JEGYZŐKÖNYV

Adatkezelés XML környezetben

# Féléves feladat

Könyvtári nyilvántartás létrehozása

Készítette: **Bartók-Balog Péter**

Neptunkód: **ASNQPH**

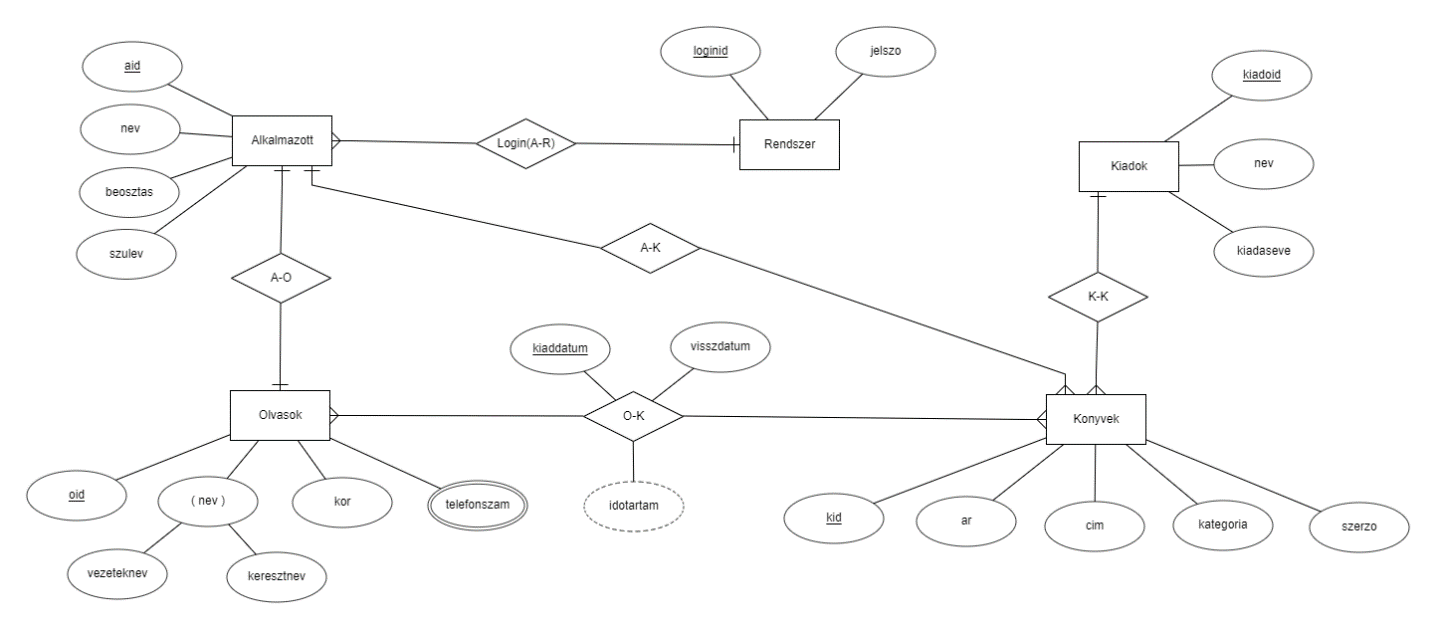
**A feladat leírása:**

A feladat egy hiteles adatbázis létrehozása, különböző lekérdezések kialakítása XML nyelven keresztül. A feladat megoldásának feltétele, hogy az adatbázis tartalmazzon legalább 5 egyedet, egyedenként 4 tulajdonságot, valamint többféle kapcsolatot. A probléma megoldása kezdődik az adatbázis témájának kiválasztásával, ami jelen esetben egy könyvtári rendszer adatbázisa. Ezt követi az adatbázis szerkezetét leíró ER, valamint XDM modelljének kialakítása. Amint a megfelelő konvertálások megtörténetek a feladat folytatódik az xml fájl létrehozásával és feltöltésével. Miután a dokumentum feltöltődött adattal, létre kell hozni hozzá egy XML Schema fájlt az előzőleg létrehozott dokumentum alapján. A feladat befejezéshez szükséges egy jegyzőkönyv, illetve a feladat megvédése a határidőben feltüntetett gyakorlat időpontjáig.

**ER modell leírása:**

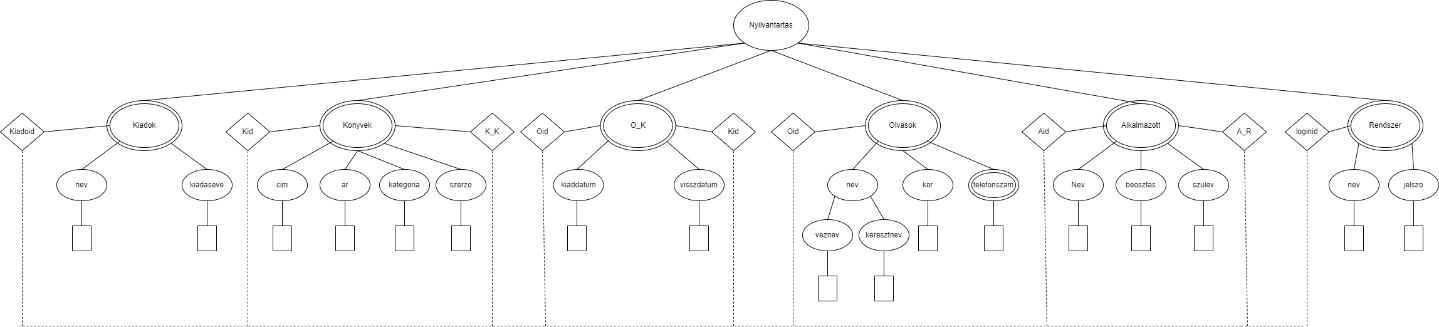
A feladatban egy reális, és valós E-R adatbázist hozok létre. Az adatbázisnak a következőek a jellemzői:

