

JEGYZŐKÖNYV

Adatkezelés XML környezetben

Féléves feladat

Étterem adatbázis

Készítette: Butella Bence Kristóf

Neptunkód: IVLJQO

Dátum:

Tartalom

1.Feladat leírása	2
A feladat leírása:.....	2
1.1 ER modell:.....	3
1.2Az adatbázis konvertálása XDM modellre:	4
1.3Az XDM modell alapján XML dokumentum készítése:.....	5
1.4 AZ XML dokumentum alapján XMLSchema készítése:.....	12
2 Feladat:.....	19
Feladat leírása:	19
2.1Adatbeolvasás:	19
2.1 Adatátmódosítás:	22
2.3Adatlekérdezés:.....	24

1.Feladat leírása

A feladat leírása:

A beadandóm témája egy étel rendelő app adatbázisa ami kezeli a leadott rendeléseket a rendszer eltárolja az adott személy adatait illetve a rendelést hogy melyik étterem készítette és hogy mikor mit. Az adatokat értelem szerűen úgy készítettem hogy könnyen értelmezhető legyen a program készítése során. Ez úgy terveztem hogy ne csak egy étteremhez lehessen benne eltárolni hanem egy szerre többet is de jelenleg csak három van benne, de az a következőkben ki is derül az étterem pontos meg határozása érdekében egyedi adatokat lehet benne eltárolni étteremID ami ugye alpból egy egyedi azonosító de az id-t azt a többi egyednél használtam hogy meg különböztessék őket mert a nevel pontosan nem lehet meg különböztetni mindent. Mert egy étterme név lehet bármi, van hogy nevén viseli néha egy város, utca, vagy esetleg egy egyén nevét is és ilyenkor kellene az egyedi azonosítók hogy meg különböztessük az adatokat. Az egyedek nagy része alap adatokat tartalmaz egy adott dologról például vegyük a vevőt aminek kell hogy legyen neve mert a rendelések névre szólnak kell hogy legyen egy telefonszáma stb. A feladatom során törekedtem a pontoságra hogy minél életszerűbb legyen az egésznek a kialakítása.

Étterem egyed tulajdonságai:

- ÉtteremID: Az étterem egyed elsődleges kulcsa, azonosítója.
- Név: Az étterem neve.
- Telefonszám: Az étterem telefonszáma.

- Weblap: Az étterem weblap elérhetőséget tárolom benne.

Vevő egyed tulajdonságai:

- Id: A vevő id-át tartalmazza mivel egy vevő több rendelés is adhat le így tudjuk meg különböztetni.
- Név: A vevő neve.
- Telefonszám: A vevő telefonszáma.
- Fizetés: Összetett elem ami a készpénzből áll össze meg a kártyás fizetésből.

Lakcím egyed:

- Id: Egy vevő több helyről is rendelheti az ételét ezért külön id-val kell rendelkeznie.
- Házsám: Fontos a rendelés pontosítása érdekében.
- Utca: Fontos a rendelés pontosítása érdekében.
- Város: Fontos a rendelés pontosítása érdekében mert nem biztos hogy azonos helyen van az étterem mint a rendelő lakcíme.

Étlap:

- Név: Az étel neve hogy mit rendelt a vevő.
- Darab: Meg rendelt ételek mennyisége.
- Összeg: Mennyibe került az adott rendel mennyiség.
- Id: Több rendelés leadás megkülönböztetése érdekében.

Rendelés adatok:

- Dátum: Mikor adtál le a rendelést.
- RendelésId: A rendelés Id-ja.
- ÉtteremId: Az étterem Id-ja.
- Érkezés: Összetett elem ami abból az érkezés idő intervallumijából áll ettől meddig fog meg érkezni az adott rendelés.

