Jegyzőkönyv

**Adatkezelés XML környezetben**

**Féléves feladat**

Készítette: **Mészáros Marcell**

Nepunkód: **X0R8S7**

**A feladat leírása:**

Az én feladatomban egy olyan ER modellt kellett készítenem, amellyel egy étterem vezetni tudja a készleten lévő alapanyagait, ételeit, italait, kiszolgálóit, avagy futárjait az ügyfelekkel és az általuk leadott megrendelésekkel együtt. Az ER modellemben 6 egyed található

Étterem, Alapanyagok, Étel, Vendég, Kiszolgálók, illetve Rendelések

Az Étterem egyed tartalmazza azokat az adatokat, melyeket az adott étteremről tudni kell, úgy mint a Cégjegyzékszám (ez az egyedi azonosító), név, cím és telefonszám.

Az alapanyagok egyed megmutatja, hogy milyen alapanyagok vannak az ételekhez készleten, milyen mennyiségben vannak raktáron és mikor fog elkövetkezni a legelső szavatossági idő, amire a dolgozóknak oda kell figyelniük.

Az étel egyedben a menün lévő készételek találhatóak meg, ezekhez lesznek rendelve a raktáron lévő alapanyagok, és ezek eladásával lehet csökkenteni a készletet. Ami még nagyon fontos, hogy itt van tárolva minden eladható termék (étel) ára is. Itt a készlet egyed származtatott, mivel a készételek hozzávalóinak a mennyiségéből lehet megadni, hogy mennyi étel készülhet el az adott raktári mennyiségből.

A rendelések egyedben vannak a vendégek által az étterem felé leadott megrendelések készételekre, melyeket majd a kiszolgálók segítségével kaphatnak meg. Itt meg kell majd adni a rendelések fizetési módját, illetve a fogyasztást is, hogy helyben fogyasztják el, elviszi a vendég, vagy házhoz kell neki szállítani.

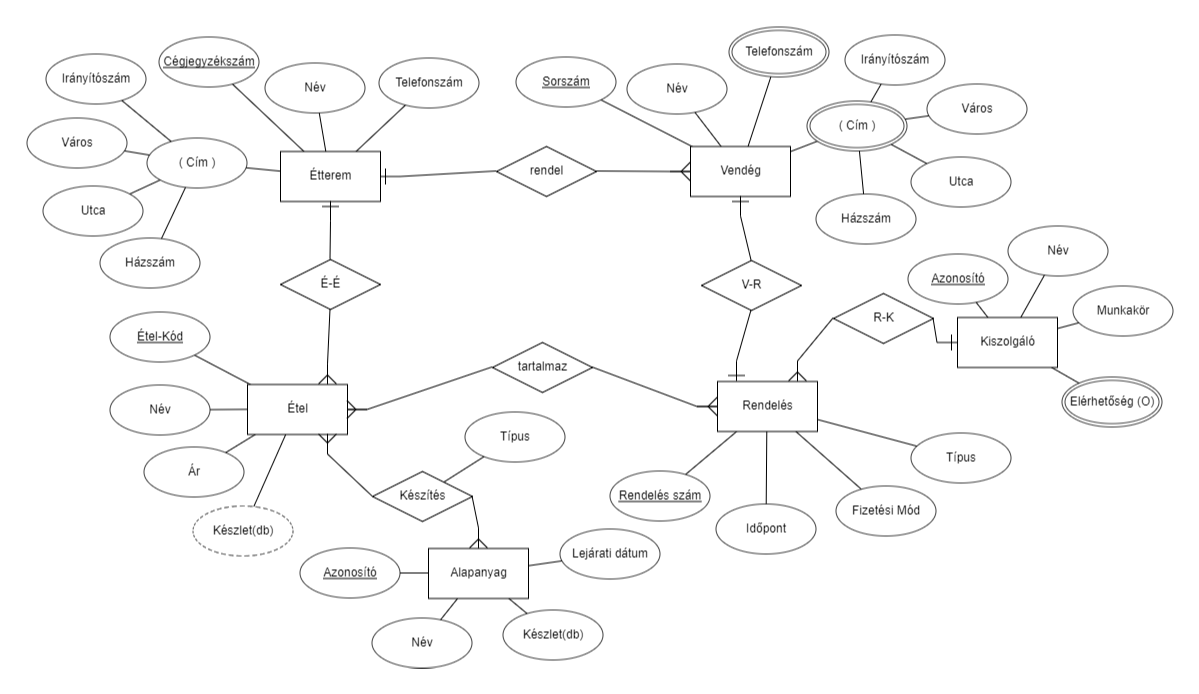
A kiszolgálók egyed mutatja meg, hogy milyen ki-, illetve felszolgálók, valamint futárok dolgoznak, típus szerint megadva, akik a felvett rendeléseket átadják a vendégeknek.

