

# JEGYZŐKÖNYV

Adatkezelés XML környezetben  
Féléves Feladat

Készítette: **Csehi Máté**

Neptunkód: **GKDM8D**

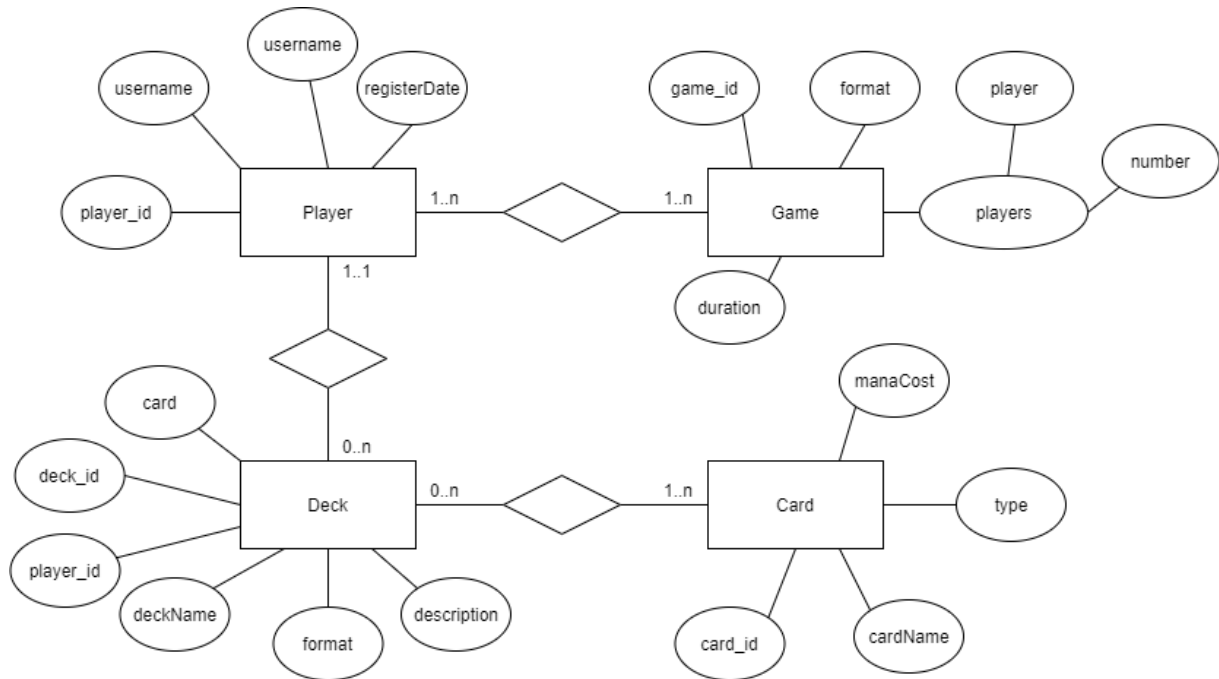
**A feladat leírása:**

A Magic the Gathering(mtg) kártyajáték leegyszerűsített adatbázisrendszerét készítettem el, amely négy egyedből áll. Az alap ötletet a Cockatrice nevű nyílt forráskódú mtg szimulátor adta, ahol a paklikat hasonló xml fájlokban tárolják.

