

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Gépkocsi nyilvántartás

Varga-Molnár Bertalan

PY7QFH

2022

Tartalomjegyzék

| | |
|---|----|
| Témakör leírása | 3 |
| Egyedek, attribútumok, kapcsolatok | 3 |
| GitHub repository elérése..... | 5 |
| Első feladat | 6 |
| Az adatbázis ER modell | 6 |
| Az adatbázis konvertálása XDM modellre | 6 |
| Az XDM modell alapján XML dokumentum készítése..... | 8 |
| Az XML dokumentum alapján XMLSchema készítése | 11 |
| Második feladat | 16 |
| Adatolvasás | 16 |
| Adatmódosítás | 20 |
| Adatlekérdezés | 23 |

Témakör leírása

Az általam választott témakör: Személygépjárművek és tulajdonságaik. Ez a modell 5 egyeddel rendelkezik, melyek mindegyike reguláris elem, vagyis a modell nem rendelkezik gyengén típusos komponenssel. Ha a kapcsolatokat figyeljük meg, akkor megkülönböztethetünk identifikáló, és nem identifikáló kapcsolatokat.

Egyedek, attribútumok, kapcsolatok

Az alábbi kapcsolatok egyike sem identifikáló:

- Motor-Személygépjármű (HAS_A, egy a többhöz)
- Gyártó-Személygépjármű (HAS_A, egy a többhöz)
- Karosszéria-Személygépjármű (HAS_A, egy az egyhez)
- Tulajdonos-Személygépjármű (HAS_A, egy a többhöz)

Személygépjármű

- Rendszám (elsődleges kulcs), három betű, három szám
- alváz száma (idegen kulcs)
- tulaj_jogsi (idegen kulcs)
- motor_száma (idegen kulcs)
- gyártás_sorszám (idegen kulcs)
- Kor (a Gyártási év mezőből származtatott)
- CASCO azonosító (opcionális) – min előfordulás 0

Tulajdonos

- Jogosítvány száma (elsődleges kulcs), két betű, hat szám
- Név (összetett), elemei: Vezetéknév, Keresztnév

Motor:

- Motorszám (elsődleges kulcs), 14 betű vagy szám
- üzemanyag (lehet benzin, dízel, hibrid vagy elektromos)
- Lóerő
- Nyomaték

Karosszéria:

- Alvázszám (elsődleges kulcs), 17 betű vagy szám
- Felület (matt, részleges, teljes fényezés)
- Szín (többértékű)
- Matricák (van vagy nincs)

Gyártás:

- GyártásID (elsődleges kulcs), 2 szám 1 betű
- márka
- modell
- Gyártási év (valós évszám kell legyen)

