

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Egy ruhákat készítő vállalkozás adatbázisa

Készítette: Demján Csongor

Neptunkód: SZ7MGG

Dátum: 2022/11/27

Tartalom

Feladat leírása.....	3
1.feladat	3
1a) Az adatbázis ER modell:	3
1b) Az adatbázis konvertálása XDM modellre:.....	4
1c) Az XDM modell alapján XML dokumentum készítése:	4
1d) Az XML dokumentum alapján XMLSchema készítése:	6
2.feladat	10
2a) Adatolvasás.....	10
2b) Adatmódosítás.....	11
2c) Adatlekérdezés.....	12

Feladat leírása

Eljátszottam a gondolattal, hogy milyen lenne, ha alapítanék egy márkát, természetesen nem egyedül. Valakire mindenképpen szükség lenne, aki ért a marketinghez, hiszen abból rengeteg szükséges egy sikeres márká felfuttatásához. Természetesen nem árt, ha vannak jó megfigyelései is.:) Ezen gondolat felépítésében nagy segítségre volt az ER modell elkészítése, ahol kézenfekvőek voltak a „márka” és „tulajdonos” egyedek. Szabadidőmben szívesen festek, előfordult néhány alkalommal, hogy ezt a tevékenységet ruhákon is alkalmaztam. Szóval miért is ne legyen egy ruha márká, így jött létre a „ruha” egyed. A ruhákat, ha van rá kereslet valaki megvenné, ezt a „vásárló” egyeddel reprezentáljuk és már ha van egyfajta tranzakció természetesen a számlát is ki kell állítani. Így jutottunk el az utolsó egyedünkhöz, a „számla”-hoz. Ezt összefoglaló címmel „Boltom”-nak neveztem el.

1.feladat

1a) Az adatbázis ER modell:

5 egyedet tartalmaz az adatbázis: Tulajdonos, Márka, Ruha, Vásárló, Számla.

A Tulajdonosnak ID, név, profit, cím tulajdonságai vannak és tulajdona a márká.

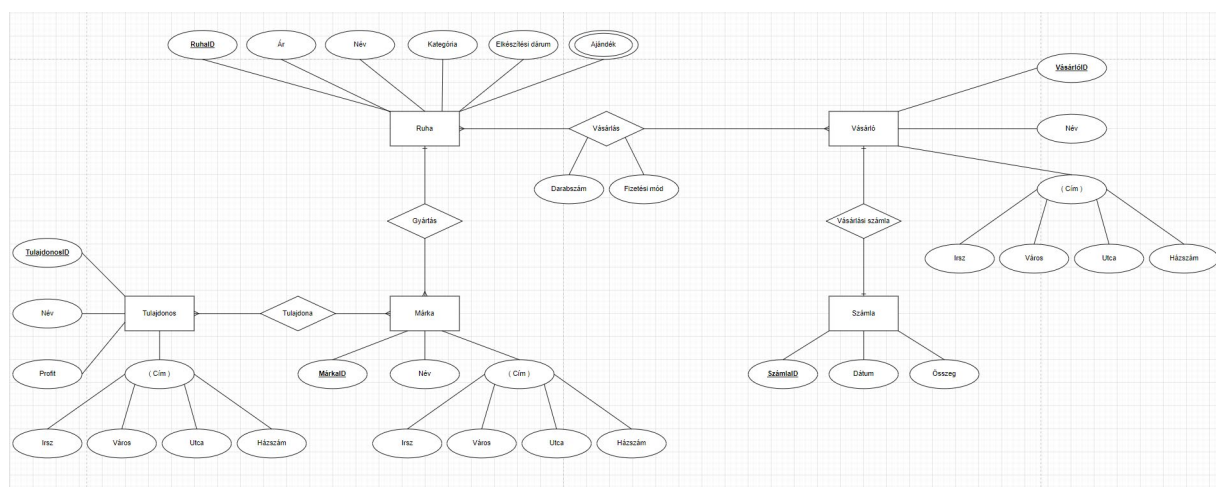
A Márkának ID, név, cím tulajdonságai vannak és ruhákat gyártanak.

A Ruhának ID, ár, név, kategória, elkészítési dátuma van és a vásárlók vásárolják.

