

# **Jegyzőkönyv**

**Adatkezelés XML környezetben**

**Féléves feladat**

**Cinemagic mozi jegyeladó rendszer  
adatbázisa**

**Készítette: Dudás Dániel  
Neptun kód: VUXFKS**

## A feladat leírása:

A Cinemagic új mozihálózat adatbázisának felvázolása jegypénztári megközelítésből. Ebben találhatóak a jegyadatok, az eladáshoz szükséges belépési azonosítók, a mozik adatai és a már megvalósult vásárlások nyilvántartása.

## Az egyedek és tulajdonságaik:

### A Mozi egyed:

- **MozID:** elsődleges kulcs
- **Mozinev:** egy adott mozi nevét tárolja
- **Cim:** az adatbázisban található mozi címe, összetett tulajdonság, tárolja az irányítószámot, utcát és házszámot.

### Az Elado egyed:

- **EladoID:** elsődleges kulcs
- **Felhasznalonev:** az eladó felhasználóneve
- **Jelszo:** az eladó jelszava
- **Admin:** az eladó adminisztrátori jogosultságát tárolja

### A Jegy egyed:

- **JegyID:** elsődleges kulcs
- **Vetites:** idegen kulcs, a Vetites egyeddel való kapcsolatot azonosítóját tárolja
- **Sor:** az adott jegy ülőhelyének sorát tárolja
- **Oszlop:** az adott jegy ülőhelyének sorát tárolja

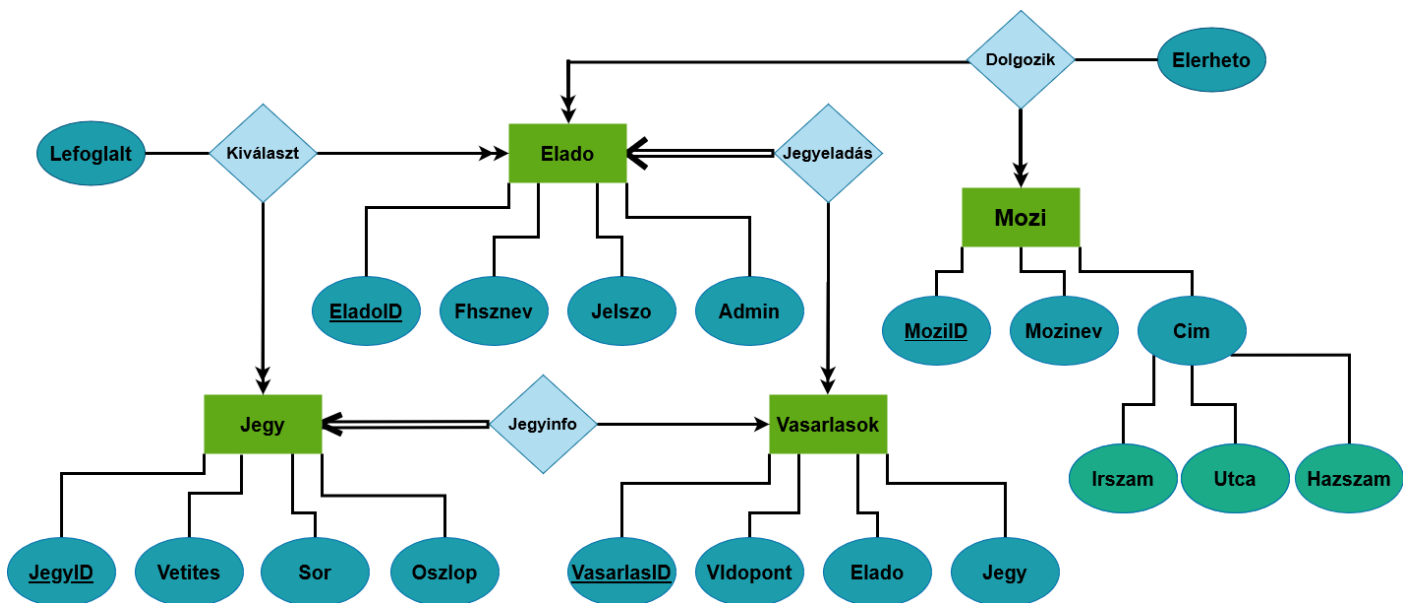
### A Vasarlasok egyed:

- **VasarlasID:** elsődleges kulcs
- **Vldopont:** egy normál tulajdonságú mező, ami a vásárlás idejét tárolja
- **Elado:** egy idegen kulcs, ami az EladoID-vel áll kapcsolatban és az eladást lebonyolító alkalmazott azonosítóját tárolja
- **Jegy:** idegen kulcs, ami a JegyID-vel áll kapcsolatban és a megvásárolt jegy részleteit érhetjük el

