

JEGYZŐKÖNYV

Adatkezelés XML környezetben
Féléves feladat

Készítette: **Baumel Márton Benedek**
Neptunkód: **O09CTQ**

1, Feladat

A feladat leírása:

Az általam kidolgozásra választott feladat egy olyan képzeletbeli iroda adatbázisának a megtervezése ahol a részleg adatait, a dolgozó információit, a dolgozó számítógépekhez tartozó engedélyeit és az irodai eszközöket is tudjuk tárolni egy raktárba.

Különböző részlegek (**Department**) léteznek amik a világ bármely pontján elhelyezkedhetnek. Egy alkalmazott (**Employee**) csak egy részlegen dolgozhat ezért az alkalmazottnál rögzíteni kell a részleg azonosítóját.

Részlegnek van egy raktár (**Warehouse**) ahol is az ott lévő irodai eszközöket tudjuk tárolni, a részleg és a raktár hely információja különbözhet. A raktáron belül több fajta irodai eszközt is tudunk tárolni. A raktár azonosítója viszont a részleg azonosítójával megegyezik ezért egy részleghez csak egy raktár tartozik.

Az alkalmazotaknak lehet hozzáférése adott számítógépekhez, amit az engedélyeknél (**Permission**) lesz tárolva, annak érdekében, hogy egy engedély fajtából többet lehessen kiadni több embernek. Egy engedély csak egy számítógépre lehet specifikusan megadva.

Repository elérése:

https://github.com/marcibaume/O09CTQ_XMLGyak/tree/main/XMLTaskO09CTQ

Adatbázis Egyedei:

Employee, Department, Permission, Warehouse

Eyedekből létrejött táblák:

Employee

<i>EmployeeId</i>	-Dolgozó Azonosítója (3 darab számnak kell lennie)
Name	-Dolgozó neve
Phone	-Dolgozó telefonszáma
Email	-Dolgozó email címe
DepartmentId_FK	-A részleg azonosítója (idegen kulcs)

Department

<i>DepartmentId</i>	-A részleg azonosítója (2 darab számnak kell lennie)
Country	-A részleg ország helye
Postcode	-A részleg irányítószáma
City	-A részleg város hely
Street	-A részleg utca címe
Email	-A részleg email címe
Website	-A részleg email címe

Permission

<i>PermissionId</i>	-Engedély azonosító
Role	-Engedély szerepkör
Modified	-Az engedély mikor volt utoljára módosítva
Computer	-Az engedély melyik gépre szól

Warehouse

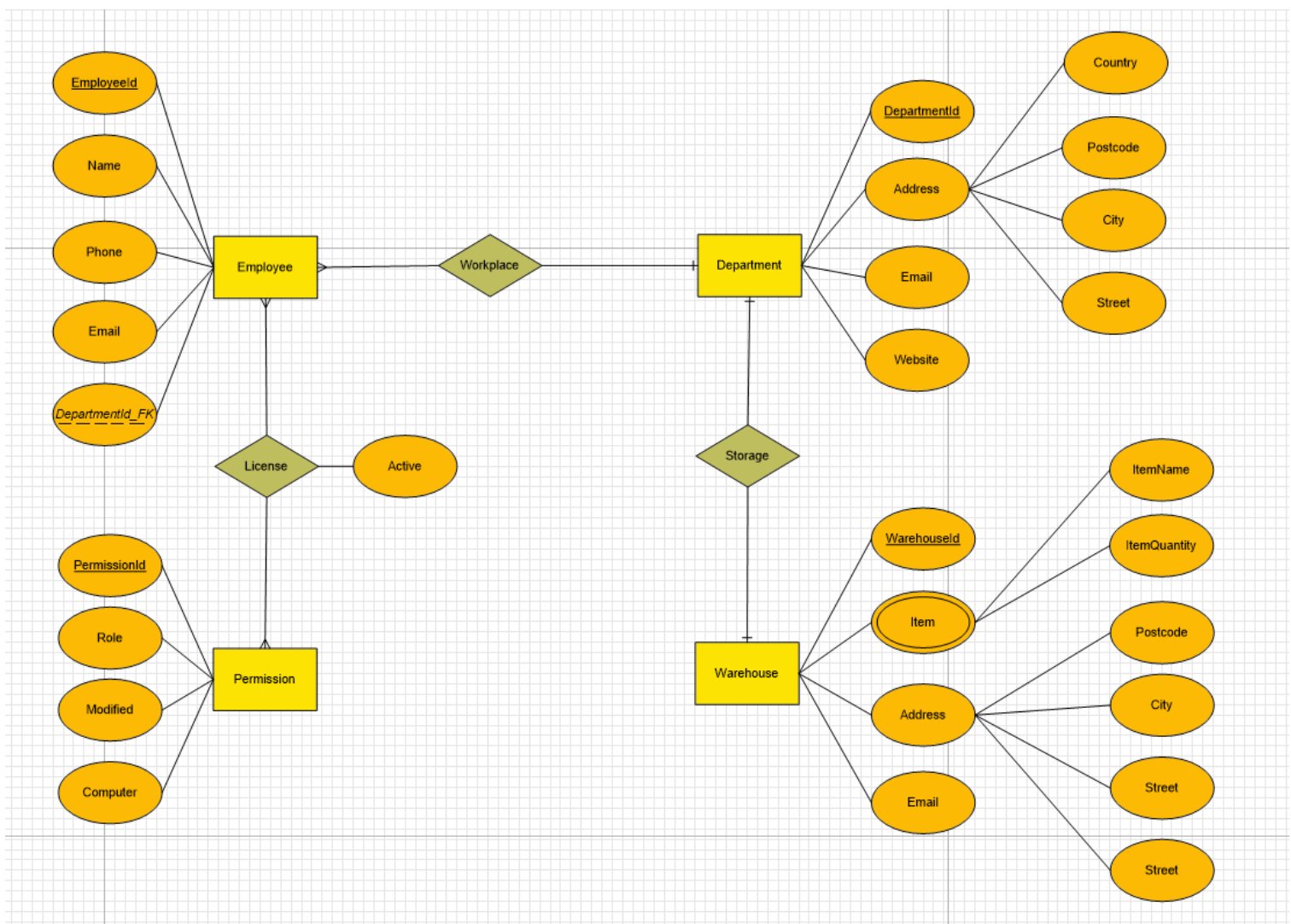
<i>WarehouseId</i>	-A raktár azonosítója
ItemName	-A termék neve
ItemQuantity	-A termék mennyisége
Country	-A raktár ország helye
Postcode	-A raktár irányítószáma
City	-A raktár város hely
Street	-A raktár utca címe
Email	-A raktár email címe

