**Lehet félre értettem a jegyzőkönyvet. Megpróbáltam kihozni valamit.**

# 1a. ER modell

## Egyedek (min. 5)

### 1. HALLGATÓ

- **hallgato_id** (PK)
- név *(összetett)*
    - vezetéknév
    - keresztnév
- születési_dátum
- email *(többértékű)*
- évfolyam

### 2. OKTATÓ

- **oktato_id** (PK)
- név *(összetett)*
- tanszék
- beosztás
- email

### 3. KURZUS

- **kurzus_id** (PK)
- kurzus_név
- kredit
- félév
- óraszám

### 4. TANSZÉK

- **tanszek_id** (PK)
- tanszek_név
- iroda
- telefon
- kar

### 5. TEREM

- **terem_id** (PK)
- épület
- terem_szám
- férőhely
- típus

## Kapcsolatok

### OKTATÓ — TANSZÉK

- **1:N**
  (Egy tanszékhez több oktató tartozik)

### KURZUS — OKTATÓ

- **1:N**
  (Egy oktató több kurzust tarthat)

### HALLGATÓ — KURZUS

- **M:N → FELVÉTEL** kapcsolóegyed
  Attribútumai:
- **jegy**
- teljesítés_dátuma
- státusz

### KURZUS — TEREM

- **1:1**
  (Egy kurzus egy fix teremben van megtartva

A hierarchia: XDM modell

egyetem

├─ tanszekek

| └─ tanszek

├─ oktatok

├─ kurzusok

| └─ felvetelek

├─ hallgatok

└─ termek

XML dokumentum

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

<egyetem>


  <!-- Tanszékek -->

  <tanszekek>
```

```
  <tanszek id="T1">
    <nev>Informatikai Tanszék</nev>
    <iroda>B/203</iroda>
    <telefon>+361234567</telefon>
    <kar>TTK</kar>
  </tanszek>

  <tanszek id="T2">
    <nev>Matematika Tanszék</nev>
    <iroda>C/101</iroda>
    <telefon>+361987654</telefon>
    <kar>TTK</kar>
  </tanszek>
</tanszekek>

<!-- Oktatók -->
<oktatok>
  <oktato id="O1" tanszek_ref="T1">
    <nev>
      <vezeteknev>Kiss</vezeteknev>
      <keresztnev>Péter</keresztnev>
    </nev>
    <beosztas>docens</beosztas>
    <email>kiss.peter@uni.hu</email>
  </oktato>

  <oktato id="O2" tanszek_ref="T2">
    <nev>
      <vezeteknev>Nagy</vezeteknev>
      <keresztnev>Anna</keresztnev>
    </nev>
```

```
      <beosztas>adjunktus</beosztas>
      <email>nagy.anna@uni.hu</email>
   </oktato>
</oktatok>

<!-- Hallgatók -->
<hallgatok>
   <hallgato id="H1">
      <nev>
         <vezeteknev>Kovács</vezeteknev>
         <keresztnev>Bence</keresztnev>
      </nev>
      <szuletesi_datum>2002-05-12</szuletesi_datum>
      <email>kovacs.bence@gmail.com</email>
      <email>bence@uni.hu</email>
      <evfolyam>3</evfolyam>
   </hallgato>

   <hallgato id="H2">
      <nev>
         <vezeteknev>Szabó</vezeteknev>
         <keresztnev>Dóra</keresztnev>
      </nev>
      <szuletesi_datum>2001-11-03</szuletesi_datum>
      <email>szabo.dora@gmail.com</email>
      <evfolyam>4</evfolyam>
   </hallgato>
</hallgatok>