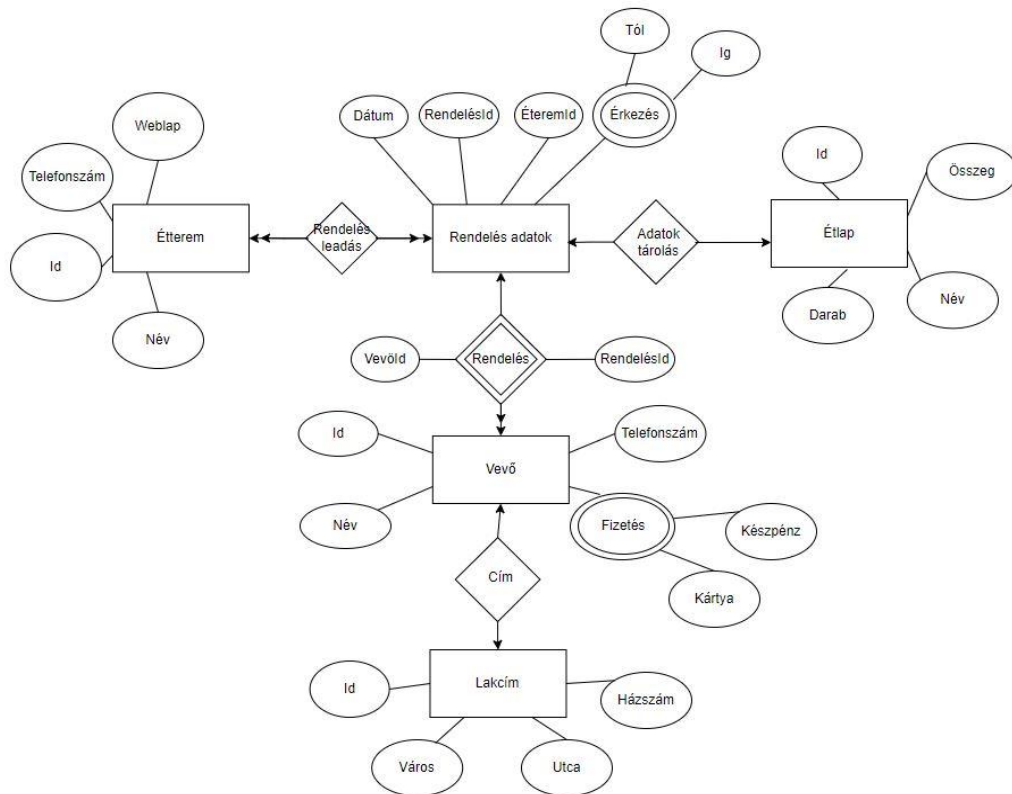
1.1 ER modell:

Az ER modellben mint az látható 5 egyed van mint azt ahogy fent említettem a leírásomban (Étterem, Vevő, Lakcím, Étlap, Rendelési adatok). Ezek között egy az egy 1:1 egy a többhöz 1:N illetve több a többhöz N:M kapcsolat is megfigyelhető az ábra alapján.

1:1 Az a vevőnél áll fent mivel egy vevő csak egy címre kérheti az adott ételét.

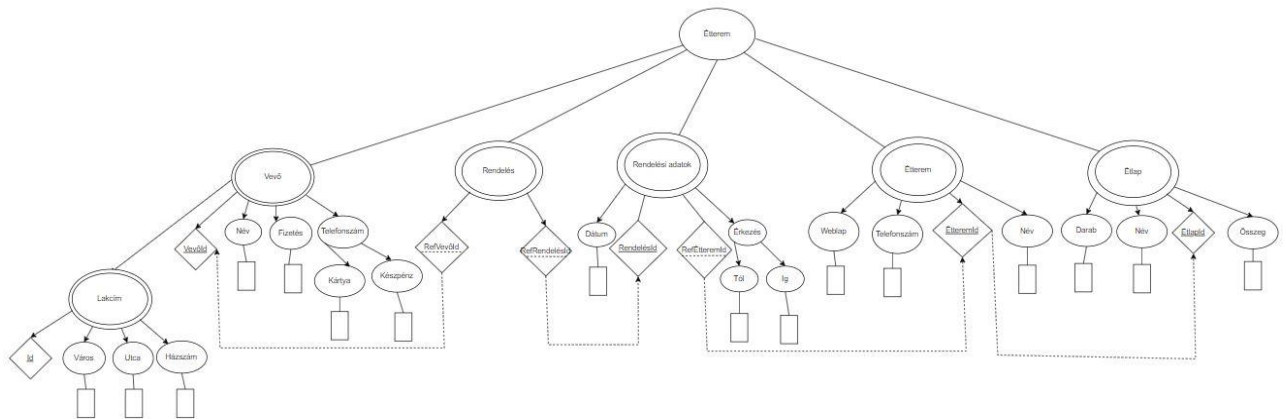
1:N Itt a rendelés és az étlapon jön létre mert egy étlapról csak egy rendelés jön létre de viszont a rendelések tartalmazhatnak több étlapot is.

