

Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Fesztivál és közvélemény-kutatás

Készítette: **Szabó András**

Neptunkód: **V9RN7C**

Dátum: **2022.12.04.**

Tartalomjegyzék

A feladat leírása:	3
Egyedek	3
Fesztivál	3
Tulajdonos	3
Fellépő	3
Bár	3
Alkalmazott	3
Résztevő	3
Közvélemény	3
Nyelvtudás	3
Kapcsolatok	3
1. feladat	4
1a) Az adatbázis ER modell	4
1b) Az adatbázis konvertálása XDM modellre	4
1c) Az XDM modell alapján XML dokumentum készítése:	4
1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok, ref, key, keyref, speciális elemek)	7
2. feladat	11
Dom Read	11
Dom Query	13
Első lekérdezés	14
Második lekérdezés	15
Harmadik lekérdezés	15
Negyedik lekérdezés	16
Ötödik lekérdezés	16
Dom Modify	17
Elemérték módosítás	18
Attribútum módosítás	18
Gyerekelem törlés	19
Gyerekelem hozzáadás	20
Attribútum hozzáadás	20

A feladat leírása:

Egy fesztivál és hozzátartozó egyedek az XML környezetbe implementálásáról és hozzátartozó Java programokról szól a féléves feladatom. Szerepel még XSD validáció is. 7 darab egyed szerepel az adatbázis és több-több kapcsolat miatt még egy egyed.

Egyedek

Fesztivál

Egy fesztivált jelöl ez az egyed, sok másik egyed hozzákapcsolódik, egy darab lehet belőle, attribútumai: **fesz_id** (kulcs), név, helyszín és év. Kapcsolódik hozzá a fellépő, résztvevő, bár és tulajdonos egyed.

Tulajdonos

Egy darab lehet belőle, attribútumai: **t_id** (kulcs), cégnév, adószám, alapítás, bevétel. Idegen kulcs a **fesz_t**.

Fellépő

Több darab lehet belőle, attribútumai: **fel_id** (kulcs), név, ország (melyik országból származik) és díj (mennyi pénzt kap fellépésért). Idegen kulcs a **fesz_fel**.

Bár

Attribútumai: **b_id** (kulcs), név (a bár neve), ital (többször szerepelhet), cégnév (az üzemeltető neve). Idegen kulcs a **fesz_b** és kapcsolódik a bár egyedhez az alkalmazott egyed.

Alkalmazott

Attribútumai: **a_id** (kulcs), név, fizetés, műszak (többször szereplhet). Idegen kulcs a **b_a**.

Résztvevő

Attribútumai: **r_id** (kulcs), név, lakcím (összetett tulajdonság: város, utca és házzámból áll), diákigazolvány (boolean). Idegen kulcs a **fesz_r** és hozzákapcsolódik a közvélemény kapcsoló egyed.

Közvélemény

Kapcsoló egyed, két idegenkulccsal rendelkezik: résztvevő: **r_k_r** nyelvtudás: **ny_k_ny**. Egy attribútum: kérdező (aki elvégezte a kutatást).

Nyelvtudás

Attribútumai: **ny_id** (kulcs), nyelv (melyik nyelven beszél), szint. Hozzákapcsolódik a közvélemény kapcsoló egyed.

Kapcsolatok

Fesz_T: Egy-egy kapcsolat a fesztivál és a tulajdonos között.

Fesz_Fel: Egy-több kapcsolat a fesztivál és fellépő között.

Fesz_B: Egy-több kapcsolat a fesztivál és a bár között.

