

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **László Levente**

Neptunkód: **ESTJ78**

A feladat leírása:

A feladatom bemutatja az online vásarlast egy viragboltbol.

Az ER modell vegigvezeti a vasarlas menetet egészen a termék kiválasztásától a megrendelésig.

A vásárló miután kiválasztotta a terméket amelynek van neve, ára, készlete és kódja (nid), elhelyezi a kosárba ami itt megfelel a rendelés részleteknek.

A rendelés részleteknek lesz id-je és egy mennyiség mezoje amelyben a nid (növény id-je) alapján fogjuk tudni, hogy miből melyik termékből mennyit szeretnénk.

Itt természetesen úgy tudja változtatni a mennyiséget, majd leadja a rendelést.

A rendelést a rendelés tábla fogja rögzíteni amelynek lesz egy id-je, végösszege, dátuma és hogy kinek mit a többi tábla idegenkulcsaiból fogja tudni.

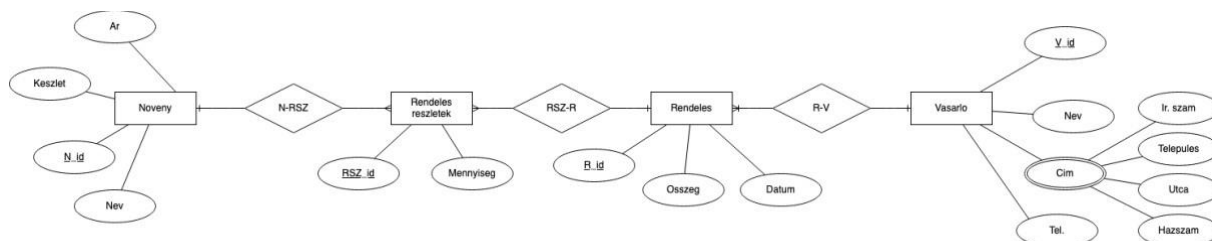
Ezek az idegenkulcsok a rendelés részleteinek az id-je és a vásárló id-je lesz.

Meg fogja kapni a rendelés részleteit és a vásárló adatait.

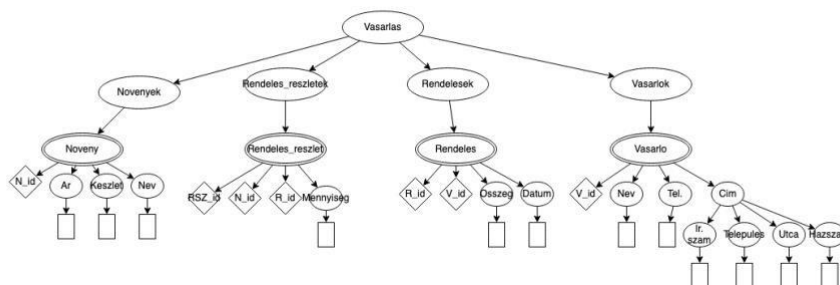
A vásárlónak természetesen van neve, telefonszáma, címe ami egy összetett típus: tartalmazza az irányítószámot, település nevet, utcát és a házszámot.

1 feladat

1a) Az adatbázis ER modell:



1b) Az adatbázis konvertálása XDM modellre:



1c) Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<vasarlas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XSDestj78.xsd">
  <novenyek>
    <noveny nid="1">
      <ar>550</ar>
      <keszlet>10</keszlet>
      <nev>lilium</nev>
    </noveny>
    <noveny nid="2">
      <ar>1200</ar>
      <keszlet>5</keszlet>
      <nev>fikusz</nev>
    </noveny>
    <noveny nid="3">
      <ar>670</ar>
      <keszlet>15</keszlet>
      <nev>pafrany</nev>
    </noveny>
  </novenyek>
  <rendeles_reszletek>
    <rendeles_reszlet rszid="1" nid="1" rid="1">
      <mennyiseg>2</mennyiseg>
    </rendeles_reszlet>
    <rendeles_reszlet rszid="2" nid="2" rid="1">
      <mennyiseg>1</mennyiseg>
    </rendeles_reszlet>
    <rendeles_reszlet rszid="3" nid="3" rid="2">
      <mennyiseg>2</mennyiseg>
    </rendeles_reszlet>
    <rendeles_reszlet rszid="4" nid="1" rid="2">
      <mennyiseg>1</mennyiseg>
    </rendeles_reszlet>
    <rendeles_reszlet rszid="5" nid="2" rid="3">
      <mennyiseg>2</mennyiseg>
    </rendeles_reszlet>
    <rendeles_reszlet rszid="6" nid="3" rid="3">
      <mennyiseg>1</mennyiseg>
    </rendeles_reszlet>
  </rendeles_reszletek>
  <rendelesek>
    <rendeles rid="1" vid="1">
      <osszeg>2300</osszeg>
      <datum>
        <ev>2020</ev>
        <honap>10</honap>
        <nap>22</nap>
      </datum>
    </rendeles>
    <rendeles rid="2" vid="2">
      <osszeg>6700</osszeg>
      <datum>
        <ev>2020</ev>
        <honap>11</honap>
        <nap>27</nap>
      </datum>
    </rendeles>
    <rendeles rid="3" vid="3">
      <osszeg>4500</osszeg>
      <datum>
        <ev>2020</ev>
        <honap>06</honap>
        <nap>10</nap>
      </datum>
    </rendeles>
  </rendelesek>
  <vasarlok>
    <vasarlo vid="1">
      <nev>Kis Matyas</nev>
      <tel>06201237654</tel>
      <cim>
        <irszam>3535</irszam>
        <telepuless>Miskolc</telepuless>
        <utca>Polgar Jenő</utca>
        <hazszam>11</hazszam>
      </cim>
    </vasarlo>
    <vasarlo vid="2">
      <nev>Nagy Andras</nev>
      <tel>06304561254</tel>
      <cim>
        <irszam>3520</irszam>
        <telepuless>Miskolc</telepuless>
        <utca>Petőfi</utca>
        <hazszam>43</hazszam>
      </cim>
    </vasarlo>
    <vasarlo vid="3">
      <nev>Fekete Janos</nev>
      <tel>06709472649</tel>
      <cim>
        <irszam>3530</irszam>
        <telepuless>Miskolc</telepuless>
        <utca>Arany</utca>
        <hazszam>2</hazszam>
      </cim>
    </vasarlo>
  </vasarlok>
</vasarlas>
```

1d) Az XML dokumentum alapján XMLSchema készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs=
"http://www.w3.org/2001/XMLSchema">
  <xs:element name="vasarlas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="novenyek">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="noveny" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:short" name="ar"/>
                    <xs:element type="xs:byte" name="keszlet"/>
                    <xs:element type="xs:string" name="nev"/>
                  </xs:sequence>
                  <xs:attribute type="xs:byte" name="nid" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="rendeles_reszletek">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="rendeles_reszlet" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:byte" name="mennyiseg"/>
                  </xs:sequence>
                  <xs:attribute type="xs:byte" name="rszid" use="optional"/>
                  <xs:attribute type="xs:byte" name="nid" use="optional"/>
                  <xs:attribute type="xs:byte" name="rid" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="rendelesek">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="rendeles" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:short" name="osszeg"/>
                    <xs:element name="datum">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:short" name="ev"/>
                          <xs:element type="xs:byte" name="honap"/>
                          <xs:element type="xs:byte" name="nap"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute type="xs:byte" name="rid" use="optional"/>
                  <xs:attribute type="xs:byte" name="vid" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="vasarlok">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="vasarlo" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="nev"/>
                    <xs:element type="xs:long" name="tel"/>
                    <xs:element name="cim">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:short" name="irszam"/>
                          <xs:element type="xs:string" name="telepules"/>
                          <xs:element type="xs:string" name="utca"/>
                          <xs:element type="xs:byte" name="hazszam"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute type="xs:byte" name="vid" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

2 feladat

2a)

