

Adatkezelés XML-es környezetben Jegyzőkönyv



Név: Huzynets Erik

Neptun kód: BJ8BDJ

Szak: gazdaságinformatika

Tankör: G3-BGI

Gyakorlat ideje: Kedd 14-16

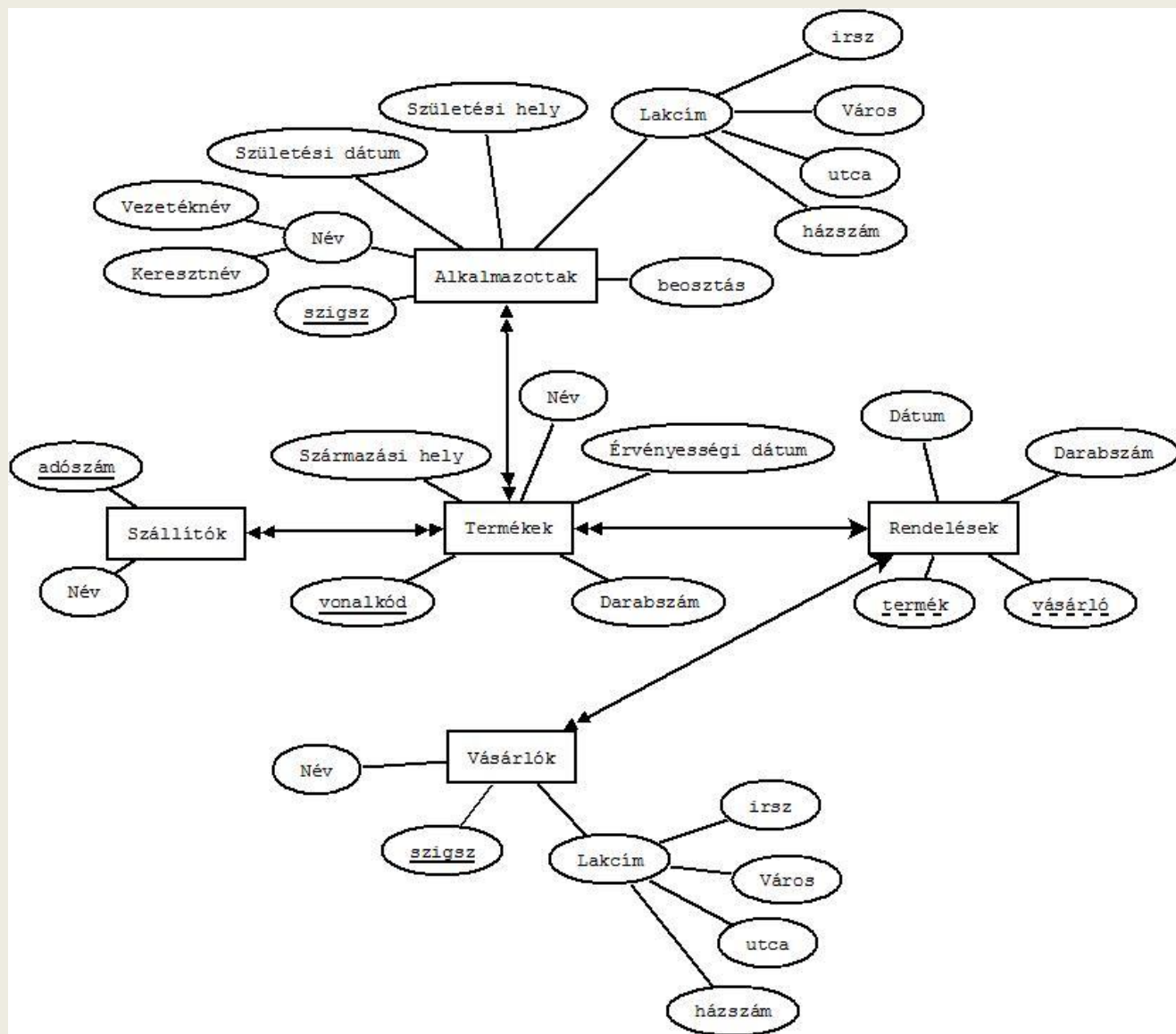
Gyakorlatvezető: Barabás Péter

Feladat leírás

Beadandó feladtkomban egy zöldségbolt adatbázisát tervezem meg. Az adatbázisban szerepelnek majd az alkalmazottokról, a szállítókról, termékekről, vásárlókról és rendelésekről adatok.