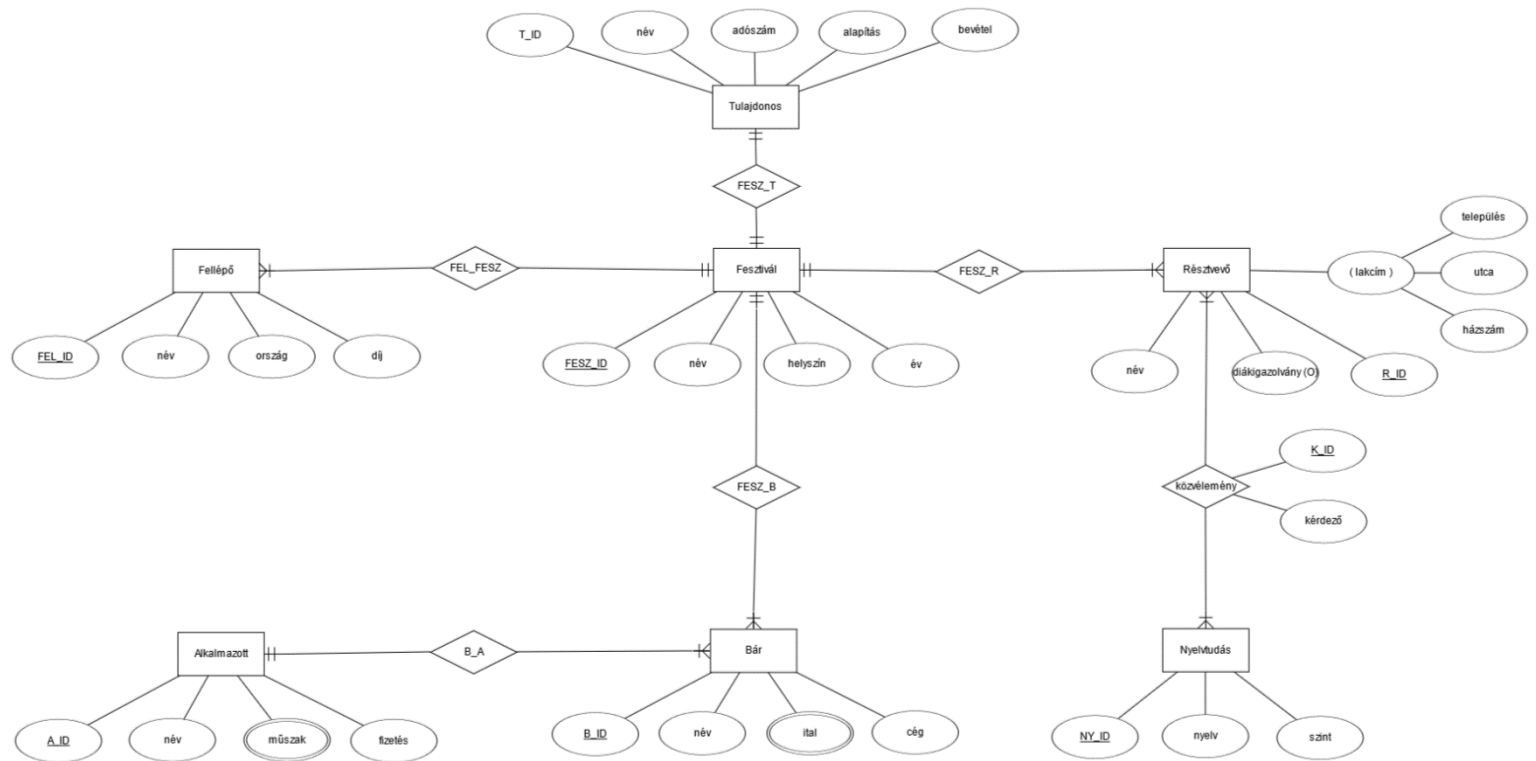
B_A: Egy-több kapcsolat a bár és az alkalmazott között.

Fesz_R: Egy-több kapcsolat a fesztivál és a résztvevő között.

Közvélemény: Több-több kapcsolat a résztvevő és nyelvtudás között.

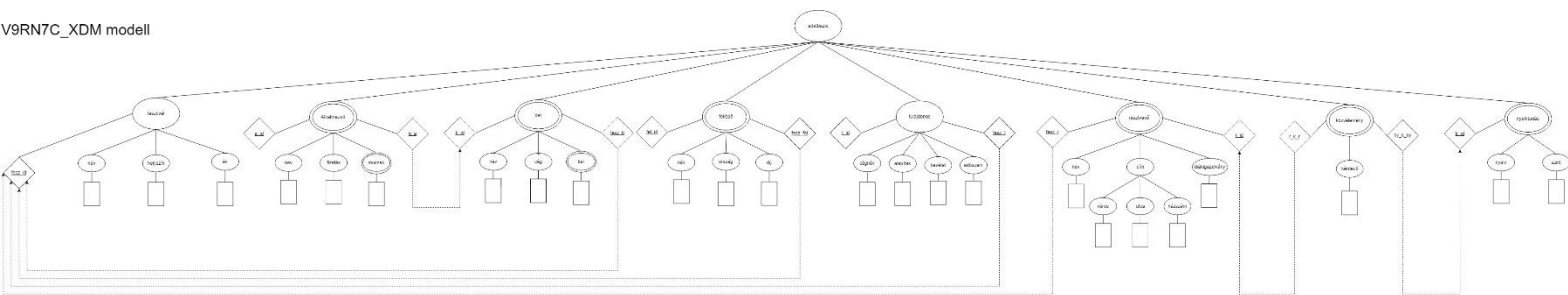
1. feladat

1a) Az adatbázis ER modell



1b) Az adatbázis konvertálása XDM modellre

V9RN7C_XDM modell



1c) Az XDM modell alapján XML dokumentum készítése:

