JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Füredi Gábor**

Neptunkód: **YRNWBP**

**A feladat leírása:**

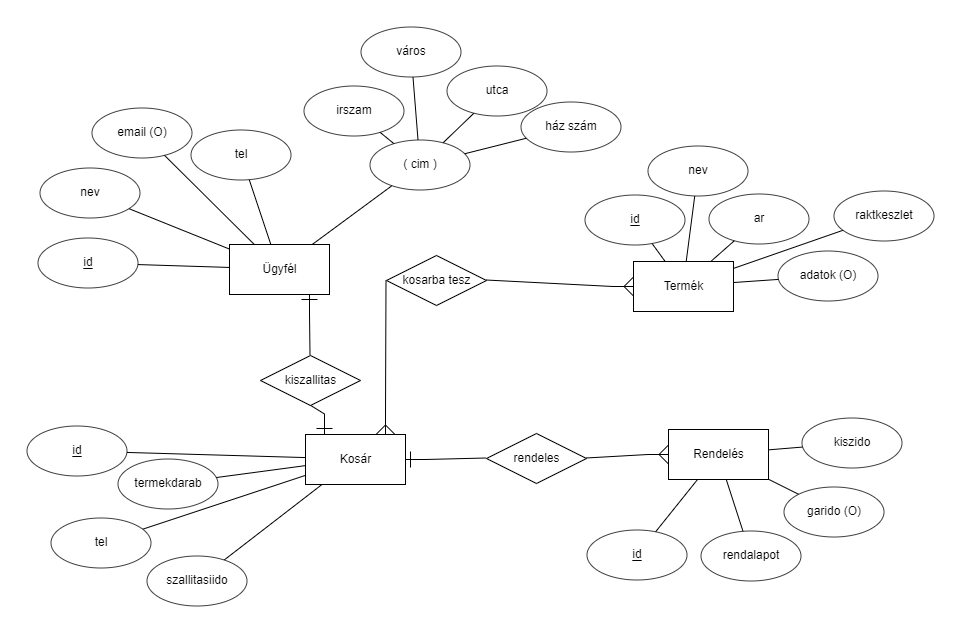
Az adatkezelés XML-ben című tárgy féléves feladatának jegyzőkönyvét egy Online bolt rendelés nyilvántartásáról szóló adatbázis megtervezéséről írom.

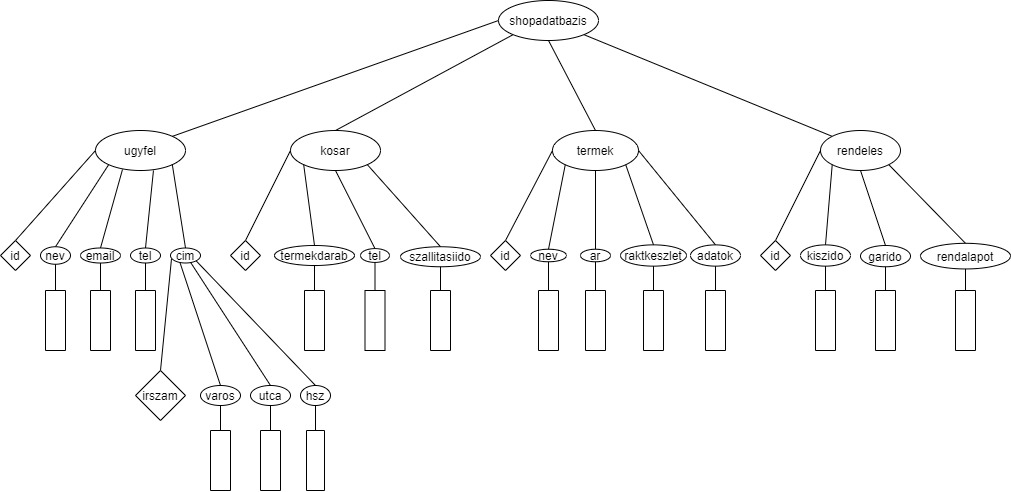
Az ER modell az Ügyfél, Termék, Kosár, Rendelés egyedekből áll.

Termék és a Kosár között N:M kapcsolat áll fenn. A Kosár és a Rendelés között 1:N kapcsolat áll fenn, míg az Ügyfél és Kosár között 1:1 kapcsolat áll fenn.

Az ER modell elkészítése után az adatbázis XDM modellre konvertálásnál nagy segítség volt a moodle oldalon megtalálható XML tanagyag.