Természetesen ez nem egy sima zöldségbolt adatbázisa, hanem egy középszerű vállalat ahonnan a vásárlók nagy tételben vásárolhatnak majd (a kis tétel is megengedett vásárlás a boltban). Minden zöldség be van csomagolva és van vonalkódja, így tegyük fel, ha egy vásárló vásárol 2db paprikát (az adott boltban feltétel hogy 1kg-os csomagokban vannak a zöldségek), akkor az adatbázisból történő darabszámot értelemszerűen csökkenteni kell viszont a vásárló adatait nem kell felegetezni. Szállító zöldség szállatása esetén az adott termék darabszámát növelni kell. Minden zöldségnek van származási helye és érvényességi dátuma. Rendelés esetén a vásárló neve és a megrendelt terméket rögzíteni kell az adatbázisban, ha a rendelt termék elfogyott, akkor a rendelést nem lehet felvenni.

ER modell



XSD fájl

```

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

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

  <xs:complexType name="lakcimtipus">
    <xs:sequence>
      <xs:element name="irsz">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:minInclusive value="1000" />
            <xs:maxInclusive value="9999" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="varos" type="xs:string" />
      <xs:element name="utca" type="xs:string" />
      <xs:element name="hazzsam">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:maxInclusive value="1" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="datum_tipus">
    <xs:sequence>
      <xs:element name="ev">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:maxLength value="4" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="honap">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:maxInclusive value="12" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="nap">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:maxInclusive value="31" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

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

    <xs:complexType name="alkalmazottak_tipus">
        <xs:sequence>
            <xs:element name="nev" type="nevtipus" />
            <xs:element name="szuletesi_hely" type="xs:string" />
            <xs:element name="szuletesi_datum" type="datum_tipus" />
            <xs:element name="lakcim" type="lakcimtipus" />
            <xs:element name="beosztas">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="főnök" />
                        <xs:enumeration value="eladó" />
                        <xs:enumeration value="raktáros" />
                        <xs:enumeration value="minőségellenőrző"
/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="alkalmazott_szigsz" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="[0-9]{6}[A-Z]{2}" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="termekref" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9]{13}" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="szallitok_tipus">
        <xs:sequence>
            <xs:element name="nev" type="xs:string" />
        </xs:sequence>
        <xs:attribute name="adoszam" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9]{10}" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="termek_vonalkod" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9]{13}" />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

    <xs:complexType name="termekek_tipus">
        <xs:sequence>
            <xs:element name="nev" type="xs:string" />

```

```

        <xs:element name="darabszam" type="xs:integer" />
        <xs:element name="ervenyesseg" type="datum_tipus" />
        <xs:element name="szarmazasi_hely" type="xs:string" />
    </xs:sequence>

    <xs:attribute name="vonalkod" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:pattern value="[0-9]{13}" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="alkalmazottref" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="[0-9]{6}[A-Z]{2}" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>

<xs:complexType name="vasarlok_tipus">
    <xs:sequence>
        <xs:element name="nev" type="nevtipus" />
        <xs:element name="lakcim" type="lakcimtipus" />
    </xs:sequence>
    <xs:attribute name="vasarlo_szigsz" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="[0-9]{6}[A-Z]{2}" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>

<xs:complexType name="rendelesek_tipus">
    <xs:sequence>
        <xs:element name="datum" type="datum_tipus" />
        <xs:element name="termek_db" type="xs:integer" />
    </xs:sequence>
    <xs:attribute name="termekref" type="xs:integer" use="required"
/>
    <xs:attribute name="vasarloref" type="xs:integer"
use="required">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="[0-9]{6}[A-Z]{2}" />
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>

<xs:element name="Zöldseges">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="alkalmazott"
type="alkalmazottak_tipus" maxOccurs="unbounded" />