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

```
<adatbázis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:noNamespaceSchemaLocation="V9RN7Cxml.xsd">
```

```
  <!--Tulajdonos -->
```

```
  <tulajdonos t_id="sgh-32h-j2" feszt_t="dh2-82k-ds">
```

```
    <cegnev>Providence Equity Partners</cegnev>
```

```
    <adoszam>312316312</adoszam>
```

```

        <alapitas>1989</alapitas>
        <bevetel>31000000000</bevetel>
    </tulajdonos>

    <!--Fesztivál -->
    <fesztival fesz_id="dh2-82k-ds">
        <nev>Balaton Sound</nev>
        <helyszin>Zamárdi</helyszin>
        <ev>2022</ev>
    </fesztival>

    <!--Fellépő -->
    <fellepo fel_id="312-sda-ma" fesz_fel="dh2-82k-ds">
        <nev>Paul Kalkbrenner</nev>
        <orszag>Németország</orszag>
        <di>3000000</di>
    </fellepo>
    <fellepo fel_id="wdn-w72-as" fesz_fel="dh2-82k-ds">
        <nev>Alesso</nev>
        <orszag>Svédország</orszag>
        <di>6000000</di>
    </fellepo>
    <fellepo fel_id="km7-h7g-2s" fesz_fel="dh2-82k-ds">
        <nev>Martin Garrix</nev>
        <orszag>Hollandia</orszag>
        <di>7000000</di>
    </fellepo>

    <!--Bár -->
    <bar b_id="ads-h2w-w2" fesz_b="dh2-82k-ds">
        <nev>Dreher bár</nev>
        <ital>Sör</ital>
        <ital>Vodka</ital>
        <ital>Whisky</ital>
        <cegnev>Dreher Sörgyárak Zrt.</cegnev>
    </bar>
    <bar b_id="mub-322-ds" fesz_b="dh2-82k-ds">
        <nev>Jana Aqua bár</nev>
        <ital>Ásványvíz</ital>
        <ital>Jägermeister</ital>
        <ital>Vodka</ital>
        <cegnev>AQUA Kft.</cegnev>
    </bar>
    <bar b_id="mnd-hb7-20" fesz_b="dh2-82k-ds">
        <nev>Foods and Drinks</nev>
        <ital>Vodka</ital>
        <ital>Energiaital</ital>
    <ital>Gin</ital>
        <cegnev>Zamardi Foods Kft.</cegnev>

```

```

</bar>

<!--Alkalmazott -->
<alkalmazott a_id="mnh-hgf-j7" b_a="mub-322-ds">
  <nev>Dobos Sára</nev>
  <fizetes>321200</fizetes>
  <muszak>péntek-szombat</muszak>
  <muszak>szerda</muszak>
</alkalmazott>
<alkalmazott a_id="sha-dw7-wa" b_a="mnd-hb7-20">
  <nev>Papp Márton</nev>
  <fizetes>400000</fizetes>
  <muszak>szerda-péntek</muszak>
  <muszak>szombat</muszak>
</alkalmazott>
<alkalmazott a_id="dsa-wa6-42" b_a="mnd-hb7-20">
  <nev>Varga István</nev>
  <fizetes>410000</fizetes>
  <muszak>szombat</muszak>
  <muszak>csütörtök</muszak>
</alkalmazott>

<!--Résztvevő -->
<resztvevo r_id="wda-312-s2" fész_r="dh2-82k-ds">
  <nev>Pataki Patrícia</nev>
  <diakigazolvany>true</diakigazolvany>
  <lakcim>
    <telepules>Szombathely</telepules>
    <utca>Tolnai Lajos utca</utca>
    <hazszam>23</hazszam>
  </lakcim>
</resztvevo>
<resztvevo r_id="daw-29k-ws" fész_r="dh2-82k-ds">
  <nev>Halász Milán</nev>
  <diakigazolvany>true</diakigazolvany>
  <lakcim>
    <telepules>Miskolc</telepules>
    <utca>Herend utca</utca>
    <hazszam>100</hazszam>
  </lakcim>
</resztvevo>
<resztvevo r_id="dwj-82w-sa" fész_r="dh2-82k-ds">
  <nev>Somogyi Linda</nev>
  <diakigazolvany>false</diakigazolvany>
  <lakcim>
    <telepules>Karcag</telepules>
    <utca>Kardhegy utca</utca>
    <hazszam>33</hazszam>
  </lakcim>

