JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Üzletlánc

Készítette: **Danyi Kristóf Milán**

Neptunkód: **GQOKMW**

Dátum: 2023-12-03

Tartalomjegyzék

Bevezetés

A feladatom egy üzletlánc adazbázisáról szól, amelynek különböző üzletei vannak, ezek az üzletek minden egy cég alatt vannak. Az adatbázisban eltároljuk a Vásárlók adatait, hozzájuk tartozik egy Megrendelés. A Megrendeléshez külön táblában tároljuk a hozzá tartozó Megrendelt Termékeket. Mivel nincs minden boltban minden termék, így el kell tárolnunk azt is, hogy melyik boltból fogjuk összeszedni a termékeket, hogy sikeresen teljesíteni tudjuk a megrendelést. Az Üzletekről is tárolunk információkat, ugyanis különböző helyen és név alatt lehetnek a cég tulajdonában. Ezek mellett még fontos a Dolgozókat eltárolni, hogy tudjuk ki hol dolgozik, hogyan érhetjük el stb.

Vásárló

A Vásárló adatait eltároljuk: A Telefonszámát, mivel egy vásárlónak több telefonszáma is lesz, így ez egy többértékű tulajdonság lesz. Nevét, azonosítóját, majd a Címét, hogy tudjuk hova kell szállítani a termékeket. A Cím egy összetett tulajdonság lesz, így külön lehet szedni a városokat, utcákat az effektívebb kiszállítás érdekében

Megrendelés

Határidő, amíg legkésőbb a kiszállítást el kell végezni, a megrendelés státuszát, ezzel nyomon lehet követni hogy mely megrendelések vannak kiszállítás alatt / teljesítve. Ugyanitt a megrendelés idejét és a megrendelés azonosítóját. A Vásárló és Megrendelés között 1:1 kapcsolat van, a Megrendelés egy vásárlóhoz tartozik, és egy Vásárlóhoz egy egyedi Megrendelés kulcs.

Megrendelt termék

Itt tároljuk magát a szállítani kívánt termékeket. Eltároljuk a Listaárát, ez az hogy alapból mennyibe kerül a termék, valamint a Leárazást, így később ki tudjuk számítani a tényleges összeget amit a vásárlónak fizetni kell. Emellett a Mennyiséget is, hogy egy Megrendelésbe ugyanabból a termékből mennyit tegyünk, valamint az egyedi kulcsát. A Megrendelés és a Megrendelt termék közt 1:N kapcsolat van, egy Megrendelt termék egy rendeléshez tartozhat, de egy Megrendelés több termékből is állhat.

Üzlet

Eltároljuk az üzletről a releváns információkat. Mivel ez egy üzletlánc adatbázisa, így a cég tulajdonában különböző nevű üzletek lehetnek, ezért a Nevét eltároljuk, Címét hogy könnyen megtaláljuk a Megrendelt terméket, valamint az Üzlet elérhetőségét a Telefonszám alatt, és egy egyedi kulcsot. A Megrendelt termék és az üzlet között 1:N kapcsolat van, az adott termék csak egy boltban van, viszont egy üzletben több termék is lehet amelyet megrendeltek.