```

```

        <xs:element name="vasarlo" type="vasarlok_tipus"
maxOccurs="unbounded" />
        <xs:element name="termek" type="termekek_tipus"
maxOccurs="unbounded" />
        <xs:element name="rendeles" type="rendelesek_tipus"
/>
        <xs:element name="szallito" type="szallitok_tipus"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:key name="key1">
    <xs:selector xpath="alkalmazott" />
    <xs:field xpath="@alkalmazott_szigsz" />
</xs:key>

<xs:key name="key2">
    <xs:selector xpath="vasarlo" />
    <xs:field xpath="@vasarlo_szigsz" />
</xs:key>

<xs:key name="key3">
    <xs:selector xpath="termek" />
    <xs:field xpath="@vonalkod" />
</xs:key>

<xs:key name="key4">
    <xs:selector xpath="szallito" />
    <xs:field xpath="@adoszam" />
</xs:key>

<xs:keyref name="keyref1" refer="key3">
    <xs:selector xpath="szallito" />
    <xs:field xpath="@termek_vonalkod" />
</xs:keyref>

<xs:keyref name="keyref2" refer="key3">
    <xs:selector xpath="rendeles" />
    <xs:field xpath="@termekref" />
</xs:keyref>

<xs:keyref name="keyref3" refer="key2">
    <xs:selector xpath="rendeles" />
    <xs:field xpath="@vasarloref" />
</xs:keyref>

<xs:keyref name="keyref5" refer="key1">
    <xs:selector xpath="termek" />
    <xs:field xpath="@alkalmazottref" />
</xs:keyref>

<xs:keyref name="keyref6" refer="key3">
    <xs:selector xpath="alkalmazott" />
    <xs:field xpath="@termekref" />
</xs:keyref>

</xs:element>

</xs:schema>

```

XML fájl

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="beadando.xsl"?>
<Zöldseges xmlns="huzynets/beadando"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="huzynets/beadando beadando.xsd ">
  <alkalmazott alkalmazott_szigsz="111111RR" termékref="1234567891234">
    <nev>
      <vezeteknev>Domokos</vezeteknev>
      <keresztnev>Lajos</keresztnev>
    </nev>
    <szuletési_hely>Miskolc</szuletési_hely>
    <szuletési_datum>
      <ev>1970</ev>
      <honap>4</honap>
      <nap>1</nap>
    </szuletési_datum>
    <lakcim>
      <irsz>5000</irsz>
      <varos>Tiszafüred</varos>
      <utca>Homok utca</utca>
      <hazszam>9</hazszam>
    </lakcim>
    <beosztas>raktáros</beosztas>
  </alkalmazott>
  <alkalmazott alkalmazott_szigsz="111113RR" termékref="1234567891234">
    <nev>
      <vezeteknev>Molnár</vezeteknev>
      <keresztnev>Gábor</keresztnev>
    </nev>
    <szuletési_hely>Miskolc</szuletési_hely>
    <szuletési_datum>
      <ev>1969</ev>
      <honap>3</honap>
      <nap>5</nap>
    </szuletési_datum>
    <lakcim>
      <irsz>4400</irsz>
      <varos>Nyíregyháza</varos>
      <utca>Dózsa György út</utca>
      <hazszam>20</hazszam>
    </lakcim>
    <beosztas>főnök</beosztas>
  </alkalmazott>
  <alkalmazott alkalmazott_szigsz="111115QR" termékref="1234567891234">
    <nev>
      <vezeteknev>Kovács</vezeteknev>
      <keresztnev>Péter</keresztnev>
    </nev>
    <szuletési_hely>Miskolc</szuletési_hely>
    <szuletési_datum>
      <ev>1975</ev>
      <honap>1</honap>
      <nap>28</nap>
    </szuletési_datum>
    <lakcim>
      <irsz>3515</irsz>
      <varos>Miskolc</varos>
      <utca>Boldog utca</utca>
  </alkalmazott>

```



```
    <hazszam>10</hazszam>
  </lakcim>
  <beosztas>raktáros</beosztas>
</alkalmazott>
<alkalmazott alkalmazott_szigsz="111665QY" termékref="1234567891234">
  <nev>
    <vezeteknev>Magyar</vezeteknev>
    <keresztnev>László</keresztnev>
  </nev>
  <szuletési_hely>Vásárosnamény</szuletési_hely>
  <szuletési_datum>
    <ev>1991</ev>
    <honap>06</honap>
    <nap>20</nap>
  </szuletési_datum>
  <lakcim>
    <irsz>3515</irsz>
    <varos>Miskolc</varos>
    <utca>Március utca</utca>
    <hazszam>55</hazszam>
  </lakcim>
  <beosztas>minőségellenőrző</beosztas>
</alkalmazott>
<vasarlo vasarlo_szigsz="623456AA">
  <nev>
    <vezeteknev>Milner</vezeteknev>
    <keresztnev>Gábor</keresztnev>
  </nev>
  <lakcim>
    <irsz>3515</irsz>
    <varos>Miskolc</varos>
    <utca>Egyetemváros</utca>
    <hazszam>4</hazszam>
  </lakcim>
</vasarlo>
<vasarlo vasarlo_szigsz="623499CA">
  <nev>
    <vezeteknev>Ács</vezeteknev>
    <keresztnev>Miklós</keresztnev>
  </nev>
  <lakcim>
    <irsz>1350</irsz>
    <varos>Budapest</varos>
    <utca>Parlament út</utca>
    <hazszam>22</hazszam>
  </lakcim>
</vasarlo>
<vasarlo vasarlo_szigsz="188499QQ">
  <nev>
    <vezeteknev>Barabás</vezeteknev>
    <keresztnev>Péter</keresztnev>
  </nev>
  <lakcim>
    <irsz>3535</irsz>
    <varos>Miskolc</varos>
    <utca>Magyar út</utca>
    <hazszam>33</hazszam>
  </lakcim>
</vasarlo>
<termek alkalmazottref="891234AE" vonalkod="1234567890123">
  <nev>Paprika</nev>
```