```

```

</resztvevo>

<!--Közvélemény (kapcsolótábla) -->
<kozvelemen y_r_k_r="dwj-82w-sa" ny_k_ny="dag-213-sd">
  <kerdezo>Szűcs Vilmos</kerdezo>
</kozvelemen y>
<kozvelemen y_r_k_r="dwj-82w-sa" ny_k_ny="kjg-mhz-2d">
  <kerdezo>Szűcs Vilmos</kerdezo>
</kozvelemen y>
<kozvelemen y_r_k_r="daw-29k-ws" ny_k_ny="sk6-wss-31">
  <kerdezo>Jónás Judit</kerdezo>
</kozvelemen y>
<kozvelemen y_r_k_r="wda-312-s2" ny_k_ny="jnu-21g-bs">
  <kerdezo>Szűcs Vilmos</kerdezo>
</kozvelemen y>

<!--Nyelvtudás -->
<nyelvtudas ny_id="dag-213-sd">
  <nyelv>angol</nyelv>
  <szint>közép</szint>
</nyelvtudas>
<nyelvtudas ny_id="sk6-wss-31">
  <nyelv>angol</nyelv>
  <szint>emelt</szint>
</nyelvtudas>
<nyelvtudas ny_id="kjg-mhz-2d">
  <nyelv>német</nyelv>
  <szint>kezdő</szint>
</nyelvtudas>
<nyelvtudas ny_id="jnu-21g-bs">
  <nyelv>francia</nyelv>
  <szint>közép</szint>
</nyelvtudas>
</adatbazis>

```

1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok, ref, key, keyref, speciális elemek)

```

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

  <!--Egyszerű típusok-->
  <xs:simpleType name="hazszam_type" >
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{1,3}(A|a|B|b|C|c|D|d)?" />
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="utca_type" >

```

```

    <xs:restriction base="xs:string">
        <xs:pattern value="[a-zA-Z]{3,}( utca| út| tér)" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="szint_type" >
    <xs:restriction base="xs:string">
        <xs:pattern value="(kezdő|közép|emelt)" />
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="fizetes_type">
    <xs:restriction base="xs:positiveInteger">
        <xs:minInclusive value="200000" />
    </xs:restriction>
</xs:simpleType>

<!-- Komplex típusok-->
<xs:complexType name="tulajdonos_type">
    <xs:sequence>
        <xs:element name="cegnev" />
        <xs:element name="adoszam"/>
        <xs:element name="alapitas" />
        <xs:element name="bevetel" />
    </xs:sequence>
    <xs:attribute name="t_id" use="required" />
    <xs:attribute name="fesz_t" use="required" />
</xs:complexType>

<xs:complexType name="fesztival_type">
    <xs:sequence>
        <xs:element name="nev" />
        <xs:element name="helyszin" />
        <xs:element name="ev" />
    </xs:sequence>
    <xs:attribute name="fesz_id" use="required" />
</xs:complexType>

<xs:complexType name="fellepo_type">
    <xs:sequence>
        <xs:element name="nev" />
        <xs:element name="orszag" />
        <xs:element name="dij"/>
    </xs:sequence>
    <xs:attribute name="fel_id" use="required" />
    <xs:attribute name="fesz_fel" use="required" />
</xs:complexType>

<xs:complexType name="bar_type">

```



```

<xs:sequence>
  <xs:element name="nev" type="xs:string" />
  <xs:element name="ital" maxOccurs="unbounded" />
  <xs:element name="cegnev" />
</xs:sequence>
<xs:attribute name="b_id" use="required" />
<xs:attribute name="fesz_b" use="required" />
</xs:complexType>

<xs:complexType name="alkalmazott_type">
  <xs:sequence>
    <xs:element name="nev" />
    <xs:element name="fizetes" type="fizetes_type" />
    <xs:element name="muszak" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="a_id" use="required" />
  <xs:attribute name="b_a" use="required" />
</xs:complexType>

<xs:complexType name="lakcim_type">
  <xs:sequence>
    <xs:element name="telepules" type="xs:string"/>
    <xs:element name="utca" type="utca_type"/>
    <xs:element name="hazszam" type="hazszam_type" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="resztvevo_type">
  <xs:sequence>
    <xs:element name="nev" />
    <xs:element name="diakigazolvany" type="xs:boolean" />
    <xs:element name="lakcim" type="lakcim_type"></xs:element>
  </xs:sequence>
  <xs:attribute name="r_id" use="required" />
  <xs:attribute name="fesz_r" use="required" />
</xs:complexType>

<xs:complexType name="kozvelemen_type">
  <xs:sequence>
    <xs:element name="kerdezo" />
  </xs:sequence>
  <xs:attribute name="r_k_r" use="required" />
  <xs:attribute name="ny_k_ny" use="required" />
</xs:complexType>

<xs:complexType name="nyelvtudas_type">
  <xs:sequence>
    <xs:element name="nyelv" type="xs:string"/>
    <xs:element name="szint" type="szint_type"/>
  </xs:sequence>

