

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **László Levente**

Neptunkód: **ESTJ78**

## A feladat leírása:

A feladatom bemutatja az online vásárlást egy viragboltból.

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

Itt ugye tudja változtatni a mennyiséget, majd leadja a rendelest.

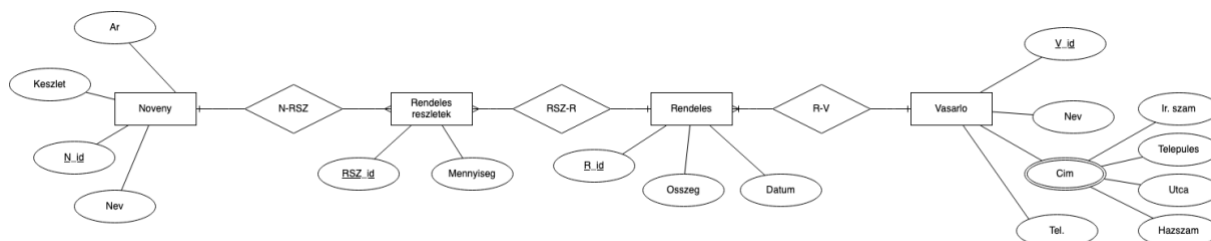
A rendelset a rendelés tábla fogja rögzíteni amelynek lesz egy vegösszege, dátuma és hogy kinek mit a többi tábla idegenkulcsaiból fogja tudni.

Meg fogja kapni a rendelés részleteit és a vásárló adatait.

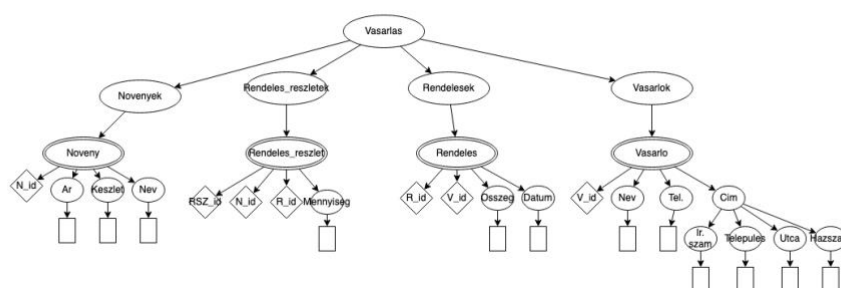
A vásárlónak természetesen van neve, telefonszáma, címe ami egy összetett típus: tartalmazza az irányítószámot, település nevet, utcát és a házszámot.

## 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>liliom</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>
      <telepules>Miskolc</telepules>
      <utca>Polgar Jeno</utca>
      <hazszam>11</hazszam>
    </cim>
  </vasarlo>
  <vasarlo vid="2">
    <nev>Nagy Andras</nev>
    <tel>06304561254</tel>
    <cim>
      <irszam>3520</irszam>
      <telepules>Miskolc</telepules>
      <utca>Petofi</utca>
```

```

        <hazszam>43</hazszam>
    </cim>
</vasarlo>
<vasarlo vid="3">
    <nev>Fekete Janos</nev>
    <tel>06709472649</tel>
    <cim>
        <irszam>3530</irszam>
        <telepules>Miskolc</telepules>
        <utca>Arany</utca>
        <hazszam>2</hazszam>
    </cim>
</vasarlo>
</vasarlok>
</vasarlas>

```

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

```

<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: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:complexType>
      <xs:attribute type="xs:byte" name="vid" use="optional"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## 2 feladat

**2a)**

```

package hu.domparse.estj78;

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;

import hu.domparse.estj78.DOMModifyEstj78;;

public class DOMReadEstj78 {
    private static void printNode(Node r) {
        NodeList els = r.getChildNodes();
        for (int j = 0; j < els.getLength(); j++) {
            if (els.item(j).getNodeName() != "#text") {
                Node n = els.item(j);
                System.out.println(n.getNodeName());
                if (n.hasChildNodes()) {
                    for (int i = 0; i < n.getChildNodes().getLength(); i++) {
                        Node s = n.getChildNodes().item(i);
                        if (s.getNodeType() == Node.ELEMENT_NODE) {
                            System.out.println("\t" + s.getNodeName() + ": " + s.getTextContent());
                        }
                    }
                }
            }
        }
    }

    public static void main(String[] args) {
        try {
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db;
            db = dbf.newDocumentBuilder();
            Document document = db.parse("webaruhaz.xml");
            // root data
            Element root = document.getDocumentElement();
            printNode(root);
            DOMModifyEstj78.modifyPrices(root);
            printNode(root);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

2b)



```

package hu.dompars.e stj78;

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 DOMModifyEstj78 {
    public static void modifyPrices(Node root) {
        try {
            System.out.println(root.getNodeType());
            NodeList els = root.getChildNodes();
            for (int j = 0; j < els.getLength(); j++) {
                if (els.item(j).getNodeName() == "novenyek") {
                    for (int i = 0; i < els.item(j).getChildNodes().getLength(); i++) {
                        if (els.item(j).getChildNodes().item(i).getNodeName() == "noveny") {
                            for (int k = 0; k < els.item(j).getChildNodes().item(i).getChildNodes().getLength(); k++) {
                                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).setTextContent(String.valueOf(ar + 200));
                                }
                            }
                        }
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```