* Alkalmazott egyed: „**aid**” **egyedi tulajdonság**, „**nev**”, „**beosztas**”, „**szulev**” tulajdonság.
* Rendszer egyed: „**loginid**” **egyedi tulajdonság, „nev”** és „**jelszo**”, azaz a belépéshez szükséges adatok.
* Olvasok egyed: „**oid**” **egyedi tulajdonság**, „**nev**” **összetett tulajdonság** („vezeteknev”, „keresztnev”), „**kor**”, „**telefonszam**” **többértékű tulajdonság**, hiszen többet telefonszám is lehetséges.
* Konyvek egyed: „**kid**” **egyedi tulajdonság**, „**ar**”, „**cím**”, „**kategoria**”, „**szerzo**” tulajdonságok magától értetődőek.
* Kiadok egyed: „**kiadoid**” **egyedi tulajdonság**, „**nev**”, „**kiadaseve**” magától értetődő.
* O-K kapcsolat: **N-M** kapcsolat „**idotartam**” tulajdonsága **származtatott**, „kiaddatum” és „visszdatum” hányadosa adja.
* K-K kapcsolat: **1-N** kapcsolat „Konyvek”, és „Kiadok” között, hiszen egy kiadó több könyvet is kiadhat.
* A-K kapcsolat: **1-N** kapcsolat. Az alkalmazottak több könyvet is ápolhatnak.
* A-R kapcsolat: **1-N** kapcsolat. A „Rendszer” egyed több alkalmazottnál is előfordulhat.



**Adatbázis konvertálása XDM modellre:**

* „Alkalmazott” egyedből egy elem lesz, a meglévő tulajdonságokon kívül lesz még, egy „loginid” idegen kulcs is. „aid” mező folyamatosan növekszik.
* „Olvasok” egyedből egy elem lesz, tulajdonságai közül a „telefonszam” tulajdonság többértékű. A „nev” összetett tulajdonság feloszlik „keresztnev”, és „vezeteknev” tulajdonságra. „oid” folyamatosan növekszik.
* „Kiadok” egyedből egy elem lesz, tulajdonságai maradnak ugyan azok. „kiadoid” folyamatosan növekszik, „nev” mező nem lehet üres.
* „Rendszer” egyedből szintén egy elem lesz, tulajdonságai maradnak. „loginid” nem lehet üres.
* „Konyvek” egyedből egy elem lesz meglévő tulajdonságain kívül három idegenkulcs tartja a kapcsolatot „Olvasok”, „Kiadok”, és „Alkalmazott” tábla között. „kid” folyamatosan növekszik, „cim”, „kategoria”, „szerzo”, nem lehet üres.
* O-K kapcsolat N-M kapcsolat ezért egy külön elembe kerül. Meglévő tulajdonságokon kívül két idegenkulcs tartja a kapcsolatot „Konyvek”, és „Olvasók” között. „idotartam” tulajdonság származtatott, így az kimarad az elemből.

****

**Adatbázis relációs sémájal:**

* Alkalmazott [aid, nev, loginid]
* Rendszer [loginid, jelszo]
* O-K [kiaddatum, visszdatum, kid, oid]
* Olvasok [oid, vezeteknev, keresztnev, kor]
* Telefonszam [oid]
* Konyvek [kid, ar, cím, kategoria, szerzo, oid, kiadoid, aid]
* Kiadok [kiadoid, nev, kiadaseve]

**Adatbázis konvertálása XDM modellről XML-re:**

Fejlesztőkörnyezet segítségével létrehozom az xml fájlt. Létrehozom a modellt átfogó nyilvantartas elemet, majd az XMD modellt felhasználva létrehozom az első példányt, aminek mintájára létrehozom a maradék kettőt, majd lezárom az elemet.

