

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

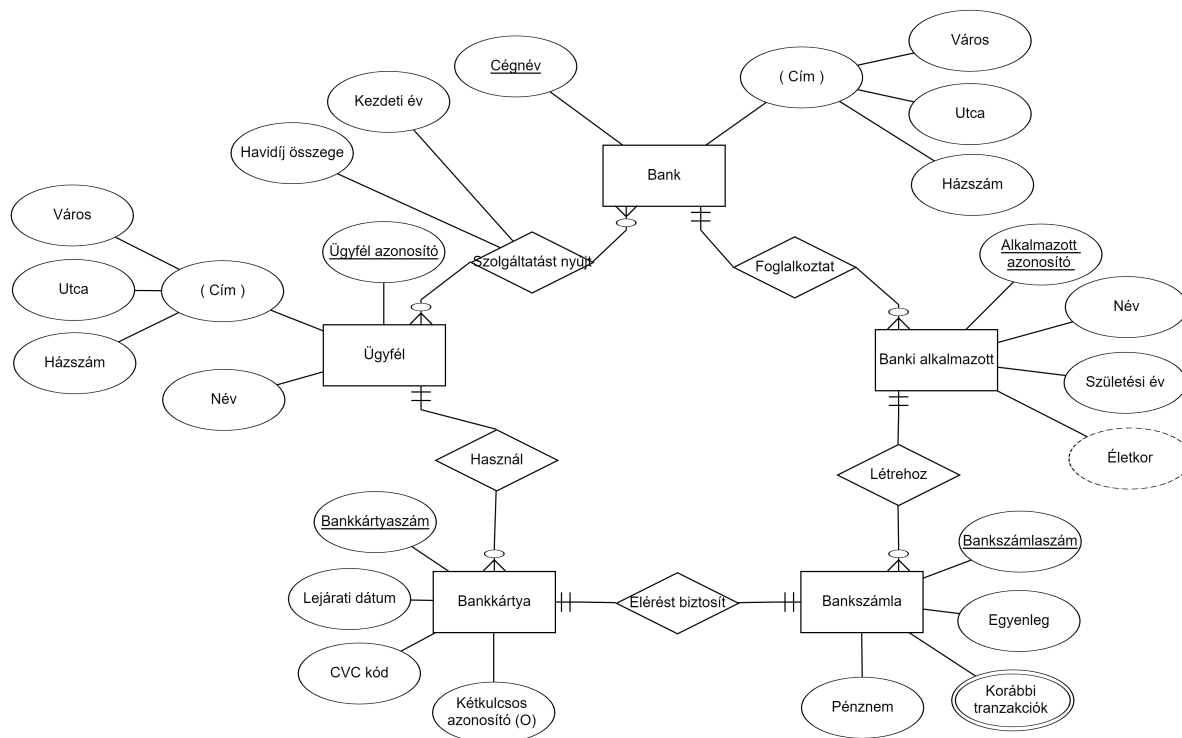
Készítette: Zán Martin Márk
Neptunkód: N16IAF

A feladat leírása: A modellem a bankok működését jellemző adatokat ír le.

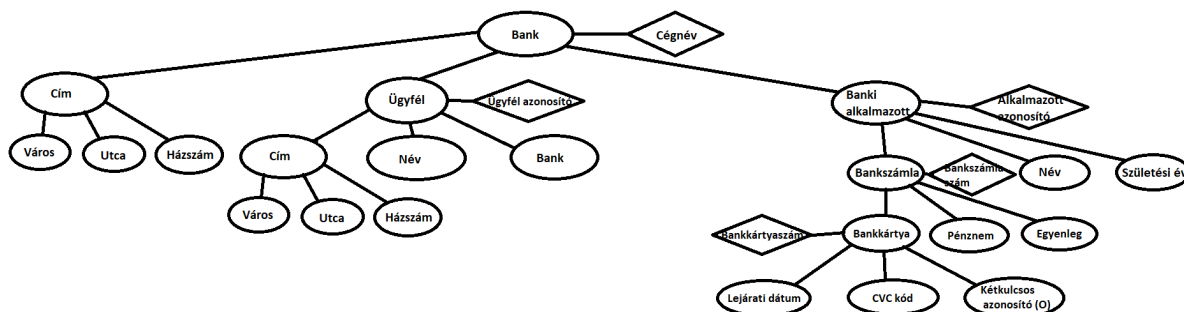
- **Bank:** Maga a bank, a legközpontibb elem. Ez a gyökérellem, neki vannak alkalmazottjai, számlái, ügyfelei. Azonosítja őt a neve, és ezentúl van címe amely városból, utcából és házszámból áll.
- **Ügyfél:** Szolgáltatást kap a banktól. Őt egy ügyfélazonosító azonosítja, ezen-túl van neve és ugyanolyan összetett címe, mint a banknak.
- **Banki alkalmazott:** A bankban dolgozó emberek összessége. Egy alkalmazott azonosító azonosítja, van neve, születési éve, és számíthatunk életkort neki.
- **Bankszámla:** A banki alkalmazottak hozhatják létre. Bankszámlaszám azo-nosítja. Egyenlege, pénzneme és korábbi tranzakciói írják le. Korábbi tranz-akciói többértékű, azaz több is lehet belőle.
- **Bankkártya:** Ez biztosítja az elérést a bankszámlához. Bankkártyaszám azo-nosítja. Lejárat dátuma, CVC kódja írja le, és opcionálisan tartozhat hozzá kétkulcsos azonosítás használatául szolgáló azonosító kód.
- **Szolgáltatást nyújt:** A szolgáltatást a bank és az ügyfél között adja meg, melynek van kezdeti éve és havidíjának egy összege.