```
package hu.domparse.estj78;

import java.io.File;

import javax.xml.*;
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 javax.xml.xpath.XPath;
import javax.xml.xpath.XPathFactory;

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

import hu.domparse.estj78.DOMModifyEstj78;;

public class DOMReadEstj78 {
    private static void printNode(Node r) {
        // Node nevének kiírása
        if (r.getNodeName() != "#text") {
            System.out.println(r.getNodeName());
        }
        // Gyerekek Node listába helyezése
        NodeList children = r.getChildNodes();
        for (int i = 0; i < children.getLength(); i++) {
            Node child = children.item(i);
            boolean isComplex = child.getTextContent().contains("\n");
            // Attribute kiírása
            if (child.hasAttributes()) {
                NamedNodeMap attributes = child.getAttributes();
                int numAttrs = attributes.getLength();
                for (int j = 0; j < numAttrs; j++) {
                    Attr attr = (Attr) attributes.item(j);
                    String attrName = attr.getNodeName();
                    String attrValue = attr.getNodeValue();
                    System.out.println(" " + attrName + " : " + attrValue);
                }
            }
            // Név és tartalom kiírása
            if (isComplex) {
                printNode(child);
            } else {
                System.out.print(" " + child.getNodeName());
                System.out.println(": " + child.getTextContent());
            }
        }
    }

    public static void main(String[] args) {
        try {

            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            Document doc = dbBuilder.parse("webaruhaz.xml");
            String filepath = "./webaruhaz.xml";
            doc.getDocumentElement().normalize();

            // XPath
            XPath xPath = XPathFactory.newInstance().newXPath();

            Element root = doc.getDocumentElement();
            printNode(root);
            DOMModifyEstj78.modifyPrices(root);
            printNode(root);
            // Módosított xml mentése
            TransformerFactory transformerFactory = TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            DOMSource source = new DOMSource(doc);
            StreamResult result = new StreamResult(new File(filepath));
            transformer.transform(source, result);

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

2b)

```
package hu.dompars.ejt78;

import javax.xml.*;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class DOMModifyEjt78 {
    public static void modifyPrices(Node root) {
        try {
            System.out.println(root.getNodeType());
            NodeList els = root.getChildNodes();
            for (int j = 0; j < els.getLength(); j++) {
                // Vegigmegy a root gyerekein
                if (els.item(j).getNodeName() == "novenyek") {
                    for (int i = 0; i < els.item(j).getChildNodes().getLength(); i++) {
                        // Megtalalta a novenyeket most annak gyerekein megy végig
                        if (els.item(j).getChildNodes().item(i).getNodeName() == "noveny") {
                            for (int k = 0; k < els.item(j).getChildNodes().item(i).getChildNodes().getLength(); k++) {
                                // Noveny gyerekeinek megkeresi az arakat majd modositja 200-al
                                if (els.item(j).getChildNodes().item(i).getChildNodes().item(k).getNodeName() == "ar") {
                                    int ar = Integer.parseInt(els.item(j).getChildNodes().item(i).getChildNodes().item(k).getTextContent());
                                    els.item(j).getChildNodes().item(i).getChildNodes().item(k).setContents(String.valueOf(ar + 200));
                                }
                            }
                        }
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```