```

```

        </xs:sequence>
        <xs:attribute name="ny_id" use="required" />
    </xs:complexType>

    <!-- Gyökérelemtől az egyedek felhasználása -->
    <xs:element name="adatbazis">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="tulajdonos" type="tulajdonos_type"/>
                <xs:element name="fesztival" type="fesztival_type"/>
                <xs:element name="fellepo" type="fellepo_type"
maxOccurs="unbounded" />
                <xs:element name="bar" type="bar_type" maxOccurs="unbounded" />
                <xs:element name="alkalmazott" type="alkalmazott_type"
maxOccurs="unbounded" />
                <xs:element name="resztvevo" type="resztvevo_type"
maxOccurs="unbounded" />
                <xs:element name="kozvelemen" type="kozvelemen_type"
maxOccurs="unbounded" />
                <xs:element name="nyelvtudas" type="nyelvtudas_type"
maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>

    <!-- Elsődleges kulcsok-->
    <xs:unique name="unique_tulajdonos">
        <xs:selector xpath="tulajdonos" />
        <xs:field xpath="@fesz_t" />
    </xs:unique>

    <xs:key name="fesztival_kulcs">
        <xs:selector xpath="fesztival" />
        <xs:field xpath="@fesz_id" />
    </xs:key>

    <xs:key name="fellepo_kulcs">
        <xs:selector xpath="fellepo" />
        <xs:field xpath="@fel_id" />
    </xs:key>

    <xs:key name="bar_kulcs">
        <xs:selector xpath="bar" />
        <xs:field xpath="@b_id" />
    </xs:key>

    <xs:key name="alkalmazott_kulcs">
        <xs:selector xpath="alkalmazott" />
        <xs:field xpath="@a_id" />
    </xs:key>

```

```

<xs:key name="resztvevo_kulcs">
  <xs:selector xpath="resztvevo" />
  <xs:field xpath="@r_id" />
</xs:key>

<xs:key name="nyelvtudas_kulcs">
  <xs:selector xpath="nyelvtudas" />
  <xs:field xpath="@ny_id" />
</xs:key>

<!-- Idegen kulcsok-->
<xs:keyref refer="fesztival_kulcs" name="fellepo_idegens_kulcs">
  <xs:selector xpath="fellepo" />
  <xs:field xpath="@feszt_fel" />
</xs:keyref>

<xs:keyref refer="fesztival_kulcs" name="bar_idegens_kulcs">
  <xs:selector xpath="bar" />
  <xs:field xpath="@feszt_b" />
</xs:keyref>

<xs:keyref refer="bar_kulcs" name="alkalmazott_idegens_kulcs">
  <xs:selector xpath="alkalmazott" />
  <xs:field xpath="@b_a" />
</xs:keyref>

<xs:keyref refer="fesztival_kulcs" name="resztvevo_idegens_kulcs">
  <xs:selector xpath="resztvevo" />
  <xs:field xpath="@feszt_r" />
</xs:keyref>

<xs:keyref refer="resztvevo_kulcs" name="resztvevo_k_idegens_kulcs">
  <xs:selector xpath="kozvelemenye" />
  <xs:field xpath="@r_k_r" />
</xs:keyref>

<xs:keyref refer="nyelvtudas_kulcs" name="nyelvtudas_idegens_kulcs">
  <xs:selector xpath="kozvelemenye" />
  <xs:field xpath="@ny_k_ny" />
</xs:keyref>
</xs:element>
</xs:schema>

```

2. feladat

Dom Read

```
package hu.domparse.v9rn7c;
```

```
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;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;

class DomReadV9RN7C {
    public static void main(String[] args) throws IOException,
ParserConfigurationException, SAXException {
        //Xml fájl megnyitása és DOM inicializálása
        File xmlFile = new File("V9RN7Cxml.xml");
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        //Node listát csinálás a kiválasztott elem gyerekelemeiből
        NodeList nList = doc.getElementsByTagName("bar");
        //Bejárja a gyerekelemeket
        for (int i = 0; i < nList.getLength(); i++) {
            //Megkönnyíti a különböző node-ok megkülönböztetését ez a sor
            Node nNode = nList.item(i);
            //Gyökérelem neve
            System.out.println("\nCurrent Element: " + nNode.getNodeName());

            //Aktuális node elemmé konvertálás
            Element elem = (Element) nNode;
            //Attribútumok kigyűjtése
            String b_id = elem.getAttribute("b_id");
            String fész_b = elem.getAttribute("fész_b");
            System.out.println("Bár kódja: " + b_id);
            System.out.println("Fesztivál kódja: " + fész_b);

            //Gyerekelemek kiválasztása és megfelelő adattípus szerint konvertálja
            Node node1 = elem.getElementsByTagName("nev").item(0);
            String nev = node1.getTextContent();

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

            System.out.println("Bár neve: " + nev);
            for (int j = 0; j < 3; j++) {
                Node node2 = elem.getElementsByTagName("ital").item(j);
                String ital = node2.getTextContent();
            }
        }
    }
}