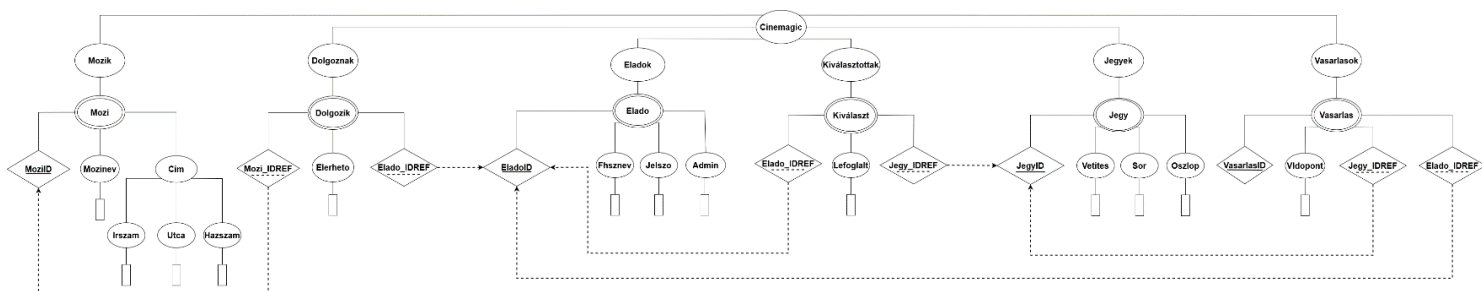
## Egyedek közötti kapcsolat:

- **Mozi és Elado között:**  
N:M típusú a kapcsolat, mivel egy moziban több eladó is dolgozhat és egy eladó több moziban is dolgozhat.
- **Elado és Jegy között:**  
N:M típusú kapcsolat, mert egy jegyet több eladó is kiválaszthat és egy eladó több jegyet is kiválaszthat.
- **Elado és Vasarlasok között:**  
1:N típusú kapcsolat van, mert egy eladó több jegyet is eladhat, de egy adott jegyet csak egy eladó adhat el.
- **Vasarlasok és Jegy között:**  
1:1 típusú kapcsolat, mert egy megvásárolt jegy jegyinformációjának lekérdezésével egy jegyinformációt kaphatunk és egy jegy foglaltságának lekérdezésekor nem lehet eredmény több jegy információja.

## Az adatbázis ER- modellje:



## Az ER-modell konvertálása XDM- modellre:



## XDM- modell alapján készült XML fájl:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <Cinemagic xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XMLSchemaVUXFKS.xsd">
4   <mozik>
5     <mozi MoziID="01">
6       <mozinev>CinemaMiskolc</mozinev>
7       <cim>
8         <irszam>3500</irszam>
9         <utca>Rákóczi</utca>
10        <hazszam>15.</hazszam>
11      </cim>
12    </mozi>
13    <mozi MoziID="02">
14      <mozinev>CinemaBudapest</mozinev>
15      <cim>
16        <irszam>1100</irszam>
17        <utca>Fő</utca>
18        <hazszam>4.</hazszam>
19      </cim>
20    </mozi>
21    <mozi MoziID="03">
22      <mozinev>CinemaSalgótarján</mozinev>
23      <cim>
24        <irszam>3100</irszam>
25        <utca>Kiss</utca>
26        <hazszam>88.</hazszam>
27      </cim>
28    </mozi>
29  </mozik>
30  <dolgoznak>
31    <dolgozik Mozi_IDREF="01" Elado_IDREF="01" >
32      <elerheto>0</elerheto>
33    </dolgozik>
34    <dolgozik Mozi_IDREF="02" Elado_IDREF="02">
35      <elerheto>1</elerheto>
36    </dolgozik>
37    <dolgozik Mozi_IDREF="03" Elado_IDREF="03">
38      <elerheto>1</elerheto>
39    </dolgozik>
40  </dolgoznak>
```

```
41● <eladok>
42●   <elado EladoID="01" >
43       <fhsznev>Béla05</fhsznev>
44       <jelszo>ujjelszo1</jelszo>
45       <admin>0</admin>
46   </elado>
47●   <elado EladoID="02" >
48       <fhsznev>Anna01</fhsznev>
49       <jelszo>ujjelszo2</jelszo>
50       <admin>0</admin>
51   </elado>
52●   <elado EladoID="03" >
53       <fhsznev>Judit08</fhsznev>
54       <jelszo>ujjelszo3</jelszo>
55       <admin>0</admin>
56   </elado>
57●   <elado EladoID="04" >
58       <fhsznev>Péter04</fhsznev>
59       <jelszo>ujjelszo4</jelszo>
60       <admin>1</admin>
61   </elado>
62   </eladok>
63● <kivalasztottak>
64●   <kivalaszt Elado_IDREF="03" Jegy_IDREF="01">
65       <lefoglalt>0</lefoglalt>
66   </kivalaszt>
67●   <kivalaszt Elado_IDREF="02" Jegy_IDREF="02">
68       <lefoglalt>0</lefoglalt>
69   </kivalaszt>
70●   <kivalaszt Elado_IDREF="03" Jegy_IDREF="04">
71       <lefoglalt>1</lefoglalt>
72   </kivalaszt>
73●   <kivalaszt Elado_IDREF="02" Jegy_IDREF="06">
74       <lefoglalt>1</lefoglalt>
75   </kivalaszt>
76 </kivalasztottak>
```

