewis Hamilton</nev>
  <pont>50</pont>
  <csapatid>1</csapatid>
</versenyzo>
<versenyzo>
  <rajtszam>55</rajtszam>
  <nev>Carlos Sainz</nev>
  <pont>20</pont>
  <csapatid>4</csapatid>
</versenyzo>
<versenyzo>
  <rajtszam>77</rajtszam>
  <nev>Valtteri Bottas</nev>
  <pont>50</pont>
  <csapatid>1</csapatid>
</versenyzo>
</f1>
```

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

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="f1">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="csapat" maxOccurs="10">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id">
                <xs:simpleType>
                  <xs:restriction base="xs:int">
                    <xs:minInclusive value="1"/>
                    <xs:maxInclusive value="10"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="csapatnev" nillable="true" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="200"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="csapatpont" nillable="true" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:int">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="1000"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
          <xs:key name="csapat_PrimaryKey_1">
            <xs:selector xpath="."/>
            <xs:field xpath="id"/>
          </xs:key>
        </xs:element>
        <xs:element name="eredmeny" maxOccurs="1000">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="id">
                <xs:simpleType>
                  <xs:restriction base="xs:int">
                    <xs:minInclusive value="1"/>
                    <xs:maxInclusive value="1000"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

    </xs:simpleType>
  </xs:element>
  <xs:element name="nagydiijd" nillable="true" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="25"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="helyezes" nillable="true" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="20"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="rajtszam" nillable="true" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="99"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="csapatid" nillable="true" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="1"/>
        <xs:maxInclusive value="10"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="pontszam" nillable="true" minOccurs="0">
    <xs:simpleType>
      <xs:restriction base="xs:int">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="26"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:sequence>
</xs:complexType>
<xs:key name="eredmeny_PrimaryKey_1">
  <xs:selector xpath="."/>
  <xs:field xpath="id"/>
</xs:key>
</xs:element>
<xs:element name="kapcs" maxOccurs="unbounded">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="csapat" nillable="true" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:minInclusive value="1"/>
          <xs:maxInclusive value="10"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="nagydij" nillable="true" minOccurs="0">
      <xs:simpleType>
        <xs:restriction base="xs:int">
          <xs:minInclusive value="1"/>
          <xs:maxInclusive value="25"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="nagydij" maxOccurs="25">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="id">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="25"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="nev" nillable="true" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="300"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="idopont" nillable="true" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:date">
            <xs:minInclusive value="2019-01-01"/>
            <xs:maxInclusive value="2019-12-31"/>
            <xs:pattern value="\p{Nd}{4}-\p{Nd}{2}-\p{Nd}{2}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

<xs:key name="nagydij_PrimaryKey_1">
  <xs:selector xpath="."/>
  <xs:field xpath="id"/>
</xs:key>
</xs:element>
<xs:element name="versenyzo" maxOccurs="20">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="rajtszam">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="99"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="nev" nillable="true" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="100"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="pont" nillable="true" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="500"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="csapatid" nillable="true" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:int">
            <xs:minInclusive value="1"/>
            <xs:maxInclusive value="10"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:key name="versenyzo_PrimaryKey_1">
    <xs:selector xpath="."/>
    <xs:field xpath="rajtszam"/>
  </xs:key>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

## 2. feladat: DOM

### 2a) Adatolvasás – DOM\_Adatolvasas\_LHS2S6.java

```
import java.io.File;

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;

public class DOM_Adatolvasas_LHS2S6 {

    public static void main(String[] args) {
        try {
            // Az XML fájl beolvasása:
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document xmlfile = db.parse(new File("C:\\Users\\poczo\\Desktop\\xml
beadando\\f1.xml"));

            // Az XML dokumentum gyökérelemének lekérdezése és kiírása:
            Element gyokerelem = xmlfile.getDocumentElement();
            System.out.println("Az XML dokumentum gyökéreleme: " +
gyokerelem.getNodeName());

            // A gyerekelemek megkeresése majd NodeList-be rakása:
            NodeList gyerekelemek = gyokerelem.getChildNodes();

            // A gyerekelemek lista feldolgozása:
            for (int i = 0; i < gyerekelemek.getLength(); i++) {