```

```

        System.out.println("Ital neve: " + ital);
    }
    System.out.println("Cég neve: " + cegnev);
}
}
}

```

Eredmény:

Current Element: bar

Bár kódja: ads-h2w-w2

Fesztivál kódja: dh2-82k-ds

Bár neve: Dreher bár

Ital neve: Sör

Ital neve: Vodka

Ital neve: Whisky

Cég neve: Dreher Sörgyárak Zrt.

Current Element: bar

Bár kódja: mub-322-ds

Fesztivál kódja: dh2-82k-ds

Bár neve: Jana Aqua bár

Ital neve: Ásványvíz

Ital neve: Jägermeister

Ital neve: Cognac

Cég neve: AQUA Kft.

Current Element: bar

Bár kódja: mnd-hb7-20

Fesztivál kódja: dh2-82k-ds

Bár neve: Foods and Drinks

Ital neve: Vodka

Ital neve: Energiail

Ital neve: Gin

Cég neve: Zamardi Foods Kft.

Dom Query

```
package hu.domparse.v9rn7c;
```

```
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;
```

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.parsers.ParserConfigurationException;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```

class DomQueryV9RN7C {
    public static void main(String[] args) throws ParserConfigurationException,
IOException, SAXException {
        //Xml fájl megnyitása és DOM inicializálása
        File xmlFile = new File("V9RN7Cxml.xml");
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);
        doc.getDocumentElement().normalize();

        System.out.println("Azoknak az alkalmazottak neve, akik többet kapnak,
mint 400.000:");
        getBySalary(doc);
        System.out.println("Azoknak a bároknak a neve, akik árulnak vodkát:");
        getIfContainsSpecificDrink(doc);
        System.out.println("Azok a fellépőknek a neve, akik német
állampolgárok:");
        getByCountryName(doc);
        System.out.println("Azoknak az alkalmazottnak neve, akik dolgoztak
szombaton:");
        getByShift(doc);
        System.out.println("Azoknak a résztvevőknek a neve, akik diákok:");
        getByBoolean(doc);
    }
}

```

Első lekérdezés

```

private static void getByBoolean(Document doc) {
    NodeList nList = doc.getElementsByTagName("resztvevo");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element elem = (Element) nNode;

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

        Node node2 = elem.getElementsByTagName("diakigazolvany").item(0);
        Boolean diak = Boolean.parseBoolean(node2.getTextContent());
        if (diak)
            System.out.println('\t' + nev);
    }
}

```

Eredmény:

Azoknak a résztvevőknek a neve, akik diákok:

Pataki Patrícia

Halász Milán

Második lekérdezés

```
private static void getByShift(Document doc) {
    NodeList nList = doc.getElementsByTagName("alkalmazott");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element elem = (Element) nNode;

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

        Node node2 = elem.getElementsByTagName("muszak").item(0);
        String muszak1 = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("muszak").item(1);
        String muszak2 = node3.getTextContent();

        if (muszak1.contains("szombat") || muszak2.contains("szombat"))
            System.out.println('\t' + nev);
    }
}
```

Eredmény:

Azoknak az alkalmazottnak neve, akik dolgoztak szombaton:

Dobos Sára
Papp Márton
Varga István

Harmadik lekérdezés

```
private static void getByCountryName(Document doc) {
    NodeList nList = doc.getElementsByTagName("fellepo");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element elem = (Element) nNode;

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

        Node node2 = elem.getElementsByTagName("ország").item(0);
        String orszag = node2.getTextContent();

        if (orszag.equalsIgnoreCase("németország"))
            System.out.println('\t' + nev);
    }
}
```

Eredmény:

Azok a fellepőknek a neve, akik német állampolgárok:

Paul Kalkbrenner

Negyedik lekérdezés

```
private static void getIfContainsSpecificDrink(Document doc) {
    NodeList nList = doc.getElementsByTagName("bar");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element elem = (Element) nNode;

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

        for (int j = 0; j < 3; j++) {
            Node node2 = elem.getElementsByTagName("ital").item(j);
            String ital = node2.getTextContent();
            if (ital.equalsIgnoreCase("vodka"))
                System.out.println('\t' + nev);
        }
    }
}
```

Eredmény:

Azoknak a bároknak a neve, akik árulnak vodkát:

Dreher bár

Foods and Drinks

Ötödik lekérdezés