<?**xml** version=*"1.0"* encoding=*"UTF-8"*?>

<**nyilvantartas**>

<!-- Első példány létrehozása -->

<**kiadok** kiadoid = *"1"*>

<**nev**>Legjobb Kiadó</**nev**>

<**kiadaseve**>1900</**kiadaseve**>

</**kiadok**>

<**konyvek** kid = *"1"* kiadoid = *"1"*>

<**cim**>Egy könyv</**cim**>

<**ar**>2500</**ar**>

<**kategoria**>Misztikus</**kategoria**>

<**szerzo**>Hegedűs Gábor</**szerzo**>

</**konyvek**>

<**O\_K** oid = *"1"* kid = *"1"*>

<**kiaddatum**>2022.10.27</**kiaddatum**>

<**visszdatum**>2022.10.29</**visszdatum**>

</**O\_K**>

<**olvasok** oid = *"1"*>

<**veznev**>Balog</**veznev**>

<**keresztnev**>Levente</**keresztnev**>

<**kor**>18</**kor**>

<**telefonszam**>06703325674</**telefonszam**>

</**olvasok**>

<**alkalmazott** aid = *"1"* loginid = *"1"*>

<**nev**>Kiss Eszter</**nev**>

<**beosztas**>Penztaros</**beosztas**>

<**szulev**>2001</**szulev**>

</**alkalmazott**>

<**rendszer** loginid = *"1"*>

<**nev**>Kiss</**nev**>

<**jelszo**>jelszo1</**jelszo**>

</**rendszer**>

<!-- Második példány létrehozása -->

<**kiadok** kiadoid = *"2"*>

<**nev**>Kevesbe Jo Kiado</**nev**>

<**kiadaseve**>1970</**kiadaseve**>

</**kiadok**>

<**konyvek** kid = *"2"* kiadoid = *"2"*>

<**cim**>Masik Konyv</**cim**>

<**ar**>5000</**ar**>

<**kategoria**>Drama</**kategoria**>

<**szerzo**>Toth Miklos</**szerzo**>

</**konyvek**>

<**O\_K** oid = *"2"* kid = *"2"*>

<**kiaddatum**>2022.09.13.</**kiaddatum**>

<**visszdatum**>2022.09.17</**visszdatum**>

</**O\_K**>

<**olvasok** oid = *"2"*>

<**veznev**>Kovacs</**veznev**>

<**keresztnev**>Denes</**keresztnev**>

<**kor**>33</**kor**>

<**telefonszam**>06302324586</**telefonszam**>

</**olvasok**>

<**alkalmazott** aid = *"2"* loginid = *"2"*>

<**nev**>Varga Zsombor</**nev**>

<**beosztas**>Asszisztens</**beosztas**>

<**szulev**>1998</**szulev**>

</**alkalmazott**>

<**rendszer** loginid = *"2"*>

<**nev**>Kovacs</**nev**>

<**jelszo**>jelszo2</**jelszo**>

</**rendszer**>

<!-- Harmadik példány létrehozása -->

<**kiadok** kiadoid = *"3"*>

<**nev**>Rossz Kiado</**nev**>

<**kiadaseve**>2011</**kiadaseve**>

</**kiadok**>

<**konyvek** kid = *"3"* kiadoid = *"3"*>

<**cim**>Harmadik Konyv</**cim**>

<**ar**>12000</**ar**>

<**kategoria**>Tortenelmi</**kategoria**>

<**szerzo**>Kantor Bela</**szerzo**>

</**konyvek**>

<**O\_K** oid = *"3"* kid = *"3"*>

<**kiaddatum**>2022.08.23</**kiaddatum**>

<**visszdatum**>2022.09.01</**visszdatum**>

</**O\_K**>

<**olvasok** oid = *"3"*>

<**veznev**>Gyuran</**veznev**>

<**keresztnev**>Elek</**keresztnev**>

<**kor**>23</**kor**>

<**telefonszam**>06207767854</**telefonszam**>

</**olvasok**>

<**alkalmazott** aid = *"3"* loginid = *"3"*>

<**nev**>Szabo Bence</**nev**>

<**beosztas**>Raktaros</**beosztas**>

<**szulev**>2000</**szulev**>

</**alkalmazott**>

<**rendszer** loginid = *"3"*>

<**nev**>Szabo</**nev**>

<**jelszo**>jelszo3</**jelszo**>

</**rendszer**>

</**nyilvantartas**>

**Adatbázis konvertálása XML modellről XML Schema-ra:**

Kezdetben létrehozom a modellhez szükséges egyszerű típusokat a kulcsok, illetve a dátumok kezeléséhez. Ezt követően létrehozom az elsődleges, valamint az idegen kulcsokat. Létrehozom a fő elemet, amiben hivatkozok az alelemekre ref segítségével, és beállítom az előfordulásukat korlátlanra. Ezt követően létrehozom az alelemeket.

<?**xml** version=*"1.0"* encoding=*"UTF-8"*?>

<**xs:schema** xmlns:xs=*"http://www.w3.org/2001/XMLSchema"* elementFormDefault=*"qualified"* >

<**xs:simpleType** name=*"idTipus"*>

<**xs:restriction** base=*"xs:string"*>

<**xs:minLength** value=*"1"*/>

<**xs:maxLength** value=*"3"*/>

</**xs:restriction**>

</**xs:simpleType**>

<**xs:simpleType** name=*"datumTipus"*>

<**xs:restriction** base = *"xs:string"*>

</**xs:restriction**>

</**xs:simpleType**>

<**xs:key** name = *"kiadok\_kulcs"*>

<**xs:selector** xpath = *"kiadok"*/>

<**xs:field** xpath = *"@kiadoid"* />

</**xs:key**>

<**xs:key** name = *"konyvek\_kulcs"*>

<**xs:selector** xpath = *"konyvek"*/>

<**xs:field** xpath = *"@kid"* />

</**xs:key**>

<**xs:key** name = *"olvasok\_kulcs"*>

<**xs:selector** xpath = *"olvasok"*/>

<**xs:field** xpath = *"@oid"* />

</**xs:key**>

<**xs:key** name = *"alkalmazott\_kulcs"*>

<**xs:selector** xpath = *"alkalmazott"*/>

<**xs:field** xpath = *"@kaid"* />

</**xs:key**>

<**xs:key** name = *"rendszer\_kulcs"*>

<**xs:selector** xpath = *"rendszer"*/>

<**xs:field** xpath = *"@loginid"* />

</**xs:key**>

<**xs:keyref** refer = *"kiadok\_kulcs"* name= *"konyvek\_kiadok\_idegenkulcs"*>

<**xs:selector** xpath = *"konyvek"* />

<**xs:field** xpath = *"k\_k"* />

</**xs:keyref**>

<**xs:keyref** refer = *"konyvek\_kulcs"* name= *"ok\_konyvek\_idegenkulcs"*>

<**xs:selector** xpath = *"O\_K"* />

<**xs:field** xpath = *"o\_k\_k"* />

</**xs:keyref**>

<**xs:keyref** refer = *"olvasok\_kulcs"* name= *"ok\_olvasok\_idegenkulcs"*>

<**xs:selector** xpath = *"O\_K"* />

<**xs:field** xpath = *"o\_k\_o"* />

</**xs:keyref**>

<**xs:keyref** refer = *"rendszer\_kulcs"* name= *"alkalmazott\_rendszer\_idegenkulcs"*>