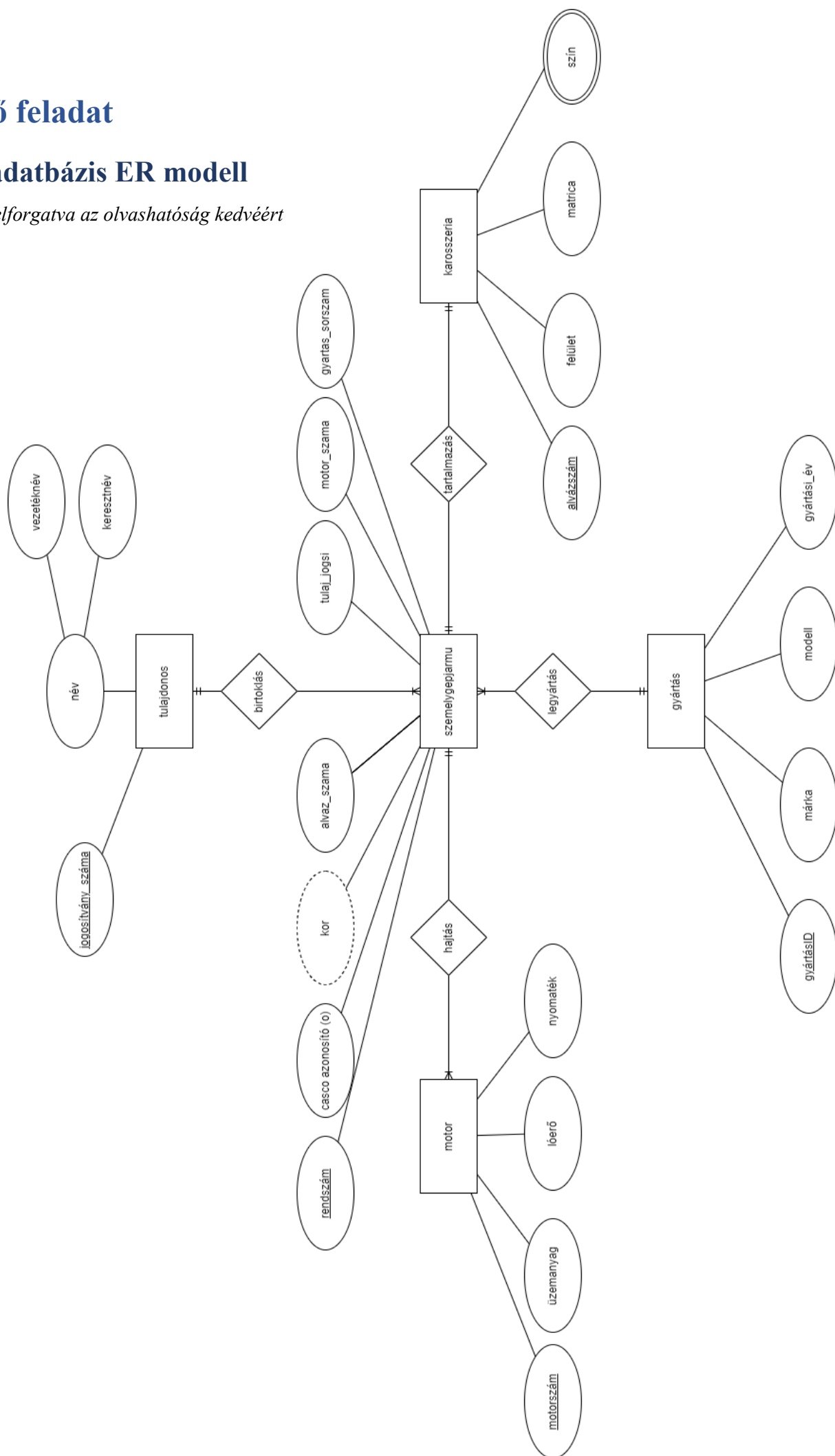
GitHub repository elérése

https://github.com/vargamolnarbertalan/PY7QFH_XMLGyak/tree/main/XMLTaskPY7QFH

Első feladat

Az adatbázis ER modell

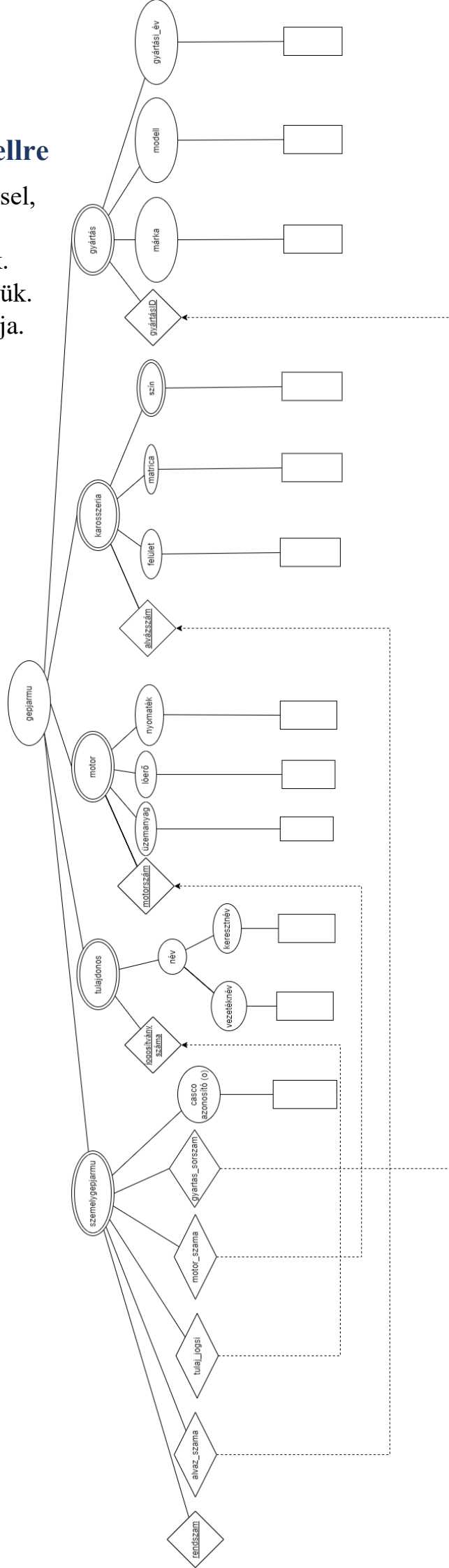
Ábra elforgatva az olvashatóság kedvéért