```
76 </Kivadasztottak>
77 <jegyek>
78   <jegy JegyID="01">
79     <vetites>01</vetites>
80     <sor>10</sor>
81     <oszlop>1</oszlop>
82   </jegy>
83   <jegy JegyID="02">
84     <vetites>01</vetites>
85     <sor>10</sor>
86     <oszlop>2</oszlop>
87   </jegy>
88   <jegy JegyID="03">
89     <vetites>01</vetites>
90     <sor>5</sor>
91     <oszlop>4</oszlop>
92   </jegy>
93   <jegy JegyID="04">
94     <vetites>01</vetites>
95     <sor>5</sor>
96     <oszlop>5</oszlop>
97   </jegy>
98   <jegy JegyID="05">
99     <vetites>02</vetites>
100     <sor>11</sor>
101     <oszlop>7</oszlop>
102   </jegy>
103   <jegy JegyID="06">
104     <vetites>02</vetites>
105     <sor>8</sor>
106     <oszlop>9</oszlop>
107   </jegy>
108 </jegyek>
109 <vasarlasok>
110   <vasarlas VasarlasID="01" Jegy_IDREF="05" Elado_IDREF="02">
111     <vidopont>2021.12.01/10:00</vidopont>
112   </vasarlas>
113   <vasarlas VasarlasID="02" Jegy_IDREF="03" Elado_IDREF="03">
114     <vidopont>2021.12.01/10:08</vidopont>
115   </vasarlas>
116 </vasarlasok>
117 </Cinemagic>
```

## Az XML fájl alapján készült XML Schema:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3   <xs:element name="Cinemagic">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="mozik" type="mozik_type"/>
7         <xs:element name="dolgoznak" type="dolgoznak_type"/>
8         <xs:element name="eladok" type="eladok_type"/>
9         <xs:element name="kivalasztottak" type="kivalasztottak_type"/>
10        <xs:element name="jegyek" type="jegyek_type"/>
11        <xs:element name="vasarlasok" type="vasarlasok_type"/>
12      </xs:sequence>
13    </xs:complexType>
14
15    <xs:key name="Mozi_PK">
16      <xs:selector xpath="mozik/mozi"/>
17      <xs:field xpath="@MoziID"/>
18    </xs:key>
19    <xs:key name="Elado_PK">
20      <xs:selector xpath="eladok/elado"/>
21      <xs:field xpath="@EladoID"/>
22    </xs:key>
23    <xs:key name="Jegy_PK">
24      <xs:selector xpath="jegyek/jegy"/>
25      <xs:field xpath="@JegyID"/>
26    </xs:key>
27    <xs:key name="VasarLas_PK">
28      <xs:selector xpath="vasarlasok/vasarlas"/>
29      <xs:field xpath="@VasarLasID"/>
30    </xs:key>
31
32    <xs:keyref name="Mozi_FK" refer="Mozi_PK">
33      <xs:selector xpath="dolgoznak/dolgozik"/>
34      <xs:field xpath="@Mozi_IDREF"/>
35    </xs:keyref>
36    <xs:keyref name="DElado_FK" refer="Elado_PK">
37      <xs:selector xpath="dolgoznak/dolgozik"/>
38      <xs:field xpath="@Elado_IDREF"/>
39    </xs:keyref>
40    <xs:keyref name="KElado_FK" refer="Elado_PK">
41      <xs:selector xpath="kivalasztottak/kivalaszt"/>
42      <xs:field xpath="@Elado_IDREF"/>
43    </xs:keyref>
44    <xs:keyref name="VELado_FK" refer="Elado_PK">
45      <xs:selector xpath="vasarlasok/vasarlas"/>
46      <xs:field xpath="@Elado_IDREF"/>
47    </xs:keyref>
48    <xs:keyref name="KJegy_FK" refer="Jegy_PK">
49      <xs:selector xpath="kivalasztottak/kivalaszt"/>
50      <xs:field xpath="@Jegy_IDREF"/>
51    </xs:keyref>
52    <xs:keyref name="VJegy_FK" refer="Jegy_PK">
53      <xs:selector xpath="vasarlasok/vasarlas"/>
54      <xs:field xpath="@Jegy_IDREF"/>
55    </xs:keyref>
56
57  </xs:element>
58
59 </xs:schema>
```

```
61<xs:complexType name="mozik_type">
62  <xs:sequence>
63    <xs:element name="mozi" type="mozi_type" maxOccurs="unbounded"/>
64  </xs:sequence>
65</xs:complexType>
66<xs:complexType name="dolgoznak_type">