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"?>
```

```
<bank cegnev="OTP">
```

```
<cim>
```

```
<varos>Miskolc</varos>
```

```
<utca>Uitz Béla</utca>
```

```
<hazszam>6</hazszam>
```

```
</cim>
```

```
<ugyfel ugyfel_azonosito="1">
```

```
<cim>
```

```
<varos>Miskolc</varos>
```

```
<utca>József Attila</utca>
```

```
<hazszam>215</hazszam>
```

```
</cim>
```

```
<nev>Kulcsár Ádám</nev>
```

```
<bank>OTP</bank>
```

```
</ugyfel>
```

```
<ugyfel ugyfel_azonosito="2">
```

```
<cim>
```

```
<varos>Miskolc</varos>
```

```
<utca>Vörösmarty Mihály</utca>
```

```
<hazszam>454</hazszam>
```

```
</cim>
```

```
<nev>Szabó Henrietta</nev>
```

<bank>OTP</bank>
<bank>ERSTE</bank>
</ugyfel>

<banki_alkalmazott alkalmazott_azonosito="1">
<nev>Visegrádi Iván</nev>
<szuletesi_ev>1988</szuletesi_ev>

<bankszamla bankszamla_szam="1343598542375482">
<penznem>EUR</penznem>
<egyenleg>540</egyenleg>

<bankkartya bankkartya_szam="5012443244328394">
<lejarati_datum>20220830</lejarati_datum>
<cvckod>632</cvckod>
<ketkulcsos_azonosito>7892</ketkulcsos_azonosito>
</bankkartya>

</bankszamla>

</banki_alkalmazott>

<banki_alkalmazott alkalmazott_azonosito="2">
<nev>Galambos Lajos</nev>
<szuletesi_ev>1965</szuletesi_ev>

<bankszamla bankszamla_szam="1343698541375482">
<penznem>KRW</penznem>
<egyenleg>2200000</egyenleg>

<bankkartya bankkartya_szam="1912443244328394">
<lejarati_datum>20241121</lejarati_datum>
<cvckod>112</cvckod>
<ketkulcsos_azonosito>1234</ketkulcsos_azonosito>
</bankkartya>

</bankszamla>

<bankszamla bankszamla_szam="1343498344375482">
<penznem>HUF</penznem>
<egyenleg>-54</egyenleg>

<bankkartya bankkartya_szam="1612475254308394">
<lejarati_datum>20191121</lejarati_datum>
<cvckod>748</cvckod>

</bankkartya>

</bankszamla>

</banki_alkalmazott>

</bank>

1d) Az XML dokumentum alapján XMLSchema készítése (saját típusok):

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd"
  elementFormDefault="qualified"
  xmlns="http://tempuri.org/XMLSchema.xsd"
  xmlns:mstns="http://tempuri.org/XMLSchema.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  >

  <xs:element name="bank" type="bank_tipus"/>

  <xs:simpleType name="penznem_tipus">
    <xs:restriction base="xs:string">
      <xs:enumeration value="HUF"/>
      <xs:enumeration value="EUR"/>
      <xs:enumeration value="GBP"/>
      <xs:enumeration value="CHF"/>
      <xs:enumeration value="USD"/>
      <xs:enumeration value="SKW"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="cvc_kod_tipus">
    <xs:restriction base="xs:int">
      <xs:minInclusive value="100"/>
      <xs:maxInclusive value="999"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="bank_tipus" id="cegnev">
    <xs:sequence>
      <xs:element name="cim" type="cim_tipus"/>
      <xs:element name="ugyfel" type="ugyfel_tipus" maxOccurs="unbounded"/>
      <xs:element name="banki_alkalmazott"
        type="banki_alkalmazott_tipus" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