<**xs:selector** xpath = *"alkalmazott"* />

<**xs:field** xpath = *"a\_r"* />

</**xs:keyref**>

<**xs:element** name=*"nyilvantartas"*>

<**xs:complexType**>

<**xs:sequence**>

<**xs:element** ref = *"kiadok"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** ref = *"konyvek"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** ref = *"O\_K"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** ref = *"olvasok"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** ref = *"alkalmazott"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** ref = *"rendszer"* maxOccurs=*"unbounded"*></**xs:element**>

</**xs:sequence**>

</**xs:complexType**>

</**xs:element**>

<**xs:element** name= *"kiadok"*>

<**xs:complexType**>

<**xs:sequence**>

<**xs:element** name = *"nev"* type = *"xs:string"* maxOccurs=*"unbounded"*/>

<**xs:element** name = *"kiadaseve"* type = *"xs:integer"* maxOccurs=*"unbounded"*/>

</**xs:sequence**>

<**xs:attribute** name =*"kiadoid"* type = *"idTipus"* use = *"required"*></**xs:attribute**>

</**xs:complexType**>

</**xs:element**>

<**xs:element** name = *"konyvek"*>

<**xs:complexType**>

<**xs:sequence**>

<**xs:element** name = *"cim"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"ar"* type = *"xs:integer"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"kategoria"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"szerzo"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

</**xs:sequence**>

<**xs:attribute** name = *"kid"* type = *"idTipus"* use= *"required"* ></**xs:attribute**>

<**xs:attribute** name =*"k\_k"* type = *"idTipus"* use = *"required"*></**xs:attribute**>

</**xs:complexType**>

</**xs:element**>

<**xs:element** name =*"O\_K"*>

<**xs:complexType**>

<**xs:sequence**>

<**xs:element** name = *"kiaddatum"* type = *"datumTipus"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"visszdatum"* type = *"datumTipus"* maxOccurs=*"unbounded"*></**xs:element**>

</**xs:sequence**>

<**xs:attribute** name =*"o\_k\_k"* type =*"idTipus"* use=*"required"*></**xs:attribute**>

<**xs:attribute** name =*"o\_k\_o"* type =*"idTipus"* use=*"required"*></**xs:attribute**>

</**xs:complexType**>

</**xs:element**>

<**xs:element** name =*"olvasok"*>

<**xs:complexType**>

<**xs:sequence**>

<**xs:element** name = *"veznev"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"keresztnev"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"kor"* type = *"xs:integer"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"telefonszam"* type = *"xs:integer"* maxOccurs=*"unbounded"*></**xs:element**>

</**xs:sequence**>

<**xs:attribute** name = *"oid"* type =*"idTipus"* use= *"required"*></**xs:attribute**>

</**xs:complexType**>

</**xs:element**>

<**xs:element** name = *"alkalmazott"*>

<**xs:complexType**>

<**xs:sequence**>

<**xs:element** name = *"nev"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"beosztas"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"szulev"* type = *"xs:integer"* maxOccurs=*"unbounded"*></**xs:element**>

</**xs:sequence**>

<**xs:attribute** name = *"aid"* type= *"idTipus"* use= *"required"*></**xs:attribute**>

<**xs:attribute** name = *"a\_r"* type = *"idTipus"* use = *"required"*></**xs:attribute**>

</**xs:complexType**>

</**xs:element**>

<**xs:element** name = *"rendszer"*>

<**xs:complexType**>

<**xs:sequence**>

<**xs:element** name = *"nev"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

<**xs:element** name = *"jelszo"* type = *"xs:string"* maxOccurs=*"unbounded"*></**xs:element**>

</**xs:sequence**>

<**xs:attribute** name = *"loginid"* type = *"idTipus"* use = *"required"*></**xs:attribute**>

</**xs:complexType**>

</**xs:element**>

</**xs:schema**>

**Adatbázis beolvasása, kiíratása DOM segítségével:**

Először a megfelelő importok használata, fájlok megnyitása, létrehozása, majd a builderek felépítése. Dokumentum, és a gyökérelem kezelése. Ezt követően a gyermekelemek hozzáadása a gyökérelemhez, majd formázott output-ba rendezés, és kiíratás.

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

//Fileok megnyitása

File xmlFile = new File("XMLASNQPH.xml");

File myFile = new File("XMLASNQPH1.xml");

//Builderek létehozása

DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();

DocumentBuilder dBuilder = factory.newDocumentBuilder();

//Dokumentumok beállítása

Document doc = dBuilder.parse(xmlFile);

Document doc2 = dBuilder.newDocument();

//Gyökérelem meghatározása

Element root = doc2.createElementNS("XMLASNQPH", "nyilvantartas");

doc2.appendChild(root);

//Gyerekelemek létrehozása XML alapján

root.appendChild(*createKiadok*(doc2,"1", "Legjobb Kiadó", "1900"));

root.appendChild(*createKiadok*(doc2,"2", "Kevesbe Jo Kiado","1970"));

root.appendChild(*createKiadok*(doc2,"3", "Rossz Kiado", "2011"));

root.appendChild(*createKonyvek*(doc2,"1","1", "Egy Konyv", "2500", "Misztikus", "Hegedus Gabor"));

root.appendChild(*createKonyvek*(doc2,"2", "2", "Masik Konyv","5000", "Drama", "Toth Miklos"));

root.appendChild(*createKonyvek*(doc2,"3","3", "Harmadik Konyv", "12000", "Tortenelmi","Kantor Bela"));

root.appendChild(*createO\_K*(doc2,"1","1", "2022.10.17", "2022.10.29"));

root.appendChild(*createO\_K*(doc2,"2", "2", "2022.09.13","2022.09.17"));

root.appendChild(*createO\_K*(doc2,"3","3", "2022.08.23", "2022.09.01"));