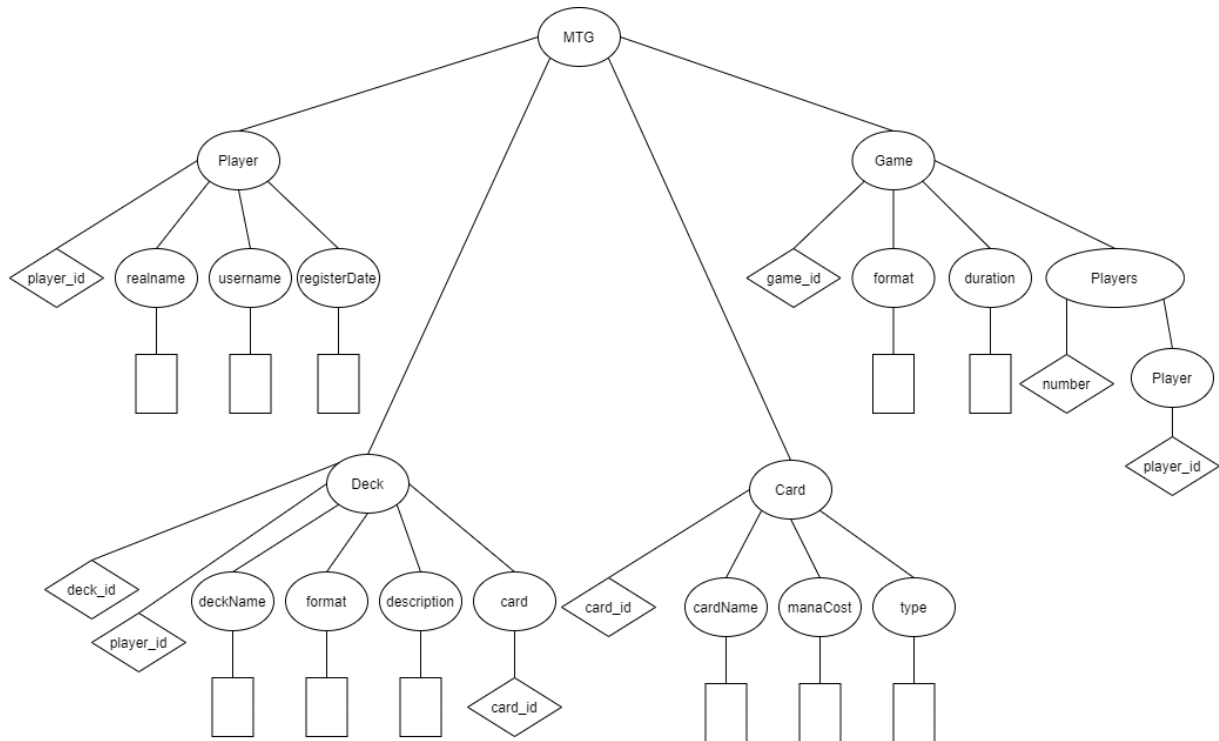
A Cockatrice-től eltérően a kártyák tulajdonságait nem a paklikban tárolom el, hanem külön egyedként kezelem őket, valamint az én adatbázisom nem csak a paklilat és lapokat tárolja el, hanem nyilvántartja a játékosokat is, és a lejátszott játékokat is.

## 1 feladat

### 1 a) Az adatbázis ER modellje:



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



### 1 c) Az XDM modell alapján XML dokumentum készítése:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="XMLSchemaGKDM8D.xsl"?>
```

```
<mtg>
  <player player_id="01">
    <username>Tsehy</username>
    <realname>Winch Eszter</realname>
    <registerDate>2020-05-28</registerDate>
  </player>
  <player player_id="02">
    <username>Wass</username>
    <realname>Zsíros B. Ödön</realname>
    <registerDate>2019-01-01</registerDate>
  </player>
  <player player_id="03">
    <username>Mate77</username>
    <realname>Fá Zoltán</realname>
    <registerDate>1969-07-13</registerDate>
  </player>
  <card card_id="01">
    <cardName>Solar Blaze</cardName>
    <manaCost>2RW</manaCost>
    <type>Sorcery</type>
  </card>
  <card card_id="02">
    <cardName>Lightning Bolt</cardName>
    <manaCost>R</manaCost>
    <type>Instant</type>
  </card>
  <card card_id="03">
    <cardName>Syr Konrad, the Grim</cardName>
    <manaCost>3BB</manaCost>
    <type>Legendary Creature - Human Knight</type>
  </card>
  <card card_id="04">
    <cardName>Chandra, Flame's Fury</cardName>
    <manaCost>4RR</manaCost>
    <type>Legendary Planeswalker - Chandra</type>
  </card>
  <card card_id="05">
    <cardName>Colossal Dreadmaw</cardName>
    <manaCost>4GG</manaCost>
    <type>Creature - Dinosaur</type>
  </card>
  <card card_id="06">
    <cardName>Arclight Phoenix</cardName>
    <manaCost>3R</manaCost>
    <type>Creature - Phoenix</type>
  </card>
  <card card_id="07">
    <cardName>Opt</cardName>
```

```

        <manaCost>U</manaCost>
        <type>Instant</type>
    </card>
    <card card_id="08">
        <cardName>Kenrith, the Returned King</cardName>
        <manaCost>4W</manaCost>
        <type>Legendary Creature - Human Noble</type>
    </card>
    <card card_id="09">
        <cardName>Ajani's Pridemate</cardName>
        <manaCost>1W</manaCost>
        <type>Creature - Cat</type>
    </card>
    <deck deck_id="01" player_id="01">
        <format>Commander</format>
        <deckName>You will die</deckName>
        <description>Syr Konrad self mill EDH</description>
        <card card_id="03"/>
        <card card_id="02"/>
        <card card_id="06"/>
        <card card_id="04"/>
    </deck>
    <deck deck_id="02" player_id="01">
        <format>Standard</format>
        <deckName>Blazing Fire</deckName>
        <description>Solar Blaze and Chandra deck 4fun</description>
        <card card_id="01"/>
        <card card_id="01"/>
        <card card_id="01"/>
        <card card_id="01"/>
        <card card_id="04"/>
        <card card_id="04"/>
    </deck>
    <deck deck_id="03" player_id="03">
        <format>Commander</format>
        <deckName>Broken Kenrith</deckName>
        <description>Kenrith, but not budget</description>
        <card card_id="08"/>
        <card card_id="07"/>
        <card card_id="05"/>
    </deck>
    <deck deck_id="04" player_id="02">
        <format>Standard</format>
        <deckName>We are the dinosaurs</deckName>
        <description>marching, marching</description>
        <card card_id="05"/>
        <card card_id="05"/>
        <card card_id="05"/>
        <card card_id="05"/>
    </deck>
    <game game_id="01">
        <format>Commander</format>
        <duration>00:35:12</duration>

```