67  <xs:sequence>
68    <xs:element name="dolgozik" type="dolgozik_type" maxOccurs="unbounded"/>
69  </xs:sequence>
70</xs:complexType>
71<xs:complexType name="eladok_type">
72  <xs:sequence>
73    <xs:element name="elado" type="elado_type" maxOccurs="unbounded"/>
74  </xs:sequence>
75</xs:complexType>
76<xs:complexType name="kivalasztottak_type">
77  <xs:sequence>
78    <xs:element name="kivalaszt" type="kivalaszt_type" maxOccurs="unbounded"/>
79  </xs:sequence>
80</xs:complexType>
81<xs:complexType name="jegyek_type">
82  <xs:sequence>
83    <xs:element name="jegy" type="jegy_type" maxOccurs="unbounded"/>
84  </xs:sequence>
85</xs:complexType>
86<xs:complexType name="vasarlasok_type">
87  <xs:sequence>
88    <xs:element name="vasarlas" type="vasarlas_type" maxOccurs="unbounded"/>
89  </xs:sequence>
90</xs:complexType>
91
92<xs:complexType name="mozi_type">
93  <xs:sequence>
94    <xs:element name="mozinev" type="xs:string"/>
95    <xs:element name="cim" type="cim_type"/>
96  </xs:sequence>
97  <xs:attribute name="MoziID" type="xs:unsignedByte" use="required"/>
98</xs:complexType>
99
100<xs:complexType name="dolgozik_type">
101  <xs:sequence>
102    <xs:element name="elerheto" type="xs:integer"/>
103  </xs:sequence>
104  <xs:attribute name="Mozi_IDREF" type="xs:unsignedByte" use="required"/>
105  <xs:attribute name="Elado_IDREF" type="xs:unsignedByte" use="required"/>
106</xs:complexType>
```



```

108< xs:complexType name="elado_type">
109<   <xs:sequence>
110<     <xs:element name="fhsznev" type="xs:string"/>
111<     <xs:element name="jelszo" type="xs:string"/>
112<     <xs:element name="admin" type="xs:integer"/>
113<   </xs:sequence>
114<   <xs:attribute name="EladoID" type="xs:unsignedByte" use="required"/>
115< </xs:complexType>
116
117< xs:complexType name="kivalaszt_type">
118<   <xs:sequence>
119<     <xs:element name="lefoglalt" type="xs:integer"/>
120<   </xs:sequence>
121<   <xs:attribute name="Elado_IDREF" type="xs:unsignedByte" use="required"/>
122<   <xs:attribute name="Jegy_IDREF" type="xs:unsignedByte" use="required"/>
123< </xs:complexType>
124
125< xs:complexType name="jegy_type">
126<   <xs:sequence>
127<     <xs:element name="vetites" type="xs:string"/>
128<     <xs:element name="sor" type="xs:integer"/>
129<     <xs:element name="oszló" type="xs:integer"/>
130<   </xs:sequence>
131<   <xs:attribute name="JegyID" type="xs:unsignedByte" use="required"/>
132< </xs:complexType>
133
134< xs:complexType name="vasarlas_type">
135<   <xs:sequence>
136<     <xs:element name="vidopont" type="vasarlasidopontType"/>
137<   </xs:sequence>
138<   <xs:attribute name="VasarlasID" type="xs:unsignedByte" use="required"/>
139<   <xs:attribute name="Jegy_IDREF" type="xs:unsignedByte" use="required"/>
140<   <xs:attribute name="Elado_IDREF" type="xs:unsignedByte" use="required"/>
141< </xs:complexType>
142
143< xs:complexType name="cim_type">
144<   <xs:sequence>
145<     <xs:element name="irszam" type="xs:integer"/>
146<     <xs:element name="utca" type="xs:string"/>
147<     <xs:element name="hazszam" type="xs:string"/>
148<   </xs:sequence>
149< </xs:complexType>
150
151< xs:simpleType name="vasarlasidopontType">
152<   <xs:restriction base="xs:string">
153<     <xs:pattern value="[0-9][0-9][0-9][0-9].[0-9][0-9].[0-9][0-9]/[0-9][0-9]*:[0-9][0-9]"/>
154<   </xs:restriction>
155< </xs:simpleType>
156
157</xs:schema>