root.appendChild(*createOlvasok*(doc2,"1","Balog", "Levente", "18", "06703325674"));

root.appendChild(*createOlvasok*(doc2,"2", "Kovacs","Denes", "33", "06302324586"));

root.appendChild(*createOlvasok*(doc2,"3","Gyuran", "Elek", "23","06207767854"));

root.appendChild(*createAlkalmazott*(doc2,"1","1", "Kiss Eszter", "Penztaros", "2001"));

root.appendChild(*createAlkalmazott*(doc2,"2", "2", "Varga Zsombor","Asszisztens", "1998"));

root.appendChild(*createAlkalmazott*(doc2,"3","3", "Szabo Bence", "Raktaros", "2000"));

root.appendChild(*createRendszer*(doc2,"1", "Kiss", "jelszo1"));

root.appendChild(*createKiadok*(doc2,"2", "Kovacs","jelszo2"));

root.appendChild(*createKiadok*(doc2,"3", "Szabo", "jelszo3"));

TransformerFactory transformerFactory = TransformerFactory.*newInstance*();

Transformer transf = transformerFactory.newTransformer();

//Output megformázása

transf.setOutputProperty(OutputKeys.***ENCODING***, "UTF-8");

transf.setOutputProperty(OutputKeys.***INDENT***, "yes");

transf.setOutputProperty("{http://xml.apache.org/xslt}indent-amount", "2");

DOMSource source = new DOMSource(doc2);

//Kiíratás

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

StreamResult file = new StreamResult (myFile);

transf.transform(source, console);

transf.transform(source, file);

//Megnyitott xml file kezelése

doc.getDocumentElement().normalize();

//Gyökér elem kiíratás

System.***out***.println("\nGyoker elem: " + doc.getDocumentElement().getNodeName());

//Gyermek elemek listába rendezése

NodeList kiadoList = doc.getElementsByTagName("kiadok");

NodeList konyvekList = doc.getElementsByTagName("konyvek");

NodeList O\_KList = doc.getElementsByTagName("O\_K");

NodeList olvasokList = doc.getElementsByTagName("olvasok");

NodeList alkalmazottList = doc.getElementsByTagName("alkalmazott");

NodeList rendszerList = doc.getElementsByTagName("rendszer");

//File-ba írás

StringWriter sw = new StringWriter();

transf.transform(source, new StreamResult(sw));

FileWriter fw = new FileWriter("file.txt");

fw.write(sw.toString());

fw.close();

//Listák feltöltése

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

Node kiadoNode = kiadoList.item(i);

System.***out***.println("\nJelenlegi elem: " + kiadoNode.getNodeName());

if (kiadoNode.getNodeType() == Node.***ELEMENT\_NODE***)

{

Element elem = (Element) kiadoNode;

String kiadoid = elem.getAttribute("kiadoid");

Node node1 = elem.getElementsByTagName("nev").item(0);

String nev = node1.getTextContent();

Node node2 = elem.getElementsByTagName("kiadaseve").item(0);

String kiadaseve = node2.getTextContent();

System.***out***.println("Kiadoid: " + kiadoid);

System.***out***.println("Nev: " + nev);

System.***out***.println("Kiadaseve: " + kiadaseve);

}

}

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

Node konyvekNode = konyvekList.item(i);

System.***out***.println("\nJelenlegi elem: " + konyvekNode.getNodeName());

if (konyvekNode.getNodeType() == Node.***ELEMENT\_NODE***)

{

Element elem = (Element) konyvekNode;

String kid = elem.getAttribute("kid");

Element elem2 = (Element) konyvekNode;

String kiadoid = elem2.getAttribute("kiadoid");

Node node1 = elem.getElementsByTagName("cim").item(0);

String cim = node1.getTextContent();

Node node2 = elem.getElementsByTagName("ar").item(0);

String ar = node2.getTextContent();

Node node3 = elem.getElementsByTagName("kategoria").item(0);

String kategoria = node3.getTextContent();

Node node4 = elem.getElementsByTagName("szerzo").item(0);

String szerzo = node4.getTextContent();

System.***out***.println("Kid: " + kid);

System.***out***.println("Kiadoid: " + kiadoid);

System.***out***.println("Cim: " + cim);

System.***out***.println("Ar: " + ar);

System.***out***.println("Kategoria: " +kategoria);

System.***out***.println("Szerzo: " + szerzo);

}

}

for (int i = 0; i < O\_KList.getLength(); i++) {

Node O\_KNode = O\_KList.item(i);

System.***out***.println("\nJelenlegi elem: " + O\_KNode.getNodeName());

if (O\_KNode.getNodeType() == Node.***ELEMENT\_NODE***)

{

Element elem = (Element) O\_KNode;

String kid = elem.getAttribute("kid");

Element elem2 = (Element) O\_KNode;

String oid = elem2.getAttribute("oid");

Node node1 = elem.getElementsByTagName("kiaddatum").item(0);

String kiaddatum = node1.getTextContent();

Node node2 = elem.getElementsByTagName("visszdatum").item(0);

String visszdatum = node2.getTextContent();

System.***out***.println("Kid: " + kid);

System.***out***.println("Oid: " + oid);

System.***out***.println("Kiaddatum: " + kiaddatum);

System.***out***.println("Visszdatum: " + visszdatum);

}

}

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

Node olvasokNode = olvasokList.item(i);

System.***out***.println("\nJelenlegi elem: " + olvasokNode.getNodeName());

if (olvasokNode.getNodeType() == Node.***ELEMENT\_NODE***)

{

Element elem = (Element) olvasokNode;

String oid = elem.getAttribute("oid");

Node node1 = elem.getElementsByTagName("veznev").item(0);

String veznev = node1.getTextContent();