```
private static void getBySalary(Document doc) {
    //Node listát csinálás a kiválasztott elem gyerekelemeiből
    NodeList nList = doc.getElementsByTagName("alkalmazott");
    //Bejárja a gyerekelemeket
    for (int i = 0; i < nList.getLength(); i++) {
        //Megkönnyíti a különböző node-ok megkülönböztetését ez a sor
        Node nNode = nList.item(i);
        //Aktuális node elemmé konvertálás
        Element elem = (Element) nNode;

        //Gyerekelemek kiválasztása és megfelelő adattípus szerint konvertálja
        Node node1 = elem.getElementsByTagName("nev").item(0);
        String nev = node1.getTextContent();

        Node node2 = elem.getElementsByTagName("fizetes").item(0);
        int fizetes = Integer.parseInt(node2.getTextContent());
        //Feltétel megadása
        if (fizetes >= 400000)
            System.out.println('\t' + nev);
    }
}
```



```
    }  
}
```

Eredmény:

Azoknak az alkalmazottak neve, akik többet kapnak, mint 400.000:

Papp Márton

Varga István

Dom Modify

```
package hu.domparse.v9rn7c;
```

```
import org.w3c.dom.*;
```

```
import org.xml.sax.SAXException;
```

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.parsers.ParserConfigurationException;
```

```
import javax.xml.transform.Transformer;
```

```
import javax.xml.transform.TransformerConfigurationException;
```

```
import javax.xml.transform.TransformerException;
```

```
import javax.xml.transform.TransformerFactory;
```

```
import javax.xml.transform.dom.DOMSource;
```

```
import javax.xml.transform.stream.StreamResult;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
class DomModifyV9RN7C {
```

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

```
        File xmlFile = new File("V9RN7Cxml.xml");
```

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

```
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
```

```
        Document doc = dBuilder.parse(xmlFile);
```

```
        doc.getDocumentElement().normalize();
```

```
        //Elem érték átnevezés: A bárokban a vodkák átnevezése vodkaszódának:
```

```
        elementValueModify(doc);
```

```
        //Attribútum átnevezés: Egyik alkalmazott elsődleges kulcsát, az a_id-  
jának új értékadás
```

```
        attributeValueModify(doc);
```

```
        //Gyerekelem törlés: Résztvevőknél a diákigazolvány elemet kitörli
```

```
        childElementDelete(doc);
```

```
        //Új gyerekelem hozzáadás: Alkalmazott egy új kor nevű gyerekelemet
```

```
        childElementAdd(doc);
```

```
        //Új attribútumot hozzáadás: Közvélemény kapcsoló táblának új id: k_id
```

```
        attributeAddNew(doc);
```

```
        writeToConsole(doc);
```

```
    }
```

Elemérték módosítás

```
private static void elementValueModify(Document doc) {
    NodeList nList = doc.getElementsByTagName("bar");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        //Átmegy az összes ital elemen
        for (int j = 0; j < 3; j++) {
            Node node = element.getElementsByTagName("ital").item(j);
            String ital = node.getTextContent();
            //Ha az ital megegyezik a kívánt értékkel, akkor új értékere
módosítja
            if (ital.equalsIgnoreCase("vodka")) {
                ital = "vodkaszóda";
                Element modifyElement = (Element) node;
                modifyElement.setTextContent(ital);
            }
        }
    }
}
```

Eredmény:

```
<bar b_id="ads-h2w-w2" fesz_b="dh2-82k-ds">
    <nev>Dreher bár</nev>
    <ital>Sör</ital>
    <ital>vodkaszóda</ital>
    <ital>Whisky</ital>
    <cegnev>Dreher Sörgyárak Zrt.</cegnev>
</bar>
<bar b_id="mub-322-ds" fesz_b="dh2-82k-ds">
    <nev>Jana Aqua bár</nev>
    <ital>Ásványvíz</ital>
    <ital>Jägermeister</ital>
    <ital>Cognac</ital>
    <cegnev>AQUA Kft.</cegnev>
</bar>
<bar b_id="mnd-hb7-20" fesz_b="dh2-82k-ds">
    <nev>Foods and Drinks</nev>
    <ital>vodkaszóda</ital>
    <ital>Energiaital</ital>
    <ital>Gin</ital>
    <cegnev>Zamardi Foods Kft.</cegnev>
</bar>
```

Attribútum módosítás

```
private static void attributeValueModify(Document doc) {
```

```

NodeList nList = doc.getElementsByTagName("alkalmazott");
//Első a sorban alkalmazottat kiválasztjuk
Node nNode = nList.item(0);
//Kigyűjtük az attribútumokat
NamedNodeMap attributes = nNode.getAttributes();
//a_id kigyűjtése az attribútum listából
Node a_idNode = attributes.getNamedItem("a_id");
//új érték adás
Attr modifyAttr = (Attr) a_idNode;
modifyAttr.setNodeValue("kpw-77a-ns");
}

