

# 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.

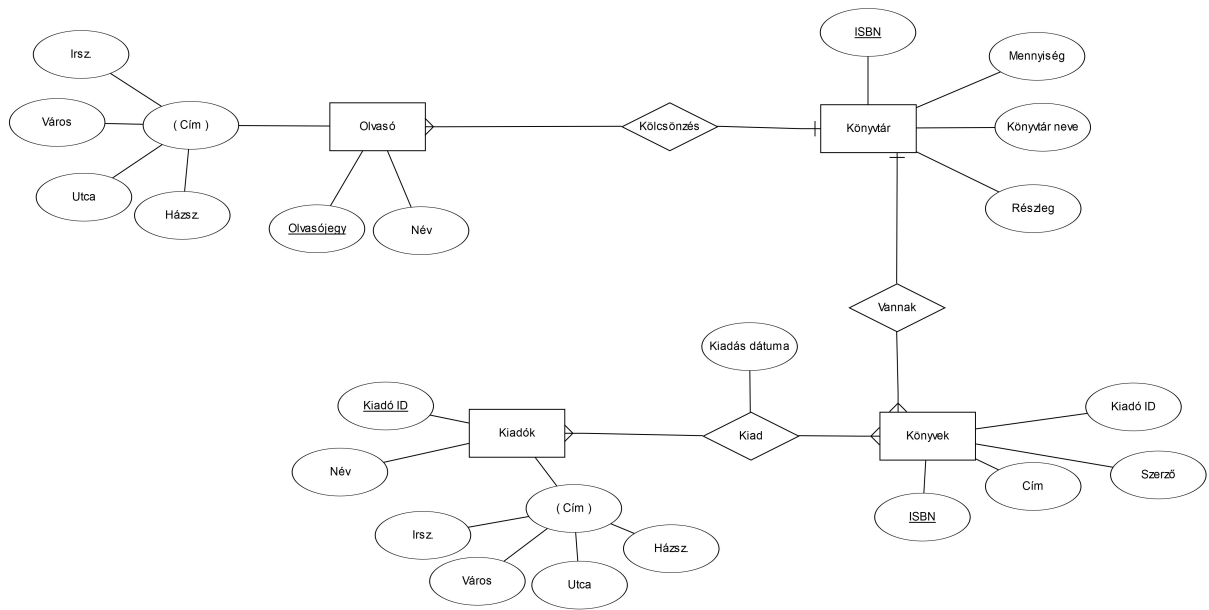
Ez N:N kapcsolatban áll a könyvekkel, mivel egy kiadóhoz több könyv is tartozik és egy könyvnek is lehet több kiadója.

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..

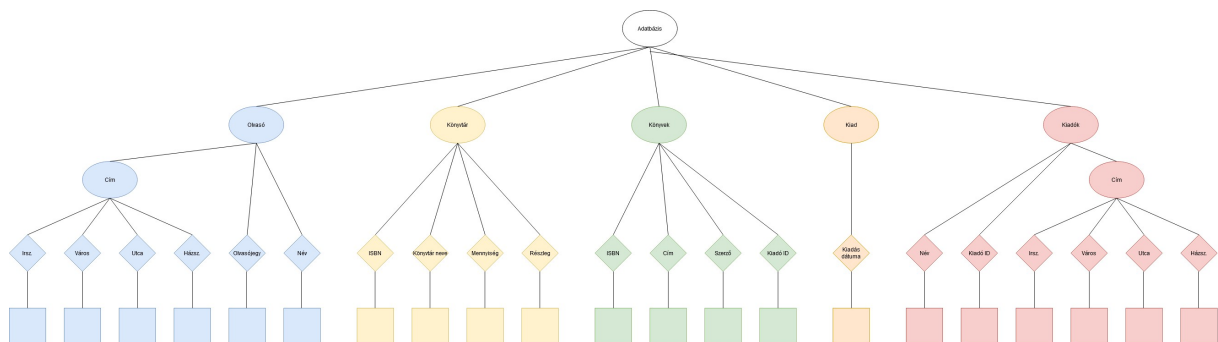
Ez 1:N kapcsolatban áll a könyvekkel, mivel az összes könyv ebben a könyvtárban van és ebben a könyvtárban sok könyv van.

