

# JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Készítette: **Erdei Botond**

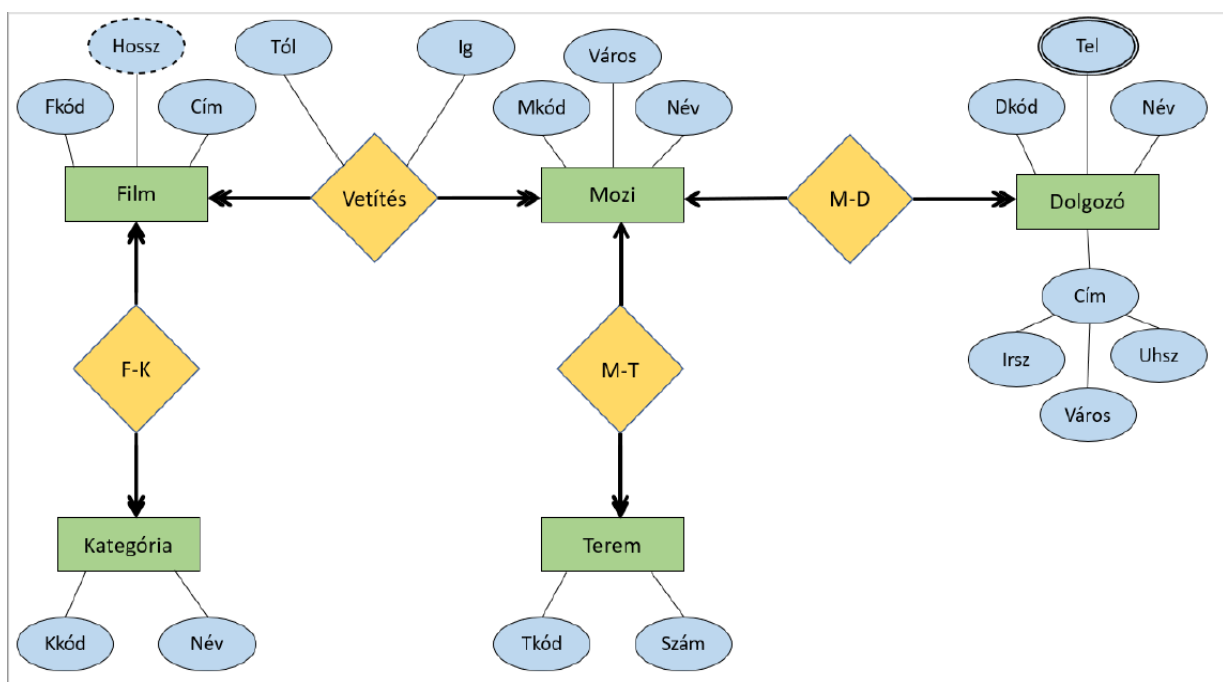
Neptunkód: **CF94SO**

## 1. Feladat: A feladat leírása:

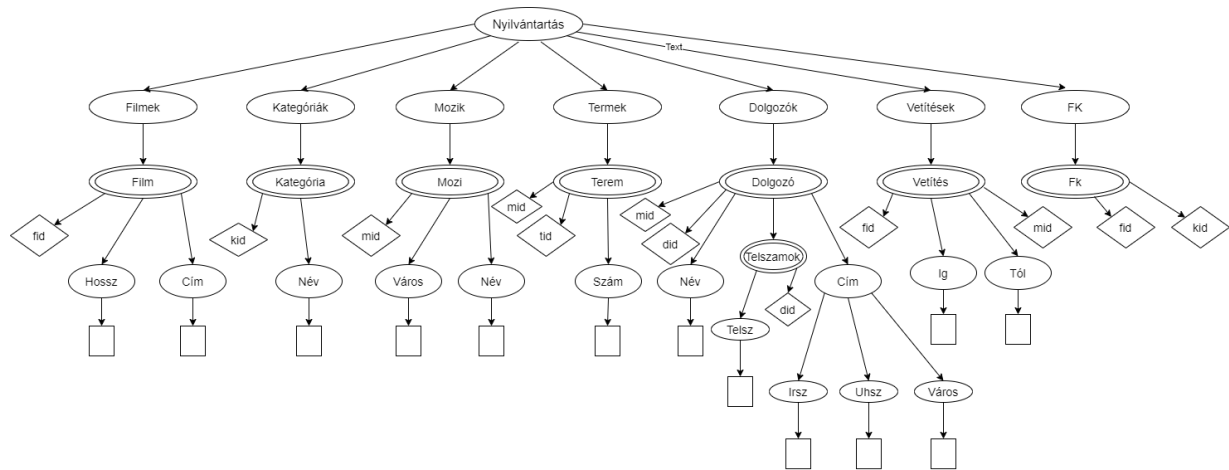
A feladatban egy öt egyedű adatbázist hozok létre mozi témakörben, az alábbiak szerint:

- A Film egyed három tulajdonsággal rendelkezik, Fkód mely az ID-je, Cím, ami a film címe és a Hossz.
- A Kategória egyednek két tulajdonsága van, Kkód az ID és Név, ami a kategóriák neve. Egy filmhez több kategória és egy kategóriába több film is beletartozhat, ezért az F-K egy N:M kapcsolat.
- A Mozi három tulajdonságú egyed Mkód, Város, Név. Mivel a Moziban több filmet is vetítenek és egy Filmet több Moziban is vetíthetnek, a Vetítés szintén egy N:M kapcsolat lesz, valamint két tulajdonsággal rendelkezik. Ezek a Tól és az Ig. A film vetítésének kezdetét és végét adják meg.
- A teremnek két tulajdonsága van, Tkód és Szám. A Tkód az ID-je a szám pedig a terem sorszáma. A Mozi és a Terem között M-T, egy 1:N kapcsolat van.
- A Dolgozó egyednek két egyszerű tulajdonsága van, a Tel egy többértékű tulajdonság a telefonszámok tárolására. A Cím pedig összetett tulajdonság, mely három részből áll: Irsz (Irányítószám), Város, Uhsz(Utca Házszám). M-D pedig egy 1:N kapcsolat hiszen több ember is dolgozhat ugyan abban a moziban.
- A Film egyed Hossz tulajdonsága egy származtatott tulajdonság, A Vetítés egyed Tól és Ig tulajdonságával kiszámítható.

## 1.a: Az adatbázis ER modell készítése:



## 1.b: Az adatbázis konvertálása XDM modellre:



1.c: Az XDM modell alapján validált XML dokumentum készítése:

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

<nyilvantartas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaCf94so.xsd">
  <filmek>
    <film fid="0">
      <hossz>120 perc</hossz>
      <cim>Terminator</cim>
    </film>
  </filmek>
  <kategoriak>
    <kategoria kid="0">
      <nev>Akcio</nev>
    </kategoria>
  </kategoriak>
  <mozik>
    <mozi mid="0">
      <varos>Miskolc</varos>
      <nev>A Mozi</nev>
    </mozi>
  </mozik>
  <termek>
    <terem tid="0" mid="0">
      <szam>1</szam>
    </terem>
  </termek>
  <dolgozok>
    <dolgozo did="0" mid="0">
      <nev>Erdei Botond</nev>
      <telefonszamok>
        <telefonszam did="0">
          <szam>06122342342</szam>
        </telefonszam>
      </telefonszamok>
      <cim>
        <irsz>1234</irsz>
        <uhsz>Rozsa utca 1.</uhsz>
        <varos>Miskolc</varos>
      </cim>
    </dolgozo>
  </dolgozok>
  <vetitesek>
    <vetites fid="0" mid="0">
      <tol>13:00</tol>
      <ig>14:20</ig>
    </vetites>
  </vetitesek>
  <fk>
    <fk fid="0" mid="0"></fk>
  </fk>
</nyilvantartas>
```

## 1.d: Az XML dokumentum alapján XML Schema készítése:

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

  <xs:element name="nyilvantartas" type="nyilvantartasType"/>

  <xs:complexType name="filmType">
    <xs:sequence>
      <xs:element type="xs:string" name="hossz"/>
      <xs:element type="xs:string" name="cim"/>
    </xs:sequence>
    <xs:attribute type="xs:byte" name="fid"/>
  </xs:complexType>
  <xs:complexType name="filmekType">
    <xs:sequence>
      <xs:element type="filmType" name="film"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="kategoriaType">
    <xs:sequence>
      <xs:element type="xs:string" name="nev"/>
    </xs:sequence>
    <xs:attribute type="xs:byte" name="kid"/>
  </xs:complexType>
  <xs:complexType name="kategoriakType">
    <xs:sequence>
      <xs:element type="kategoriaType" name="kategoria"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="moziType">
    <xs:sequence>
      <xs:element type="xs:string" name="varos"/>
      <xs:element type="xs:string" name="nev"/>
    </xs:sequence>
    <xs:attribute type="xs:byte" name="mid"/>
  </xs:complexType>
  <xs:complexType name="mozikType">
    <xs:sequence>
      <xs:element type="moziType" name="mozi"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="teremType">
    <xs:sequence>
      <xs:element type="xs:byte" name="szam"/>
    </xs:sequence>
    <xs:attribute type="xs:byte" name="tid"/>
    <xs:attribute type="xs:byte" name="mid"/>
  </xs:complexType>
  <xs:complexType name="termekType">
    <xs:sequence>
      <xs:element type="teremType" name="terem"/>
    </xs:sequence>
  </xs:complexType>
```

```

<xs:complexType name="telefonszamType">
  <xs:sequence>
    <xs:element type="xs:Long" name="szam"/>
  </xs:sequence>
  <xs:attribute type="xs:byte" name="did"/>
</xs:complexType>
<xs:complexType name="telefonszamokType">
  <xs:sequence>
    <xs:element type="telefonszamType" name="telefonszam"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="cimType">
  <xs:sequence>
    <xs:element type="xs:short" name="irsz"/>
    <xs:element type="xs:string" name="uhsz"/>
    <xs:element type="xs:string" name="varos"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="dolgozoType">
  <xs:sequence>
    <xs:element type="xs:string" name="nev"/>
    <xs:element type="telefonszamokType" name="telefonszamok"/>
    <xs:element type="cimType" name="cim"/>
  </xs:sequence>
  <xs:attribute type="xs:byte" name="did"/>
  <xs:attribute type="xs:byte" name="mid"/>
</xs:complexType>
<xs:complexType name="dolgozokType">
  <xs:sequence>
    <xs:element type="dolgozoType" name="dolgozo"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="vetitesType">
  <xs:sequence>
    <xs:element type="xs:string" name="tol"/>
    <xs:element type="xs:string" name="ig"/>
  </xs:sequence>
  <xs:attribute type="xs:byte" name="fid"/>
  <xs:attribute type="xs:byte" name="mid"/>
</xs:complexType>
<xs:complexType name="vetitesekType">
  <xs:sequence>
    <xs:element type="vetitesType" name="vetites"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="fkType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute type="xs:byte" name="fid"/>
      <xs:attribute type="xs:byte" name="mid"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```
<xs:complexType name="fkkType">
  <xs:sequence>
    <xs:element type="fkType" name="fk"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="nyilvantartasType">
  <xs:sequence>
    <xs:element type="filmekType" name="filmek"/>
    <xs:element type="kategoriakType" name="kategoriak"/>
    <xs:element type="mozikType" name="mozik"/>
    <xs:element type="termekType" name="termek"/>
    <xs:element type="dolgozokType" name="dolgozok"/>
    <xs:element type="vetitesekType" name="vetitesek"/>
    <xs:element type="fkkType" name="fkk"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

2 feladat: Egy DOM program elkészítése az XML dokumentum adatai adminisztrálása alappján

2.a: Adatolvasás