```

Eredmény:

```

<alkalmazott a_id="kpw-77a-ns" b_a="mub-322-ds">
  <nev>Dobos Sára</nev>
  <fizetes>321200</fizetes>
  <muszak>péntek-szombat</muszak>
  <muszak>szerda</muszak>
<kor>20</kor></alkalmazott>

```

Gyerekelem törlés

```

private static void childElementDelete(Document doc) {
    NodeList nList = doc.getElementsByTagName("resztvevo");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        Element element = (Element) nNode;
        Node node = element.getElementsByTagName("diakigazolvany").item(0);
        element.removeChild(node);
    }
}

```

Eredmény:

```

<resztvevo fesz_r="dh2-82k-ds" r_id="wda-312-s2">
  <nev>Pataki Patrícia</nev>

  <lakcim>
    <telepules>Szombathely</telepules>
    <utca>Tolnai Lajos utca</utca>
    <hazszam>23</hazszam>
  </lakcim>
</resztvevo>
<resztvevo fesz_r="dh2-82k-ds" r_id="daw-29k-ws">
  <nev>Halász Milán</nev>

  <lakcim>
    <telepules>Miskolc</telepules>
    <utca>Herend utca</utca>
    <hazszam>100</hazszam>
  </lakcim>

```

```

</resztvevo>
<resztvevo fesz_r="dh2-82k-ds" r_id="dwj-82w-sa">
  <nev>Somogyi Linda</nev>

  <lakcim>
    <telepules>Karcag</telepules>
    <utca>Kardhegy utca</utca>
    <hazszam>33</hazszam>
  </lakcim>
</resztvevo>

```

Gyerekelem hozzáadás

```

private static void childElementAdd(Document doc) {
    NodeList nList = doc.getElementsByTagName("alkalmazott");
    for (int i = 0; i < nList.getLength(); i++) {
        Node nNode = nList.item(i);
        //Létrehozza az új gyerekelemet
        Element element = doc.createElement("kor");
        //Belerakja az alkalmazott elembe
        nNode.appendChild(element);
        //Add text részt a kor gyerekelemnek
        element.appendChild(doc.createTextNode(String.valueOf(20 + i)));
    }
}

```

Eredmény:

```

<alkalmazott a_id="kpw-77a-ns" b_a="mub-322-ds">
  <nev>Dobos Sára</nev>
  <fizetes>321200</fizetes>
  <muszak>péntek-szombat</muszak>
  <muszak>szerda</muszak>
  <kor>20</kor></alkalmazott>
<alkalmazott a_id="sha-dw7-wa" b_a="mnd-hb7-20">
  <nev>Papp Márton</nev>
  <fizetes>400000</fizetes>
  <muszak>szerda-péntek</muszak>
  <muszak>szombat</muszak>
  <kor>21</kor></alkalmazott>
<alkalmazott a_id="dsa-wa6-42" b_a="mnd-hb7-20">
  <nev>Varga István</nev>
  <fizetes>410000</fizetes>
  <muszak>szombat</muszak>
  <muszak>csütörtök</muszak>
  <kor>22</kor></alkalmazott>

```

Attribútum hozzáadás

```

private static void attributeAddNew(Document doc) {
    NodeList nList = doc.getElementsByTagName("kozvelemen");
}

```

```

        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);
            Element element = (Element) nNode;
            element.setAttribute("k_id", "mau-ugs-2" + i);
        }
    }

    private static void writeToConsole(Document doc) throws TransformerException {
        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);
    }
}

```

Eredmény:

```

<kozvelemeny k_id="mau-ugs-20" ny_k_ny="dag-213-sd" r_k_r="dwj-82w-sa">
    <kerdezo>Szűcs Vilmos</kerdezo>
</kozvelemeny>
<kozvelemeny k_id="mau-ugs-21" ny_k_ny="kjg-mhz-2d" r_k_r="dwj-82w-sa">
    <kerdezo>Szűcs Vilmos</kerdezo>
</kozvelemeny>
<kozvelemeny k_id="mau-ugs-22" ny_k_ny="sk6-wss-31" r_k_r="daw-29k-ws">
    <kerdezo>Jónás Judit</kerdezo>
</kozvelemeny>
<kozvelemeny k_id="mau-ugs-23" ny_k_ny="jnu-21g-bs" r_k_r="wda-312-s2">
    <kerdezo>Szűcs Vilmos</kerdezo>
</kozvelemeny>

```