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"?>  
<?xml-model href="XMLSchemaYrnwbp.xsd" type="application/xml" schematypens="http://www.w3.org/2001/XMLSchema"?>  
<shopadatbazis>  
   
 <kosar id="1253" >  
 <tel>06306358468</tel>  
 <termekdarab>2</termekdarab>  
 <szallitasiido>1-3 munkanap</szallitasiido>  
 </kosar>  
 <kosar id="1250" >  
 <tel>06706358468</tel>  
 <termekdarab>2</termekdarab>  
 <szallitasiido>1-3 munkanap</szallitasiido>  
 </kosar>  
 <termek id="5263">  
 <nev>AMD Ryzen 3 3100 100000284BOX processzor</nev>  
 <ar>25000</ar>  
 <raktkeszlet>10</raktkeszlet>  
 <adatok>Fogyasztás (W): 65</adatok>  
 </termek>  
 <termek id="5261">  
 <nev>Kingston 2x8GB DDR4 2666MHz HyperX FURY fekete HX426C16FB3K2/16 memória</nev>  
 <ar>32000</ar>  
 <raktkeszlet>4</raktkeszlet>  
 <adatok>Memória típus: DDR4</adatok>  
 </termek>  
 <termek id="5262">  
 <nev>Seagate 2TB BarraCuda 256MB ST2000DM008 merevlemez</nev>  
 <ar>19000</ar>  
 <raktkeszlet>10</raktkeszlet>  
 <adatok>Belső cache (MB): 256</adatok>  
 </termek>  
 <termek id="5260">  
 <nev>Samsung 500GB 970 Evo Plus MZ-V7S500BW M.2 SSD meghajtó</nev>  
 <ar>35000</ar>  
 <raktkeszlet>10</raktkeszlet>  
 <adatok>Formátum: M.2 PCIe</adatok>  
 </termek>  
 <ugyfel id="7952">  
 <nev>Kiss Elemér</nev>  
 <tel>06306358468</tel>  
 <email></email>  
 <cim irszam="3142">  
 <varos>Sopron</varos>  
 <utca>TóthJenő</utca>  
 <hsz>12</hsz>  
 </cim>  
 </ugyfel>  
 <ugyfel id="7950">  
 <nev>Tatár Endre</nev>  
 <tel>06706358468</tel>  
 <email>endre.tatar@gmal.com</email>  
 <cim irszam="3525">  
 <varos>Miskolc</varos>  
 <utca>TóthJenő</utca>  
 <hsz>35</hsz>  
 </cim>  
 </ugyfel>  
 <rendeles megrido="2012-09-24" id="6834">  
 <kiszido>2002-09-24</kiszido>  
 <garido>2</garido>  
 <rendalapot>kiszállítás alatt</rendalapot>  
 </rendeles>  
 <rendeles megrido="2015-09-24" id="6835">  
 <kiszido>2002-09-24</kiszido>  
 <garido>1 év</garido>  
 <rendalapot>beszállítás alatt</rendalapot>  
 </rendeles>  
 <ugyfel\_kosar\_kapcsolok>  
 <ugyfel\_kosar ugyfelref="7952" kosarref="1253"/>  
 <ugyfel\_kosar ugyfelref="7950" kosarref="1250"/>  
 </ugyfel\_kosar\_kapcsolok>  
 <kosar\_rendeles\_kapcsolok>  
 <kosar\_rendeles kosarref="1253" rendelesref="6834"/>  
 <kosar\_rendeles kosarref="1250" rendelesref="6834"/>  
 <kosar\_rendeles kosarref="1253" rendelesref="6835"/>  
 </kosar\_rendeles\_kapcsolok>  
 <termek\_kosar\_kapcsolok>  
 <termek\_kosar termekref="5263" kosarref="1253"></termek\_kosar>  
 <termek\_kosar termekref="5260" kosarref="1253"></termek\_kosar>  
 <termek\_kosar termekref="5263" kosarref="1250"></termek\_kosar>  
 <termek\_kosar termekref="5262" kosarref="1250"></termek\_kosar>  
 <termek\_kosar termekref="5262" kosarref="1250"></termek\_kosar>  
 </termek\_kosar\_kapcsolok>  
</shopadatbazis>