Az adatbázis konvertálása XDM modellre

Az ER modellben szereplő egyedeket ellipszissel, a tulajdonságokat ellipszissel és téglalappal, az elsődleges kulcsokat rombuszsal ábrázoljuk. A többszörös előfordulást dupla vonallal jelöljük. Az adatbázis root eleme a *gepjarmu* nevet kapja.

Ábra elforgatva az olvashatóság kedvéért



Az XDM modell alapján XML dokumentum készítése

Az XDM modell alapján készítettem el az XML dokumentumot, kezdve a root elementtel. Figyeltem arra, hogy a gyerekelemekből legalább 3 legyen, mert úgy szemléletes, illetve az opcionális elementből valahol van, valahol nincs, a többértékűnél pedig előfordul egy, kettő illetve 3 is egy parenten belül.

Forráskód

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

<gepjarmu xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="XMLSchemaPY7QFH.xsd">

  <!-- személygépjárművek -->

  <szemelygepjarmu rendszam="ABC-123">
    <tulaj_jogsi>HU345678</tulaj_jogsi>
    <motor_szama>100FDSAF008F1G</motor_szama>
    <alvaz_szama>0FSFDSFFF70000019</alvaz_szama>
    <gyartas_sorszam>15B</gyartas_sorszam>
    <casco_azon>FSDAG34</casco_azon>
  </szemelygepjarmu>

  <szemelygepjarmu rendszam="LMX-534">
    <tulaj_jogsi>HU346544</tulaj_jogsi>
    <motor_szama>100FDFGDS58F1G</motor_szama>
    <alvaz_szama>0FSFDSCCC70000019</alvaz_szama>
    <gyartas_sorszam>10B</gyartas_sorszam>
    <casco_azon>HSDAG34</casco_azon>
  </szemelygepjarmu>

  <szemelygepjarmu rendszam="TBS-820">
    <tulaj_jogsi>R0378544</tulaj_jogsi>
    <motor_szama>540FDFGDS58F1G</motor_szama>
    <alvaz_szama>3HSFDSCCC70000019</alvaz_szama>
    <gyartas_sorszam>98C</gyartas_sorszam>
  </szemelygepjarmu>

  <!-- tulajdonosok -->

  <tulajdonos jogositvany_szama="HU345678">
    <nev>
      <vezeteknev>Példa</vezeteknev>
      <keresztnev>Aladár</keresztnev>
    </nev>
  </tulajdonos>

  <tulajdonos jogositvany_szama="HU346544">
```



```

    <nev>
      <vezeteknev>Molnár</vezeteknev>
      <keresztnev>Jolán</keresztnev>
    </nev>
  </tulajdonos>

  <tulajdonos jogositvany_szama="R0378544">
    <nev>
      <vezeteknev>Mutu</vezeteknev>
      <keresztnev>Adrian</keresztnev>
    </nev>
  </tulajdonos>

  <!-- motorok -->

  <motor motorszam="100FDSAF008F1G">
    <uzemanyag>dízel</uzemanyag>
    <loero>68</loero>
    <nyomatek>160 Nm</nyomatek>
  </motor>

  <motor motorszam="100FDFGDS58F1G">
    <uzemanyag>benzin</uzemanyag>
    <loero>75</loero>
    <nyomatek>107 Nm</nyomatek>
  </motor>

  <motor motorszam="540FDFGDS58F1G">
    <uzemanyag>elektromos</uzemanyag>
    <loero>150</loero>
    <nyomatek>310 Nm</nyomatek>
  </motor>

  <!-- karosszériák -->

  <karosszeria alvazszam="0FSFDSFFF70000019">
    <felulet>matt</felulet>
    <matrica>nincs</matrica>
    <szin>fehér</szin>
    <szin>szürke</szin>
  </karosszeria>

  <karosszeria alvazszam="0FSFDSCCC70000019">
    <felulet>teljes</felulet>
    <matrica>van</matrica>
    <szin>fehér</szin>
  </karosszeria>