Egyedek között kapcsolatból létrejött tábla:

License

EmployeeId_FK	-Dolgozó Azonosítója (idegen kulcs)
PermissionId_FK	-Engedély azonosító (idegen kulcs)
Active	-Jelenleg aktív-e a kapott engedély (boolean)

Az adatbázis ER modell:



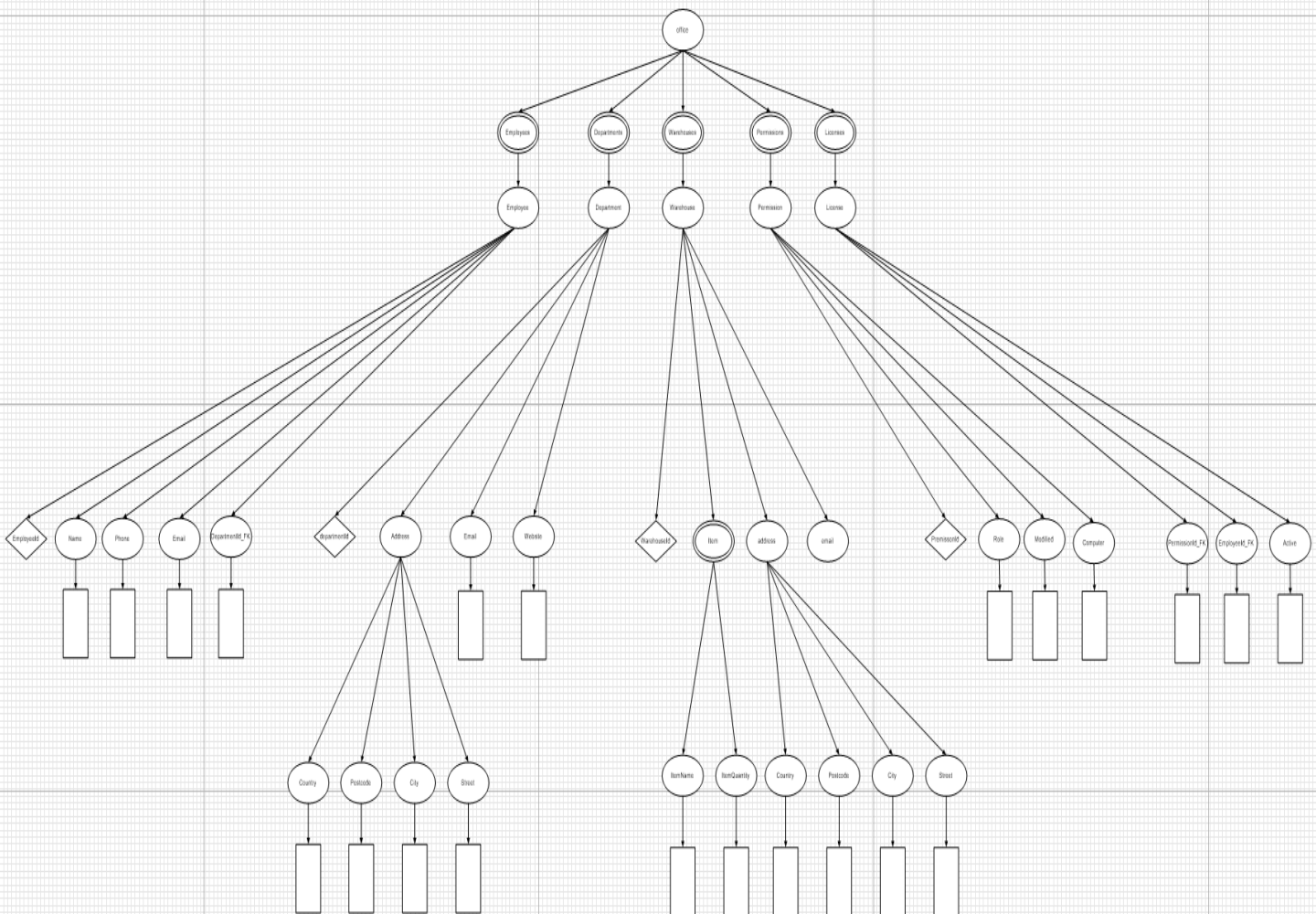
ERO09CTQ.PNG: [Dokumentum elérése](#)