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

<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">  
 <xs:element name="shopadatbazis">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="kosar"/>  
 <xs:element ref="termek"/>  
 <xs:element ref="ugyfel"/>  
 <xs:element ref="rendeles"/>  
 </xs:sequence>  
 </xs:complexType>  
 </xs:element>  
 <!-- csomopont elemeim -->  
 <xs:element name="rendeles">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="kiszido"/>  
 <xs:element ref="garido"/>  
 <xs:element ref="rendalapot"/>  
 </xs:sequence>  
 <xs:attribute name="megrido" type="xs:date" use="required"/>  
 <xs:attribute name="id" type="azon" use="required"/>  
 </xs:complexType>  
 </xs:element>   
 <xs:element name="termek">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="nev"/>  
 <xs:element ref="ar"/>  
 <xs:element ref="raktkeszlet"/>  
 <xs:element ref="adatok"/>  
 </xs:sequence>  
 <xs:attribute name="id" type="azon" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="kosar">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="tel"/>  
 <xs:element ref="termekdarab"/>  
 <xs:element ref="szallitasiido"/>  
 </xs:sequence>  
 <xs:attribute name="id" type="azon" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 <xs:element name="ugyfel">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="nev"/>  
 <xs:element ref="tel"/>  
 <xs:element ref="email"/>  
 <xs:element ref="cim"/>  
 </xs:sequence>  
 <xs:attribute name="id" type="azon" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 <!-- alap elemeim -->  
 <xs:element name="tel" type="xs:integer"/>  
 <xs:element name="email" type="xs:string"/>  
 <xs:element name="varos" type="xs:NCName"/>  
 <xs:element name="utca" type="xs:NCName"/>  
 <xs:element name="hsz" type="xs:integer"/>  
 <xs:element name="garido" type="xs:string"/>  
 <xs:element name="nev" type="xs:string"/>  
 <xs:element name="termekdarab" type="xs:integer"/>  
 <xs:element name="szallitasiido" type="xs:integer"/>  
 <xs:element name="adatok" type="xs:string"/>  
 <xs:element name="ar" type="xs:integer"/>  
 <xs:element name="raktkeszlet" type="xs:integer"/>  
 <xs:element name="kiszido" type="xs:date"/>  
 <xs:element name="rendalapot" type="rendalapot"/>  
 <!-- komplex egyedi tipusaim -->  
 <xs:element name="cim">  
 <xs:complexType>  
 <xs:sequence>  
 <xs:element ref="varos"/>  
 <xs:element ref="utca"/>  
 <xs:element ref="hsz"/>  
 </xs:sequence>  
 <xs:attribute name="irszam" type="irszam" use="required"/>  
 </xs:complexType>  
 </xs:element>  
 <!-- Simple egyedi tipusaim -->  
 <xs:simpleType name="irszam">  
 <xs:restriction base="xs:integer">  
 <xs:pattern value="\d{4}"/>  
 </xs:restriction>  
 </xs:simpleType>  
 <xs:simpleType name="azon">  
 <xs:restriction base="xs:string">  
 <xs:pattern value="\d{8}"/>  
 </xs:restriction>  
 </xs:simpleType>  
 <xs:simpleType final="restriction" name="rendalapot">  
 <xs:restriction base="xs:string">  
 <xs:enumeration value="beszállítás alatt"/>  
 <xs:enumeration value="kiszállítás alatt"/>  
 <xs:enumeration value="kézbesítve"/>  
 </xs:restriction>  
 </xs:simpleType>  
</xs:schema>

1. **feladat** 
   1. **adatolvasás - DOMReaderYRNWBP.java**

package hu.domparse.yrnwbp;

import java.io.\*;

import javax.xml.parsers.\*;

import org.w3c.dom.\*;

import org.w3c.dom.traversal.\*;

import org.xml.sax.\*;

public class DOMReaderYRNWBP {

public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {

File xml = new File("src\\hu\\domparse\\yrnwbp\\XMLyrnwbp.xml");

// XML fájl DOM document alakítása

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

Document document = builder.parse(xml);

// DOM document átalakítása DOM DocumentTraversal formába

DocumentTraversal traversal = (DocumentTraversal) document;

//TreeWalker inicializálása

TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),

NodeFilter.SHOW\_ELEMENT | NodeFilter.SHOW\_TEXT, null, true);

//DOM bejárása és kiíratása

DomTraverser.traverseLevel(walker, "");

}

private static class DomTraverser {

public static void traverseLevel(TreeWalker walker, String indent) {

// aktuális csomópont

Node node = walker.getCurrentNode();

if (node.getNodeType() == Node.ELEMENT\_NODE) {

printElementNode(node, indent);

} else {

printTextNode(node, indent);