```

<xs:complexType name="cim_tipus">
  <xs:sequence>
    <xs:element name="varos" type="xs:string"/>
    <xs:element name="utca" type="xs:string"/>
    <xs:element name="hazszam" type="xs:unsignedInt"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="ugyfel_tipus" id="ugyfel_azonosito">
  <xs:sequence>
    <xs:element name="cim" type="cim_tipus"/>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="bank" type="xs:string" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="banki_alkalmazott_tipus" id="alkalmazott_azonosito">
  <xs:sequence>
    <xs:element name="nev" type="xs:string"/>
    <xs:element name="szuletesi_ev" type="xs:gYear"/>
    <xs:element name="bankszamla" type="bankszamla_tipus"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="bankszamla_tipus" id="bankszamla_szam">
  <xs:sequence>
    <xs:element name="egyenleg" type="xs:int"/>
    <xs:element name="penznem" type="penznem_tipus"/>
    <xs:element name="bankkartya" type="bankkartya_tipus" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="bankkartya_tipus" id="bankkartya_szam">
  <xs:sequence>
    <xs:element name="lejarati_datum" type="xs:date"/>
    <xs:element name="cvc_kod" type="cvc_kod_tipus"/>
    <xs:element name="ketkulcsos_azonosito" type="xs:int" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

</xs:schema>

```

2. feladat

2a) adatolvasás - DOMReadN16IAF.java

```
package hu.domparse.N16IAF;
```

```

import java.io.File;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import org.w3c.dom.*;
import org.xml.sax.SAXException;

public class DomReadN16IAF {

    public static void main(String[] args) throws
        ParserConfigurationException, SAXException, IOException {
        // TODO Auto-generated method stub

        // File elérésének biztosítása, gyár létrehozása,
        // DocumentBuilder létrehozása és maga a dokumentum létrehozása.
        File xmlFile = new File("D:\\Programming\\Github\\N16IAF_XMLGyak\\XMLTaskN16IAF\\XMLN16IAF.xml");
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);

        // A dokumentum normalizálása.
        doc.getDocumentElement().normalize();

        // Kiírjuk a gyökérelemet.
        System.out.println("Gyökérelem: " + doc.getDocumentElement().getNodeName());

        // Lekérdezzük az "ugyfel"-eket, először egy listába téve azokat.
        NodeList nList = doc.getElementsByTagName("ugyfel");

        // Majd ezen a listán végigmegyünk.
        for (int i = 0; i < nList.getLength(); i++) {
            Node nNode = nList.item(i);

            // Kiíratjuk a nevüket a node-oknak.
            System.out.println("\nJelenlegi elem: " + nNode.getNodeName());

            // Ha ez egy elem akkor kiírjuk róluk az adatokat.
            if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) nNode;

                // Attribútum (jellemző) lekérdezése.

```

```

String uid = elem.getAttribute("ugyfel_azonosito");

// Elemek lekérdezése.
Node node1 = elem.getElementsByTagName("nev").item(0);
String name = node1.getTextContent();

// Itt for ciklus szükséges, mert több bank is tartozhat egy ügyfélhez.
String bank = "";
Node node2;
for ( int j = 0; j < elem.getElementsByTagName("bank").getLength() ; j++ ) {
node2 = elem.getElementsByTagName("bank").item(j);
bank = bank + " " + node2.getTextContent();
}

Node node3 = elem.getElementsByTagName("cim").item(0);
String adress = node3.getTextContent();

// Majd a legvégén kiíratjuk azokat.
System.out.printf("Ügyfél azonosító: %s%n", uid);
System.out.printf("Név: %s%n", name);
System.out.printf("Bank: %s%n", bank);
System.out.printf("Cím: %s%n", adress);
}
}
}

}

```

2b) adatlekérdezés - DOMQueryN16IAF.java

2b) adatmódosítás - DOMModifyN16IAF.java