```

# DOM

## Adatolvasás:

```
1 package hu.dompars.vuxfks;
2
3 import java.io.File;
4 import java.io.IOException;
5
6 import javax.xml.parsers.DocumentBuilder;
7 import javax.xml.parsers.DocumentBuilderFactory;
8 import javax.xml.parsers.ParserConfigurationException;
9
10 import org.xml.sax.SAXException;
11
12 import org.w3c.dom.Document;
13 import org.w3c.dom.NodeList;
14 import org.w3c.dom.Node;
15 import org.w3c.dom.Element;
16
17 public class DomReadVUXFKS {
18
19     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException {
20
21         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
22         DocumentBuilder dBuilder = factory.newDocumentBuilder();
23
24         File xmlFile = new File("XMLVUXFKS.xml");
25         Document doc = dBuilder.parse(xmlFile);
26
27         doc.getDocumentElement().normalize();
28
29         System.out.println("Root elem: " + doc.getDocumentElement().getNodeName());
30
31         NodeList nList = doc.getElementsByTagName("mozi");
32
33     }
34 }
```

```
35
36     for (int i = 0; i < nList.getLength(); i++) {
37
38
39         Node nNode = nList.item(i);
40         System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
41
42         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
43             Element elem = (Element) nNode;
44
45
46             String MoziID = elem.getAttribute("MoziID");
47
48
49             Node n1 = elem.getElementsByTagName("mozinev").item(0);
50             String mozinev = n1.getTextContent();
51
52             Node n2 = elem.getElementsByTagName("irszam").item(0);
53             String irszam = n2.getTextContent();
54
55             Node n3 = elem.getElementsByTagName("utca").item(0);
56             String utca = n3.getTextContent();
57
58             Node n4 = elem.getElementsByTagName("hazszam").item(0);
59             String hazszam = n4.getTextContent();
60
61             System.out.println("Mozi azonosito: " + MoziID);
62             System.out.println("Mozinév: " + mozinev);
63             System.out.println("Irányítószám: " + irszam);
64             System.out.println("Utca: " + utca);
65             System.out.println("Házszám: " + hazszam);
66         }
67     }
68
```

```
68
69     nList = doc.getElementsByTagName("dolgozik");
70     for (int i = 0; i < nList.getLength(); i++) {
71
72         Node nNode = nList.item(i);
73         System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
74
75         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
76             Element elem = (Element) nNode;
77
78             String Mozi_IDREF = elem.getAttribute("Mozi_IDREF");
79             String Elado_IDREF = elem.getAttribute("Elado_IDREF");
80
81             Node n1 = elem.getElementsByTagName("elerheto").item(0);
82             String elerheto = n1.getTextContent();
83
84             System.out.println("Mozi azonosito: " + Mozi_IDREF);
85             System.out.println("Dolgozik: " + elerheto);
86             System.out.println("Eladó azonosítója: " + Elado_IDREF);
87
88         }
89     }
90
91     nList = doc.getElementsByTagName("elado");
92     for (int i = 0; i < nList.getLength(); i++) {
93
94         Node nNode = nList.item(i);
95         System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
96
97         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
98             Element elem = (Element) nNode;
99
100             String EladoID = elem.getAttribute("EladoID");
101
102             Node n1 = elem.getElementsByTagName("fhsznev").item(0);
103             String fhsznev = n1.getTextContent();
104             Node n2 = elem.getElementsByTagName("jelszo").item(0);
105             String jelszo = n2.getTextContent();
106             Node n3 = elem.getElementsByTagName("admin").item(0);
107             String admin = n3.getTextContent();
108
109             System.out.println("Elado azonosító: " + EladoID);
110             System.out.println("Felhasználónév: " + fhsznev);
111             System.out.println("Jelszó: " + jelszo);
112             System.out.println("Adminjog: " + admin);
113         }
114     }
115
```

```

115
116     nList = doc.getElementsByTagName("jegy");
117     for (int i = 0; i < nList.getLength(); i++) {
118
119         Node nNode = nList.item(i);
120         System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
121
122         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
123             Element elem = (Element) nNode;
124
125             String JegyID = elem.getAttribute("JegyID");
126
127
128             Node n1 = elem.getElementsByTagName("vetites").item(0);
129             String vetites = n1.getTextContent();
130
131             Node n2 = elem.getElementsByTagName("sor").item(0);
132             String sor = n2.getTextContent();
133
134             Node n3 = elem.getElementsByTagName("oszlop").item(0);
135             String oszlop = n3.getTextContent();
136
137             System.out.println("Jegy azonosítója: " + JegyID);
138             System.out.println("Vetítés: " + vetites);
139             System.out.println("Sor: " + sor);
140             System.out.println("Oszlop: " + oszlop);
141         }
142     }
143
144     nList = doc.getElementsByTagName("vasarlas");
145     for (int i = 0; i < nList.getLength(); i++) {
146
147         Node nNode = nList.item(i);
148         System.out.println("\nKiválasztott elem: " + nNode.getNodeName());
149
150         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
151             Element elem = (Element) nNode;
152
153             String VasarlasID = elem.getAttribute("VasarlasID");
154             String Jegy_IDREF = elem.getAttribute("Jegy_IDREF");
155             String Elado_IDREF = elem.getAttribute("Elado_IDREF");
156
157             Node n1 = elem.getElementsByTagName("vidopont").item(0);
158             String vidopont = n1.getTextContent();
159
160             System.out.println("Vásárlási azonosító: " + VasarlasID);
161             System.out.println("Vásárlás időpontja: " + vidopont);
162             System.out.println("Jegy azonosítója: " + Jegy_IDREF);
163             System.out.println("Eladó azonosítója: " + Elado_IDREF);
164
165         }
166     }
167 }
168 }