Node node2 = elem.getElementsByTagName("keresztnev").item(0);

String keresztnev = node2.getTextContent();

Node node3 = elem.getElementsByTagName("kor").item(0);

String kor = node3.getTextContent();

Node node4 = elem.getElementsByTagName("telefonszam").item(0);

String telefonszam = node4.getTextContent();

System.***out***.println("Oid: " + oid);

System.***out***.println("Vezeteknev: " + veznev);

System.***out***.println("Keresztnev: " + keresztnev);

System.***out***.println("Kor: " + kor);

System.***out***.println("Telefonszam: " + telefonszam);

}

}

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

Node alkalmazottNode = alkalmazottList.item(i);

System.***out***.println("\nJelenlegi elem: " + alkalmazottNode.getNodeName());

if (alkalmazottNode.getNodeType() == Node.***ELEMENT\_NODE***)

{

Element elem = (Element) alkalmazottNode;

String aid = elem.getAttribute("aid");

Element elem2 = (Element) alkalmazottNode;

String loginid = elem2.getAttribute("loginid");

Node node1 = elem.getElementsByTagName("nev").item(0);

String nev = node1.getTextContent();

Node node2 = elem.getElementsByTagName("beosztas").item(0);

String beosztas = node2.getTextContent();

Node node3 = elem.getElementsByTagName("szulev").item(0);

String szulev = node3.getTextContent();

System.***out***.println("Aid: " + aid);

System.***out***.println("Loginid: " + loginid);

System.***out***.println("Nev: " + nev);

System.***out***.println("Beosztas: " + beosztas);

System.***out***.println("Szulev: " + szulev);

}

}

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

Node rendszerNode = rendszerList.item(i);

System.***out***.println("\nJelenlegi elem: " + rendszerNode.getNodeName());

if (rendszerNode.getNodeType() == Node.***ELEMENT\_NODE***)

{

Element elem = (Element) rendszerNode;

String loginid = elem.getAttribute("loginid");

Node node1 = elem.getElementsByTagName("nev").item(0);

String nev = node1.getTextContent();

Node node2 = elem.getElementsByTagName("jelszo").item(0);

String jelszo = node2.getTextContent();

System.***out***.println("Loginid: " + loginid);

System.***out***.println("Nev: " + nev);

System.***out***.println("Jelszo: " + jelszo);

}

}

}

//Függvények a feltöltéshez

private static Node createKiadok(Document doc2, String kiadoid, String nev, String kiadaseve)

{

Element kiadok = doc2.createElement("kiadok");

kiadok.setAttribute("kiadoid", kiadoid);

kiadok.appendChild(*createKiadokElement*(doc2,"nev", nev));

kiadok.appendChild(*createKiadokElement*(doc2,"kiadaseve", kiadaseve));

return kiadok;

}

private static Node createKiadokElement(Document doc2, String name, String value)

{

Element node = doc2.createElement(name);

node.appendChild(doc2.createTextNode(value));

return node;

}

private static Node createKonyvek(Document doc2, String kiadoid, String kid, String cim, String ar, String kategoria, String szerzo)

{

Element konyvek = doc2.createElement("konyvek");

konyvek.setAttribute("kiadoid", kiadoid);

konyvek.setAttribute("kid", kid);

konyvek.appendChild(*createKonyvekElement*(doc2,"cim", cim));

konyvek.appendChild(*createKonyvekElement*(doc2,"ar", ar));

konyvek.appendChild(*createKonyvekElement*(doc2,"kategoria", kategoria));

konyvek.appendChild(*createKonyvekElement*(doc2,"szerzo", szerzo));

return konyvek;

}

private static Node createKonyvekElement(Document doc2, String name, String value)

{

Element node = doc2.createElement(name);

node.appendChild(doc2.createTextNode(value));

return node;

}

private static Node createO\_K(Document doc2, String kid, String oid, String kiaddatum, String visszdatum)

{

Element O\_K = doc2.createElement("O\_K");

O\_K.setAttribute("kid", kid);

O\_K.setAttribute("oid", oid);

O\_K.appendChild(*createO\_KElement*(doc2,"kiaddatum", kiaddatum));

O\_K.appendChild(*createO\_KElement*(doc2,"kiadaseve", visszdatum));

return O\_K;

}

private static Node createO\_KElement(Document doc2, String name, String value)

{

Element node = doc2.createElement(name);

node.appendChild(doc2.createTextNode(value));

return node;

}

private static Node createOlvasok(Document doc2, String oid, String veznev, String keresztnev, String kor, String telefonszam)

{

Element olvasok = doc2.createElement("olvasok");

olvasok.setAttribute("oid", oid);

olvasok.appendChild(*createOlvasokElement*(doc2,"veznev", veznev));

olvasok.appendChild(*createOlvasokElement*(doc2,"keresztnev", keresztnev));

olvasok.appendChild(*createOlvasokElement*(doc2,"kor", kor));

olvasok.appendChild(*createOlvasokElement*(doc2,"telefonszam", telefonszam));

return olvasok;

}

private static Node createOlvasokElement(Document doc2, String name, String value)

{

Element node = doc2.createElement(name);

node.appendChild(doc2.createTextNode(value));

return node;

}

private static Node createAlkalmazott(Document doc2, String aid, String loginid, String nev, String beosztas, String szulev)

{

Element alkalmazott = doc2.createElement("alkalmazott");

alkalmazott.setAttribute("aid", aid);

alkalmazott.setAttribute("loginid", loginid);

alkalmazott.appendChild(*createAlkalmazottElement*(doc2,"nev", nev));

alkalmazott.appendChild(*createAlkalmazottElement*(doc2,"beosztas", beosztas));

alkalmazott.appendChild(*createAlkalmazottElement*(doc2,"szulev", szulev));

return alkalmazott;