<!-- Termek -->
<termek>
```

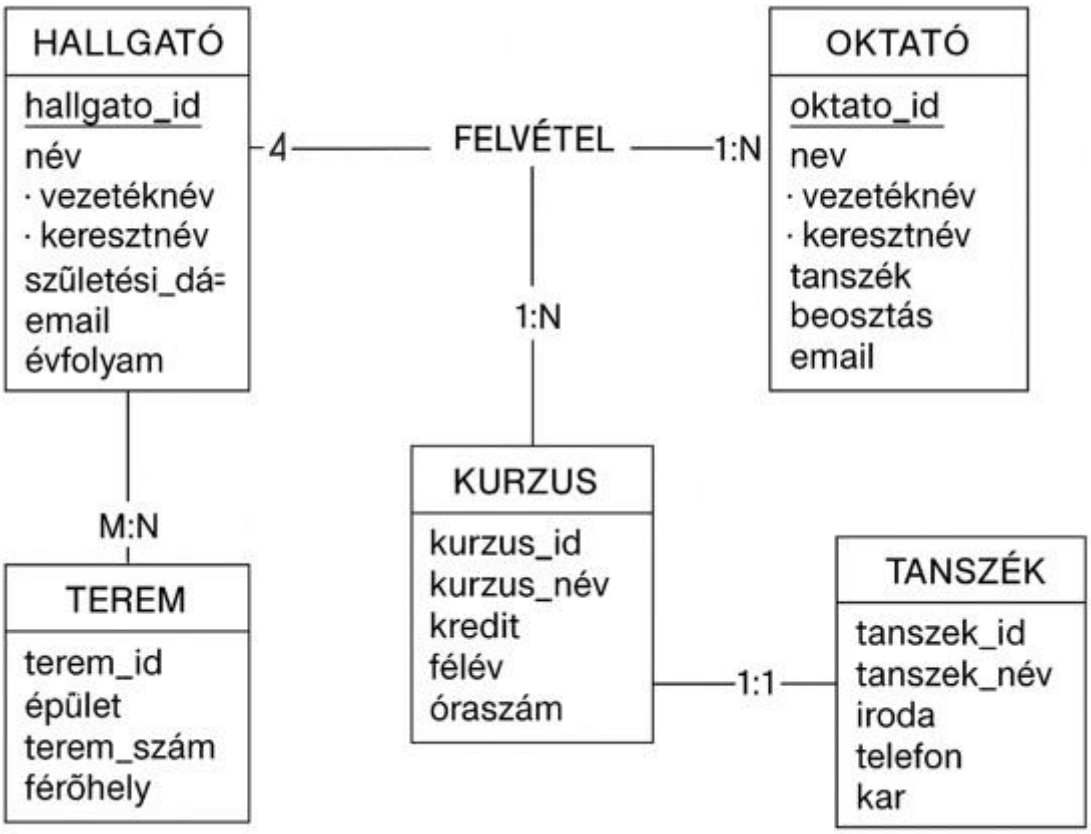
```
    <terem id="R1">
      <epulet>A</epulet>
      <terem_szam>101</terem_szam>
      <ferohely>40</ferohely>
      <tipus>előadó</tipus>
    </terem>

    <terem id="R2">
      <epulet>B</epulet>
      <terem_szam>202</terem_szam>
      <ferohely>25</ferohely>
      <tipus>számítógépes</tipus>
    </terem>
</termek>

<!-- Kurzusok és felvételek -->
<kurzusok>
    <kurzus id="K1" oktato_ref="O1" terem_ref="R1">
      <nev>XML adatkezelés</nev>
      <kredit>5</kredit>
      <felev>2024/25/1</felev>
      <oraszam>4</oraszam>

      <!-- Felvételek -->
      <felvetelek>
        <felvetel hallgato_ref="H1">
          <jegy>5</jegy>
          <teljesites_datuma>2024-12-15</teljesites_datuma>
          <statusz>teljesítve</statusz>
        </felvetel>
```

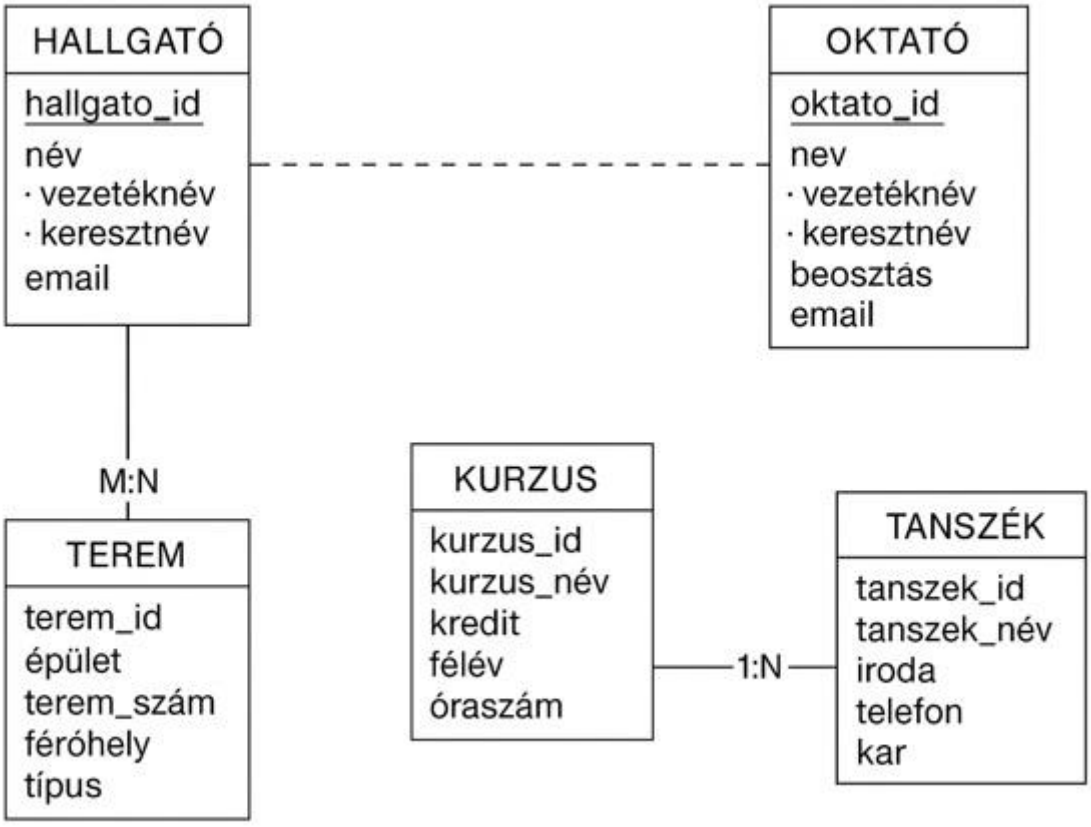
```
            <felvetel hallgato_ref="H2">
               <jegy>4</jegy>
               <teljesites_datuma>2024-12-16</teljesites_datuma>
               <statusz>teljesítve</statusz>
            </felvetel>
         </felvetelek>
      </kurzus>
   </kurzusok>