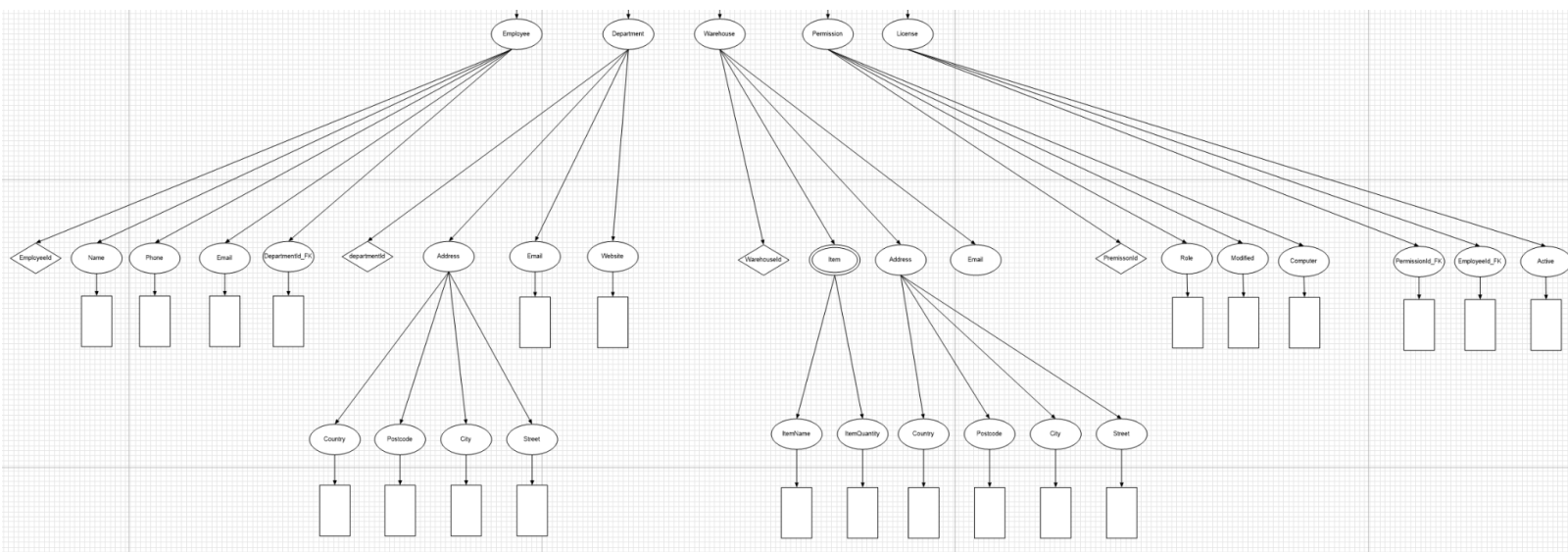
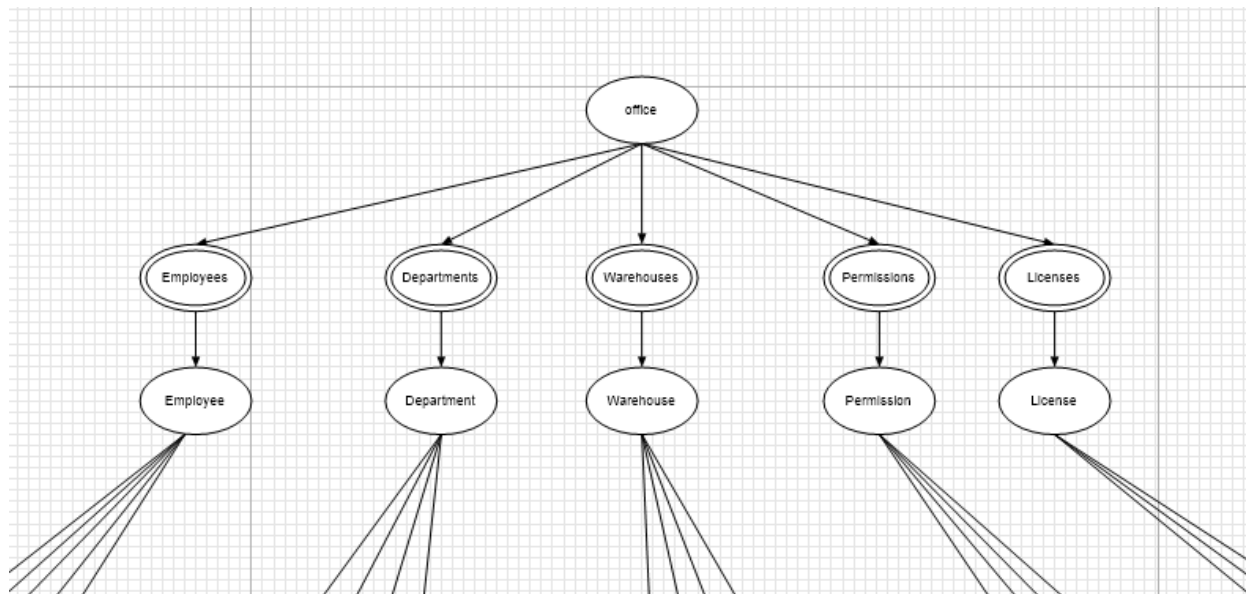
Az adatbázis konvertálása XDM modellre:

Annak érdekében, hogy XML-ben tudjam tárolni az adataimat a különböző egyedeket belehelyeztem egy olyan XML egyedbe amik több hasonló elemet tudnak el tárolni. A név konvenció az lett, hogy az egyed nevének többes számával lett jelölve. Ez a fajta jelölés látszik az XDM modellelmen.