A Vásárlás, amely a ruha és vásárló közötti kapcsolat darabszáma és fizetési mód tulajdonságai vannak.

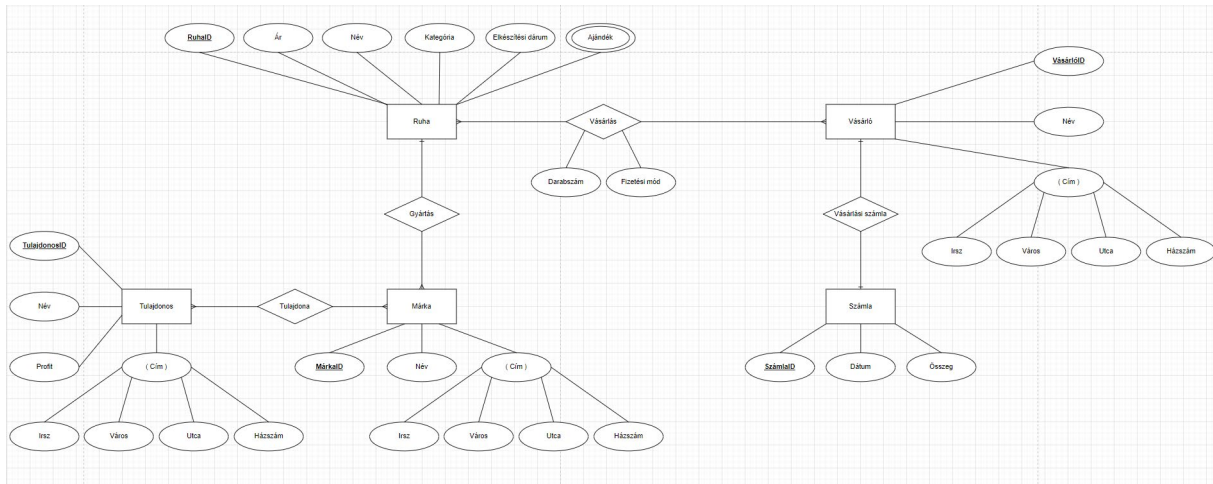
A Vásárló Id-val, névvel, címmel rendelkezik, a vásárlási számla kapcsolattal a számla kapcsolódik hozzá.

A Számla ID, dátum, összeg tulajdonságokkal rendelkezik.



1b) Az adatbázis konvertálása XDM modellre:

XDMSZ7MGG.png, megtalálható XDMSZ7MGG.drawio, ahol további szerkesztési lehetőség van.



1c) Az XDM modell alapján XML dokumentum készítése:

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

<boltom xsi:noNamespaceSchemaLocation="XMLSchemaSZ7MGG.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <tulajdonos>
    <tulajdonosId>1</tulajdonosId>
    <nev>Demján Csongor</nev>
    <profit>50000</profit>
    <cim>
      <irsz>3335</irsz>
      <varos>Miskolc</varos>
      <utca>Geró</utca>
      <hazzam>1</hazzam>
    </cim>
  </tulajdonos>
  <tulajdonos>
    <tulajdonosId>2</tulajdonosId>
    <nev>Másik Tulaj</nev>
    <profit>50000</profit>
    <cim>
      <irsz>3500</irsz>
      <varos>Miskolc</varos>
      <utca>Csabai</utca>
      <hazzam>2</hazzam>
    </cim>
  </tulajdonos>
  <marka>
    <markaId>1</markaId>
    <nev>Jocucc</nev>
    <cim>
      <irsz>3500</irsz>
      <varos>Miskolc</varos>
      <utca>Csabai</utca>
    </cim>
  </marka>
</boltom>
```