  <karosszeria alvazszam="3HSFDSCCC70000019">

```

```
<felulet>részleges</felulet>
<matrica>nincs</matrica>
<szin>zöld</szin>
<szin>fekete</szin>
<szin>szürke</szin>
</karosszeria>

<!-- gyártási infók -->

<gyartas gyartasID="15B">
  <marka>Ford</marka>
  <modell>Fusion</modell>
  <gyartasi_ev>2004</gyartasi_ev>
</gyartas>

<gyartas gyartasID="10B">
  <marka>Renault</marka>
  <modell>Twingo</modell>
  <gyartasi_ev>2011</gyartasi_ev>
</gyartas>

<gyartas gyartasID="98C">
  <marka>Volkswagen</marka>
  <modell>ID.3</modell>
  <gyartasi_ev>2021</gyartasi_ev>
</gyartas>

</gepjarmu>
```

Forráskód vége

Az XML dokumentum alapján XMLSchema készítése

Az XML dokumentum validálásához saját sémát készítettem, melyben próbáltam kreatív lenni, validációs megkötésnek használtam saját típusokat és olyat is, amit az xsd nyelv tartalmaz alapvetően (például xs:gYear). A séma szerkezetét az alábbiak szerint alakítottam: egyszerű elemek, amikre később referálok, saját típusok, amikkel a témakör sajátos megkötéseit érvényesítem, felépítés, elsődleges kulcsok majd végül idegen kulcsok.

Forráskód:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">

  <!--Egyszerű elemek-->

  <xs:element name="tulaj_jogsi" type="jogositvany_szama_tipus"/>
  <xs:element name="motor_szama" type="motorszam_tipus"/>
  <xs:element name="alvaz_szama" type="alvazszam_tipus"/>
  <xs:element name="gyartas_sorszam" type="gyartoID_tipus"/>
  <xs:element name="casco_azon" type="xs:string"/>

  <xs:element name="vezeteknev" type="xs:string"/>
  <xs:element name="keresztnev" type="xs:string"/>

  <xs:element name="uzemanyag" type="uzemanyag_tipus"/>
  <xs:element name="loero" type="xs:string"/>
  <xs:element name="nyomatek" type="xs:string"/>

  <xs:element name="felulet" type="felulet_tipus"/>
  <xs:element name="matrica" type="matrica_tipus"/>
  <xs:element name="szin" type="xs:string"/>

  <xs:element name="marka" type="xs:string"/>
  <xs:element name="modell" type="xs:string"/>
  <xs:element name="gyartasi_ev" type="xs:gYear"/>

  <!--Saját típusok-->

  <xs:simpleType name="rendszam_tipus">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Z][A-Z]-[0-9][0-9][0-9]"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="jogositvany_szama_tipus">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Z][0-9][0-9][0-9][0-9][0-9][0-9]"/>
    </xs:restriction>
  </xs:simpleType>
```

```

        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="motorszam_tipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="[A-Z0-9]{14}" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="uzemanyag_tipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="benzin|dízel|hibrid|elektromos" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="alvazszam_tipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="[A-Z0-9]{17}" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="felulet_tipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="matt|részleges|teljes" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="matrica_tipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="van|nincs" />
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="gyartoID_tipus">
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9][0-9][A-Z]" />
        </xs:restriction>
    </xs:simpleType>

    <!--Felépítés-->

    <xs:element name="gepjarmu">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="szemelygepjarmu" minOccurs="1"
maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>

```

```

        <xs:element ref="tulaj_jogsi" />
        <xs:element ref="motor_szama"/>
        <xs:element ref="alvaz_szama"/>
        <xs:element ref="gyartas_sorszam" />
        <xs:element ref="casco_azon" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="rendszam" type="rendszam_tipus"
use="required"/>
</xs:complexType>
</xs:element>

    <xs:element name="tulajdonos" minOccurs="1"
maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="nev">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="vezeteknev" />
                        <xs:element ref="keresztnev" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="jogositvany_szama"
type="jogositvany_szama_tipus" use="required"/>
    </xs:complexType>
</xs:element>

    <xs:element name="motor" minOccurs="1" maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="uzemanyag" />
            <xs:element ref="loero" />
            <xs:element ref="nyomatek" />
        </xs:sequence>
        <xs:attribute name="motorszam" type="motorszam_tipus"
use="required"/>
    </xs:complexType>
</xs:element>

    <xs:element name="karosszeria" minOccurs="1"
maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="felulet" />
            <xs:element ref="matrica" />
            <xs:element ref="szin" maxOccurs="unbounded"/>
        </xs:sequence>