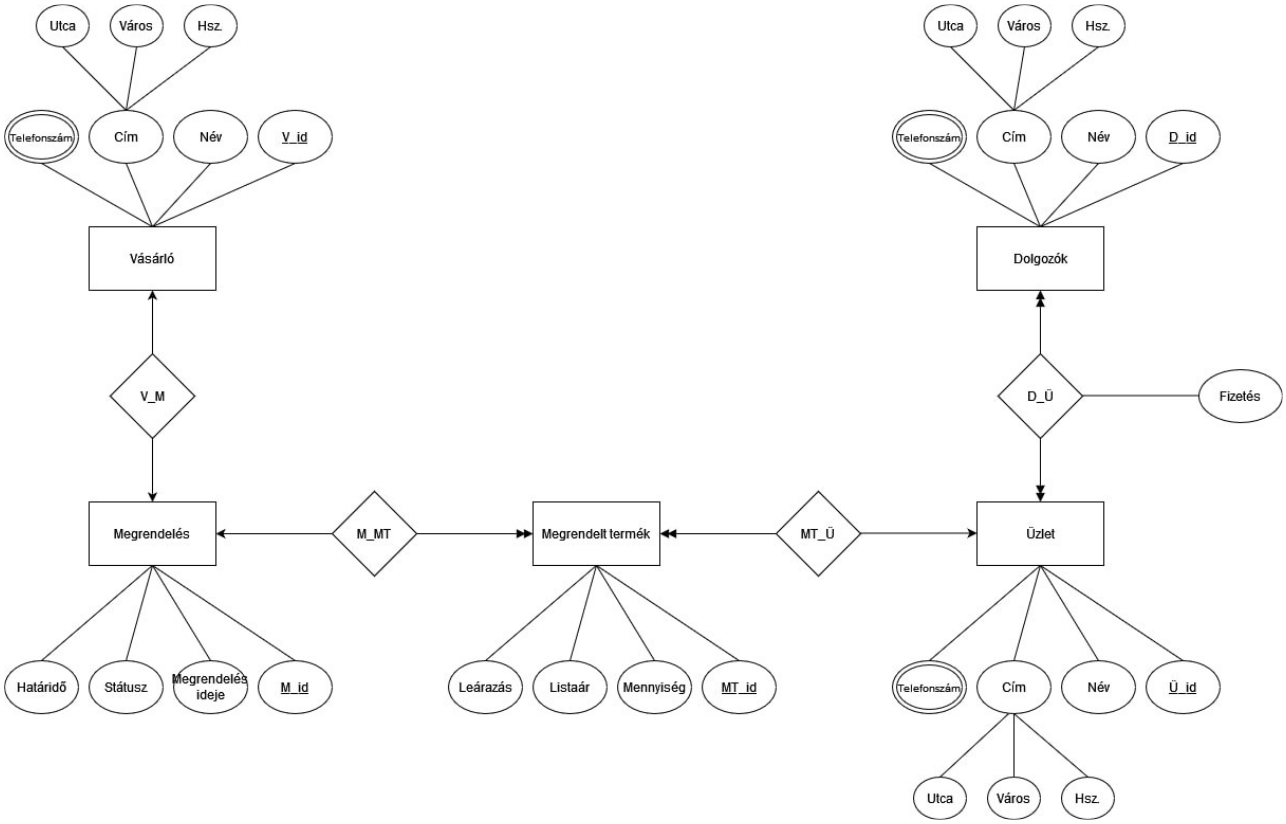
Dolgozók

Eltároljuk a Dolgogozókat akik az üzletláncnak dolgoznak, Nevüket, elérhetőségeiket valamint lakcímüket. A Fizetésüket az Üzlet és Dolgozók kapcsolótábla tulajdonságába tároljuk. N:N kapcsolat lesz a két tábla között, ugyanis egy Üzletben többen is dolgozhatnak, de nincs kizárva, hogy egy ember több Üzletben is dolgozzon. A kapcsolótábla tulajdonsága lesz a Fizetés, így ha valaki több üzletben is dolgozik tudni fogjuk hogy melyik üzletből mennyi fizetést kap.

1. feladat

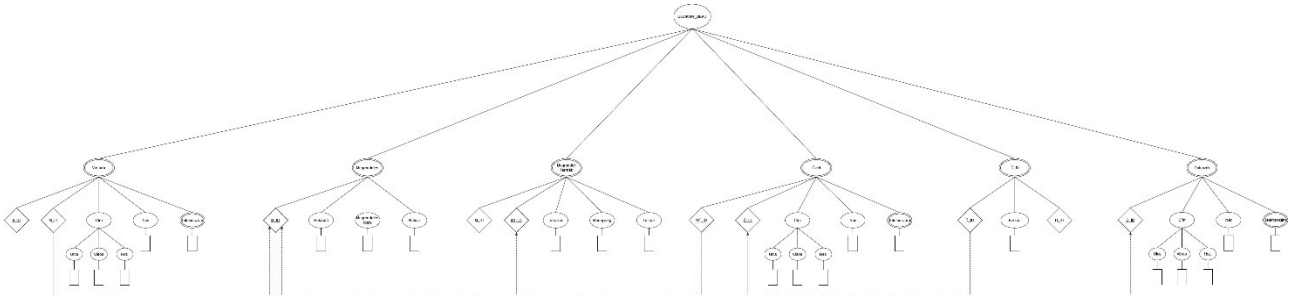
1a) Az adatbázis ER modell tervezése (Legyen legalább 5 egyed, többféle kapcsolat (1:1; 1:N; M:N), tulajdonságok - normál, kulcs, összetett, többértékű. (Csak szerkesztő programmal rajzolt ábra megfelelő, szabványos szimbólumok használata)

A modelljeimet az online Draw.io segítségével készítettem el. Itt látható a fent leírt adatbázis ER modellje (nagyobb méretben a Github-on elérhető):



1b) Az adatbázis konvertálása XDM modellre (Csak szerkesztő programmal rajzolt ábra megfelelő, szabványos szimbólumok használata)

Ezután konvertáltam ezt az ER modellt szintén Draw.io - val egy XDM modellé (szintén elérhető Githubon nagyobb formában)



Az ER modell és XDM hasonló, de az XDM-ben XML dokumentummá alakítjuk.

Az egyedünket egy elemmé alakítjuk,

Egyszerű tulajdonságot szöveg elemmé,

Kulcs tulajdonságot egy elemjellemzővel,

Összetett tulajdonságot a szülőtulajdonságban, gyerekelemekkel tároljuk,

Többértékű tulajdonság egy gyermekelem, amelyet minden értékkel ismétlünk,

Idegen kulcs egy elemjellemző lesz,

1:N kapcsolat szintén elemjellemző,

N:N kapcsolatnak külön kapcsoló elemeket, az idegen kulcsok elemjellemzők lesznek.

1c) Az XDM modell alapján XML dokumentum készítése: (Ide kerül az XML kódja!)

Itt teszem meg az egyedek példányosítását, elemjellemzőbe kulcsaikat, XDM-nek megfelelő struktúrában.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<GQOKMW_BEADANDO xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="XMLSchemaGQOKMW.xsd">

    <!-- Vásárlók -->
    <Vasarlo V_id="1" M_id="01">
        <Nev>Ádám</Nev>
        <Cim>
            <Utca>Árpád utca</Utca>
            <Varos>Budapest</Varos>
            <Hazszam>1</Hazszam>
        </Cim>
        <Telefonszam>0630-1234567</Telefonszam>
    </Vasarlo>

    <Vasarlo V_id="2" M_id="02">
        <Nev>Ádám</Nev>
        <Cim>
            <Utca>Árpád utca</Utca>
            <Varos>Budapest</Varos>
            <Hazszam>1</Hazszam>
        </Cim>
        <Telefonszam>0630-1234567</Telefonszam>
    </Vasarlo>

    <Vasarlo V_id="3" M_id="03">
        <Nev>Ádám</Nev>
        <Cim>
            <Utca>Árpád utca</Utca>
            <Varos>Budapest</Varos>
            <Hazszam>1</Hazszam>
        </Cim>
        <Telefonszam>0630-1234567</Telefonszam>
        <Telefonszam>0630-7654321</Telefonszam>
    </Vasarlo>
```

```xml
<!-- Megrendelés -->

<Megrendeles M_id="01">
    <Hatarido>2021-02-03</Hatarido>
    <Megrendelesideje>2021-02-01</Megrendelesideje>
    <Statusz>Teljesítve</Statusz>
</Megrendeles>

<Megrendeles M_id="02">
    <Hatarido>2021-02-07</Hatarido>
    <Megrendelesideje>2021-02-02</Megrendelesideje>
    <Statusz>Szállítás alatt</Statusz>
</Megrendeles>

<Megrendeles M_id="03">
    <Hatarido>2021-03-05</Hatarido>
    <Megrendelesideje>2021-03-01</Megrendelesideje>
    <Statusz>Teljesítve</Statusz>
</Megrendeles>
```

```xml
<!-- Megrendelt termék-->

<Megrendelttermék M_id="01" MT_id="11">
    <Learazas>0.2</Learazas>
    <Mennyiség>2</Mennyiség>
    <Listaar>25000</Listaar>
</Megrendelttermék>

