

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: Kelemen Bence

Neptunkód: B8S0JQ

A feladat leírása:

A feladatban egy könyvtár adatbázisát készítettem el, az alábbi adatokkal:

Az olvasó egyedben megtalálható az Olvasójegy, mely Primary Key, az olvasó neve(ez nem lehet NULL érték) és címe összetett adattípusként, mely tartalmazza az irányítószámot, a várost, az utcát és a házszámot.

1:N kapcsolatban áll a Könyvtárral, mivel a könyvtárnak sok olvasója lehet.

A kiadók egyedben találhatóak a kiadók nevei és címei összetett adattípusként, mely tartalmazza az irányítószámot, a várost, az utcát és a házszámot.

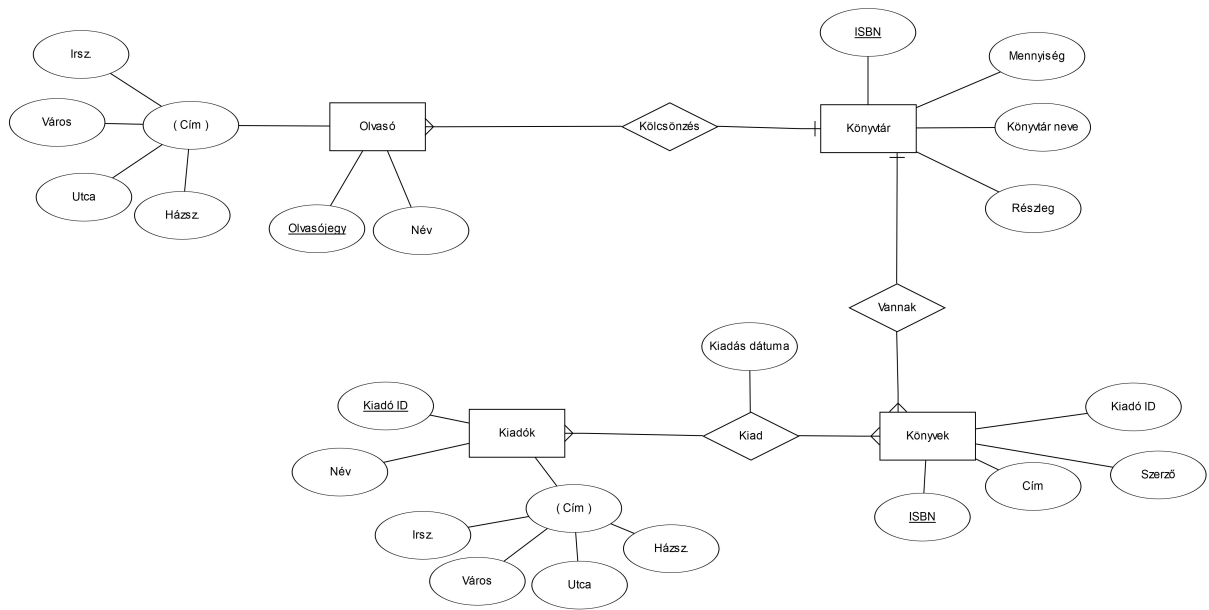
Ez N:N kapcsolatban áll a könyvekkel, mivel egy kiadóhoz több könyv is tartozik és egy könyvnek is lehet több kiadója.

A könyvtár egyed tartalmazza hogy melyik könyvből hány darab van a könyvtárban, a könyvtár nevét, a részleget ahol a könyv található és az ISBN-t..

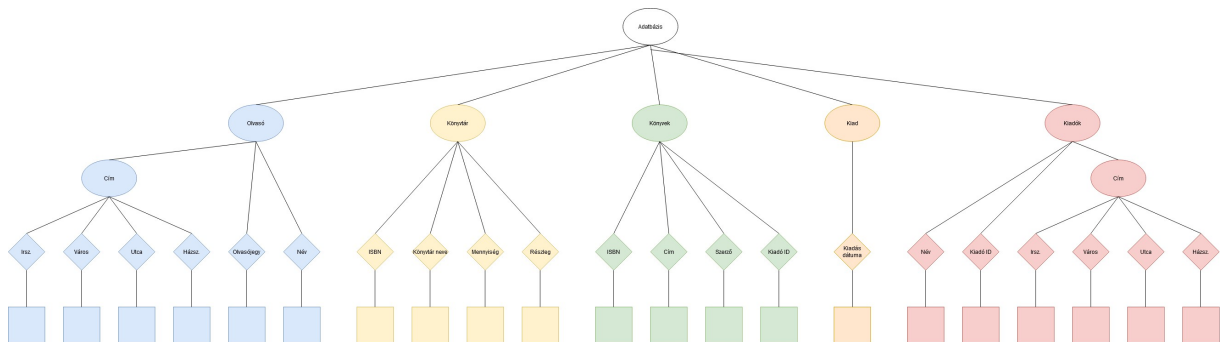
Ez 1:N kapcsolatban áll a könyvekkel, mivel az összes könyv ebben a könyvtárban van és ebben a könyvtárban sok könyv van.