A Vendég egyed tartalmazza minden fontos adatot az étterem vendégeiről, amik szükségesek adott esetben a kiszolgálásukhoz, itt a telefonszám, illetve a cím tulajdonság többértékű, mivel rendelkezhetnek több telefonszámmal is, illetve házhozszállítás esetében nem feltétlen mindig ugyan arra a címre rendelnek, otthon mellett akár munkahelyre is kérhetnek kiszállítást.

Röviden az egyedek közötti kapcsolatokról. Az Étterem és Vendég kapcsolat 1:N, mivel az adott étterem szemszögéből több vendégük is lehet, viszont a szempontjukból irreleváns, hogy másik étterembe is elmehetnek vagy rendelhetnek.  
Az Étterem és Étel között szintén 1:N kapcsolat van, az étteremnek több étele is van, viszont a szempontjukból fontos módon azt az ételt csak abban az egy étteremben készítik el.  
Étel és Alapanyagok egyedek esetében N:M kapcsolat áll fenn, mivel egy alapanyagok több készételhez is szükséges lehet, illetve egy ételhez is több alapanyagra lehet szükség.  
Ezen logika mentén az Étel és Rendelés egyedek között is N:M kapcsolat van, egy rendelésben több étel is szerepelhet, mint ahogy egy étel több rendelésben.  
Rendelés, illetve Vendég egyedek esetében már más a helyzet, itt 1:1 kapcsolat van a feladatban először, mivel egy fogyasztás vagy rendelést bizonyosan egy vendéghez tartozik, mint ahogy egy alkalommal egy vendég egyszer is ad le megrendelést egy étterem számára.  
Elérkeztünk utolsó kapcsolatunkhoz, ez a Rendelés és a Kiszolgáló egyedek között köttetett meg, itt is 1:N kapcsolatról van szó, mivel egy kiszolgáló egyszerre több rendeléssel is el tud bánni, de egy adott megrendelést bizonyosan egyetlen kiszolgáló fog elvégezni.

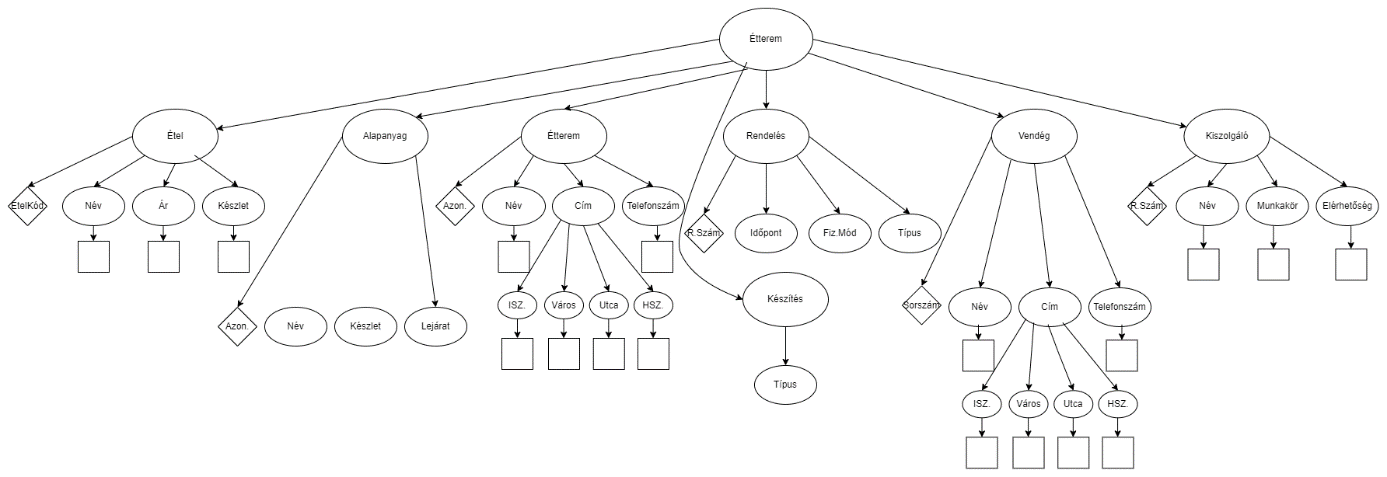
**1. feladat**

**1a, Az adatbázis ER modell:**



**1b, Az adatbázis konvertálása XDM modellre:**

Ebben a feladatrészletben elkészítettem a már megalkotott ER modell XDM modellre való konvertálását, ami alább látható is. Ezen modell segítségével könnyeden készíthettünk a feladatra megfelelő XML dokumentumot.