<Megrendelttermék M_id="02" MT_id="12">
    <Learazas>0.2</Learazas>
    <Mennyiség>2</Mennyiség>
    <Listaar>25000</Listaar>
</Megrendelttermék>

<Megrendelttermék M_id="03" MT_id="13">
    <Learazas>0.2</Learazas>
    <Mennyiség>2</Mennyiség>
    <Listaar>25000</Listaar>
</Megrendelttermék>
```

```xml
<!-- Üzlet -->

<Üzlet Ü_id="21" MT_id="11">
    <Cim>
        <Utca>Lajos utca</Utca>
        <Varos>Miskolc</Varos>
        <Hazszam>6</Hazszam>
    </Cim>
    <Nev>Hervis</Nev>
    <Telefonszam>0670-5312490</Telefonszam>
    <Telefonszam>0630-6587421</Telefonszam>
</Üzlet>

<Üzlet Ü_id="22" MT_id="12">
    <Cim>
        <Utca>Lajos utca</Utca>
        <Varos>Miskolc</Varos>
        <Hazszam>6</Hazszam>
    </Cim>
    <Nev>Intersport</Nev>
    <Telefonszam>0670-5312490</Telefonszam>
    <Telefonszam>0630-6587421</Telefonszam>
</Üzlet>

<Üzlet Ü_id="23" MT_id="13">
    <Cim>
        <Utca>Lajos utca</Utca>
        <Varos>Miskolc</Varos>
        <Hazszam>6</Hazszam>
    </Cim>
    <Nev>Dechatlon</Nev>
    <Telefonszam>0670-5312490</Telefonszam>
    <Telefonszam>0630-6587421</Telefonszam>
</Üzlet>
```

```xml
<!-- Ü_D -->
<Ü_D Ü_id="21" D_id="31">
    <Fizetes>420000</Fizetes>
</Ü_D>
<Ü_D Ü_id="22" D_id="32">
    <Fizetes>530000</Fizetes>
</Ü_D>
<Ü_D Ü_id="23" D_id="33">
    <Fizetes>370000</Fizetes>
</Ü_D>

<!-- Dolgozók -->

<Dolgozo D_id="31">
    <Cim>
        <Utca>Oláh Lajos</Utca>
        <Varos>Miskolc</Varos>
        <Hazszam>6</Hazszam>
    </Cim>
    <Nev>Nagy Gergő</Nev>
    <Telefonszam>0670-5342490</Telefonszam>
    <Telefonszam>0630-6596521</Telefonszam>
</Dolgozo>

<Dolgozo D_id="32">
    <Cim>
        <Utca>Deaák ferenc</Utca>
        <Varos>Arnót</Varos>
        <Hazszam>8</Hazszam>
    </Cim>
    <Nev>Kiss Árpád</Nev>
    <Telefonszam>0670-5643905</Telefonszam>
    <Telefonszam>0630-6734524</Telefonszam>
</Dolgozo>

<Dolgozo D_id="33">
```

1d) Az XML dokumentum alapján XMLSchema készítése - saját típusok, ref, key, keyref, speciális elemek. (Ide kerül az XML Schema kódja!)

Először definiáltam az egyszerű elemeket. Ezután a komplex típusokat, amin belül referáltam az egyszerű típusokra, valamint az többértékű tulajdonsághoz egy egyszerű értéket, ahol a telefonszám helyességét regex-el ellenőriztem.

Eztán az egyedek komplex típusát hoztam létre, felhasználva az eddigi element, komplex és egyserű típusokat, valamint attribútumba helyeztem az egyedi és idegen kulcsokat, használatukat kötelezővé tettem. Amikor ezzel kész voltam, megcsináltam a sémát a gyökérelemből kiindulva, referálva az eddigi típusokra, majd létre hoztam a kulcsok, egyedi kulcsok és az egy-egy kapcsolat megvalósítását.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Nev" type="xs:string"/>
    <xs:element name="Hatarido" type="xs:date"/>
    <xs:element name="Utca" type="xs:string"/>
    <xs:element name="Varos" type="xs:string"/>
    <xs:element name="Hazszam" type="xs:string"/>
    <xs:element name="Megrendelesideje" type="xs:date"/>
    <xs:element name="Statusz" type="xs:string"/>
    <xs:element name="Learazas" type="xs:float"/>
    <xs:element name="Mennyiség" type="xs:positiveInteger"/>
    <xs:element name="Listaar" type="xs:int"/>
    <xs:element name="Fizetes" type="xs:positiveInteger"/>

    <xs:complexType name="cimType">
        <xs:sequence>
            <xs:element ref="Utca"/>
            <xs:element ref="Varos"/>
            <xs:element ref="Hazszam"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="telefonszamType">
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{4}-\d{7}"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="Vasarlo">
        <xs:sequence>
            <xs:element ref="Nev"/>
            <xs:element type="cimType" name="Cim"/>
            <xs:element type="telefonszamType" name="Telefonszam" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="V_id" type="xs:positiveInteger" use="required"/>
        <xs:attribute name="M_id" type="xs:positiveInteger" use="required"/>
    </xs:complexType>
```

```xml
<xs:complexType name="Megrendeles">
    <xs:sequence>
        <xs:element ref="Hatarido"/>
        <xs:element ref="Megrendelesideje"/>
        <xs:element ref="Statusz"/>
    </xs:sequence>
    <xs:attribute name="M_id" type="xs:positiveInteger" use="required"/>
</xs:complexType>

<xs:complexType name="Megrendelttermék">
    <xs:sequence>
        <xs:element ref="Learazas"/>
        <xs:element ref="Mennyiség"/>
        <xs:element ref="Listaar"/>
    </xs:sequence>
    <xs:attribute name="M_id" type="xs:positiveInteger" use="required"/>
    <xs:attribute name="MT_id" type="xs:positiveInteger" use="required"/>
</xs:complexType>

<xs:complexType name="Üzlet">
    <xs:sequence>
        <xs:element type="cimType" name="Cim"/>
        <xs:element ref="Nev"/>
        <xs:element type="telefonszamType" name="Telefonszam" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="Ü_id" type="xs:positiveInteger" use="required"/>
    <xs:attribute name="MT_id" type="xs:positiveInteger" use="required"/>
</xs:complexType>

<xs:complexType name="Ü_D">
    <xs:sequence>
        <xs:element ref="Fizetes"/>
    </xs:sequence>
    <xs:attribute name="Ü_id" type="xs:positiveInteger" use="required"/>
    <xs:attribute name="D_id" type="xs:positiveInteger" use="required"/>
</xs:complexType>
```

```
<!-- Egyedi kulcsok -->
<xs:key name="Vasarlo_V_id">
    <xs:selector xpath="Vasarlo"/>
    <xs:field xpath="@V_id"/>
</xs:key>
<xs:key name="Megrendeles_M_id">
    <xs:selector xpath="Megrendeles"/>
    <xs:field xpath="@M_id"/>
</xs:key>
<xs:key name="Megrendelttermék_MT_id">
    <xs:selector xpath="Megrendelttermék"/>
    <xs:field xpath="@MT_id"/>
</xs:key>
<xs:key name="Üzlet_Ü_id">
    <xs:selector xpath="Üzlet"/>
    <xs:field xpath="@Ü_id"/>
</xs:key>
<xs:key name="Dolgozok_D_id">
    <xs:selector xpath="Dolgozo"/>
    <xs:field xpath="@D_id"/>
</xs:key>
```

```
    <!-- Idegen kulcsok -->
    <xs:keyref name="Vasarlo_Megrendeles_id_ref" refer="Megrendeles_M_id">
        <xs:selector xpath="Vasarlo"/>
        <xs:field xpath="@M_id"/>
    </xs:keyref>
    <xs:keyref name="Megrendelttermék_Megrendeles_id_ref" refer="Megrendeles_M_id">
        <xs:selector xpath="Megrendelttermék"/>
        <xs:field xpath="@M_id"/>
    </xs:keyref>
    <xs:keyref name="Üzlet_Megrendelttermék_id_ref" refer="Megrendelttermék_MT_id">
        <xs:selector xpath="Üzlet"/>
        <xs:field xpath="@MT_id"/>
    </xs:keyref>
    <xs:keyref name="Ü_D_Üzlet_id_ref" refer="Üzlet_Ü_id">
        <xs:selector xpath="Ü_D"/>
        <xs:field xpath="@Ü_id"/>
    </xs:keyref>
    <xs:keyref name="Ü_D_Dolgozo_id_ref" refer="Dolgozok_D_id">
        <xs:selector xpath="Ü_D"/>
        <xs:field xpath="@D_id"/>
    </xs:keyref>

    <!-- Egy-egy kapcsolat -->
    <xs:unique name="Megrendeles_M_id_unique">
        <xs:selector xpath="Vasarlo"/>
        <xs:field xpath="@M_id"/>
    </xs:unique>


</xs:element>
```