```

```

        <xs:attribute name="alvazszam" type="alvazszam_tipus"
use="required"/>
    </xs:complexType>
</xs:element>

    <xs:element name="gyartas" minOccurs="1"
maxOccurs="unbounded">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="marka" />
            <xs:element ref="modell" />
            <xs:element ref="gyartasi_ev"
/>
        </xs:sequence>
        <xs:attribute name="gyartasID" type="gyartoID_tipus"
use="required"/>
    </xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>

<!--Kulcsok-->

<xs:key name="szemelygepjarmu_kulcs">
    <xs:selector xpath="szemelygepjarmu" />
    <xs:field xpath="@rendszam" />
</xs:key>

<xs:key name="tulajdonos_kulcs">
    <xs:selector xpath="tulajdonos" />
    <xs:field xpath="@jogositvany_szama" />
</xs:key>

<xs:key name="motor_kulcs">
    <xs:selector xpath="motor" />
    <xs:field xpath="@motorszam" />
</xs:key>

<xs:key name="karosszeria_kulcs">
    <xs:selector xpath="karosszeria" />
    <xs:field xpath="@alvazszam" />
</xs:key>

<xs:key name="gyartas_kulcs">
    <xs:selector xpath="gyartas" />
    <xs:field xpath="@gyartasID" />
</xs:key>

```

```
<!--Idegen kulcsok-->

<xs:keyref refer="tulajdonos_kulcs" name="tulaj_idegen_kulcs">
  <xs:selector xpath="szemelygpjarmu" />
  <xs:field xpath="tulaj_jogsi" />
</xs:keyref>

<xs:keyref refer="motor_kulcs" name="motor_idegen_kulcs">
  <xs:selector xpath="szemelygpjarmu" />
  <xs:field xpath="motor_szama" />
</xs:keyref>

<xs:keyref refer="karosszeria_kulcs" name="karosszeria_idegen_kulcs">
  <xs:selector xpath="szemelygpjarmu" />
  <xs:field xpath="alvaz_szama" />
</xs:keyref>

<xs:keyref refer="gyartas_kulcs" name="gyartas_idegen_kulcs">
  <xs:selector xpath="szemelygpjarmu" />
  <xs:field xpath="gyartas_sorszam" />
</xs:keyref>

</xs:element>

</xs:schema>
```

Forráskód vége

Második feladat

Adatolvasás

A teljes XML dokumentum feldolgozásához létrehoztam egy osztályszintű metódust, amit kétszer hívok meg a kódban, eredményét egyszer a konzolra írjuk ki, egyszer pedig szöveges dokumentumba. A *Feldolgozas()* metóduson belül először inicializáljuk a beolvasandó fájlt, majd a parsert és kiírjuk a root elementet. Ezután követve az XML dokumentum struktúráját, minden eltárolt adatot megjelenítünk. Az opcionális elemnél nincset írunk, ha nem található abban a parent elementben, míg a többszörös előfordulásoknál for ciklussal számozzuk és kiírjuk az összes értéket.

Forráskód:

```
package hu.domparse.py7qfh;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;

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;
import org.xml.sax.SAXException;

public class DomReadPY7QFH {

    public static void main(String argv[]) throws ParserConfigurationException,
    SAXException, IOException {

        // Adatok kiirasa console-ra

        Feldolgozas();

        // Adatok mentése TXT-be

        PrintStream out = new PrintStream(new
        FileOutputStream("XML_Parse_output.txt"));
        System.setOut(out);

        Feldolgozas();

    }

    private static void Feldolgozas() throws ParserConfigurationException,
    SAXException, IOException {
        File xmlFile = new File("XMLPY7QFH.xml");
```



```

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
DocumentBuilder dBuilder = factory.newDocumentBuilder();

Document doc = dBuilder.parse(xmlFile);

doc.getDocumentElement().normalize();

System.out.println("Root element: " +
doc.getDocumentElement().getNodeName());

// személygépjarmuvek kiirasa

NodeList nList = doc.getElementsByTagName("szemelygepjarmu");

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

    Node nNode = nList.item(i);
    System.out.println("\nCurrent element: " + nNode.getNodeName());

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

        Element elem = (Element) nNode;
        String rendszam = elem.getAttribute("rendszam");

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

        Node node2 = elem.getElementsByTagName("motor_szama").item(0);
        String msz = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("alvaz_szama").item(0);
        String asz = node3.getTextContent();

        Node node4 = elem.getElementsByTagName("gyartas_sorszam").item(0);
        String gys = node4.getTextContent();

        String ca = "nincs";
        Node node5 = elem.getElementsByTagName("casco_azon").item(0);
        if (node5 != null) {
            ca = node5.getTextContent();
        }

        System.out.println("Rendszam: " + rendszam);
        System.out.println("Tulaj jogsi: " + tj);
        System.out.println("Motor szama: " + msz);
        System.out.println("Alvaz szama: " + asz);
        System.out.println("Gyartasi sorszam: " + gys);
        System.out.println("Casco azonosito: " + ca);

    }

}

// tulajdonosok kiirasa

nList = doc.getElementsByTagName("tulajdonos");

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