```

Kiválasztott elem: mozi  
Mozi azonosító: 01  
Mozi név: CinemaMiskolc  
Irányítószám: 3500  
Utca: Rákóczi  
Házszám: 15.

Kiválasztott elem: mozi  
Mozi azonosító: 02  
Mozi név: CinemaBudapest  
Irányítószám: 1100  
Utca: Fő  
Házszám: 4.

Kiválasztott elem: mozi  
Mozi azonosító: 03  
Mozi név: CinemaSalgótarján  
Irányítószám: 3100  
Utca: Kiss  
Házszám: 88.

Kiválasztott elem: dolgozik  
Mozi azonosító: 01  
Dolgozik: 0  
Eladó azonosítója: 01

Kiválasztott elem: dolgozik  
Mozi azonosító: 02  
Dolgozik: 1  
Eladó azonosítója: 02

Kiválasztott elem: dolgozik  
Mozi azonosító: 03  
Dolgozik: 1  
Eladó azonosítója: 03

## Adatlekérdezések:

```
1 package hu.dompars.vuxfks;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.text.ParseException;
6
7 import javax.xml.parsers.DocumentBuilder;
8 import javax.xml.parsers.DocumentBuilderFactory;
9 import javax.xml.parsers.ParserConfigurationException;
10
11 import org.w3c.dom.Document;
12 import org.w3c.dom.Element;
13 import org.w3c.dom.Node;
14 import org.w3c.dom.NodeList;
15 import org.xml.sax.SAXException;
16
17 public class DomQueryVUXFKS {
18
19     public static void main(String[] args) throws ParserConfigurationException, SAXException, IOException, ParseException {
20
21         DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
22         DocumentBuilder dBuilder = factory.newDocumentBuilder();
23
24         File xmlFile = new File("XMLVUXFKS.xml");
25         Document doc = dBuilder.parse(xmlFile);
26
27         doc.getDocumentElement().normalize();
28
29         System.out.println("Root element: " + doc.getDocumentElement().getNodeName());
30
31         System.out.println("\nAz éppen dolgozó eladók adatai:");
32         NodeList eladolist = doc.getElementsByTagName("elado");
33         NodeList dolgoziklist = doc.getElementsByTagName("dolgozik");
34     }
35 }
```

```

35
36     for (int i = 0; i < eladolist.getLength(); i++) {
37
38         Node nNode = eladolist.item(i);
39
40         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
41             Element elem = (Element) nNode;
42
43             String EladoID = elem.getAttribute("EladoID");
44
45             Node n1 = elem.getElementsByTagName("fhsznev").item(0);
46             String fhsznev = n1.getTextContent();
47
48             Node n2 = elem.getElementsByTagName("jelszo").item(0);
49             String jelszo = n2.getTextContent();
50
51             Node n3 = elem.getElementsByTagName("admin").item(0);
52             String admin = n3.getTextContent();
53
54             for (int j = 0; j < dolgoziklist.getLength(); j++) {
55
56                 Node nNode2 = dolgoziklist.item(j);
57
58                 if (nNode2.getNodeType() == Node.ELEMENT_NODE) {
59                     Element elem2 = (Element) nNode2;
60
61                     String Elado_IDREF = elem2.getAttribute("Elado_IDREF");
62
63                     Node n4 = elem2.getElementsByTagName("elerheto").item(0);
64                     String elerheto = n4.getTextContent();
65
66                     if(EladoID.equals(Elado_IDREF) && elerheto.toString().contains("1")) {
67
68                         System.out.println("Eladó azonosítója: " + EladoID);
69                         System.out.println("Felhasználónév: " + fhsznev);
70                         System.out.println("Jelszó: " + jelszo);
71                         System.out.println("Adminjog: " + admin);
72                     }
73                 }
74             }
75         }
76     }
77 }
78
79 System.out.println("\nAz első vetítésre szóló jegyek részletei:");
80 NodeList jegylist = doc.getElementsByTagName("jegy");
81