2. feladat

2a) adatolvasás (kód – comment együtt) – fájlnév: DOMReadGQOKMW.java

Beolvastam a fájlomat, a megfelelő Document osztályokat példányosítottam, készítettem egy dokumentumot, parseoltam az xml fájlomat, kiirattam a gyökérelemet, majd listába kigyűjtöttem a példányokat, és a segédfüggvények segítségével formázva kiírtam a konzolra és egy fájlba

```
 1. package hu.domparse.gqokmw;
 2.
 3. import java.io.File;
 4. import java.io.FileWriter;
 5. import java.io.PrintWriter;
 6. import java.util.StringJoiner;
 7.
 8. import javax.xml.parsers.DocumentBuilder;
 9. import javax.xml.parsers.DocumentBuilderFactory;
10.
11.
12. import org.w3c.dom.*;
13.
14. public class DOMReadGQOKMW {
15.
16.      public static void main(String[] args) {
17.         try {
18.             File xmlFile = new File("XMLGQOKMW.xml");
19.             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
20.             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
21.             Document doc = dBuilder.parse(xmlFile);
22.             doc.getDocumentElement().normalize();
23.
24.             File outputFile = new File("DomReadOutPut_GQOKMW.xml");
25.             PrintWriter writer = new PrintWriter(new FileWriter(outputFile, true));
26.
27.             // Kiírjuk az XML főgyökér elemét a konzolra és fájlba
28.             Element rootElement = doc.getDocumentElement();
29.             String rootName = rootElement.getTagName();
30.             StringJoiner rootAttributes = new StringJoiner(" ");
31.             NamedNodeMap rootAttributeMap = rootElement.getAttributes();
32.
33.             for (int i = 0; i < rootAttributeMap.getLength(); i++) {
34.                 Node attribute = rootAttributeMap.item(i);
35.                 rootAttributes.add(attribute.getNodeName() + "=\"" +
attribute.getNodeValue() + "\"");
36.             }
37.
38.
39.
40.             System.out.print("<?xml version=\"1.0\" encoding=\"UTF-8\"?> \n");
41.             writer.print("<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
42.
43.             System.out.print("<" + rootName + " " + rootAttributes.toString() + "> \n");
44.             writer.print("<" + rootName + " " + rootAttributes.toString() + "> \n");
45.
46.             NodeList vasarloList = doc.getElementsByTagName("Vasarlo");
47.             NodeList megrendelesList = doc.getElementsByTagName("Megrendeles");
48.             NodeList megrendelttermékList = doc.getElementsByTagName("Megrendelttermek");
49.             NodeList üzletList = doc.getElementsByTagName("Üzlet");
50.             NodeList Ü_DList = doc.getElementsByTagName("Ü_D");
51.             NodeList dolgozoList = doc.getElementsByTagName("Dolgozo");
52.
53.             // Kiírjuk az XML-t a konzolra megtartva az eredeti formázást
54.             printNodeList(vasarloList, writer);
55.             System.out.println("");
56.             writer.println("");
57.             printNodeList(megrendelesList, writer);
58.             System.out.println("");
59.             writer.println("");
```

```
60.             printNodeList(megrendelttermékList, writer);
61.             System.out.println("");
62.             writer.println("");
63.             printNodeList(üzletList, writer);
64.             System.out.println("");
65.             writer.println("");
66.             printNodeList(Ü_DList, writer);
67.             System.out.println("");
68.             writer.println("");
69.             printNodeList(dolgozoList, writer);
70.
71.             // Zárjuk le az XML gyökér elemét
72.             System.out.println("</" + rootName + ">");
73.             writer.append("</" + rootName + ">");
74.
75.             writer.close();
76.         } catch (Exception e) {
77.             e.printStackTrace();
78.         }
79.     }
80.
81.     // Rekurzív függvény a NodeList tartalmának kiírására
82.     private static void printNodeList(NodeList nodeList, PrintWriter writer) {
83.         for (int i = 0; i < nodeList.getLength(); i++) {
84.             Node node = nodeList.item(i);
85.             printNode(node, 0, writer);
86.             System.out.println(""); // Üres sor hozzáadása az elemek között
87.             writer.println(""); // Üres sor hozzáadása a fájlban az elemek között
88.         }
89.     }
90.
91.     // Rekurzív függvény a Node tartalmának kiírására
92.     private static void printNode(Node node, int indent, PrintWriter writer) {
93.         if (node.getNodeType() == Node.ELEMENT_NODE) {
94.             Element element = (Element) node;
95.             String nodeName = element.getTagName();
96.             StringJoiner attributes = new StringJoiner(" ");
97.             NamedNodeMap attributeMap = element.getAttributes();
98.
99.             for (int i = 0; i < attributeMap.getLength(); i++) {
100.                Node attribute = attributeMap.item(i);
101.                attributes.add(attribute.getNodeName() + "=\"" + attribute.getNodeValue() +
"\"");
102.            }
103.
104.            System.out.print(getIndentString(indent));
105.            System.out.print("<" + nodeName + "" + attributes.toString() + ">");
106.
107.            writer.print(getIndentString(indent));
108.            writer.print("<" + nodeName + "" + attributes.toString() + ">");
109.
110.            NodeList children = element.getChildNodes();
111.            if (children.getLength() == 1 && children.item(0).getNodeType() ==
Node.TEXT_NODE) {
112.                System.out.print(children.item(0).getNodeValue());
113.                writer.print(children.item(0).getNodeValue());
114.            } else {
115.                System.out.println();
116.                writer.println();
117.                for (int i = 0; i < children.getLength(); i++) {
118.                    printNode(children.item(i), indent + 1, writer);
119.                }
120.                System.out.print(getIndentString(indent));
121.                writer.print(getIndentString(indent));
122.            }
123.            System.out.println("</" + nodeName + ">");
124.            writer.println("</" + nodeName + ">");
125.        }
126.    }
127.
```

```
128.     // Segédmetódus az indentáláshoz
129.     private static String getIndentString(int indent) {
130.         StringBuilder sb = new StringBuilder();
131.         for (int i = 0; i < indent; i++) {
132.             sb.append("  "); // 4 spaces per indent level
133.         }
134.         return sb.toString();
135.     }
136. 
137. }
138. 
```