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>
  <Olvaso>
    <Olvasojegy>1</Olvasojegy>
    <Nev>Tóth Ferenc</Nev>
    <Cim>
      <Irsz>3501</Irsz>
      <Varos>Miskolc</Varos>
      <Utca>Kossuth utca</Utca>
      <Hatsz>12</Hatsz>
    </Cim>
  </Olvaso>
  <Konyvtar>
    <ISBN>12345678</ISBN>
    <Mennyiseg>3</Mennyiseg>
    <Konyvtar_neve>Móra Ferenc</Konyvtar_neve>
    <Reszleg>Horror</Reszleg>
  </Konyvtar>
  <Konyvek>
    <ISBN>12345678</ISBN>
    <KCim>A ragyogás</KCim>
    <Szerzo>Stephen King</Szerzo>
    <KiadoID>3456</KiadoID>
  </Konyvek>
  <Kiad>
    <Kiadas_datuma>1991</Kiadas_datuma>
  </Kiad>
  <Kiadok>
    <Nev>Árkádia</Nev>
    <KiadoID>978</KiadoID>
    <Cim>
      <Irsz>1051</Irsz>
      <Varos>Budapest</Varos>
```

<Utca>Rákóczi út</Utca>

<Hatsz>123</Hatsz>

</Cim>

</Kiadok>

</Adatbazis>

1d Az XML dokumentum alapján XMLSchema készítése:

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

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

  <xs:element name="Adatbazis">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Olvaso">
          <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:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="Konyvtar">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:int" name="ISBN"/>
        <xs:element type="xs:int" name="Mennyiseg"/>
        <xs:element type="xs:string" name="Konyvtar_neve"/>
        <xs:element type="xs:string" name="Reszleg"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
</xs:element>
<xs:element name="Konyvek">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:int" name="ISBN"/>
      <xs:element type="xs:string" name="Cim"/>
      <xs:element type="xs:string" name="Szerzo"/>
      <xs:element type="xs:short" name="KiadoID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Kiad">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:short" name="Kiadas_datuma"/>
    </xs:sequence>
  </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="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:complexType>
</xs:sequence>
```

```
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## 2 feladat

A feladat egy DOM program készítése az XML dokumentum adatainak adminisztrálása alapján:  
package domB8S0JQ;

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import org.w3c.dom.Document;
```

```
import org.w3c.dom.Element;
```

```
import org.w3c.dom.Node;
```

```
import org.w3c.dom.NodeList;
```

```
public class DomParserB8S0JQ {
```

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

```
        try {
```

```
            DocumentBuilderFactory dbFactory =  
DocumentBuilderFactory.newInstance();
```

```
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
```

```
            Document doc = dBuilder.parse("D:\\KB_B8S0JQ\\KB_B8S0JQ\\student.xml");
```

```
            doc.getDocumentElement().normalize();
```

```
            System.out.println("Root element :" + doc.getDocumentElement().getNodeName());
```

```
            NodeList nList = doc.getElementsByTagName("student");
```

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

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

```
                Node nNode = nList.item(temp);
```

```
                System.out.println("\nCurrent Element :" + nNode.getNodeName());
```



```

if (nNode.getNodeType() == Node.ELEMENT_NODE) {
    Element eElement = (Element) nNode;
    System.out.println ("Student roll no : "
        + eElement.getAttribute("rollno"));

    System.out.println("First Name : "
        + eElement
        .getElementsByTagName("firstname")
        .item(0)
        .getTextContent());
    System.out.println("Last Name : "
        + eElement
        .getElementsByTagName("lastname")
        .item(0)
        .getTextContent());
    System.out.println("Nick Name : "
        + eElement
        .getElementsByTagName("nickname")
        .item(0)
        .getTextContent());
    System.out.println("Age : "
        + eElement
        .getElementsByTagName("age")
        .item(0)
        .getTextContent());
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```