                // Aktuális node lekérése:
                Node curNode = gyerekelemek.item(i);

                // Az aktuális node elemvizsgálata:
                if (curNode.getNodeType() == Node.ELEMENT_NODE) {

                    // Az aktuális node elemmé alakítása:
                    Element curElement = (Element) curNode;

                    // Kiírás:
                    System.out.println("Aktuális elem: " + curElement.getTagName());

                    // Egy újabb node ba lekérjük az aktuális elem gyerekelemeit:
                    Node cur = curElement.getFirstChild();
                }
            }
        }
    }
}
```

```

// Ciklus ami addig megy amíg talál a cur node-dal egy szinten lévő (testvér)
// node-okat:
while (cur != null) {
    // ha az aktuális node ismét elem akkor elemmé alakítjuk:
    if (cur.getNodeType() == Node.ELEMENT_NODE) {
        Element current = (Element) cur;
        // Majd kiírjuk az elem nevét és a tartalmát:
        System.out.println(" " + current.getTagName() + ": " +
current.getTextContent());
    }
    // a következő testvér node lekérése, ha nincs akkor null-t ad és a ciklus kilép
    cur = cur.getNextSibling();
}

    System.out.println();
}
}

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

```



## 2b) Adatmódosítás – DOM\_Adatmodositas\_LHS2S6.java

```
import java.io.File;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
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.Node;
import org.w3c.dom.NodeList;

public class DOM_Adatmodositas_LHS2S6 {

    public static void main(String[] args) {

        // A második nagydíj elmaradt, ezért az adatait frissíteni kell új helyszínre és
        // időpontra

        try {
            // Az XML fájl beolvasása:
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document xmlfile = db.parse(new File("C:\\Users\\poczo\\Desktop\\xml
beadando\\f1.xml"));

            Element gyokerelem = xmlfile.getDocumentElement();
            System.out.println("Az XML dokumentum gyökéreleme: " +
            gyokerelem.getNodeName());

            // A gyerekelemek megkeresése majd NodeList-be rakása:
            NodeList gyerekelemek = gyokerelem.getChildNodes();

            // A gyerekelemek lista feldolgozása:
            for (int i = 0; i < gyerekelemek.getLength(); i++) {

                // Aktuális node lekérése:
                Node curNode = gyerekelemek.item(i);

                // Az aktuális node elemvizsgálata:
                if (curNode.getNodeType() == Node.ELEMENT_NODE) {

                    // Az aktuális node elemmé alakítása:
                    Element curElement = (Element) curNode;
```

```

        if (curElement.getTagName() == "nagydi") {

            // Egy újabb node ba lekérjük az aktuális elem gyerekelemeit:
            Node cur = curElement.getFirstChild();

            // Ciklus ami addig megy amíg talál a cur node-dal egy szinten lévő (testvér)
            // node-okat:
            while (cur != null) {
                // ha az aktuális node ismét elem akkor elemmé alakítjuk:
                if (cur.getNodeType() == Node.ELEMENT_NODE) {
                    Element current = (Element) cur;
                    // vizsgáljuk a fent megadott feltételt majd módosítjuk az adatokat
                    if (current.getTagName() == "id" &&
Integer.parseInt(current.getTextContent()) == 2) {
                        cur = cur.getNextSibling();
                        cur = cur.getNextSibling();
                        cur.setTextContent("Olasz nagydíj");
                        cur = cur.getNextSibling();
                        cur = cur.getNextSibling();
                        cur.setTextContent("2019-04-07");

                    }
                }
                // a következő testvér node lekérése, ha nincs akkor null-t ad és a ciklus kilép
                cur = cur.getNextSibling();
            }

        }
    }
}

TransformerFactory tff = TransformerFactory.newInstance();
Transformer tf = tff.newTransformer();
DOMSource ds = new DOMSource(xmlfile);
StreamResult sr = new StreamResult(new File("C:\\Users\\poczo\\Desktop\\xml
beadando\\flout.xml"));
tf.transform(ds, sr);

System.out.println("Módosítás sikeres. A módosított xml fájl mentve az flout.xml
fájlba");

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

```