**1c, Az XDM modell alapján XML dokumentum készítése:**

A képen szöveg látható

Automatikusan generált leírás  
A képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírás

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

**A képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírásA képen szöveg látható

Automatikusan generált leírás**

**2. feladat**

**2a, Adatolvasás:**

**package hu.domparse.x0r8s7;**

**import org.w3c.dom.Element;**

**import org.w3c.dom.Document;**

**import org.xml.sax.SAXException;**

**import org.w3c.dom.Node;**

**import org.w3c.dom.NodeList;**

**import java.io.File;**

**import java.io.IOException;**

**import javax.xml.parsers.DocumentBuilder;**

**import javax.xml.parsers.DocumentBuilderFactory;**

**import javax.xml.parsers.ParserConfigurationException;**

**public class DOMReadx0r8s7 {**

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

**try {**

**File xmlFile = new File("XMLx0r8s7.xml"); // fájl beolvasása**

**DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();**

**DocumentBuilder dBuilder = factory.newDocumentBuilder();**

**Document doc = dBuilder.parse(xmlFile);**

**doc.getDocumentElement().normalize();**

**Read(doc);**

**//kivételkezelés**

**} catch (ParserConfigurationException pce) {**

**pce.printStackTrace();**

**} catch (IOException ioe) {**

**ioe.printStackTrace();**

**} catch (SAXException sae) {**

**sae.printStackTrace();**

**}**

**}**

**public static void Read(Document doc) {**

**NodeList nList = doc.getElementsByTagName("etterem1"); //NoteList létrehozása, amiben tároljuk az elemeket**

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

**Node nNode = nList.item(i);**

**Element element = (Element) nNode;**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) { //az azonosito kodok**

**String kod = element.getAttribute("kod");**

**String azon = element.getAttribute("azon");**

**String kk = element.getAttribute("kk");**

**ReadEtteremById(doc, kod);**

**ReadAlapanyagById(doc, azon);**

**ReadKiszolgaloById(doc, kk);**

**}**

**}**

**}**

**public static void ReadEtteremById(Document doc, String kod) {**

**NodeList nList = doc.getElementsByTagName("etterem1"); //étterem adatait tartalmazó nodeList létrehozása**

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

**Node nNode = nList.item(i);**

**Element element = (Element) nNode;**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) {**

**if (element.getAttribute("kod").equals(eid)) { //kiirandó adatok kiválasztása**

**String telefonszam = element.getElementsByTagName("telefonszam").item(0).getTextContent();**

**String etel = element.getElementsByTagName("etel").item(0).getTextContent();**

**String nev = element.getElementsByTagName("nev").item(0).getTextContent();**

**String cim = element.getElementsByTagName("cim").item(0).getTextContent();**

**System.out.println(" Az étterem adatai: \n\tNév:\t" + nev**

**+ "\n\tTelefonszám:\t" + telefonszam + "\n\tÉtelek::\t" + etel + "\n\tCim:\t" + cim); //kiíratás**

**}**

**}**

**}**

**}**

**public static void ReadAlapanyagById(Document doc, String azon) {**

**NodeList nList = doc.getElementsByTagName("alapanyag"); //az alapanyagokat tartalmazo nodelist létrehozása**

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