```

```

Node nNode = nList.item(i);
System.out.println("\nCurrent element: " + nNode.getNodeName());

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

    Element elem = (Element) nNode;
    String jogszi = elem.getAttribute("jogositvany_szama");

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

    Node node2 = elem.getElementsByTagName("keresztnev").item(0);
    String kn = node2.getTextContent();

    System.out.println("Jogositvany szama: " + jogszi);
    System.out.println("Vezeteknev: " + vn);
    System.out.println("Keresztnev: " + kn);

}
}

// motorok kiirasa

nList = doc.getElementsByTagName("motor");

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

    Node nNode = nList.item(i);
    System.out.println("\nCurrent element: " + nNode.getNodeName());

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

        Element elem = (Element) nNode;
        String msz = elem.getAttribute("motorszam");

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

        Node node2 = elem.getElementsByTagName("loero").item(0);
        String le = node2.getTextContent();

        Node node3 = elem.getElementsByTagName("nyomatek").item(0);
        String ny = node3.getTextContent();

        System.out.println("Motorszam: " + msz);
        System.out.println("Uzemanyag: " + ua);
        System.out.println("Loero: " + le);
        System.out.println("Nyomatek: " + ny);

    }

}

// karosszeriak kiirasa

nList = doc.getElementsByTagName("karosszeria");

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

    Node nNode = nList.item(i);

```

```

        System.out.println("\nCurrent element: " + nNode.getNodeName());

        if (nNode.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) nNode;
            String asz = elem.getAttribute("alvazszam");

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

            Node node2 = elem.getElementsByTagName("matrica").item(0);
            String ma = node2.getTextContent();

            System.out.println("Alvazszam: " + asz);
            System.out.println("Felulet: " + ft);
            System.out.println("Matrica: " + ma);

            for (int a = 0; a < elem.getElementsByTagName("szin").getLength();
a++) {
                Node node3 = elem.getElementsByTagName("szin").item(a);
                String sz = node3.getTextContent();
                System.out.println("Szín #" + (a + 1) + ": " + sz);
            }
        }
    }
    // gyártási infók kiirasa

    nList = doc.getElementsByTagName("gyartas");

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

        Node nNode = nList.item(i);
        System.out.println("\nCurrent element: " + nNode.getNodeName());

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

            Element elem = (Element) nNode;
            String gyi = elem.getAttribute("gyartasID");

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

            Node node2 = elem.getElementsByTagName("modell").item(0);
            String me = node2.getTextContent();

            Node node3 = elem.getElementsByTagName("gyartasi_ev").item(0);
            String gye = node3.getTextContent();

            System.out.println("GyartasID: " + gyi);
            System.out.println("Marka: " + mk);
            System.out.println("Modell: " + me);
            System.out.println("Gyartasi ev: " + gye);

        }
    }
    return;
}
}

```

Adatmódosítás

Az adatmódosítás során cél, hogy az eredeti XML dokumentumból olvassunk be, de egy másikban (XMLPY7QFH2.xml) tároljuk el a módosításokat, így nem vesz el az első fájlunk. Az alábbi 5 változtatást végezzük el:

- A második személygépjármű elem rendszámát átírjuk
- Az első tulajdonos jogosítvány számát átírjuk
- Az összes karosszérián beállítjuk, hogy van matrica
- A benzines motorokat hibridre állítjuk
- A *gyartasi_ev* elemnevet *evjaratra* módosítjuk