2b) adatmódosítás (kód – comment együtt) – fájlnév: DOMModifyGQOKMW.java

Ugyanúgy felépíítettem a dokumentumot, majd a setTextContent függvénnyel néhány elemet átállítottam, ezúttal viszont a Transformer osztállyal irattam ki fájlba és konzolra az adatokat, amely könnyen és gyorsan formázza .

```
 1. package hu.domparse.gqokmw;
 2.
 3. import java.io.File;
 4.
 5. import javax.xml.parsers.DocumentBuilder;
 6. import javax.xml.parsers.DocumentBuilderFactory;
 7. import javax.xml.transform.Transformer;
 8. import javax.xml.transform.TransformerFactory;
 9. import javax.xml.transform.dom.DOMSource;
10. import javax.xml.transform.stream.StreamResult;
11.
12.
13. import org.w3c.dom.*;
14.
15. public class DOMModifyGQOKMW {
16.
17.       public static void main(String[] args) {
18.         try {
19.             File xmlFile = new File("XMLGQOKMW.xml");
20.             DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
21.             DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
22.             //létrehozom a doc-ot amit később a transformhoz használok fel
23.             Document doc = dBuilder.parse(xmlFile);
24.
25.             // lekérjük egy adott típushoz tartozó összes elemet amit egy listában tárolunk
el
26.             NodeList vasarloList = doc.getElementsByTagName("Vasarlo");
27.             //lekérjük azt az elemet a listából amelyiket módosítani szeretnénk, itt index
alapján történik a módosítás
28.             Element vasarlo = (Element) vasarloList.item(0);
29.             //az elemnek megkeressük azt a tagjét amit módosítani szeretnénk, majd a
tartalmát átállítjuk
30.             vasarlo.getElementsByTagName("Nev").item(0).setTextContent("Türk Viktor");
31.
32.             NodeList megrendelesList = doc.getElementsByTagName("Megrendeles");
33.             Element megrendeles = (Element) megrendelesList.item(0);
34.             megrendeles.getElementsByTagName("Statusz").item(0).setTextContent("Hiba");
35.
36.             NodeList megrendelttermekList = doc.getElementsByTagName("Megrendelttermek");
37.             Element megrendelttermek = (Element) megrendelttermekList.item(1);
38.             megrendelttermek.getElementsByTagName("Learazas").item(0).setTextContent("0.6");
39.
40.             NodeList üzletList = doc.getElementsByTagName("Üzlet");
41.             Element üzlet = (Element) üzletList.item(1);
42.             üzlet.getElementsByTagName("Varos").item(0).setTextContent("Kiskunfélegyháza");
43.
44.
45.             NodeList dolgozoList = doc.getElementsByTagName("Dolgozo");
46.             Element dolgozo = (Element) dolgozoList.item(1);
47.             dolgozo.getElementsByTagName("Nev").item(0).setTextContent("Lakatos Rómeo");
48.
49.             //TransformerFactory-val iratom ki a fájlt
50.             TransformerFactory transformerFactory = TransformerFactory.newInstance();
51.             //Beállítom a transformert
52.             Transformer transformer = transformerFactory.newTransformer();
53.             //Megadom a forrás fájlt amit fent létrehoztam
54.             DOMSource source = new DOMSource(doc);
55.             //Megnyitom a streamet és konzolra kiíratom sys.out-al a fájlt
56.             StreamResult consoleResult = new StreamResult(System.out);
57.             transformer.transform(source, consoleResult);
58.
59.         } catch (Exception e) {
```

```
60.             e.printStackTrace();
61.         }
62.     }
63. }
64.
```