A könyvek egyed tartalmazza a könyvek adatait: Az ISBN azonosítójukat, a címüket, a kiadó azonosítóját és a szerzőt.

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"?>
```

```
<Adatbazis xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaB8S0JQ.xsd">
```

```
  <Olvaso id="1">
```

```
    <Olvasojegy>1</Olvasojegy>
```

```
    <Nev>Tóth Ferenc</Nev>
```

```
    <Cim>
```

```
      <Irsz>3501</Irsz>
```

```
      <Varos>Miskolc</Varos>
```

```
      <Utca>Kossuth utca</Utca>
```

```
      <Hatsz>12</Hatsz>
```

```
    </Cim>
```

```
  </Olvaso>
```

```
  <Olvaso id="2">
```

```
    <Olvasojegy>2</Olvasojegy>
```

```
    <Nev>Molnár Éva</Nev>
```

```
    <Cim>
```

```
      <Irsz>3502</Irsz>
```

```
      <Varos>Miskolc</Varos>
```

```
      <Utca>Széchenyi utca</Utca>
```

```
      <Hatsz>56</Hatsz>
```

```
    </Cim>
```

```
  </Olvaso>
```

```
  <Konyvtar id="1">
```

```
    <ISBN>12345678</ISBN>
```

```
    <Mennyiseg>3</Mennyiseg>
```

```
    <Konyvtar_neve>Móra Ferenc</Konyvtar_neve>
```

```
    <Reszleg>Horror</Reszleg>
```

```
  </Konyvtar>
```

```
  <Konyvtar id="2">
```

```
    <ISBN>87654321</ISBN>
```

```
    <Mennyiseg>4</Mennyiseg>
```

```
<Konyvtar_neve>Egyetemi könyvtár</Konyvtar_neve>
<Reszleg>Szakirodalom</Reszleg>
</Konyvtar>
<Konyvek id="1">
  <ISBN>12345678</ISBN>
  <KCim>A ragyogás</KCim>
  <Szerzo>Stephen King</Szerzo>
  <KiadoID>3456</KiadoID>
</Konyvek>
<Konyvek id="2">
  <ISBN>87654321</ISBN>
  <KCim>Az XML kézikönyv</KCim>
  <Szerzo>N. Bradley</Szerzo>
  <KiadoID>9786</KiadoID>
</Konyvek>
<Kiad id="1">
  <Kiadas_datuma>1991</Kiadas_datuma>
</Kiad>
<Kiad id="2">
  <Kiadas_datuma>2000</Kiadas_datuma>
</Kiad>
<Kiadok id="1">
  <Nev>Árkádia</Nev>
  <KiadoID>3456</KiadoID>
  <KCim>
    <KIrsz>1051</KIrsz>
    <KVaros>Budapest</KVaros>
    <KUtea>Rákóczi út</KUtea>
    <KHasz>123</KHasz>
  </KCim>
</Kiadok>
<Kiadok id="2">
```

<Nev>SZAK</Nev>

<KiadoID>9786</KiadoID>

<KCim>

<KIrsz>3515</KIrsz>

<KVaros>Miskolc</KVaros>

<KUtca>Egyetemváros</KUtca>

<KHazsz>0000</KHazsz>

</KCim>

</Kiadok>

</Adatbazis>

1d Az XML dokumentum alapján XMLSchema készítése:

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

```
<xs:element name="Adatbazis">
```

```
<xs:complexType>
```

```
<xs:choice maxOccurs="unbounded" minOccurs="0">
```

```
<xs:element name="Olvaso" maxOccurs="unbounded" minOccurs="0">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element type="xs:int" name="Olvasojegy"/>
```

```
<xs:element type="xs:string" name="Nev"/>
```

```
<xs:element name="Cim">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element type="xs:short" name="Irsz"/>
```

```
<xs:element type="xs:string" name="Varos"/>
```

```
<xs:element type="xs:string" name="Utca"/>
```

```
<xs:element type="xs:int" name="Hazsz"/>
```

```
</xs:sequence>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
</xs:sequence>
```

```
<xs:attribute type="xs:byte" name="id" use="optional"/>
```

```
</xs:complexType>
```

```
</xs:element>
```

```
<xs:element name="Konyvtar" maxOccurs="unbounded" minOccurs="0">
```

```
<xs:complexType>
```

```
<xs:sequence>
```

```
<xs:element type="xs:int" name="ISBN"/>
```

```
<xs:element type="xs:byte" name="Mennyiseg"/>
```

```
<xs:element type="xs:string" name="Konyvtar_neve"/>
```

```
<xs:element type="xs:string" name="Reszleg"/>
```

```
</xs:sequence>
```

```

        <xs:attribute type="xs:byte" name="id" use="optional"/>
    </xs:complexType>
</xs:element>