```

        <hazszam>2</hazszam>
    </cim>
</marka>
<marka>
    <markaId>2</markaId>
    <nev>Asd</nev>
    <cim>
        <irsz>3500</irsz>
        <varos>Miskolc</varos>
        <utca>Csabai</utca>
        <hazszam>2</hazszam>
    </cim>
</marka>
<ruha>
    <ruhaId>001</ruhaId>
    <nev>DamnTheBestT-Shirt</nev>
    <ar>30000</ar>
    <kategoria>T-Shirts</kategoria>
    <elkeszitesidatum>2022-12-12</elkeszitesidatum>
    <ajandek>Csokornyakkendő</ajandek>
</ruha>
<ruha>
    <ruhaId>002</ruhaId>
    <nev>DamnTheBestHoodie</nev>
    <ar>70000</ar>
    <kategoria>Hoodie</kategoria>
    <elkeszitesidatum>2022-10-10</elkeszitesidatum>
    <ajandek>Sapka</ajandek>
</ruha>
<ruha>
    <ruhaId>003</ruhaId>
    <nev>NiceFlowers</nev>
    <ar>50000</ar>
    <kategoria>T-Shirt</kategoria>
    <elkeszitesidatum>2022-11-11</elkeszitesidatum>
    <ajandek>Zokni</ajandek>
</ruha>
<vasarlas>
    <ruhaFk>1</ruhaFk>
    <darabszam>1</darabszam>
    <fizetesimod>Kartyas</fizetesimod>
    <vasarloFK>1</vasarloFK>
</vasarlas>
<vasarlas>
    <ruhaFk>2</ruhaFk>
    <darabszam>1</darabszam>
    <fizetesimod>Utánvét</fizetesimod>
    <vasarloFK>2</vasarloFK>
</vasarlas>
<vasarlas>
    <ruhaFk>3</ruhaFk>
    <darabszam>1</darabszam>
    <fizetesimod>Utánvét</fizetesimod>
    <vasarloFK>3</vasarloFK>
</vasarlas>
<vasarlo>
    <vasarloId>1</vasarloId>
    <nev>Nagy Ákos</nev>
    <cim>
        <irsz>4000</irsz>
        <varos>Budapest</varos>

```

```

        <utca>Déry</utca>
        <hazszam>12</hazszam>
    </cim>
</vasarlo>
<vasarlo>
    <vasarloId>2</vasarloId>
    <nev>Cserepes Bence</nev>
    <cim>
        <irsz>9000</irsz>
        <varos>Sopron</varos>
        <utca>Kedves</utca>
        <hazszam>3</hazszam>
    </cim>
</vasarlo>
<vasarlo>
    <vasarloId>3</vasarloId>
    <nev>Kiss Ibolya</nev>
    <cim>
        <irsz>6000</irsz>
        <varos>Miskolc</varos>
        <utca>Bencések</utca>
        <hazszam>30</hazszam>
    </cim>
</vasarlo>
<szamla Sz_V_Id="01">
    <szamlaId>1</szamlaId>
    <datum>2020-12-13</datum>
    <osszeg>30000</osszeg>
</szamla>
<szamla Sz_V_Id="02">
    <szamlaId>2</szamlaId>
    <datum>2020-11-12</datum>
    <osszeg>70000</osszeg>
</szamla>
<szamla Sz_V_Id="03">
    <szamlaId>3</szamlaId>
    <datum>2020-10-11</datum>
    <osszeg>50000</osszeg>
</szamla>
</boltom>

```

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="qualified">

    <xs:attribute name="Sz_V_Id" type="xs:string" />

    <xs:complexType name="boltomTipus">
        <xs:sequence>
            <xs:element name="tulajdonos" type="tulajdonosTipus"
minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="marka" type="markaTipus" minOccurs="0"
maxOccurs="unbounded"/>

```

```

        <xs:element name="ruha" type="ruhaTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="vasarlas" type="vasarlasTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="vasarlo" type="vasarloTipus" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="szamla" type="szamlaTipus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="tulajdonosTipus">
    <xs:sequence>
        <xs:element name="tulajdonosId" type="xs:string"/>
        <xs:element name="nev" type="xs:string" maxOccurs="unbounded"/>
        <xs:element name="profit" type="xs:integer"/>
        <xs:element name="cim" type="cimTipus"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="markaTipus">
    <xs:sequence>
        <xs:element name="markaId" type="xs:integer"/>
        <xs:element name="nev" type="xs:string"/>
        <xs:element name="cim" type="cimTipus"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="ruhaTipus">
    <xs:sequence>
        <xs:element name="ruhaId" type="xs:integer"/>
        <xs:element name="nev" type="xs:string"/>
        <xs:element name="ar" type="xs:integer"/>
        <xs:element name="kategoria" type="xs:string"
maxOccurs="unbounded"/>
        <xs:element name="elkeszitesidatum" type="xs:date"/>
        <xs:element name="ajandek" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="vasarlasTipus">
    <xs:sequence>
        <xs:element name="ruhaFk" type="xs:integer"/>
        <xs:element name="darabszam" type="xs:integer"/>
        <xs:element name="fizetesimod" type="xs:string"/>
        <xs:element name="vasarloFK" type="xs:integer"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="vasarloTipus">
    <xs:sequence>
        <xs:element name="vasarloId" type="xs:integer"/>
        <xs:element name="nev" type="xs:string"/>
        <xs:element name="cim" type="cimTipus"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="cimTipus">
    <xs:sequence>
        <xs:element name="irsz" type="xs:integer"/>
        <xs:element name="varos" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