2c) adatlekérdezés (kód – comment együtt) – fájlnév: DOMQueryGQOKMW.java

Szokásos módon felépítem a dokumentumot, majd egy adott kirtéria szerint kikeresek adatokat, és kiiratom a konzolra. A lekérdezésekben van egyszerű lekérdezés, több adatos lekérdezés, többelem feltételes lekérdezés, majd idegen kulcson keresztüli lekérdezés, és egy komplex elem adatainak lekérdezése

```
 1. package hu.domparse.gqokmw;
 2.
 3. import java.io.File;
 4.
 5. import javax.xml.parsers.DocumentBuilder;
 6. import javax.xml.parsers.DocumentBuilderFactory;
 7.
 8.
 9. import org.w3c.dom.*;
10.
11. public class DOMQueryGQOKMW {
12.
13.       public static void main(String[] args) {
14.         try {
15.                  // XML fájl beolvasása és DOM létrehozása
16.             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
17.             DocumentBuilder builder = factory.newDocumentBuilder();
18.             Document document = builder.parse(new File("XMLGQOKMW.xml"));
19.
20.             // 1. Lekérdezés: "13"-as ID-s Megrendelttermek minden adata
21.             String MT_Id = "13";
22.             NodeList mttermekList = document.getElementsByTagName("Megrendelttermek");
23.             for (int i = 0; i < mttermekList.getLength(); i++) {
24.
25.                 Element mttermek = (Element) mttermekList.item(i);
26.                 String mttermekId = mttermek.getAttribute("MT_id");
27.                 if (mttermekId.equals(MT_Id)) {
28.                     String learazas =
mttermek.getElementsByTagName("Learazas").item(0).getTextContent();
29.                     String mennyiseg =
mttermek.getElementsByTagName("Mennyiseg").item(0).getTextContent();
30.                     String listaar =
mttermek.getElementsByTagName("Listaar").item(0).getTextContent();
31.                     System.out.println("Lekérdezés 1:");
32.                     System.out.println("Az '" + MT_Id + "' ID-jű megrendelt termék
leárazása: " + learazas);
33.                     System.out.println("Megrendelés mennyisége: " + mennyiseg);
34.                     System.out.println("Az eredeti listaára: " + listaar);
35.                     System.out.println();
36.                     break;
37.                 }
38.             }
39.          // Lekérdezés 2: Az "21"-es ID-s Üzlet címe
40.             String Ü_Id = "21";
41.             NodeList üzletList = document.getElementsByTagName("Üzlet");
42.             for (int i = 0; i < üzletList.getLength(); i++) {
43.
44.                 Element vasarlo = (Element) üzletList.item(i);
45.                 if (vasarlo.getAttribute("Ü_id").equals(Ü_Id)) {
46.
47.                     Element cim = (Element) vasarlo.getElementsByTagName("Cim").item(0);
48.                     String varos =
cim.getElementsByTagName("Varos").item(0).getTextContent().trim();
49.                     String utca =
cim.getElementsByTagName("Utca").item(0).getTextContent().trim();
50.                     String hazszam =
cim.getElementsByTagName("Hazszam").item(0).getTextContent().trim();
51.                     System.out.println("Lekérdezés 2:");
52.                     System.out.println("Cím: " + " " + varos + " " + utca + " " + hazszam);
53.                     System.out.println();
```

```
                    break;
                        }
                    }


            // Lekérdezés 3: A "03"-as Megrendelés határideje
               String M_Id = "03";
               NodeList megrendelesList = document.getElementsByTagName("Megrendeles");
               for (int i = 0; i < megrendelesList.getLength(); i++) {

                   Element megrendeles = (Element) megrendelesList.item(i);
                   if(megrendeles.getAttribute("M_id").equals(M_Id)) {
                           String hatarido =
megrendeles.getElementsByTagName("Hatarido").item(0).getTextContent().trim();
                           System.out.println("Lekérdezés 3:");
                           System.out.println("A "+ M_Id + "-as ID Megrendelés határideje : "
+ hatarido);
                           System.out.println();
                   }

               }


            // Lekérdezés 4: Azon vásárlók telefonszámának és nevének kiírása, akiknek több
telefonszámuk van
               NodeList vasarloList = document.getElementsByTagName("Vasarlo");
               for (int i = 0; i < vasarloList.getLength(); i++) {
                   Element vasarlo = (Element) vasarloList.item(i);
                   NodeList telefonszamList = vasarlo.getElementsByTagName("Telefonszam");

                   if (telefonszamList.getLength() > 1) {
                       String nev =
vasarlo.getElementsByTagName("Nev").item(0).getTextContent().trim();

                       System.out.println("Lekérdezés 4:");
                       System.out.println("Vásárló neve: " + nev);
                       System.out.println("Telefonszámai:");

                       for (int j = 0; j < telefonszamList.getLength(); j++) {
                           Element telefonszam = (Element) telefonszamList.item(j);
                           System.out.println(telefonszam.getTextContent().trim());
                       }
                       System.out.println();
                   }

               }

            // Lekérdezés 5: Azon futárok nevének kiírása, akiknél az áru értéke 2000 fölött
van
               String Vnev = "Nagy Máté";
               NodeList vasarloList2 = document.getElementsByTagName("Vasarlo");
               for (int i = 0; i < vasarloList2.getLength(); i++) {

                  Element vasarlo = (Element) vasarloList2.item(i);
                  String M_Id_Vasarlo = "";

           if(vasarlo.getElementsByTagName("Nev").item(0).getTextContent().equals(Vnev)) {

                           M_Id_Vasarlo = vasarlo.getAttribute("M_id");
                  }

                  NodeList megrendelesList2 = document.getElementsByTagName("Megrendeles");
                  for (int j = 0; j < megrendelesList2.getLength(); j++) {

                                                Element megrendeles = (Element)
megrendelesList2.item(j);

           if(megrendeles.getAttribute("M_id").equals(M_Id_Vasarlo)) {
```

```
115.                                                    String statusz =
megrendeles.getElementsByTagName("Statusz").item(0).getTextContent();
116.                                                    System.out.println("Lekérdezés 5:");
117.                                                    System.out.println(Vnev + "
megrendelésének a státusza: " + statusz);
118.                                                }
119.                                            }
120.
121.                                        }
122.
123.                 } catch (Exception e) {
124.                     e.printStackTrace();
125.                 }
126.             }
127.
128. }
129.
```