}

private static Node createAlkalmazottElement(Document doc2, String name, String value)

{

Element node = doc2.createElement(name);

node.appendChild(doc2.createTextNode(value));

return node;

}

private static Node createRendszer(Document doc2, String loginid, String nev, String jelszo)

{

Element rendszer = doc2.createElement("rendszer");

rendszer.setAttribute("loginid", loginid);

rendszer.appendChild(*createRendszerElement*(doc2,"nev", nev));

rendszer.appendChild(*createRendszerElement*(doc2,"jelszo", jelszo));

return rendszer;

}

private static Node createRendszerElement(Document doc2, String name, String value)

{

Element node = doc2.createElement(name);

node.appendChild(doc2.createTextNode(value));

return node;

}

}

**Adatbázis módosítása DOM segítségével:**

Először a megfelelő importok használata, fájlok megnyitása, létrehozása, majd a builderek felépítése. Dokumentum, és a gyökérelem kezelése. Ezt követően a gyermekelemek hozzáadása a gyökérelemhez, majd minden módosításhoz 1-1 ciklus létrehozása, amiben a módosítás történik, majd a megfelelő elem módosítása után a változás formázott kiíratása a console-ra.

//File megnyitása

File inputFile = new File("XMLASNQPH.xml");

//Builderek létrehozása

DocumentBuilderFactory docfactory = DocumentBuilderFactory.newInstance();

DocumentBuilder docBuilder = docfactory.newDocumentBuilder();

//Dokumentum beállítása

Document doc = docBuilder.parse(inputFile);

//Elemek megkeresése

Node konyvek = doc.getElementsByTagName("konyvek").item(0);

Node kiadok = doc.getElementsByTagName("kiadok").item(0);

Node olvasok = doc.getElementsByTagName("olvasok").item(0);

Node alkalmazott = doc.getElementsByTagName("alkalmazott").item(0);

Node rendszer = doc.getElementsByTagName("rendszer").item(0);

NodeList konyvekList = konyvek.getChildNodes();

NodeList kiadokList = kiadok.getChildNodes();

NodeList olvasokList = olvasok.getChildNodes();

NodeList alkalmazottList = alkalmazott.getChildNodes();

NodeList rendszerList = rendszer.getChildNodes();

//Konyvek elemenen torteno modosítas

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

{

Node node = konyvekList.item(i);

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

{

Element element = (Element) node;

if("cim".equals(element.getNodeName()))

{

if("Egy konyv".equals(element.getTextContent()))

{

element.setTextContent("Semmivalami");

}

}

}

}

//Kiadok elemen torteno valtozas

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

{

Node node = kiadokList.item(i);

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

{

Element element = (Element) node;

if("nev".equals(element.getNodeName()))

{

if("Legjobb kiado".equals(element.getTextContent()))

{

element.setTextContent("Nem jo kiado");

}

}

}

}

//Olvasok elemen torteno valtozas

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

{

Node node = olvasokList.item(i);

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

{

Element element = (Element) node;

if("veznev".equals(element.getNodeName()))

{

if("Balog".equals(element.getTextContent()))

{

element.setTextContent("Huszar");

}

}

}

}

//Alkalmazott elemen torteno valtozas

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

{

Node node = alkalmazottList.item(i);

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

{

Element element = (Element) node;

if("nev".equals(element.getNodeName()))

{

if("Kiss Eszter".equals(element.getTextContent()))

{

element.setTextContent("Kiss Gabor");

}

}

}

}

//Rendszer elemen torteno valtozas

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

{

Node node = rendszerList.item(i);

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

{

Element element = (Element) node;

if("jelszo".equals(element.getNodeName()))

{

if("jelszo1".equals(element.getTextContent()))

{

element.setTextContent("valtozottjelszo");

}

}

}

}

//Valtoztatasok kiirasa

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transf = transformerFactory.newTransformer();

DOMSource source = new DOMSource(doc);

System.out.println("Modositott fajl: ");

StreamResult consoleResult = new StreamResult(System.out);

transf.transform(source,consoleResult);

**Adatbázis lekérdezése DOM segítségével:**

Először a megfelelő importok használata, fájlok megnyitása, létrehozása, majd a builderek felépítése. Dokumentum, és a gyökérelem kezelése. Ezt követően a gyermekelemek hozzáadása a gyökérelemhez, majd minden lekérdezéshez 1-1 ciklus létrehozása, majd a megfelelő elemek lekérdezése után formázott kiíratása a console-ra.

try {

// XML file kiválasztása

File inputFile = new File("XMLASNQPH.xml");

// Dokumentum builderek létehozása

DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

// Dokumentumok beállítása

Document doc = dBuilder.parse(inputFile);

doc.getDocumentElement().normalize();

// Gyökérelem meghatározása, konzolon feltüntetése

System.out.println("---------------------");

System.out.print("Gyokerelem: ");

System.out.println(doc.getDocumentElement().getNodeName());

// 1. Lekérdezés fő részének meghatározása

NodeList nList1 = doc.getElementsByTagName("alkalmazott");

System.out.println("---------------------");

System.out.println("\n------------------------------------");

System.out.println("1. Alkalmazottak adatainak lekerdezese:");

System.out.println("------------------------------------");

// Iterálás az elemeken és adott elemek kiíratása a konzolra

for (int temp = 0; temp < nList1.getLength(); temp++) {

Node nNode = nList1.item(temp);

System.out.println("\nAktualis elem :");

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

Element element = (Element) nNode;

System.out.print("Alkalmazott ID: ");

System.out.println(element.getAttribute("aid"));

NodeList alkalmazottNevList = element.getElementsByTagName("nev");

NodeList alkalmazottBeoList = element.getElementsByTagName("beosztas");