```

        <players number="2">
            <player player_id="01"/>
            <player player_id="02"/>
        </players>
    </game>
    <game game_id="02">
        <format>Standard</format>
        <duration>00:12:07</duration>
        <players number="2">
            <player player_id="01"/>
            <player player_id="02"/>
        </players>
    </game>
    <game game_id="03">
        <format>Chaos</format>
        <duration>01:21:56</duration>
        <players number="3">
            <player player_id="01"/>
            <player player_id="02"/>
            <player player_id="03"/>
        </players>
    </game>
</mtg>

```

#### 1 d) Az XML dokumentum alapján XMLSchema készítése:

```

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

    <xs:complexType name="playerType">
        <xs:sequence>
            <xs:element name="username" type="xs:string"/>
            <xs:element name="realname" type="xs:string"/>
            <xs:element name="registerDate" type="xs:date"/>
        </xs:sequence>
        <xs:attribute name="player_id" type="xs:integer" use="required"/>
    </xs:complexType>

    <xs:complexType name="cardType">
        <xs:sequence>
            <xs:element name="cardName" type="xs:string"/>
            <xs:element name="manaCost" type="xs:string"/>
            <xs:element name="type" type="xs:string"/>
        </xs:sequence>
        <xs:attribute name="card_id" type="xs:integer" use="required"/>
    </xs:complexType>

    <xs:complexType name="deckType">
        <xs:sequence>
            <xs:element name="format" type="xs:string"/>

```

```

    <xs:element name="deckName" type="xs:string"/>
    <xs:element name="description" type="xs:string"/>
    <xs:element name="card" maxOccurs="unbounded">
      <xs:complexType>
        <xs:attribute name="card_id" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="deck_id" type="xs:integer" use="required"/>
  <xs:attribute name="player_id" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="gameType">
  <xs:sequence>
    <xs:element name="format" type="xs:string"/>
    <xs:element name="duration" type="xs:time"/>
    <xs:element name="players">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="player" maxOccurs="unbounded">
            <xs:complexType>
              <xs:attribute name="player_id" type="xs:integer" use="required"/>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="number" type="xs:integer" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="game_id" type="xs:integer" use="required"/>
</xs:complexType>

<xs:element name="mtg">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="player" type="playerType" maxOccurs="unbounded"/>
      <xs:element name="card" type="cardType" maxOccurs="unbounded"/>
      <xs:element name="deck" type="deckType" maxOccurs="unbounded"/>
      <xs:element name="game" type="gameType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <!-- kulcsok -->
  <xs:key name="key1">
    <xs:selector xpath="player"/>
    <xs:field xpath="@player_id"/>
  </xs:key>
  <xs:key name="key2">
    <xs:selector xpath="card"/>
    <xs:field xpath="@card_id"/>
  </xs:key>

```

```
</xs:key>
<xs:key name="key3">
  <xs:selector xpath="deck"/>
  <xs:field xpath="@deck_id"/>
</xs:key>
<xs:key name="key4">
  <xs:selector xpath="game"/>
  <xs:field xpath="@game_id"/>
</xs:key>
<xs:keyref name="keyref1" refer="key1">
  <xs:selector xpath="deck"/>
  <xs:field xpath="@player_id"/>
</xs:keyref>
<xs:keyref name="keyref2" refer="key1">
  <xs:selector xpath="game/players/player"/>
  <xs:field xpath="@player_id"/>
</xs:keyref>
<xs:keyref name="keyref3" refer="key2">
  <xs:selector xpath="deck/card"/>
  <xs:field xpath="@card_id"/>
</xs:keyref>
</xs:element>
</xs:schema>
```



## 2 feladat

### 2 a) adatolvasás:

```
package hu.domparse.gkdm8d;

import java.io.IOException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
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.xml.sax.SAXException;