```

```

        <xs:element name="utca" type="xs:string"/>
        <xs:element name="hazszam" type="xs:string"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="szamlaTipus">
    <xs:sequence>
        <xs:element name="szamlaId" type="xs:integer"/>
        <xs:element name="datum" type="xs:date"/>
        <xs:element name="osszeg" type="xs:integer"/>
    </xs:sequence>
    <xs:attribute ref="Sz_V_Id" use="required"/>
</xs:complexType>

<xs:element name="boltom" type="boltomTipus">
    <!-- Kulcsok -->
    <xs:key name = "tulajdonosPK">
        <xs:selector xpath="tulajdonos"/>
        <xs:field xpath="tulajdonosId"/>
    </xs:key>

    <xs:key name = "markaPK">
        <xs:selector xpath="marka"/>
        <xs:field xpath="markaId"/>
    </xs:key>

    <xs:key name = "ruhaPK">
        <xs:selector xpath="ruha"/>
        <xs:field xpath="ruhaId"/>
    </xs:key>

    <xs:key name = "vasarloPK">
        <xs:selector xpath="vasarlo"/>
        <xs:field xpath="vasarloId"/>
    </xs:key>

    <xs:key name = "szamlaPK">
        <xs:selector xpath="szamla"/>
        <xs:field xpath="szamlaId"/>
    </xs:key>

    <!-- Idegen kulcsok -->
    <xs:keyref refer="tulajdonosPK" name = "markatulajdonosFK">
        <xs:selector xpath="marka"/>
        <xs:field xpath="tulajdonosFk"/>
    </xs:keyref>

    <xs:keyref refer="ruhaPK" name = "ruhamarkaFK">
        <xs:selector xpath="marka"/>
        <xs:field xpath="ruhaFk"/>
    </xs:keyref>

    <xs:keyref refer="ruhaPK" name = "ruhavasarloFK">
        <xs:selector xpath="vasarlo"/>
        <xs:field xpath="ruhaFk"/>
    </xs:keyref>

    <xs:keyref refer="vasarloPK" name = "vasarlovasarloFK">
        <xs:selector xpath="vasarlo"/>
        <xs:field xpath="vasarloFk"/>
    </xs:keyref>

```



```
<!-- 1:1 kapcsolat -->
<xs:unique name="unique_Szamla">
  <xs:selector xpath="szamla"/>
  <xs:field xpath="Sz_V_Id"/>
</xs:unique>

</xs:element>

</xs:schema>
```

2.feladat

A második feladatban az első feladatban elkészült XMLSZ7MGG.xml file adataival dolgozunk.

Ezeket adatolvasást, adatmódosítást és adatok lekérdezését végzünk.

A kód részletes dokumentációt tartalmaz az adott elemekkel végzett műveletekről.