```
package hu.domparsed.cf94so;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NamedNodeMap;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.Attr;

public class DOMReadCf94so {

    public static void main(String[] args) {

        try {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            Document doc = dbBuilder.parse("XMLCf94so.xml");

            Element root = doc.getDocumentElement();
            root.normalize();

            print(root);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void print(Node root) {

        String nodename = root.getNodeName();
        if (!nodename.contains("text")) {
            System.out.println(nodename);
        }

        NodeList children = root.getChildNodes();

        for (int i = 0; i < children.getLength(); i++) {
            Node child = children.item(i);
        }
    }
}
```



```

        boolean isComplex = child.getTextContent().contains("\n");

        if (child.hasAttributes()) {
            NamedNodeMap attributes = child.getAttributes();
            int numAttrs = attributes.getLength();
            for (int j = 0; j < numAttrs; j++) {
                Attr attr = (Attr) attributes.item(j);

                String attrName = attr.getNodeName();
                String attrValue = attr.getNodeValue();

                System.out.println(" " + attrName + " : " + attrValue);
            }
        }

        if (isComplex) {
            print(child);
        } else {
            System.out.print(" " + child.getNodeName());
            System.out.println(": " + child.getTextContent());
        }
    }
}
}

```

## 2.b: Adatmódosítás

```
package hu.domparse.cf94so;

import java.io.IOException;

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathExpressionException;
import javax.xml.xpath.XPathFactory;

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

public class DOMModifyCf94so {

    public static void main(String[] args) {

        try {
            DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
            DocumentBuilder dbBuilder = dbFactory.newDocumentBuilder();
            Document doc = dbBuilder.parse("XMLCf94so.xml");
            doc.getDocumentElement().normalize();

            XPath xPath = XPathFactory.newInstance().newXPath();

            String expression = "/nyilvantartas/filmek/film";
            NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(
                doc, XPathConstants.NODESET);

            for (int i = 0; i < nodeList.getLength(); i++) {
                Node nNode = nodeList.item(i);

                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    if( Integer.parseInt(eElement.getAttribute("fid")) == 0) {
                        eElement.getElementsByTagName("hossz").item(0).setTextContent("111 perc");
                        System.out.println("\nCurrent Element : " + nNode.getNodeName());
                        System.out.println("Film id : " + eElement.getAttribute("fid"));
                        System.out.println("Cim : "
```

```

        + eElement
        .getElementsByTagName("cim")
        .item(0)
        .getTextContent());
    System.out.println("Hossz : "
        + eElement
        .getElementsByTagName("hossz")
        .item(0)
        .getTextContent());
    }
}
}
} catch (ParserConfigurationException e) {
    e.printStackTrace();
} catch (SAXException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (XPathExpressionException e) {
    e.printStackTrace();
}
}
}
}

```