//Az adott xml dokumentumot beolvassa, és kiírja a console-ba
public class DOMReadGKDM8D
{
    public static void main(String[] args)
    {
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        try
        {
            DocumentBuilder builder = factory.newDocumentBuilder();
            try
            {
                try
                {
                    Document document = builder.parse("XMLGKDM8D.xml");
                    Element rootNode = document.getDocumentElement();
                    printNode(rootNode, "", rootNode.getNodeName());
                }
                catch(SAXException e)
                {
                    e.printStackTrace();
                }
            }
            catch(IOException e)
            {
                e.printStackTrace();
            }
        }
        catch(ParserConfigurationException e)
        {
            e.printStackTrace();
        }
    }
}

//kiírató függvény
```

```

public static void printNode(Node rootNode, String spacer, String rootName)
{
    if(rootNode.getNodeType() == Node.ELEMENT_NODE)
    {
        if(rootNode.getParentNode().getNodeName().equals(rootName))
            System.out.println(spacer);
        System.out.print(spacer + rootNode.getNodeName());
        if(!(rootNode.getNodeName().equals(rootName)))
        {
            System.out.print(rootNode.hasAttributes()? " -":": ");
            System.out.println(rootNode.hasAttributes()?printAttributes(rootNode):rootNode.getTextContent());
        }
        else
            System.out.println();
    }
    NodeList nodeList = rootNode.getChildNodes();
    for (int i = 0; i < nodeList.getLength(); i++)
        printNode(nodeList.item(i), spacer + " |", rootName);
}

//attribútumokat lekérdező függvény
public static String printAttributes(Node mainNode)
{
    NamedNodeMap attributeList = mainNode.getAttributes();
    String attributes = " " + attributeList.item(0).getNodeName() + ":" + attributeList.item(0).getNodeValue();
    for(int i = 1; i < attributeList.getLength(); i++)
    {
        Node attribute = attributeList.item(i);
        attributes += (" , " + attribute.getNodeName() + ":" + attribute.getNodeValue());
    }
    return attributes;
}
}

```

## 2 b) adatmódosítás:

```
package hu.domparse.gkdm8d;

import java.io.File;
import java.io.IOException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
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.xml.sax.SAXException;

//A beolvasott xml fájlban megkeresi az adott node-ot, és új értéket ad meg neki
public class DOMModifyGKDM8D
{
    public static void main(String[] args)
    {
        String parentName = "card";
        String nodeName = "cardName";
        String newValue = "Infuriate";
        String id = "02";
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        try
        {
            DocumentBuilder builder = factory.newDocumentBuilder();
            try
            {
                try
                {
                    File xmlFile = new File("XMLGKDM8D.xml");
                    Document document = builder.parse(xmlFile);
                    Element rootNode = document.getDocumentElement();
                    searchParentNode(rootNode, parentName, nodeName, id, newValue);
                    modifyDocument(document, xmlFile);
                }
                catch(SAXException e)
                {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

```

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

//Szülő megkeresése
public static void searchParentNode(Node rootNode, String parentName, String nodeName,
String id, String newValue)
{
    if(rootNode.getNodeType() == Node.ELEMENT_NODE)
    {
        if((rootNode.getNodeName().equals(parentName)))
        {
            NamedNodeMap attributeList = rootNode.getAttributes();
            if(attributeList.item(0).getNodeValue().equals(id))
            {
                searchNode(rootNode, nodeName, newValue);
            }
        }
    }
    NodeList nodeList = rootNode.getChildNodes();
    for (int i = 0; i < nodeList.getLength(); i++)
        searchParentNode(nodeList.item(i), parentName, nodeName, id, newValue);
}

//node megkeresése és adatcsere
public static void searchNode(Node rootNode, String nodeName, String newValue)
{
    if(rootNode.getNodeType() == Node.ELEMENT_NODE)
    {
        if((rootNode.getNodeName().equals(nodeName)))
        {
            rootNode.setTextContent(newValue);
        }
    }
    NodeList nodeList = rootNode.getChildNodes();
    for (int i = 0; i < nodeList.getLength(); i++)
        searchNode(nodeList.item(i), nodeName, newValue);
}

//dokumentum módosító függvény
public static void modifyDocument(Document document, File xmlFile)

```

```
{
    try
    {
        TransformerFactory transformerFactory = TransformerFactory.newInstance();
        Transformer transformer = transformerFactory.newTransformer();
        DOMSource source = new DOMSource(document);
        StreamResult result = new StreamResult(xmlFile);

        transformer.transform(source, new StreamResult(System.out));
        transformer.transform(source, result);
    }
    catch(TransformerException e)
    {
        e.printStackTrace();
    }
}
```