</egyetem>
```

XML schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

   <!-- Összetett név típus -->
   <xs:complexType name="NevTipus">
      <xs:sequence>
         <xs:element name="vezeteknev" type="xs:string"/>
         <xs:element name="keresztnev" type="xs:string"/>
      </xs:sequence>
   </xs:complexType>

   <!-- Hallgató típus -->
   <xs:complexType name="HallgatoTipus">
      <xs:sequence>
         <xs:element name="nev" type="NevTipus"/>
         <xs:element name="szuletesi_datum" type="xs:date"/>
         <xs:element name="email" type="xs:string" maxOccurs="unbounded"/>
         <xs:element name="evfolyam" type="xs:integer"/>
```

```
    </xs:sequence>
    <xs:attribute name="id" type="xs:ID" use="required"/>
  </xs:complexType>

  <!-- Gyökérelem -->
  <xs:element name="egyetem">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="hallgatok" minOccurs="1"/>
        <xs:element name="oktatok" minOccurs="1"/>
        <xs:element name="kurzusok" minOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```






DOM program:

CSS

YKPX27DOMParse

└ src

└ neptunkod

└ domparse

└ hu

├ YKPX27DOMRead.java

├ YKPX27DOMQuery.java

└ YKPX27DOMModify.java

XML beolvasás:

```
package YKPX27.domparse.hu;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

import java.io.File;

public class YKPX27DOMRead {

   public static void main(String[] args) {

      try {
         // XML fájl betöltése
         File xmlFile = new File("YKPX27XML.xml");

         // DOM parser létrehozása
         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
         DocumentBuilder builder = factory.newDocumentBuilder();

         // XML dokumentum beolvasása
         Document document = builder.parse(xmlFile);
         document.getDocumentElement().normalize();

         // Gyökérelem kiírása
         System.out.println("Gyökérelem: " + document.getDocumentElement().getNodeName());
```

```
        // Összes gyermek elem bejárása
        NodeList nodeList = document.getDocumentElement().getChildNodes();

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

           if (node.getNodeType() == Node.ELEMENT_NODE) {
              System.out.println("Elem: " + node.getNodeName());
           }
        }

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

Adatlekérés:

```
package YKPX27.domparse.hu;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.*;

import java.io.File;

public class YKPX27DOMQuery {
```

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

   try {
      File xmlFile = new File("YKPX27XML.xml");

      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
      DocumentBuilder builder = factory.newDocumentBuilder();
      Document document = builder.parse(xmlFile);

      document.getDocumentElement().normalize();

      // Hallgatók lekérdezése
      NodeList hallgatoList = document.getElementsByTagName("hallgato");

      System.out.println("Hallgatók száma: " + hallgatoList.getLength());

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

         Node node = hallgatoList.item(i);

         if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element hallgato = (Element) node;

            String id = hallgato.getAttribute("id");
            String vezeteknev = hallgato
                  .getElementsByTagName("vezeteknev")
                  .item(0)
                  .getTextContent();
            String keresztnev = hallgato
                  .getElementsByTagName("keresztnev")
                  .item(0)
```

```
                    .getTextContent();
            String evfolyam = hallgato
                    .getElementsByTagName("evfolyam")
                    .item(0)
                    .getTextContent();

            System.out.println("ID: " + id);
            System.out.println("Név: " + vezeteknev + " " + keresztnev);
            System.out.println("Évfolyam: " + evfolyam);
            System.out.println("----------------------");
          }
        }

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

Adatmódosítás:

```
package YKPX27.domparse.hu;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.*;
```

```
import java.io.File;

public class YKPX27DOMModify {

   public static void main(String[] args) {

      try {
         File xmlFile = new File("YKPX27XML.xml");

         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
         DocumentBuilder builder = factory.newDocumentBuilder();
         Document document = builder.parse(xmlFile);

         document.getDocumentElement().normalize();

         // Hallgatók listája
         NodeList hallgatoList = document.getElementsByTagName("hallgato");

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

            Element hallgato = (Element) hallgatoList.item(i);

            // Konkrét hallgató kiválasztása ID alapján
            if (hallgato.getAttribute("id").equals("H1")) {

               // Évfolyam módosítása
               hallgato.getElementsByTagName("evfolyam")
                     .item(0)
                     .setTextContent("4");
```

```
                System.out.println("Hallgató évfolyama módosítva!");
            }
        }

        // XML visszaírás fájlba
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();

        transformer.setOutputProperty(OutputKeys.INDENT, "yes");

        DOMSource source = new DOMSource(document);
        StreamResult result = new StreamResult(new File("YKPX27XML.xml"));

        transformer.transform(source, result);

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