**Node nNode = nList.item(i);**

**Element element = (Element) nNode;**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) {**

**if (element.getAttribute("azon").equals(bid)) { //kiirandó adatok kiválasztása**

**String nev = element.getElementsByTagName("nev").item(0).getTextContent();**

**String keszlet = element.getElementsByTagName("keszlet").item(0).getTextContent();**

**String lejarat = element.getElementsByTagName("lejarat").item(0).getTextContent();**

**System.out.println("Alapanyagok: \n\tNév:\t" + nev + "\n\tkeszlet:\t" + keszlet**

**+ "\n\tLejárat:\t" + lejarat); //kiíratás**

**String ekod = element.getAttribute("ekod");**

**ReadTermeloById(doc, ekod);**

**}**

**}**

**}**

**}**

**public static void ReadEtelById(Document doc, String ekod) {**

**NodeList nList = doc.getElementsByTagName("etel"); //a készételek adatait tartalmazó nodeList létrehozása**

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

**Node nNode = nList.item(i);**

**Element element = (Element) nNode;**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) {**

**if (element.getAttribute("ekod").equals(ekod)) { //kiirandó adatok kiválasztása**

**String nev = element.getElementsByTagName("nev").item(0).getTextContent();**

**String ar = element.getElementsByTagName("ar").item(0).getTextContent();**

**String keszlet = element.getElementsByTagName("keszlet").item(0).getTextContent();**

**System.out.println("Készétel: \n\tNév:\t" + nev + "\n\tÁr:\t" + ar**

**+ "\n\tKészlet:\t" + keszlet); //kiíratás**

**}**

**}**

**}**

**}**

**public static void ReadVendegById(Document doc, String sorszam) {**

**NodeList nList = doc.getElementsByTagName("vendeg"); //a vendégek adatait tartalmazó nodeList létrehozása**

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

**Node nNode = nList.item(i);**

**Element element = (Element) nNode;**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) {**

**if (element.getAttribute("sorszam").equals(sorszam)) { //kiirandó adatok kiválasztása**

**String nev = element.getElementsByTagName("nev").item(0).getTextContent();**

**String telefonszam = element.getElementsByTagName("telefonszam").item(0).getTextContent();**

**System.out.println("Személyzet adatok: \n\tNév:\t" + nev + "\n\tTelefonszám:\t" + telefonszam**

**+ "\n\tCím:\t" + cim + "\n\tMunkakör:\t" + munkakor) //kiíratás**

**String rsz = element.getAttribute("rsz");**

**ReadFoglalasById(doc, rsz);**

**}**

**}**

**}**

**}**

**public static void ReadRendelesById(Document doc, String rsz) {**

**NodeList nList = doc.getElementsByTagName("rendeles"); //a rendelések adatait tartalmazó nodeList létrehozása**

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

**Node nNode = nList.item(i);**

**Element element = (Element) nNode;**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) {**

**if (element.getAttribute("rsz").equals(fid)) { //kiirandó adatok kiválasztása**

**String idopont = element.getElementsByTagName("idopont").item(0).getTextContent();**

**String fizmod = element.getElementsByTagName("fizmod").item(0).getTextContent();**

**String tipus = element.getElementsByTagName("tipus").item(0).getTextContent();**

**System.out.println("Rendelés adatok: \n\tIdőpont:\t" + idopont + "\n\tFizetési Mód:\t" + fizmod + "\n\tTípus:\t" + tipus); //kiíratás**