Emellett létrejön az egyedek között kapcsolatból létrejövő egyed, a License. Erre azért van szükség mivel az Employee és a Permission között több-több kapcsolat van ezért muszáj az elemeket külön tárolni az átláthatóság érdekében, emellett magkapja az Active tulajdonságot amivel azt tudjuk megnézni, hogy az adott alkalmazott rendelkezik-e még a megadott engedéllyel vagy már nem.

Emellett minden adat az Office root-ban fog elhelyezkedni





XDMO09CTQ.PNG: [Dokumentum elérése](#)

Az XDM modell alapján XML dokumentum:

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

<Office xsi:noNamespaceSchemaLocation="XMLSchema009CTQ.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <Employees>
    <Employee EmployeeId="100">
      <Name>Kis Béla</Name>
      <Phone>06308881983</Phone>
      <Email>bela@email.com</Email>
      <DepartmentId_FK>01</DepartmentId_FK>
    </Employee>
    <Employee EmployeeId="101">
      <Name>Nagy Ernő</Name>
      <Phone>06308881984</Phone>
      <Email>erno@email.com</Email>
      <DepartmentId_FK>01</DepartmentId_FK>
    </Employee>
    <Employee EmployeeId="102">
      <Name>Baumel Márton</Name>
      <Phone>06308881985</Phone>
      <Email>marton@email.com</Email>
      <DepartmentId_FK>02</DepartmentId_FK>
    </Employee>
  </Employees>

  <Departments>
    <Department DepartmentId="01">
      <Address>
        <Country>Hungary</Country>
```

```

        <Postcode>3900</Postcode>
        <City>Szerencs</City>
        <Street>Hársfa utca 14</Street>
    </Address>
    <Email>marcibaumel@gmail.com</Email>
    <Website>szerencswebsite.hu</Website>
</Department>

<Department DepartmentId="02">
    <Address>
        <Country>Hungary</Country>
        <Postcode>3515</Postcode>
        <City>Miskolc</City>
        <Street>Egyetemi út 1</Street>
    </Address>
    <Email>miskolc@nememail.com</Email>
    <Website>miskolciceg.hu</Website>
</Department>
</Departments>

<Warehouses>
    <Warehouse WarehouseId="01">
        <Item>
            <ItemName>Paper</ItemName>
            <ItemQuantity>100</ItemQuantity>
        </Item>
        <Item>
            <ItemName>Scissors</ItemName>
            <ItemQuantity>200</ItemQuantity>
        </Item>
        <Address>
            <Country>Hungary</Country>

```



```

        <Postcode>3515</Postcode>
        <City>Miskolc</City>
        <Street>Egyetemi út 2</Street>
    </Address>
    <Email>warehousemiskolc@email.com</Email>
</Warehouse>

<Warehouse WarehouseId="02">
    <Item>
        <ItemName>Paper</ItemName>
        <ItemQuantity>100</ItemQuantity>
    </Item>
    <Address>
        <Country>Hungary</Country>
        <Postcode>3900</Postcode>
        <City>Szerencs</City>
        <Street>Hársfa utca 14</Street>
    </Address>
    <Email>warehouseszerencs@email.com</Email>
</Warehouse>
</Warehouses>

<Permissions>

    <Permission PermissionId="1">
        <Role>admin</Role>
        <Modified>2021-06-12</Modified>
        <Computer>01</Computer>
    </Permission>

```

```

    <Permission PermissionId="2">
        <Role>user</Role>
        <Modified>2021-06-12</Modified>
        <Computer>03</Computer>
    </Permission>
</Permissions>

<Licenses>
    <License>
        <EmployeeId_FK>100</EmployeeId_FK>
        <PermissionId_FK>1</PermissionId_FK>
        <Active>true</Active>
    </License>
    <License>
        <EmployeeId_FK>101</EmployeeId_FK>
        <PermissionId_FK>2</PermissionId_FK>
        <Active>false</Active>
    </License>
    <License>
        <EmployeeId_FK>102</EmployeeId_FK>
        <PermissionId_FK>1</PermissionId_FK>
        <Active>false</Active>
    </License>
</Licenses>
</Office>

```

XMLO09CTQ.xml: [Dokumentum elérése](#)

Az XML dokumentum alapján XMLSchema

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

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="qualified">
  <xs:element name="Office">
    <xs:complexType>
      <xs:sequence>

        <xs:element name="Employees">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Employee"
type="employeeType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Departments">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Department"
type="departmentType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Warehouses">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Warehouse"
type="warehouseType" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>

        <xs:element name="Permissions">
          <xs:complexType>
```

```

        <xs:sequence>
            <xs:element name="Permission"
type="permissionType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

    <xs:element name="Licenses">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="License" type="licenseType"
minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

</xs:sequence>
</xs:complexType>

<xs:key name="EmployeeId_FK">
    <xs:selector xpath="Employees/Employee"/>
    <xs:field xpath="@EmployeeId"/>
</xs:key>

<xs:key name="DepartmentId_FK">
    <xs:selector xpath="Departments/Department"/>
    <xs:field xpath="@DepartmentId"/>
</xs:key>

<xs:key name="PermissionId_FK">
    <xs:selector xpath="Permissions/Permission"/>
    <xs:field xpath="@PermissionId"/>
</xs:key>

<xs:keyref name="EmployeeId_FK-License" refer="EmployeeId_FK">
    <xs:selector xpath="Licenses/License/EmployeeId_FK"/>
    <xs:field xpath="."/>
</xs:keyref>