2d) adatírás - készítsen egy DOM API programot, amely egy XMLNeptunkod.xml

dokumentum tartalmát fa struktúra formában kiírja a konzolra és egy XMLNeptunkod1.xml

fájlba. (kód – comment együtt) – fájlnév: DOMWriteGQOKMW.java

Dokumentumot felépítettem, hozzáadtam a rootelementet, majd minden táblához különy függvényt csináltam, amivel a main-ben könnyen hozzáadok adatokat, szintén a Transformer osztállyal kiiratom fájlba és a konzolra is.

```
 1. package hu.domparse.gqokmw;
 2.
 3. import java.io.File;
 4. import java.io.FileWriter;
 5. import java.io.PrintWriter;
 6. import java.util.StringJoiner;
 7.
 8. import javax.xml.parsers.DocumentBuilder;
 9. import javax.xml.parsers.DocumentBuilderFactory;
10. import javax.xml.transform.OutputKeys;
11. import javax.xml.transform.Transformer;
12. import javax.xml.transform.TransformerFactory;
13. import javax.xml.transform.dom.DOMSource;
14. import javax.xml.transform.stream.StreamResult;
15.
16. import org.w3c.dom.Document;
17. import org.w3c.dom.Element;
18. import org.w3c.dom.NamedNodeMap;
19. import org.w3c.dom.Node;
20. import org.w3c.dom.NodeList;
21.
22. public class DOMWriteGQOKMW {
23.
24.     public static void main(String[] args) {
25.         try {
26.             // Create a new Document
27.             DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
28.             DocumentBuilder builder = factory.newDocumentBuilder();
29.             Document doc = builder.newDocument();
30.
31.             // Create the root element
32.             Element rootElement = doc.createElement("GQOKMW_BEADANDO");
33.             rootElement.setAttribute("xmlns:xs", "http://www.w3.org/2001/XMLSchema-
instance");
34.             rootElement.setAttribute("xs:noNamespaceSchemaLocation",
"XMLSchemaGQOKMW.xsd");
35.             doc.appendChild(rootElement);
36.
37.             // Add Vasarlok
38.             addVasarlo(doc, rootElement, "1", "01", "Kiss Ádám", "Miskolc", "Kossuth Lajos
utca", "1",
39.                     "0630-1234567");
40.             addVasarlo(doc, rootElement, "2", "02", "Nagy Pista", "Szekszárd", "Árpád
utca", "5",
41.                     "0630-3214569");
42.             addVasarlo(doc, rootElement, "3", "03", "Nagy Máté", "Esztergom", "József
Attila út", "5",
43.                     "0630-1234765", "0630-7654321");
44.
45.             // Add Megrendelesek
46.             addMegrendeles(doc, rootElement, "01", "2021-02-03", "2021-02-01",
"Teljesítve");
47.             addMegrendeles(doc, rootElement, "02", "2021-02-07", "2021-02-02", "Szállítás
alatt");
48.             addMegrendeles(doc, rootElement, "03", "2021-03-05", "2021-03-01",
"Teljesítve");
49.
50.             // Add Megrendelt termek
```

```
51.             addMegrendeltTermek(doc, rootElement, "01", "11", "0.2", "2", "25000");
52.             addMegrendeltTermek(doc, rootElement, "02", "12", "0.2", "2", "25000");
53.             addMegrendeltTermek(doc, rootElement, "03", "13", "0.2", "2", "25000");
54.
55.             // Add Uzletek
56.             addUzlet(doc, rootElement, "21", "11", "Hervis", "Miskolc", "Lajos utca", "6",
"0670-5312490", "0630-6587421");
57.             addUzlet(doc, rootElement, "22", "12", "Intersport", "Miskolc", "Lajos utca",
"6", "0670-5312490", "0630-6587421");
58.             addUzlet(doc, rootElement, "23", "13", "Dechatlon", "Miskolc", "Lajos utca",
"6", "0670-5312490", "0630-6587421");
59.
60.             // Add Ü_D
61.             addUD(doc, rootElement, "21", "31", "420000");
62.             addUD(doc, rootElement, "22", "32", "530000");
63.             addUD(doc, rootElement, "23", "33", "370000");
64.
65.             // Add Dolgozok
66.             addDolgozo(doc, rootElement, "31", "Nagy Gergő", "Miskolc", "Oláh Lajos", "6",
"0670-5342490", "0630-6596521");
67.             addDolgozo(doc, rootElement, "32", "Kiss Árpád", "Arnót", "Deaák ferenc", "8",
"0670-5643905", "0630-6734524");
68.             addDolgozo(doc, rootElement, "33", "Szabó Gergő", "Miskolc", "Kossuth Lajos",
"56", "0670-5723490", "0630-9654321");
69.
70.             // Transform and save to file
71.             TransformerFactory transformerFactory = TransformerFactory.newInstance();
72.             Transformer transformer = transformerFactory.newTransformer();
73.             transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
74.             transformer.setOutputProperty(OutputKeys.INDENT, "yes");
75.             transformer.setOutputProperty("{https://xml.apache.org/xslt}indent-amount",
"2");
76.
77.             DOMSource source = new DOMSource(doc);
78.             File myFile = new File("DomWriteOutPut_GQOKMW.xml");
79.             StreamResult file = new StreamResult(myFile);
80.             StreamResult consoleResult = new StreamResult(System.out);
81.             transformer.transform(source, file);
82.             transformer.transform(source, consoleResult);
83.
84.         } catch (Exception e) {
85.             e.printStackTrace();
86.         }
87.     }
88.
89.     // Add Vasarlo
90.     private static void addVasarlo(Document doc, Element rootElement, String v_id, String
m_id, String nev,
91.             String varos, String utca, String hazszam, String... telefonszamok) {
92.         Element vasarlo = doc.createElement("Vasarlo");
93.         vasarlo.setAttribute("V_id", v_id);
94.         vasarlo.setAttribute("M_id", m_id);
95.
96.         Element nevElement = createElement(doc, "Nev", nev);
97.
98.         Element cim = doc.createElement("Cim");
99.         Element utcaElement = createElement(doc, "Utca", utca);