NodeList alkalmazottSzulevList = element.getElementsByTagName("szulev");

for (int count = 0; count < alkalmazottNevList.getLength(); count++) {

Node node1 = alkalmazottNevList.item(count);

Node node2 = alkalmazottBeoList.item(count);

Node node3 = alkalmazottSzulevList.item(count);

Element alkalmazottNev = (Element) node1;

System.out.print("Alkalmazott neve: ");

System.out.println(alkalmazottNev.getTextContent());

Element alkalmazottBeo = (Element) node2;

System.out.print("Alkalmazott beosztasa: ");

System.out.println(alkalmazottBeo.getTextContent());

Element alkalmazottSzulev = (Element) node3;

System.out.print("Alkalmazott szuletesi eve: ");

System.out.println(alkalmazottSzulev.getTextContent());

}

}

}

// 2. Lekérdezés fő részének meghatározása

NodeList nList2 = doc.getElementsByTagName("rendszer");

System.out.println("\n---------------------------------------------------------------");

System.out.println("2. Rendszerhez szükséges nevek es jelszavak lekerdezese:");

System.out.println("---------------------------------------------------------------");

// Iterálás az elemeken és adott elemek kiíratása a konzolra

for (int temp = 0; temp < nList2.getLength(); temp++) {

Node nNode = nList2.item(temp);

System.out.println("\nAktualis elem :");

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

Element element = (Element) nNode;

System.out.print("Login ID: ");

System.out.println(element.getAttribute("loginid"));

NodeList rendszerNevList = element.getElementsByTagName("nev");

NodeList rendszerJelszoList = element.getElementsByTagName("jelszo");

for (int count = 0; count < rendszerNevList.getLength(); count++) {

Node node1 = rendszerNevList.item(count);

Node node2 = rendszerJelszoList.item(count);

Element rendszerNev = (Element) node1;

System.out.print("Rendszerhasznalo neve: ");

System.out.println(rendszerNev.getTextContent());

Element rendszerJelszo = (Element) node2;

System.out.print("Rendszerhasznalo jelszava: ");

System.out.println(rendszerJelszo.getTextContent());

}

}

}

// 3. Lekérdezés fő részének meghatározása

NodeList nList3 = doc.getElementsByTagName("konyvek");

System.out.println("\n----------------------------");

System.out.println("3. Konyvek osszegei:");

System.out.println("----------------------------");

// Iterálás az elemeken és adott elemek kiíratása a konzolra

for (int temp = 0; temp < nList3.getLength(); temp++) {

Node nNode = nList3.item(temp);

System.out.println("\nAktualis elem :");

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

Element element = (Element) nNode;

System.out.print("Konyvek ID: ");

System.out.println(element.getAttribute("kid"));

NodeList konyvekNevList = element.getElementsByTagName("cim");

NodeList konyvekArList = element.getElementsByTagName("ar");

for (int count = 0; count < konyvekNevList.getLength(); count++) {

Node node1 = konyvekNevList.item(count);

Node node2 = konyvekArList.item(count);

Element konyvekNev = (Element) node1;

Element konyvekAr = (Element) node2;

System.out.print("Konyvek neve: " + konyvekNev.getTextContent());

System.out.println("\nKonyvek ara: " + konyvekAr.getTextContent());

}

}

}

// 4. Lekérdezés fő részének meghatározása

NodeList nList4 = doc.getElementsByTagName("konyvek");

System.out.println("\n---------------------------------------------------");

System.out.println("4. Konyvek azonositoja, ahol misztikus a kategoria:");

System.out.println("---------------------------------------------------");

// Iterálás az elemeken és adott elemek kiíratása a konzolra

for (int temp = 0; temp < nList4.getLength(); temp++) {

Node nNode = nList4.item(temp);

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

Element element = (Element) nNode;

NodeList konyvekList = element.getElementsByTagName("kategoria");

for (int count = 0; count < konyvekList.getLength(); count++) {

Node node = konyvekList.item(count);

Element kategoria = (Element) node;

if (kategoria.getTextContent().equalsIgnoreCase("Misztikus")) {

System.out.println("\nID-k: " + element.getAttribute("kid"));

}

}

}

}

// 5. Lekérdezés fő részének meghatározása

NodeList nList5 = doc.getElementsByTagName("olvasok");

System.out.println("\n---------------------------------------");

System.out.println("5. Olvasok, akik 18 evnel idosebbek:");

System.out.println("---------------------------------------");

// Iterálás az elemeken és adott elemek kiíratása a konzolra

for (int temp = 0; temp < nList5.getLength(); temp++) {

Node nNode = nList5.item(temp);

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

Element element = (Element) nNode;

NodeList olvasokKeresztnevList = element.getElementsByTagName("keresztnev");

NodeList olvasokVezeteknevList = element.getElementsByTagName("veznev");

NodeList olvasokKorList = element.getElementsByTagName("kor");

for (int count = 0; count < olvasokKeresztnevList.getLength(); count++) {

Node node1 = olvasokKeresztnevList.item(count);

Node node2 = olvasokVezeteknevList.item(count);

Node node3 = olvasokKorList.item(count);

Element olvasokKeresztnev = (Element) node1;

Element olvasokVezeteknev = (Element) node2;

Element olvasokKor = (Element) node3;

if (Integer.parseInt(olvasokKor.getTextContent()) > 18) {

System.out.println("\nOlvaso Vezetekneve: " + olvasokVezeteknev.getTextContent());

System.out.println("Olvaso Keresztneve: " + olvasokKeresztnev.getTextContent());

}

}

}

}

} catch (IOException e) {

e.printStackTrace();

} catch (ParserConfigurationException e) {

e.printStackTrace();

} catch (SAXException e) {

e.printStackTrace();

}