```
<darabszam>50</darabszam>
<ervenyesseg>
  <ev>2015</ev>
  <honap>1</honap>
  <nap>1</nap>
</ervenyesseg>
<szarmazasi_hely>Magyarország</szarmazasi_hely>
</termek>
<termek alkalmazottref="891234AE" vonalkod="9988867890123">
  <nev>Saláta</nev>
  <darabszam>2000</darabszam>
  <ervenyesseg>
    <ev>2012</ev>
    <honap>12</honap>
    <nap>21</nap>
  </ervenyesseg>
  <szarmazasi_hely>Magyarország</szarmazasi_hely>
</termek>
<termek alkalmazottref="891234AE" vonalkod="1234567890223">
  <nev>Uborka</nev>
  <darabszam>50</darabszam>
  <ervenyesseg>
    <ev>2012</ev>
    <honap>12</honap>
    <nap>31</nap>
  </ervenyesseg>
  <szarmazasi_hely>Magyarország</szarmazasi_hely>
</termek>
<termek alkalmazottref="891234AE" vonalkod="1234588880221">
  <nev>Kókuszdió</nev>
  <darabszam>300</darabszam>
  <ervenyesseg>
    <ev>2013</ev>
    <honap>06</honap>
    <nap>10</nap>
  </ervenyesseg>
  <szarmazasi_hely>Japán félsziget</szarmazasi_hely>
</termek>
<termek alkalmazottref="891234AE" vonalkod="1234567893123">
  <nev>Uborka (német)</nev>
  <darabszam>100</darabszam>
  <ervenyesseg>
    <ev>2012</ev>
    <honap>6</honap>
    <nap>6</nap>
  </ervenyesseg>
  <szarmazasi_hely>Németország</szarmazasi_hely>
</termek>
<rendeles termékref="1234567890123" vasarloref="623499CA">
  <datum>
    <ev>2012</ev>
    <honap>05</honap>
    <nap>05</nap>
  </datum>
  <termek_db>30</termek_db>
</rendeles>
<rendeles termékref="1234567890123" vasarloref="623456AA">
  <datum>
    <ev>2012</ev>
    <honap>03</honap>
    <nap>10</nap>
```