```

```

        <xs:keyref name="PermissionId_FK-License" refer="PermissionId_FK">
            <xs:selector xpath="Licenses/License/PermissionId_FK"/>
            <xs:field xpath="."/>
        </xs:keyref>

        <xs:keyref name="DepartmentId_FK-Employee"
refer="DepartmentId_FK">
            <xs:selector xpath="Employees/Employee/DepartmentId_FK"/>
            <xs:field xpath="."/>
        </xs:keyref>

    </xs:element>

    <xs:complexType name="addressType">
        <xs:sequence>
            <xs:element name="Country" type="xs:string"/>
            <xs:element name="Postcode" type="xs:string"/>
            <xs:element name="City" type="xs:string"/>
            <xs:element name="Street" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="employeeType">
        <xs:sequence>
            <xs:element name="Name" type="xs:string"/>
            <xs:element name="Phone" type="xs:string"/>
            <xs:element name="Email" type="xs:string"/>
            <xs:element name="DepartmentId_FK" type="xs:integer"/>

        </xs:sequence>
        <xs:attribute name="EmployeeId" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:pattern value="[0-9][0-9][0-9]"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

```

```

<xs:complexType name="departmentType">
  <xs:sequence>
    <xs:element name="Address" type="addressType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="Email" type="xs:string"/>
    <xs:element name="Website" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="DepartmentId" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:pattern value="[0-9][0-9]"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="warehouseType">
  <xs:sequence>
    <xs:element name="Item" type="itemType" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="Address" type="addressType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="Email" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="WarehouseId" type="xs:integer"
use="required"/>
</xs:complexType>

<xs:complexType name="itemType">
  <xs:sequence>
    <xs:element name="ItemName" type="xs:string" maxOccurs="1"/>
    <xs:element name="ItemQuantity" type="xs:integer"
maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="permissionType">
  <xs:sequence>

```

```

        <xs:element name="Role" type="xs:string"/>
        <xs:element name="Modified" type="xs:date"/>
        <xs:element name="Computer" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="PermissionId" type="xs:integer"
use="required"/>
</xs:complexType>

<xs:complexType name="licenseType">
    <xs:sequence>
        <xs:element name="EmployeeId_FK" type="xs:integer"/>
        <xs:element name="PermissionId_FK" type="xs:integer"/>
        <xs:element name="Active" type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>
</xs:schema>

```

XMLSchemaO09CTQ.xsd: [Dokumentum elérése](#)

2. Feladat

2a, Adatolvasás

```
package hu.domparse.O09CTQ;

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 DOMReadO09CTQ {
    private static int baseLineRule = 0;
    private final static String XMLFILE = "XMLO09CTQ.xml";
    private static StringBuilder result;

    public static void main(String[] args) {
        /*
        Minden elem beolvasása és kiírása
        */

        System.out.println("/*****");

        System.out.println("/*****");
        System.out.println("XML File Content:");

        System.out.println("/*****");

        System.out.println("/*****");
        System.out.println("\n\n");
        readOutAllElements();
    }

    //Eredmény kiírása
```



```

private static void resultToConsole() {
    System.out.println(result.toString());
}

//Minden elem beolvasása
private static void readOutAllElements() {
    Document doc;
    DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
    result = new StringBuilder();
    try {
        DocumentBuilder db = dbf.newDocumentBuilder();
        doc = db.parse(XMLFILE);

        Node er = doc.getFirstChild();
        NodeList erList = er.getChildNodes();
        getAllElements(erList);
    } catch (ParserConfigurationException | SAXException |
IOException e) {
        e.printStackTrace();
    }

    resultToConsole();
}

//Elemek összefűzés
private static void getAllElements(NodeList nl) {
    Node node;
    for (int i = 0; i < nl.getLength(); i++) {
        node = nl.item(i);
        if (node.getNodeType() == Node.ELEMENT_NODE) {

            result.append(indent(3)+node.getNodeName());

            if (node.hasAttributes()) {
                result.append("
"+getAttributes(node.getAttributes())+"
"); //
            } else {
                result.append("
");
            }
            if (node.getChildNodes().getLength() != 1) {

                NodeList childList = node.getChildNodes();
                baseLineRule++;
                getAllElements(childList);
            }
        }
    }
}

```

```

        } else {
            baseLineRule++;

result.append(indent(3)+node.getTextContent()+"\n");
            baseLineRule--;
        }
    }
}

baseLineRule--;
}

//Elemek elválasztása
private static String indent(int multiplyBy) {
    StringBuilder whiteSpaces = new StringBuilder();
    for (int i = 0; i < (baseLineRule *multiplyBy); i++) {
        //3 space-nyi hely
        whiteSpaces.append(" ");
    }
    return whiteSpaces.toString();
}

//Elemek tulajdonságának a neve
private static String getAttributes(NamedNodeMap node) {
    StringBuilder sb = new StringBuilder(" Attributes: [");
    for (int i = 0; i < node.getLength(); i++) {
        if(i != node.getLength()-1) {
            //Elem neve
            sb.append(node.item(i).getNodeName()+" :
"+node.item(i).getNodeValue()+" ; ");
        } else {
            //Elem
            sb.append(node.item(i).getNodeName()+" :
"+node.item(i).getNodeValue());
        }
    }
    sb.append("]");
    return sb.toString();
}
}