2a) Adatolvasás

```
package hu.domparse.sz7mvg;  
  
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;  
  
public class DOMReadSZ7MVG {  
    public static void main(String[] args) throws  
ParserConfigurationException, SAXException, IOException {  
        //file nev megadása  
        File file = new File("XMLSZ7MVG.xml");  
  
        //kesobb hasznalt valtoz deklaralasa  
        String prev="";  
        //parse  
        DocumentBuilderFactory factory =  
DocumentBuilderFactory.newInstance();  
        DocumentBuilder dBuilder = factory.newDocumentBuilder();  
  
        Document doc = dBuilder.parse(file);  
  
        doc.getDocumentElement().normalize();  
        //gyokerelem kiiratas  
        System.out.println("Root element: " +  
doc.getDocumentElement().getNodeName());  
        //gyerekelemek mentese listaba  
        NodeList nList = (NodeList) doc.getDocumentElement();  
  
        for (int i = 0; i < nList.getLength(); i++) {  
            Node node = nList.item(i);  
  
            if(!node.getNodeName().equals("#text")) {  
                System.out.println("\n");  
                System.out.println("Current element: " +  
node.getNodeName());  
            }  
  
            if (node.getNodeType() == Node.ELEMENT_NODE) {  
                Element elem = (Element) node;  
                //id kiiratas  
                String id = elem.getAttribute(node.getNodeName() + "Id");  
                NodeList childs = elem.getChildNodes();
```



```

        public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException {
    //file nev megadasa
    File file = new File("XMLSZ7MGG.xml");
    //parse
    DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

    Document doc = dBuilder.parse(file);
    doc.getDocumentElement().normalize();
    //gyokerelem kiiratas
    System.out.print("Root element: ");
    System.out.println(doc.getDocumentElement().getNodeName());
    //ruhak mentese listaba
    NodeList nList = doc.getElementsByTagName("ruha");
    System.out.println("-----");

    //ruhak es gyerek elemeik megjelenitese a consolon
    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i);
        System.out.println("\nCurrent Element : " +
node.getNodeName());
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) node;
            System.out.println(elem.getAttribute("ruhaId"));
            NodeList nList2 = elem.getChildNodes();
            for (int j = 0; j < nList2.getLength(); j++) {
                Node node2 = nList2.item(j);
                if (node2.getNodeType() == Node.ELEMENT_NODE) {
                    System.out.println(node2.getNodeName() + " : " +
node2.getTextContent());
                }
            }
        }
    }
}

```

2c) Adatlekérdezés

```

package hu.dompars.s7magg;

import org.w3c.dom.*;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import java.io.File;

public class DOMModifySZ7MGG {

    public static void main(String[] args) {

        try {
            //file nev megadasa, parse

```

```

        File inputFile = new File("XMLSZ7MGG_toModified.xml");
        DocumentBuilderFactory docFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
        Document doc = docBuilder.parse(inputFile);

        //az elso es a masodik ruha adatainak mentese
        Node ruha = doc.getElementsByTagName("ruha").item(0);
        Node ruha1 = doc.getElementsByTagName("ruha").item(1);
        //gyokerelem mentese
        Node boltom = doc.getFirstChild();

        //elso ruhaId modositasa
        NamedNodeMap attr = ruha.getAttributes();
        Node nodeAttr = attr.getNamedItem("ruhaId");
        nodeAttr.setTextContent("5");

        //az elso ruha kategoriajanak modositasa T-Shirts-ról Polo-ra
        NodeList list = ruha.getChildNodes();
        for (int i = 0; i < list.getLength(); i++) {
            Node node = list.item(i);
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;
                if ("kategoria".equals(elem.getNodeName())) {
                    if ("T-Shirts".equals(elem.getTextContent())) {
                        elem.setTextContent("Polo");
                    }
                }
            }
        }

        //a masodik ruha nevenek modositasa DamnTheBestHoodie-rol
        KenyelmesPulcsi-re
        NodeList list1 = ruha1.getChildNodes();
        for (int i = 0; i < list1.getLength(); i++) {
            Node node1 = list1.item(i);
            if (node1.getNodeType() == Node.ELEMENT_NODE) {
                Element elem1 = (Element) node1;
                if ("nev".equals(elem1.getNodeName())) {
                    if ("DamnTheBestHoodie".equals(elem1.getTextContent())) {
                        elem1.setTextContent("KenyelmesPulcsi");
                    }
                }
            }
        }

        //szamlak torlese
        NodeList childNodes = boltom.getChildNodes();
        for(int i = 0; i < childNodes.getLength(); i++) {
            Node node = childNodes.item(i);

            if("szamla".equals(node.getNodeName()))
                boltom.removeChild(node);
        }

        //megjelenites a consolon
        TransformerFactory transformerFactory =
TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(doc);

```

```
        System.out.println("-----New File-----");  
        StreamResult consoleResult = new StreamResult(System.out);  
        transformer.transform(source, consoleResult);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
}
```