```
</datum>
  <termek_db>30</termek_db>
</rendeles>
<rendeles termékref="1234567890223" vasarloref="623456AA">
  <datum>
    <ev>2011</ev>
    <honap>04</honap>
    <nap>30</nap>
  </datum>
  <termek_db>30</termek_db>
</rendeles>
<rendeles termékref="1234567893123" vasarloref="188499QQ">
  <datum>
    <ev>2012</ev>
    <honap>11</honap>
    <nap>05</nap>
  </datum>
  <termek_db>30</termek_db>
</rendeles>
<szallito adoszam="1234512345" termék_vonalkod="1234588880221">
  <nev>Géza kft</nev>
</szallito>
<szallito adoszam="1234511345" termék_vonalkod="1234567890123">
  <nev>Semmi zrt</nev>
</szallito>
<szallito adoszam="1234515345" termék_vonalkod="1234588880221">
  <nev>Kaja zrt</nev>
</szallito>
<szallito adoszam="1666666665" termék_vonalkod="1234567893123">
  <nev>Uborka zrt</nev>
</szallito>
<szallito adoszam="1662222665" termék_vonalkod="1234567890123">
  <nev>Wubi kft</nev>
</szallito>
<szallito adoszam="1666664665" termék_vonalkod="1234567893123">
  <nev>Galóca kft</nev>
</szallito>
</Zöldseges>
```

XSL lekérdezések

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fn="http://www.w3.org/2005/xpath-functions"
xmlns:e="huzynets/beadando"
>
<xsl:template match="e:Zöldseges">
<html>
  <head></head>
  <body>
    <h1>XSL lekérdezések</h1>
    <h3>Készítette: Huzynets Erik</h3><small>Neptun kód: BJ8BDJ<br
/>szak:gazdaságinformatika</small>
    <h2>1.Szállítók nevei és hogy milyen zöldséget szállítottak</h2>
    <table border="1">
      <tr>
        <th>Szállítók neve</th>
        <th>Szállított termék neve</th>
      </tr>
      <xsl:for-each select="e:szallito">
        <tr>
          <td><xsl:value-of select="e:nev" /></td>
          <xsl:variable name="termezkod"
select="@termek_vonalkod" />
          <xsl:for-each select="../e:termek">
            <xsl:if test="$termezkod = @vonalkod">
              <td><xsl:value-of select="e:nev"
/></td>
            </xsl:if>
          </xsl:for-each>
        </tr>
      </xsl:for-each>
    </table>
    <br />
    <h2>2.Azok az alkalmazottak akik Miskolcon laknak és mellé hogy ott is
születtek-e</h2>
    <table border="1">
      <tr>
        <th>Miskolci alkalmazottak</th>
        <th>Ott született?</th>
      </tr>
      <xsl:for-each select="e:alkalmazott[e:lakcim/e:varos =
'Miskolc']">
        <tr>
          <td><xsl:value-of select="e:nev/e:vezeteknev" />
          <xsl:value-of select="e:nev/e:keresztnev"
/></td>
          <xsl:choose>
            <xsl:when test="e:szulesesi_hely =
'Miskolc'">
              <td>Igen</td>
            </xsl:when>
            <xsl:when test="e:szulesesi_hely !=
'Miskolc'">
              <td>Nem</td>
            </xsl:when>
          </xsl:choose>
        </tr>
      </xsl:for-each>
    </table>
  <br />

```

```

<h2>3.22Évnél idősebb alkalmazottak szigsz-a és neve</h2>
<table border="1">
  <tr>
    <th>Sz.i.szám</th>
    <th>Név</th>
  </tr>
  <xsl:for-each select="e:alkalmazott">
    <xsl:if test="e:szuletesi_datum/e:ev < 1990">
      <tr>
        <td><xsl:value-of select="@alkalmazott_szigsz">
          <td>
            <xsl:value-of select="e:nev/e:vezeteknev" />
            <xsl:value-of select="e:nev/e:keresztnev" />
          </td>
        </tr>
      </xsl:if>
    </xsl:for-each>
  </table>

<br /><h2>4.Lejárt termékek vonalkódja</h2>
<xsl:for-each select="e:termek">
  <xsl:if test="e:ervenyesseg/e:ev < 2013">
    <xsl:if test="12 >= e:ervenyesseg/e:honap">
      <xsl:if test="12 >= e:ervenyesseg/e:nap">
        <xsl:value-of select="@vonalkod" /><br />
      </xsl:if>
    </xsl:if>
  </xsl:if>
</xsl:for-each>

<br /><h2>5.Budapesti vásárlók neve</h2>
<xsl:for-each select="e:vasarlo">
  <xsl:if test="e:lakcim/e:varos='Budapest'">
    <xsl:value-of select="e:nev/e:vezeteknev" />&#160;
    <xsl:value-of select="e:nev/e:keresztnev" />
  <br />
</xsl:if>
</xsl:for-each>

<br /><h2>6.A zöldségek nevei és származási helye</h2>
<table border="1">
  <tr>
    <th>Termék neve</th>
    <th>Származási hely</th>
  </tr>
  <xsl:for-each select="e:termek">
    <tr>
      <td>
        <xsl:value-of select="e:nev" />
      </td>
      <td>
        <xsl:value-of select="e:szarmazasi_hely" />
      </td>
    </tr>
  </xsl:for-each>
</table>

<br /><h2>7.Melyik termékből(név) van a legkevesebb db</h2>
<small>Itt a sum helyére természetesen min-t kellene rakni de
valamiért nem működik</small>
<xsl:variable name="mini" select="sum(e:termek/e:darabszam)" />
<xsl:for-each select="e:termek">
  <xsl:if test="e:darabszam = $mini">
    <xsl:value-of select="e:nev" /><br />
  </xsl:if>
</xsl:for-each>