```

Output:

```
"C:\Users\Baumel Márton\.jdk\openjdk-16.0.2\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=60456:C:\Program Files\JetBrains\IntelliJ IDEA
2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath
C:\REPOS\O09TQ_XMLGyak\XMLTaskO09CTQ\DOMParseO09CTQ\target\classes
hu.domparse.O09CTQ.DOMReadO09CTQ
```

```
/******
```

```
/******
```

XML File Content:

```
/******
```

```
/******
```

Employees

Employee Attributes: [EmployeeId : 100]

Name

Kis Béla

Phone

06308881983

Email

bela@email.com

DepartmentId_FK

01

Employee Attributes: [EmployeeId : 101]

Name

Nagy Ernő

Phone

06308881984

Email

erno@email.com

DepartmentId_FK

01

Employee Attributes: [EmployeeId : 102]

Name

Baumel Márton

Phone

06308881985

Email

marton@email.com

DepartmentId_FK

02

Departments

Department Attributes: [DepartmentId : 01]

Address

Country

Hungary
Postcode
3900
City
Szerencs
Street
Hársfa utca 14
Email
marcibaumel@gmail.com
Website
szerencswebsite.hu

Department Attributes: [DepartmentId : 02]

Address
Country
Hungary
Postcode
3515
City
Miskolc
Street
Egyetemi út 1

Email
miskolc@nememail.com
Website
miskolciceg.hu

Warehouses

Warehouse Attributes: [WarehouseId : 01]

Item
ItemName
Paper
ItemQuantity
100

Item
ItemName
scissors
ItemQuantity
200

Address
Country
Hungary
Postcode
3515
City
Miskolc
Street
Egyetemi út 2

Email
warehousemiskolc@email.com

Warehouse Attributes: [WarehouseId : 02]

Item
 ItemName
 Paper
 ItemQuantity
 100
Address
 Country
 Hungary
 Postcode
 3900
 City
 Szerencs
 Street
 Hársfa utca 14
Email
 warehouseszerencs@email.com
Permissions
 Permission Attributes: [PermissionId : 1]
 Role
 admin
 Modified
 2021-06-12
 Computer
 01
 Permission Attributes: [PermissionId : 2]
 Role
 user
 Modified
 2021-06-12
 Computer
 03
Licenses
 License
 EmployeeId_FK
 100
 PermissionId_FK
 1
 Active
 true
 License
 EmployeeId_FK
 101
 PermissionId_FK
 2
 Active
 false
 License
 EmployeeId_FK
 102

PermissionId_FK
1
Active
false

Process finished with exit code 0

DOMReadO09CTQ.java: [Dokumentum elérése](#)

2b, Adatmódosítás

```
package hu.domparse.O09CTQ;

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

import org.w3c.dom.*;
import org.xml.sax.SAXException;

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

public class DOMModifyO09CTQ {
    public static void main(String[] args) {

        try {
            //Eredeti fájl
            File xmlFile = new File("XMLO09CTQ.xml");
            DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = factory.newDocumentBuilder();
            Document document = dBuilder.parse(xmlFile);
            document.getDocumentElement().normalize();

            //Egy dolgozó email címét megváltoztatjuk
            NodeList nodes = document.getElementsByTagName("Employee");
            for (int i = 0; i < nodes.getLength(); i++) {
                Node node = nodes.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    if
(node.getAttributes().getNamedItem("EmployeeId").getTextContent().equals("100")) {
                        NodeList childNodes = node.getChildNodes();
                        for (int j = 0; j < childNodes.getLength();
j++) {
                            Node childNode = childNodes.item(j);
                            if
(childNode.getNodeName().equals("Email")) {
                                childNode.setTextContent("belaÚJ@email.com");
                            }
                        }
                    }
                }
            }
        } catch (SAXException | IOException | UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
```

```

    }
}
}
}
}

//Módosítjuk a módosítás időpontját
nodes = document.getElementsByTagName("Permission");
for (int i = 0; i < nodes.getLength(); i++) {
    Node node = nodes.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        if
(node.getAttributes().getNamedItem("PermissionId").getTextContent().equals("1")) {
            NodeList childNodes = node.getChildNodes();
            for (int j = 0; j < childNodes.getLength();
j++) {
                Node childNode = childNodes.item(j);
                if
(childNode.getNodeName().equals("Modified")) {
                    childNode.setTextContent("2021-11-23");
                }
            }
        }
    }
}

//Az osztály címének teljes módosítása
nodes = document.getElementsByTagName("Department");
for (int i = 0; i < nodes.getLength(); i++) {
    Node node = nodes.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        if
(node.getAttributes().getNamedItem("DepartmentId").getTextContent().equals("02")) {
            NodeList childNodes = node.getChildNodes();
            for (int j = 0; j < childNodes.getLength();
j++) {
                Node childNode = childNodes.item(j);
                if
(childNode.getNodeName().equals("Address")) {
                    childNodes = childNode.getChildNodes();
                    for(int k = 0;k <
childNodes.getLength();k++) {
                        childNode = childNodes.item(k);

```



```

if(childNode.getNodeName().equals("City")) {

    childNode.setTextContent("Budapest");
}

if(childNode.getNodeName().equals("Street")) {

    childNode.setTextContent("Kossuth Lajos tér 1-3");
}

if(childNode.getNodeName().equals("Postcode")) {

    childNode.setTextContent("1055");
}

}

}

}

}

}

}

}

//Új fájl neve
File myFile = new File("XMLO09CTQ2.xml");
writeXml(document, myFile);
}

catch (ParserConfigurationException | SAXException |
IOException | TransformerException ex) {
    System.out.println("ERROR" + ex.getMessage());
    ex.printStackTrace();
}
}

//Az új XML fájl kiírása
private static void writeXml(Document doc, File output) throws
TransformerException, UnsupportedEncodingException {
    TransformerFactory transformerFactory =
TransformerFactory.newInstance();
    Transformer transf = transformerFactory.newTransformer();
    transf.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
    transf.setOutputProperty(OutputKeys.INDENT, "yes");

    transf.setOutputProperty("{http://xml.apache.org/xslt}indent-amunt",
"2");
}