Forráskód:

```
package hu.domparse.py7qfh;

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
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.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

public class DomModifyPY7QFH {

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

        File inputFile = new File("XMLPY7QFH.xml");

        DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

        Document doc = documentBuilder.parse(inputFile);
        doc.getDocumentElement().normalize();

        // második személygépjármű attribútumának módosítása
        Node jarmu = doc.getElementsByTagName("szemelygepjarmu").item(1);

        NamedNodeMap attr = jarmu.getAttributes();
        Node nodeAttr = attr.getNamedItem("rendszam");
        nodeAttr.setTextContent("CFN-213");
```

```

System.out.println("szemelygepjarmu attributum modositva");

// első tulajdonos attribútumának módosítása
Node tulaj = doc.getElementsByTagName("tulajdonos").item(0);

attr = tulaj.getAttributes();
nodeAttr = attr.getNamedItem("jogositvany_szama");
nodeAttr.setTextContent("FR346544");

System.out.println("tulajdonos attributum modositva");

// összes matrica beállítása van-ra

NodeList nodes = doc.getElementsByTagName("karosszeria");

for (int i = 0; i < nodes.getLength(); i++) {
    Node node = nodes.item(i);

    if (node.getNodeType() == Node.ELEMENT_NODE) {
        NodeList childNodes = node.getChildNodes();

        for (int j = 0; j < childNodes.getLength(); j++) {
            Node childNode = childNodes.item(j);
            if (childNode.getNodeName().equals("matrica")) {
                childNode.setTextContent("van");
            }
        }
    }
}
System.out.println("matrica modositások kész");

// összes benzines motor átállítása hibridre

nodes = doc.getElementsByTagName("motor");

for (int i = 0; i < nodes.getLength(); i++) {
    Node node = nodes.item(i);

    if (node.getNodeType() == Node.ELEMENT_NODE) {
        NodeList childNodes = node.getChildNodes();

        for (int j = 0; j < childNodes.getLength(); j++) {
            Node childNode = childNodes.item(j);
            if (childNode.getNodeName().equals("uzemanyag") &&
childNode.getTextContent().equals("benzin")) {
                childNode.setTextContent("hibrid");
            }
        }
    }
}
System.out.println("benzin motorok atallitva hibridre");

// gyartasi_ev element atnevezese evjaratra

nodes = doc.getElementsByTagName("gyartasi_ev");
for (int i = 0; i < nodes.getLength(); i++) {
    doc.renameNode(nodes.item(i), null, "evjarat");
}
System.out.println("gyartasi_ev atnevezeve evjaratra");

```

```

        // modosított xml dokumentum elementese
        writeXml(doc, new File("XMLPY7QFH2.xml"));
    }

    private static void writeXml(Document doc, File output) throws
TransformerException {
        Transformer transformer =
TransformerFactory.newInstance().newTransformer();
        DOMSource source = new DOMSource(doc);

        StreamResult file = new StreamResult(output);

        transformer.transform(source, file);
    }
}

```

Adatlekérdezés

Az adatlekérdezés során rendezett keretek között nyerünk ki információt az adatbázisunkból, viszont, hogy az eredmény letisztult legyen, a forráskódban a lekérdezések ki vannak kommentelve. Ezek a lekérdezések az alábbiak:

- Az adatbázisban szereplő összes tulajdonos és minden adatuk
- Az adatbázisban 10B azonosítóval szereplő gyártási infók
- Az adatbázis utolsó motor rekordja
- Az adatbázisban található összes olyan karosszéria, melynek elsődleges színe a fehér
- Az adatbázis összes olyan gyártási infója, ami 2008 utáni