```


<h2>8.Vásárlók neve városonként</h2>

```
<table border="1">
  <tr>
    <th>Név</th>
    <th>Város</th>
  </tr>
  <xsl:for-each select="e:vasarlo">
    <tr>
      <td>
        <xsl:value-of select="e:nev" />
      </td>
      <td>
        <xsl:value-of select="e:lakcim/e:varos" />
      </td>
    </tr>
  </xsl:for-each>
</table>
```


<h2>9.Vásárlók szigszáma és megrendelt termékek valamint a rendelési dátum (hónap/nap)</h2>

```
<table border="1">
  <tr>
    <th>Vásárló</th>
    <th>Termék neve</th>
    <th>Rendelési dátum</th>
  </tr>
  <xsl:for-each select="e:rendeles">
    <tr>
      <td>
        <xsl:variable name="termek_id" select="@termekref" />
        <xsl:variable name="vasarlo_id" select="@vasarloref" />
        <td><xsl:value-of select="$vasarlo_id" /></td>
        <xsl:for-each select="../e:termek">
          <xsl:if test="@vonalkod = $termek_id">
            <td>
              <xsl:value-of select="e:nev" />
            </td>
          </xsl:if>
        </xsl:for-each>
        <td>
          <xsl:value-of select="e:datum/e:honap" />.<xsl:value-of
select="e:datum/e:nap" />
        </td>
      </tr>
    </xsl:for-each>
  </table>
```


<h2>10.Azok a zöldségek vonalkódja és darabszáma amiknek a származási helye Magyarország</h2>

```
<table border="1">
  <tr>
    <th>Vonalkód</th>
    <th>Darabszám</th>
  </tr>
  <xsl:for-each select="e:termek[e:szarmazasi_hely='Magyarország']">
    <tr>
      <td><xsl:value-of select="@vonalkod" /></td>
      <td><xsl:value-of select="e:darabszam" /></td>
    </tr>
  </xsl:for-each>
</table>
```

<h2>11.A főnök beosztású alkalmazottak neve és lakcime</h2>

```
<table border="1">
  <tr>
    <th>Név</th>
    <th>Lakcim</th>
  </tr>
```

```
<xsl:for-each select="e:alkalmazott[e:beosztas='főnök']">
  <tr>
    <td>
      <xsl:value-of select="e:nev/e:vezeteknev" />#160;
      <xsl:value-of select="e:nev/e:keresztnev" />
    </td>
    <td>
      <xsl:value-of select="e:lakcim/e:varos" />, #160;
      <xsl:value-of select="e:lakcim/e:utca" />#160;
      <xsl:value-of select="e:lakcim/e:hazszam" />
    </td>
  </tr>
</xsl:for-each>
</table>
<br /><small>Opera böngészőben tesztelve</small>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Módosító DOM

```

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

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerFactoryConfigurationError;
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 DOM {

    private Document doc;

    public DOM() {
        DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder builder = factory.newDocumentBuilder();
            try {
                try {
                    doc = builder.parse("beadando.xml");
                } catch (SAXException e) {
                    e.printStackTrace();
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        } catch (ParserConfigurationException e) {
            e.printStackTrace();
        }
    }

    //az 111115QR szígszámú alkalmazott az 1234567891234 -es vonalkódú
termékhez
    public void modosit(String alk, String term) {
        NodeList list = doc.getElementsByTagName("alkalmazott");
        for (int i=0; i<list.getLength(); i++) {
            Node d = list.item(i);
            NamedNodeMap nnm = d.getAttributes();
            Node id = nnm.getNamedItem("alkalmazott_szígsz");
            if(id.getTextContent().compareTo(alk) == 0) {
                NodeList nodes = d.getChildNodes();
                for (int j=0; j<nodes.getLength(); j++) {
                    Node n = nodes.item(j);
                    if(n.getNodeName().compareTo("termekref") ==
0) {

```



```
                n.setTextContent(term);
            }
        }
    }

    public void save() {
        DOMSource source = new DOMSource(doc);
        try {
            Transformer transformer =
TransformerFactory.newInstance().newTransformer();

            transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION,
"yes");
                transformer.setOutputProperty(OutputKeys.INDENT, "yes");

            transformer.setOutputProperty("{http://xml.apache.org/xslt}indent-
amount", "2");
                StringWriter writer = new StringWriter();
                StreamResult result = new StreamResult(new
File("beadando.xml"));
                //StreamResult result = new StreamResult(writer);
                transformer.transform(source, result);
                System.out.println(writer.toString());
        } catch (TransformerFactoryConfigurationError e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (TransformerException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        DOM dt = new DOM();
        dt.modosit("111115QR", "1234567891234");
        dt.save();
    }
}
```