}

// rekurzívan meghívjuk a bejárást a DOM fában

for (Node n = walker.firstChild(); n != null; n = walker.nextSibling()) {

traverseLevel(walker, indent + " ");

}

walker.setCurrentNode(node);

}

private static void printElementNode(Node node, String indent) {

System.out.print(indent + node.getNodeName());

printElementAttributes(node.getAttributes());

}

private static void printElementAttributes(NamedNodeMap attributes) {

int length = attributes.getLength();

if (length > 0) {

System.out.print(" [ ");

for (int i = 0; i < length; i++) {

Node attribute = attributes.item(i);

System.out.printf("%s=%s%s", attribute.getNodeName(), attribute.getNodeValue(),

i != length - 1 ? ", " : "");

}

System.out.println(" ]");

} else {

System.out.println();

}

}

private static void printTextNode(Node node, String indent) {

String content\_trimmed = node.getTextContent().trim();

if (content\_trimmed.length() > 0) {

System.out.print(indent);

System.out.printf("{ %s }%n", content\_trimmed);

}

}

}

}

* 1. **adatmódosítás - DOMModifyYRNWBP.java**

package hu.domparse.yrnwbp;

import java.io.\*;

import java.text.ParseException;

import javax.xml.parsers.\*;

import javax.xml.xpath.\*;

import org.w3c.dom.\*;

import org.w3c.dom.traversal.\*;

import org.xml.sax.\*;

public class DOMReadYRNWBP {

public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException,

XPathExpressionException, DOMException, ParseException {

File xml = new File("src\\hu\\domparse\\yrnwbp\\XMLyrnwbp.xml");

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder = factory.newDocumentBuilder();

Document document = builder.parse(xml);

// a DOM document módosítása

DomModifier.modifyDom(document);

// DOM document átalakítása DOM DocumentTraversal formába

DocumentTraversal traversal = (DocumentTraversal) document;

//TreeWalker inicializálása

TreeWalker walker = traversal.createTreeWalker(document.getDocumentElement(),

NodeFilter.SHOW\_ELEMENT | NodeFilter.SHOW\_TEXT, null, true);

//DOM bejárása

DomTraverser.traverseLevel(walker, "");

}

private static class DomModifier {

public static void modifyDom(Document document) throws XPathExpressionException, DOMException, ParseException {

XPathFactory factory = XPathFactory.newInstance();

XPath xpath = factory.newXPath();

// 1.) Kiss Elemér telefon számának a megvéltoztatása

Node owner = (Node) xpath.evaluate("//ugyfel[./nev='Kiss Elemér']/tel",

document, XPathConstants.NODE);

owner.setTextContent("06706397628");

// 2.) Minden raktáron olyan termék 25% kedvezmény aminek a raktár készlete nagyobb 5 darabbnál

NodeList termekek = (NodeList) xpath.evaluate("//termek[./raktkeszlet>5]/ar", document, XPathConstants.NODESET);

System.out.println(termekek);

for (int i = 0; i < termekek.getLength(); i++) {

Node termek = termekek.item(i);

double price = Double.parseDouble(termek.getTextContent());

termek.setTextContent(Double.toString(price \* 0.75));

}

}

}

private static class DomTraverser {

public static void traverseLevel(TreeWalker walker, String indent) {

//aktuális csomópont

Node node = walker.getCurrentNode();

if (node.getNodeType() == Node.ELEMENT\_NODE) {

printElementNode(node, indent);

} else {

printTextNode(node, indent);

}

// rekurzívan meghívjuk a bejárást a DOM fában

for (Node n = walker.firstChild(); n != null; n = walker.nextSibling()) {

traverseLevel(walker, indent + " ");

}

walker.setCurrentNode(node);

}

private static void printElementNode(Node node, String indent) {

System.out.print(indent + node.getNodeName());

printElementAttributes(node.getAttributes());

}

private static void printElementAttributes(NamedNodeMap attributes) {

int length = attributes.getLength();

if (length > 0) {

System.out.print(" [ ");

for (int i = 0; i < length; i++) {

Node attribute = attributes.item(i);

System.out.printf("%s=%s%s", attribute.getNodeName(), attribute.getNodeValue(),

i != length - 1 ? ", " : "");

}

System.out.println(" ]");

} else {

System.out.println();

}

}

private static void printTextNode(Node node, String indent) {

String content\_trimmed = node.getTextContent().trim();

if (content\_trimmed.length() > 0) {

System.out.print(indent);

System.out.printf("{ %s }%n", content\_trimmed);

}

}

}

}