**}**

**}**

**}**

**}**

**}**

**//#CreatedByMeszarosMarcell**

**2b, Adatmódosítás:**

**package hu.domparse.x0r8s7;**

**import org.w3c.dom.Element;**

**import org.w3c.dom.Document;**

**import org.w3c.dom.NamedNodeMap;**

**import org.w3c.dom.Node;**

**import org.w3c.dom.NodeList;**

**import java.io.File;**

**import javax.xml.transform.Transformer;**

**import javax.xml.transform.TransformerFactory;**

**import javax.xml.parsers.DocumentBuilder;**

**import javax.xml.parsers.DocumentBuilderFactory;**

**import javax.xml.transform.dom.DOMSource;**

**import javax.xml.transform.stream.StreamResult;**

**public class DOMModifyx0r8s7 {**

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

**try {**

**File inputFile = new File("XMLx0r8s7.xml");**

**DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();**

**DocumentBuilder docBuilder = docFactory.newDocumentBuilder();**

**Document doc = docBuilder.parse(inputFile);**

**Node vendeg = doc.getElementsByTagName("vendeg").item(0);**

**NamedNodeMap attr = vendeg.getAttributes(); // 10-re módosítjuk a sorszam-t 1-rõl**

**Node nodeAttr = attr.getNamedItem("sorszam");**

**nodeAttr.setTextContent("10");**

**NodeList list = vendeg.getChildNodes();**

**for (int temp = 0; temp < list.getLength(); temp++) { // a hibaüzenet megváltoztatása**

**Node node = list.item(temp);**

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

**Element hElement = (Element) node;**

**if ("sorszam".equals(hElement.getNodeName())) {**

**if ("123456BC".equals(hElement.getTextContent()))**

**hElement.setTextContent("123456BC");**

**}**

**}**

**}**

**NodeList list2 = doc.getElementsByTagName("alapanyag");**

**for (int j = 0; j < list2.getLength(); j++) {**

**Node alapanyag = list2.item(j);**

**if (alapanyag.getNodeType() == Node.ELEMENT\_NODE) {**

**String id = alapanyag.getAttributes().getNamedItem("azon").getTextContent();**

**if ("1".equals(id.trim())) {**

**NodeList gyerekNodes = alapanyag.getChildNodes();**

**for (int k = 0; k < gyerekNodes.getLength(); k++) {**

**Node item = gyerekNodes.item(k);**

**if (item.getNodeType() == Node.ELEMENT\_NODE) { // kitöröljük az vendégbõl a kód mezõt**

**if ("telefonszam".equalsIgnoreCase(item.getNodeName())) {**

**alapanyag.removeChild(item);**

**}**

**// módosítjuk az 1-es sorszam-mal rendelkezõ nevét**

**if ("nev".equalsIgnoreCase(item.getNodeName())) {**

**item.setTextContent("ABC");**

**}**

**}**

**}**

**}**

**}**

**}**

**// kiíratás**

**TransformerFactory transformerFactory = TransformerFactory.newInstance();**

**Transformer transformer = transformerFactory.newTransformer();**

**DOMSource source = new DOMSource(doc);**

**System.out.println("-----------Módosított File-----------");**

**StreamResult consoleResult = new StreamResult(System.out);**

**transformer.transform(source, consoleResult);**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**}**

**}**

**2c, Adat lekérdezés:**

**package hu.domparse.x0r8s7;**

**import java.io.File;**

**import java.util.Scanner;**

**import java.io.IOException;**

**import javax.xml.parsers.DocumentBuilder;**

**import javax.xml.parsers.DocumentBuilderFactory;**

**import javax.xml.transform.TransformerException;**

**import org.w3c.dom.Element;**

**import org.w3c.dom.Document;**

**import org.w3c.dom.Node;**

**import org.w3c.dom.NodeList;**

**import org.xml.sax.SAXException;**

**import javax.xml.parsers.ParserConfigurationException;**

**public class DOMQueryx0r8s7 {**

**public static void main(String[] args)**

**throws ParserConfigurationException, IOException, SAXException, TransformerException {**

**File xmlFile = new File("XMLx0r8s7.xml"); //beolvassuk a fájlt**

**DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();**

**DocumentBuilder dBuilder = factory.newDocumentBuilder();**

**Document doc = dBuilder.parse(xmlFile);**

**doc.getDocumentElement().normalize();**

**LoadtermeloQuery(doc);**

**}**

**public static void LoadtermeloQuery(Document doc) throws TransformerException {**

**NodeList nodeList = doc.getElementsByTagName("etel"); //etel noteList létrehoz**

**String etel;**

**Element element = null;**

**Node nNode = null;**

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

**nNode = nodeList.item(i);**

**element = (Element) nNode;**

**String nev = element.getElementsByTagName("nev").item(0).getTextContent(); //név szerint lekérdezzük**

**System.out.println(nev);**

**}**

**System.out.println("\nÍrja be annak az ételnek a nevét, amelyiknek adatait szeretné látni:");**

**Scanner sc = new Scanner(System.in); //consolról beolvas**

**termelo = sc.nextLine();**

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

**nNode = nodeList.item(i);**

**element = (Element) nNode;**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) {**

**if (etel.equals("Margaréta")) {**

**LoadEtelQuery(doc, "1");**

**break;**

**}**

**if (etel.equals("Sonkasajt")) {**

**LoadEtelQuery(doc, "2");**

**break;**

**}**

**if (etel.equals("Szalámisajt")) {**

**LoadEtelQuery(doc, "3");**

**break;**

**}**

**if (etel.equals("Hagymasajt")) {**

**LoadEtelQuery(doc, "4");**

**break;**

**}**

**}**

**}**

**sc.close();**

**}**

**public static void LoadEtelQuery(Document doc, String ekod) throws TransformerException {**

**NodeList nodeList = doc.getElementsByTagName("etel"); //etel noteList létrehozása**

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

**Node nNode = nodeList.item(i);**

**Element element = (Element) nNode;**

**String ekod = element.getAttribute("ekod"); //az ekod alapján keressük az elemet**

**if (nNode.getNodeType() == Node.ELEMENT\_NODE) {**

**if (ekod.equals(azon)) {**

**String Bid = element.getAttribute("ekod"); //ekod alapján kivesszük az elemet**

**DOMReadx0r8s7.ReadEtelById(doc, ekod); //meghívjuk az etel függvényt**

**}**

**}**

**}**

**}**

**}**

**//CreatedByMeszarosMarcell**