Lekérdezések eredménye (a fenti sorrendben):

```
Aktuális elem: tulajdonos  
Jogosítvány száma: HU345678  
Keresztnév: Aladár  
Vezetéknév: Példa
```

```
Aktuális elem: tulajdonos  
Jogosítvány száma: HU346544  
Keresztnév: Jolán  
Vezetéknév: Molnár
```

```
Aktuális elem: tulajdonos  
Jogosítvány száma: R0378544  
Keresztnév: Adrian  
Vezetéknév: Mutu
```

```
Aktuális elem: gyartas  
ID: 10B  
Márka: Renault  
Modell: Twingo  
Gyártási év: 2011
```

```
Aktuális elem: motor  
Motor száma: 540FDFGDS58F1G  
Üzemanyag: elektromos  
Lóerő: 150  
Nyomaték: 310 Nm
```

```
Aktuális elem: karosszeria  
Alvázsám: 0FSFDSFFF70000019  
Felület: matt  
Matrica: nincs  
Elsődleges szín: fehér
```

```
Aktuális elem: karosszeria  
Alvázsám: 0FSFDSCC70000019  
Felület: teljes  
Matrica: van  
Elsődleges szín: fehér
```

```
Aktuális elem: gyartas  
ID: 10B  
Márka: Renault  
Modell: Twingo  
Gyártási év: 2011
```

```
Aktuális elem: gyartas  
ID: 98C  
Márka: Volkswagen  
Modell: ID.3  
Gyártási év: 2021
```

Forráskód:

```
package hu.domparse.py7qfh;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;

import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;

import org.xml.sax.SAXException;

public class DomQueryPY7QFH {

    public static void main(String[] args) {
        try {
            DocumentBuilderFactory documentBuilderFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder documentBuilder =
documentBuilderFactory.newDocumentBuilder();

            Document document = documentBuilder.parse("XMLPY7QFH.xml");

            document.getDocumentElement().normalize();

            XPath xPath = XPathFactory.newInstance().newXPath();
            //Lekerdezések
            String expression = "";
            //expression = "/gepjarmu/tulajdonos";
            //expression = "/gepjarmu/gyartas[@gyartasID='10B']";
            //expression = "/gepjarmu/motor[last()]";
            //expression = "/gepjarmu/karosszeria[szin='fehér']";
            expression = "/gepjarmu/gyartas[gyartasi_ev>2008]";
            NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(document,
XPathConstants.NODESET);

            for (int i = 0; i < nodeList.getLength(); i++) {
                Node node = nodeList.item(i);
                System.out.println("\nAktuális elem: " + node.getNodeName());

                if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("szemelygepjarmu")) {
                    Element element = (Element) node;

                    System.out.println("Rendszám: " + element.getAttribute("rendszam"));
                    System.out.println("Tulaj jogsi: " +
element.getElementsByTagName("tulaj_jogsi").item(0).getTextContent());
                    System.out.println("Álváz száma: " +
element.getElementsByTagName("alvaz_szama").item(0).getTextContent());
```



```

        System.out.println("Motor száma: " +
element.getElementsByTagName("motor_szama").item(0).getTextContent());
        System.out.println("Gyártási szám: " +
element.getElementsByTagName("gyartas_sorszam").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("tulajdonos")) {
        Element element = (Element) node;

        System.out.println("Jogosítvány száma: " +
element.getAttribute("jogositvany_szama"));
        System.out.println("Keresztnév: " +
element.getElementsByTagName("keresztnev").item(0).getTextContent());
        System.out.println("Vezetéknév: " +
element.getElementsByTagName("vezeteknev").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("motor")) {
        Element element = (Element) node;

        System.out.println("Motor száma: " + element.getAttribute("motorszam"));
        System.out.println("Üzemanyag: " +
element.getElementsByTagName("uzemanyag").item(0).getTextContent());
        System.out.println("Lóerő: " +
element.getElementsByTagName("loero").item(0).getTextContent());
        System.out.println("Nyomatek: " +
element.getElementsByTagName("nyomatek").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("karosszeria")) {
        Element element = (Element) node;

        System.out.println("Alvázzszám: " + element.getAttribute("alvazszam"));
        System.out.println("Felület: " +
element.getElementsByTagName("felulet").item(0).getTextContent());
        System.out.println("Matrica: " +
element.getElementsByTagName("matrica").item(0).getTextContent());
        System.out.println("Elsődleges szín: " +
element.getElementsByTagName("szin").item(0).getTextContent());
    }

    if (node.getNodeType() == Node.ELEMENT_NODE &&
node.getNodeName().equals("gyartas")) {
        Element element = (Element) node;

        System.out.println("ID: " + element.getAttribute("gyartasID"));
        System.out.println("Márka: " +
element.getElementsByTagName("marka").item(0).getTextContent());
        System.out.println("Modell: " +
element.getElementsByTagName("modell").item(0).getTextContent());
        System.out.println("Gyártási év: " +
element.getElementsByTagName("gyartasi_ev").item(0).getTextContent());
    }
}

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

```

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

Forráskód vége