<xs:element name="Konyvek" maxOccurs="unbounded" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:int" name="ISBN"/>
            <xs:element type="xs:string" name="KCim"/>
            <xs:element type="xs:string" name="Szerzo"/>
            <xs:element type="xs:short" name="KiadoID"/>
        </xs:sequence>
        <xs:attribute type="xs:byte" name="id" use="optional"/>
    </xs:complexType>
</xs:element>

<xs:element name="Kiad" maxOccurs="unbounded" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:short" name="Kiadas_datuma"/>
        </xs:sequence>
        <xs:attribute type="xs:byte" name="id" use="optional"/>
    </xs:complexType>
</xs:element>

<xs:element name="Kiadok">
    <xs:complexType>
        <xs:sequence>
            <xs:element type="xs:string" name="Nev"/>
            <xs:element type="xs:short" name="KiadoID"/>
            <xs:element name="KCim">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element type="xs:short" name="KIrsz"/>
                        <xs:element type="xs:string" name="KVaros"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```
<xs:element type="xs:string" name="KUtca"/>
<xs:element type="xs:int" name="KHazsz"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute type="xs:byte" name="id" use="optional"/>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```


2a feladat

A feladat egy DOM program készítése az XML dokumentum adatainak adminisztrálása alapján:

Adatolvasás:

```
package domB8S0JQ;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.File;
import java.io.IOException;

public class DOMParseB8S0JQ {

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

        File inputFile = new File("D:\\KB_B8S0JQ\\KB_B8S0JQ\\XMLB8S0JQ.xml");//xml fájl
        megadása

        Document doc = DocumentBuilderFactory
            .newInstance()
            .newDocumentBuilder()
            .parse(inputFile);

        Element gyoker = doc.getDocumentElement();
        gyoker.normalize();// a normalize olvasható formátumot hoz létre

        kiir(gyoker, "");
    }
}
```

```

public static void kiir(Node gyoker, String bekezdés) {
    String nodename = gyoker.getNodeName();//listázza az egyedeket
    if (!nodename.contains("text")) {
        System.out.println(bekezdés + nodename);//adatbázis gyökerének kiírása
    }
    bekezdés += " ";

    NodeList children = gyoker.getChildNodes();//gyerekek feldolgozása
    for (int i = 0; i < children.getLength(); i++) {
        Node child = children.item(i);
        boolean veg = child.getTextContent().contains("\n");//van-e benne új sor

        if (veg) {
            kiir(child, bekezdés);//rekurzív módon meghívja a függvényt
        } else {
            System.out.print(bekezdés + child.getNodeName());
            System.out.println(": " + child.getTextContent());
        }
    }
}
}

```

2b Feladat:

Adat módosítás:

```
package domB8S0JQ;

import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import java.io.IOException;
import java.util.Scanner;

public class DOMModifyB8S0JQ {
    public static void main(String args[]) {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        try {
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document doc = builder.parse("D:\\KB_B8S0JQ\\KB_B8S0JQ\\XMLB8S0JQ.xml"); //xml fájl
            megadása

            NodeList nodeList = doc.getElementsByTagName("Adatbazis");

            for (int i = 0; i < nodeList.getLength(); i++) {
                Node nd = nodeList.item(i);
                if (nd.getNodeType() == Node.ELEMENT_NODE) {
                    Element ement = (Element) nd;
                    System.out.println("Melyik könyv mennyiséseég szeretné megváltoztatni?
ISBN:");

                    Scanner scanner = new Scanner(System.in);
                    String isbn = scanner.next();//Bekérem, hogy melyik könyv számát akarom
                    módosítani

                    System.out.println("Könyvek száma: ");
                    String db = scanner.next();// az új mennyiség megadása
                    scanner.close();
                    NodeList list = ement.getChildNodes();
                    for (int j = 0; j < list.getLength(); j++) {
```

könyvtárnál jár-e már a ciklus

a könyv amit keresek

igen, akkor módosítom a mynniséget

adatainak kiírása a módosítással együtt

name.getTextContent());

}

```
Node n = list.item(j);
if (n.getNodeName() == "Konyvtar") { // megnmézem, hogy a

    NodeList children = n.getChildNodes();
    int x = 0;
    for (int k = 0; k < children.getLength(); k++) {
        Node c = children.item(k);
        String co = c.getNodeName();
        String rs = c.getTextContent();
        if (rs.equals(isbn)) { //Elenőrzöm, hogy ez-e az

            x = 1;

        }
        if (co == "Mennyiseg" && x == 1) { // ha

            c.setTextContent(db);

        }
    }
}

if (n.getNodeType() == Node.ELEMENT_NODE) { // xml

    Element name = (Element) n;
    System.out.println(" " + name.getTagName() + ": " +

name.getTextContent());

}

}

}

} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

}
```