```

```
DOMSource source = new DOMSource(doc);
```

```
StreamResult console = new StreamResult(System.out);
```

```
StreamResult file = new StreamResult(output);
```

```
transf.transform(source, console);
```

```
transf.transform(source, file);
```

```
}
```

```
}
```

Output:

```
"C:\Users\Baumel Márton\.jdk\openjdk-16.0.2\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=58159:C:\Program Files\JetBrains\IntelliJ IDEA
2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath
C:\REPOS\O09TQ_XMLGyak\XMLTaskO09CTQ\DOMParseO09CTQ\target\classes
hu.domparse.O09CTQ.DOMModifyO09CTQ
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Office xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XMLSchemaO09CTQ.xsd">
```

```
<Employees>
```

```
<Employee EmployeeId="100">
```

```
<Name>Kis Béla</Name>
```

```
<Phone>06308881983</Phone>
```

```
<Email>belaÚJ@email.com</Email>
```

```
<DepartmentId_FK>01</DepartmentId_FK>
```

```
</Employee>
```

```
<Employee EmployeeId="101">
```

```
<Name>Nagy Ernő</Name>
```

```
<Phone>06308881984</Phone>
```

```
<Email>erno@email.com</Email>
```

```
<DepartmentId_FK>01</DepartmentId_FK>
```

```
</Employee>

<Employee EmployeeId="102">

  <Name>Baumel Márton</Name>

  <Phone>06308881985</Phone>

  <Email>marton@email.com</Email>

  <DepartmentId_FK>02</DepartmentId_FK>

</Employee>

</Employees>

<Departments>

  <Department DepartmentId="01">

    <Address>

      <Country>Hungary</Country>

      <Postcode>3900</Postcode>

      <City>Szerencs</City>

      <Street>Hársfa utca 14</Street>

    </Address>

    <Email>marcibaumel@gmail.com</Email>

    <Website>szerencswebsite.hu</Website>

  </Department>

  <Department DepartmentId="02">

    <Address>

      <Country>Hungary</Country>

      <Postcode>1055</Postcode>

      <City>Budapest</City>

      <Street>Kossuth Lajos tér 1-3</Street>
```

```
</Address>

<Email>miskolc@nememail.com</Email>

<Website>miskolciceg.hu</Website>

</Department>
</Departments>
<Warehouses>

  <Warehouse WarehouseId="01">

    <Item>

      <ItemName>Paper</ItemName>

      <ItemQuantity>100</ItemQuantity>

    </Item>

    <Item>

      <ItemName>scissors</ItemName>

      <ItemQuantity>200</ItemQuantity>

    </Item>

    <Address>

      <Country>Hungary</Country>

      <Postcode>3515</Postcode>

      <City>Miskolc</City>

      <Street>Egyetemi út 2</Street>

    </Address>

    <Email>warehousemiskolc@email.com</Email>

  </Warehouse>

  <Warehouse WarehouseId="02">
```

```
<Item>

  <ItemName>Paper</ItemName>

  <ItemQuantity>100</ItemQuantity>

</Item>

<Address>

  <Country>Hungary</Country>

  <Postcode>3900</Postcode>

  <City>Szerencs</City>

  <Street>Hársfa utca 14</Street>

</Address>

<Email>warehouseszerencs@email.com</Email>

</Warehouse>

</Warehouses>

<Permissions>

  <Permission PermissionId="1">

    <Role>admin</Role>

    <Modified>2021-11-23</Modified>

    <Computer>01</Computer>

  </Permission>

  <Permission PermissionId="2">

    <Role>user</Role>

    <Modified>2021-06-12</Modified>

    <Computer>03</Computer>

  </Permission>

</Permissions>
```

```
<Licenses>

  <License>

    <EmployeeId_FK>100</EmployeeId_FK>

    <PermissionId_FK>1</PermissionId_FK>

    <Active>true</Active>

  </License>

  <License>

    <EmployeeId_FK>101</EmployeeId_FK>

    <PermissionId_FK>2</PermissionId_FK>

    <Active>false</Active>

  </License>

  <License>

    <EmployeeId_FK>102</EmployeeId_FK>

    <PermissionId_FK>1</PermissionId_FK>

    <Active>false</Active>

  </License>

</Licenses>

</Office>
```

Process finished with exit code 0

DOMModifyO09CTQ.java: [Dokumentum elérése](#)