```



```

78
79 System.out.println("\nAz első vetítésre szóló jegyek részletei:");
80 NodeList jegylist = doc.getElementsByTagName("jegy");
81
82 for (int i = 0; i < jegylist.getLength(); i++) {
83
84     Node nNode = jegylist.item(i);
85
86     if (nNode.getNodeType() == Node.ELEMENT_NODE) {
87         Element elem = (Element) nNode;
88
89         String JegyID = elem.getAttribute("JegyID");
90
91         Node n1 = elem.getElementsByTagName("vetites").item(0);
92         String vetites = n1.getTextContent();
93
94         Node n2 = elem.getElementsByTagName("sor").item(0);
95         String sor = n2.getTextContent();
96
97         Node n3 = elem.getElementsByTagName("oszlop").item(0);
98         String oszlop = n3.getTextContent();
99
100        if (vetites.toString().contains("01")) {
101
102            System.out.println("Jegy azonosítója: " + JegyID);
103            System.out.println("sor: " + sor);
104            System.out.println("oszlop: " + oszlop);
105
106        }
107    }
108
109 }
110
111
112 System.out.println("\nAz eladott jegyek adatai, a jegyet eladó felhasználóneve és a mozi neve ahol történt az eladás:");
113 NodeList mozilist = doc.getElementsByTagName("mozi");
114 NodeList vasarlaslist = doc.getElementsByTagName("vasarlas");
115

```

```
115
116     for (int i = 0; i < vasarlaslist.getLength(); i++) {
117
118         Node nNode = vasarlaslist.item(i);
119
120         if (nNode.getNodeType() == Node.ELEMENT_NODE) {
121             Element elem = (Element) nNode;
122
123             String VasarlasID = elem.getAttribute("VasarlasID");
124
125             String Jegy_IDREF = elem.getAttribute("Jegy_IDREF");
126
127             String Elado_IDREF = elem.getAttribute("Elado_IDREF");
128
129             Node n1 = elem.getElementsByTagName("vidopont").item(0);
130             String vidopont = n1.getTextContent();
131
132
133             for (int j = 0; j < eladolist.getLength(); j++) {
134
135                 Node nNode2 = eladolist.item(j);
136
137                 if (nNode2.getNodeType() == Node.ELEMENT_NODE) {
138                     Element elem2 = (Element) nNode2;
139
140                     String EladoID = elem2.getAttribute("EladoID");
141
142                     Node n2 = elem2.getElementsByTagName("fhsznev").item(0);
143                     String fhsznev = n2.getTextContent();
144
145                     if(EladoID.equals(Elado_IDREF)) {
146
147                         for (int k = 0; k < dolgoziklist.getLength(); k++) {
148
149                             Node nNode3 = dolgoziklist.item(k);
150
151
```

```

152
153
154         if (nNode3.getNodeType() == Node.ELEMENT_NODE) {
155
156             Element elem3 = (Element) nNode3;
157
158             String Mozi_IDREF = elem3.getAttribute("Mozi_IDREF");
159
160             String DElado_IDREF = elem3.getAttribute("Elado_IDREF");
161
162             if(Elado_IDREF.equals(DElado_IDREF)) {
163
164
165                 for (int l = 0; l < mozilist.getLength(); l++) {
166
167                     Node nNode4 = mozilist.item(l);
168
169                     if (nNode4.getNodeType() == Node.ELEMENT_NODE) {
170
171                         Element elem4 = (Element) nNode4;
172
173                         String MoziID = elem4.getAttribute("MoziID");
174
175                         Node n3 = elem4.getElementsByTagName("mozinev").item(0);
176                         String mozinev = n3.getTextContent();
177
178                         if(Mozi_IDREF.equals(MoziID)) {
179
180                             System.out.println("Vásárlási azonosító: " + VasarlasID);
181                             System.out.println("Jegy azonosítója: " + Jegy_IDREF);
182                             System.out.println("Vásárlás időpontja: " + vidopont);
183                             System.out.println("Eladó felhasználóneve: " + fhsznev);
184                             System.out.println("Mozi neve: " + mozinev);
185                         }
186                     }
187                 }
188             }
189         }
190     }
191 }
192
193 }
194
195 }
196
197 }
198
199 }
200
201 }
202 }

```

Az éppen dolgozó eladók adatai:

Eladó azonosítója: 02

Felhasználónév: Anna01

Jelszó: ujjelszo2

Adminjog: 0

Eladó azonosítója: 03

Felhasználónév: Judit08

Jelszó: ujjelszo3

Adminjog: 0

Az első vetítésre szóló jegyek részletei:

Jegy azonosítója: 01

sor: 10

oszlop: 1

Jegy azonosítója: 02

sor: 10

oszlop: 2

Jegy azonosítója: 03

sor: 5

oszlop: 4

Jegy azonosítója: 04

sor: 5

oszlop: 5

Az eladott jegyek adatai, a jegyet eladó felhasználóneve és a mozi neve ahol történt az eladás:

Vásárlási azonosító: 01

Jegy azonosítója: 05

Vásárlás időpontja: 2021.12.01/10:00

Eladó felhasználóneve: Anna01

Mozi neve: CinemaBudapest

Vásárlási azonosító: 02

Jegy azonosítója: 03

Vásárlás időpontja: 2021.12.01/10:08

Eladó felhasználóneve: Judit08

Mozi neve: CinemaSalgótarján

## Adatmódosítás:

```
2
3 import java.io.File;
4 import javax.xml.parsers.DocumentBuilder;
5 import javax.xml.parsers.DocumentBuilderFactory;
6 import javax.xml.transform.Transformer;
7 import javax.xml.transform.TransformerFactory;
8 import javax.xml.transform.dom.DOMSource;
9 import javax.xml.transform.stream.StreamResult;
10
11 import org.w3c.dom.Document;
12 import org.w3c.dom.Element;
13 import org.w3c.dom.Node;
14 import org.w3c.dom.NodeList;
15
16
17 import javax.xml.transform.OutputKeys;
18
19 import javax.xml.transform.TransformerException;
20
21
22 public class DomModifyVUXFKS {
23
24     public static void main(String[] args) {
25
26         try {
27             File inputFile = new File("XMLVUXFKS.xml");
28             DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
29             DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
30             Document doc = docBuilder.parse(inputFile);
31             Node root = doc.getFirstChild();
32
```

```

33 // Módosítjuk az összes eladó jelszavát "ujjelszo0"-ra annak aki nem rendelkezik admin joggal
34 NodeList eladoList = doc.getElementsByTagName("elado");
35 for (int i = 0; i < eladoList.getLength(); i++) {
36
37     Node nNode1 = eladoList.item(i);
38
39     if (nNode1.getNodeType() == Node.ELEMENT_NODE) {
40         Element elem = (Element) nNode1;
41
42         Node n2 = elem.getElementsByTagName("admin").item(0);
43         String admin = n2.getTextContent();
44
45         if (admin.equals("0")) {
46
47             NodeList childNodes = nNode1.getChildNodes();
48             for (int j = 0; j < childNodes.getLength(); j++) {
49                 Node childNode = childNodes.item(j);
50                 if (childNode.getNodeName().equals("jelszo")) {
51                     childNode.setTextContent("ujjelszo0");
52
53                 }
54             }
55         }
56     }
57 }
58
59 }
60
61
62 NodeList dolgoznak = root.getChildNodes();
63 for (int i = 0; i < dolgoznak.getLength(); i++) {
64
65     Node n1 = dolgoznak.item(i);
66
67     if (n1.getNodeName().equals("dolgoznak"))
68         root.removeChild(n1);
69 }
70
71
72
73 modify(doc);
74
75 } catch (Exception e) {
76     e.printStackTrace();
77 }
78

```

```

71 private static void modify(Document doc) throws TransformerException {
72     TransformerFactory transformerFactory = TransformerFactory.newInstance();
73     Transformer transformer = transformerFactory.newTransformer();
74     System.out.println("-Modified File-");
75     transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
76     transformer.setOutputProperty(OutputKeys.INDENT, "yes");
77     transformer.setOutputProperty("{http://xml.apache.org/xslt}indent-amunt", "2");
78
79     DOMSource source = new DOMSource(doc);
80
81     StreamResult console = new StreamResult(System.out);
82
83     transformer.transform(source, console);
84
85 }
86 }

```

&lt;eladok&gt;

```
<elado EladoID="01">
```

<fhsznev>BÄ01a05</fhsznev>

&lt;jelszo&gt;ujjelszo0&lt;/jelszo&gt;

```
<admin>0</admin>
```

&lt;/elado&gt;

```
<elado EladoID="02">
```

<fhsznev>Anna01</fhsznev>

```
<jelszo>ujjelszo0</jelszo>
```

```
<admin>0</admin>
```

&lt;/elado&gt;

```
<elado EladoID="03">
```

&lt;fhsznev&gt;Judit08&lt;/fhsznev&gt;

```
<jelszo>ujjelszo0</jelszo>
```

```
<admin>0</admin>
```

&lt;/elado&gt;

```
<elado EladoID="04">
```

<fhsznev>PÄ@ter04</fhsznev>

```
<jelszo>ujjelszo4</jelszo>
```

```
<admin>1</admin>
```

&lt;/elado&gt;