100.        Element varosElement = createElement(doc, "Varos", varos);
101.        Element hazszamElement = createElement(doc, "Hazszam", hazszam);
102.
103.        cim.appendChild(utcaElement);
104.        cim.appendChild(varosElement);
105.        cim.appendChild(hazszamElement);
106.
107.        vasarlo.appendChild(nevElement);
108.        vasarlo.appendChild(cim);
109.
110.        for (String telefon : telefonszamok) {
111.            Element telefonElement = createElement(doc, "Telefonszam", telefon);
112.            vasarlo.appendChild(telefonElement);
```

```
113.         }
114.
115.         rootElement.appendChild(vasarlo);
116.     }
117.
118.     // Add Megrendeles
119.     private static void addMegrendeles(Document doc, Element rootElement, String m_id,
String hatarido,
120.             String megrendelesideje, String statusz) {
121.         Element megrendeles = doc.createElement("Megrendeles");
122.         megrendeles.setAttribute("M_id", m_id);
123.
124.         Element hataridoElement = createElement(doc, "Hatarido", hatarido);
125.         Element megrendelesidejeElement = createElement(doc, "Megrendelesideje",
megrendelesideje);
126.         Element statuszElement = createElement(doc, "Statusz", statusz);
127.
128.         megrendeles.appendChild(hataridoElement);
129.         megrendeles.appendChild(megrendelesidejeElement);
130.         megrendeles.appendChild(statuszElement);
131.
132.         rootElement.appendChild(megrendeles);
133.     }
134.
135.     // Add Megrendelt termek
136.     private static void addMegrendeltTermek(Document doc, Element rootElement, String m_id,
String mt_id, String learazas,
137.             String mennyiseg, String listaar) {
138.         Element megrendeltTermek = doc.createElement("Megrendelttermek");
139.         megrendeltTermek.setAttribute("M_id", m_id);
140.         megrendeltTermek.setAttribute("MT_id", mt_id);
141.
142.         Element learazasElement = createElement(doc, "Learazas", learazas);
143.         Element mennyisegElement = createElement(doc, "Mennyiseg", mennyiseg);
144.         Element listaarElement = createElement(doc, "Listaar", listaar);
145.
146.         megrendeltTermek.appendChild(learazasElement);
147.         megrendeltTermek.appendChild(mennyisegElement);
148.         megrendeltTermek.appendChild(listaarElement);
149.
150.         rootElement.appendChild(megrendeltTermek);
151.     }
152.
153.     // Add Uzlet
154.     private static void addUzlet(Document doc, Element rootElement, String u_id, String
mt_id, String nev,
155.             String varos, String utca, String hazszam, String... telefonszamok) {
156.         Element uzlet = doc.createElement("Üzlet");
157.         uzlet.setAttribute("Ü_id", u_id);
158.         uzlet.setAttribute("MT_id", mt_id);
159.
160.         Element nevElement = createElement(doc, "Nev", nev);
161.
162.         Element cim = doc.createElement("Cim");
163.         Element utcaElement = createElement(doc, "Utca", utca);
164.         Element varosElement = createElement(doc, "Varos", varos);
165.         Element hazszamElement = createElement(doc, "Hazszam", hazszam);
166.
167.         cim.appendChild(utcaElement);
168.         cim.appendChild(varosElement);
169.         cim.appendChild(hazszamElement);
170.
171.
172.         uzlet.appendChild(cim);
173.         uzlet.appendChild(nevElement);
174.
175.         for (String telefon : telefonszamok) {
176.             Element telefonElement = createElement(doc, "Telefonszam", telefon);
177.             uzlet.appendChild(telefonElement);
178.         }
```

```
179.
180.         rootElement.appendChild(uzlet);
181.     }
182.
183.     // Add UD
184.     private static void addUD(Document doc, Element rootElement, String u_id, String d_id,
String fizetes) {
185.         Element ud = doc.createElement("Ü_D");
186.         ud.setAttribute("Ü_id", u_id);
187.         ud.setAttribute("D_id", d_id);
188.
189.         Element fizetesElement = createElement(doc, "Fizetes", fizetes);
190.
191.         ud.appendChild(fizetesElement);
192.
193.         rootElement.appendChild(ud);
194.     }
195.
196.     // Add Dolgozo
197.     private static void addDolgozo(Document doc, Element rootElement, String d_id, String
nev,
198.             String varos, String utca, String hazszam, String... telefonszamok) {
199.         Element dolgozo = doc.createElement("Dolgozo");
200.         dolgozo.setAttribute("D_id", d_id);
201.
202.         Element nevElement = createElement(doc, "Nev", nev);
203.
204.         Element cim = doc.createElement("Cim");
205.         Element utcaElement = createElement(doc, "Utca", utca);
206.         Element varosElement = createElement(doc, "Varos", varos);
207.         Element hazszamElement = createElement(doc, "Hazszam", hazszam);
208.
209.         cim.appendChild(utcaElement);
210.         cim.appendChild(varosElement);
211.         cim.appendChild(hazszamElement);
212.
213.
214.         dolgozo.appendChild(cim);
215.         dolgozo.appendChild(nevElement);
216.
217.         for (String telefon : telefonszamok) {
218.             Element telefonElement = createElement(doc, "Telefonszam", telefon);
219.             dolgozo.appendChild(telefonElement);
220.         }
221.
222.         rootElement.appendChild(dolgozo);
223.     }
224.
225.     private static Element createElement(Document doc, String name, String value) {
226.         Element element = doc.createElement(name);
227.         element.appendChild(doc.createTextNode(value));
228.         return element;
229.     }
230.
231.
232. }
233.
```