3c, Adatlekérdezés

```
package hu.domparse.O09CTQ;

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

import static java.lang.String.valueOf;

public class DOMQueryO09CTQ {

    public static void main(String[] args) throws
ParserConfigurationException, SAXException, IOException {
        //Minden dolgozó kiírása
        getAllEmployee();

        //Minden részleg kiírása
        getAllDepartment();

        //Minden eszköz megkapása ami a raktárban van
        getAllItemsByWarehouses();
    }

    //Megkapom az összes dolgozót
    public static void getAllEmployee() throws
ParserConfigurationException, IOException, SAXException {
        //Fájl kiválasztás
        File file = new File("XMLO09CTQ.xml");
```

```

        DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
        Document doc = dBuilder.parse(file);
        doc.getDocumentElement().normalize();
        System.out.print("Root element: ");
        System.out.println(doc.getDocumentElement().getNodeName());
        NodeList nList = doc.getElementsByTagName("Employee");

System.out.println("/*****");
;

        //Eredmény kiírása
        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i);
            System.out.println("\nCurrent Element :
"+node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;

System.out.println("ID:"+elem.getAttribute("EmployeeId"));
                NodeList nList2 = elem.getChildNodes();
                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE) {
                        Element elem2 = (Element) node2;
                        System.out.println("        "+node2.getNodeName()+"
: "+node2.getTextContent());

                    }
                }
            }
        }

System.out.println("/*****");
;

System.out.println("/*****");
;
        System.out.println("\n\n");
    }

    //Megkapom az összes osztályt

```



```

    public static void getAllDepartment() throws
ParserConfigurationException, IOException, SAXException {
    //Fájl kiválasztás
    File file = new File("XML009CTQ.xml");
    DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
    DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
    Document doc = dBuilder.parse(file);
    doc.getDocumentElement().normalize();
    System.out.print("Root element: ");
    System.out.println(doc.getDocumentElement().getNodeName());
    NodeList nList = doc.getElementsByTagName("Department");

System.out.println("/*****");
;

    //Eredmény kiírása
    for (int i = 0; i < nList.getLength(); i++) {
        Node node = nList.item(i);
        System.out.println("\nCurrent Element :
"+node.getNodeName());
        if (node.getNodeType() == Node.ELEMENT_NODE) {
            Element elem = (Element) node;

System.out.println("ID:"+elem.getAttribute("DepartmentId"));
            NodeList nList2 = elem.getChildNodes();
            for (int j = 0; j < nList2.getLength(); j++) {
                Node node2 = nList2.item(j);
                if (node2.getNodeType() == Node.ELEMENT_NODE) {
                    Element elem2 = (Element) node2;
                    System.out.println("        "+node2.getNodeName()+"
: "+node2.getTextContent());
                }
            }
        }
    }

System.out.println("/*****");
;

System.out.println("/*****");
;
    System.out.println("\n\n");

```

```

    }

    public static void getAllItemsByWarehouses() throws IOException,
    SAXException, ParserConfigurationException {
        //Fájl kiválasztás
        File file = new File("XML009CTQ.xml");
        DocumentBuilderFactory dbFactory =
        DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();

        Document doc = dBuilder.parse(file);
        doc.getDocumentElement().normalize();
        System.out.print("Root element: ");
        System.out.println(doc.getDocumentElement().getNodeName());
        NodeList nList = doc.getElementsByTagName("Warehouse");

        System.out.println("/*****");

        //Eredmény kiírása
        for (int i = 0; i < nList.getLength(); i++) {
            Node node = nList.item(i);
            System.out.println("\nCurrent Element :
            "+node.getNodeName());
            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element elem = (Element) node;

                System.out.println("ID:"+elem.getAttribute("WarehouseId"));
                NodeList nList2 = elem.getChildNodes();

                for (int j = 0; j < nList2.getLength(); j++) {
                    Node node2 = nList2.item(j);
                    if (node2.getNodeType() == Node.ELEMENT_NODE) {
                        Element elem2 = (Element) node2;
                        if (node2.getNodeName().equals("Item")) {
                            System.out.println(node2.getNodeName()+" :
                            "+node2.getTextContent());
                        }
                    }
                }
            }
        }
    }
}

```

```

System.out.println("/*****"/)
;

System.out.println("/*****"/)
;

    System.out.println("\n\n");
}

}

```

Output:

```

"C:\Users\Baumel Márton\jdk\openjdk-16.0.2\bin\java.exe" "-javaagent:C:\Program
Files\JetBrains\IntelliJ IDEA 2021.2.1\lib\idea_rt.jar=58192:C:\Program Files\JetBrains\IntelliJ IDEA
2021.2.1\bin" -Dfile.encoding=UTF-8 -classpath
C:\REPOS\O09TQ_XMLGyak\XMLTaskO09CTQ\DOMParseO09CTQ\target\classes
hu.domparse.O09CTQ.DOMQueryO09CTQ
Root element: Office
/*****/

```

```

Current Element : Employee
ID:100
    Name : Kis Béla
    Phone : 06308881983
    Email : bela@email.com
    DepartmentId_FK : 01

```

```

Current Element : Employee
ID:101
    Name : Nagy Ernő
    Phone : 06308881984
    Email : erno@email.com
    DepartmentId_FK : 01

```

```

Current Element : Employee
ID:102
    Name : Baumel Márton
    Phone : 06308881985
    Email : marton@email.com
    DepartmentId_FK : 02

```

```

/*****/
/*****/

```

Root element: Office

/*****/

Current Element : Department

ID:01

Address :

Hungary

3900

Szerencs

Hársfa utca 14

Email : marcibaumel@gmai.com

Website : szerencswebsite.hu

Current Element : Department

ID:02

Address :

Hungary

3515

Miskolc

Egyetemi út 1

Email : miskolc@nememail.com

Website : miskolciceg.hu

/*****/

/*****/

Root element: Office

/*****/

Current Element : Warehouse

ID:01

Item :

Paper

100

Item :

Scissors

200

Current Element : Warehouse

ID:02

Item :

Paper

100

```
/******  
/******
```

Process finished with exit code 0

DOMQueryO09CTQ.java: [Dokumentum elérése](#)