N:M: Étterem Rendelés mert az étterem fel vehet több rendelést is és a rendelések tartalmazhatnak több éttermet is.



1.2Az adatbázis konvertálása XDM modellre:

Az XDM modellt az ER modell alapján készítettem el ez egy kicsit rendezettebb formát kölcsönöz a adatok kapcsolatára a könnyebb értelmezhetés elérésének céljából. Ahogy a képen látható hogy egymás mellett rendezetten helyezkednek el az adatok könnyebben kivehető az adat kapcsolatok formája . A gyökér elem az étterem az ezt a későbbiekben adatokra cseréltem hogy pontosan meg tudjam adni a programon írása során hogy mire gondoltam.Mert hibát okozott ugyan is a gyökér elem és az egyik elemnek azonos volt a neve így nem tudtam pontosan lekérdezni az adatokat illetve módosítani azokat. A formákból lehet kivenni minek mi a célja ebben az adatbázisban.



1.3Az XDM modell alapján XML dokumentum készítése:

Az XDM modell azaz XML Documentation Markup, segítségével készítettem el ez a modell típus segít meg értetni hogy hogyan kell majd, az xml fájlomnak adatbázisomnak kinézni-e illetve működni-e. Értelem szerűen ugyan azokat az elemeket, adatokat, egyedeket használtam próbáltam igényesen rendezetten átláthatóan készíteni az xml fájlt. Amikor kialakítottam követtem az általános kinézeti követelményeket Az elején meg adtam az alap verzió számot ami az "1.0" utána meg adtam az általános karakter kódolást amit xml fájlomnál szoktunk használni azaz "utf-8" az UTF-8 (8-bit Unicode Transformation Format, 8 bites Unicode átalakítási formátum) változó hosszúságú Unicode karakterkódolási eljárás. Az xml-ben könnyen értelmezhető tag-et használtam a könnyebb értelmezhetőség érdekében ezeket értelem szerűen lezártam. A gyökér elem neve adatok ennek az elemi a éttermek, étterem, vevők , vevő, lakcímek , lakcím , rendelésadatok , rendelés , étlapok , étlap ezeknek egyedi adatokat adtam.

<!--Verzioszám karkarkter kódolás -->

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

<!--Gyökér elem-->

<adatok xmlns="ivljqo"

xsi:schemaLocation="ivljqo XMLSchemaIVLJQO.xsd"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<!--gyerek elem-->

<ettermek>

<!--gyerek elem id tulajdonsággal ez az elsődleges kulcs-->

<etterem id="1">

<!--nev gyerek elem -->

<nev>Egyszer étterem</nev>

<!--telefonszám gyerek elem -->

```
<telefonszam>06704564653</telefonszam>

<!--weblap gyerek elem -->

<weblap>www.egyszeretterem.com</weblap>

<rendelesadatok>

  <!--gyerek elem egy másik gyerek elemre való hivatkozásra-->

  <rendeles idref="1"/>

</rendelesadatok>

</etterem>

  <!--a második gyerek elem ugyan azokkal a gyerek elemekkel ellátva ebből összesen
3 van mindegyikből -->

  <etterem id="2">

    <nev>Jó étterem</nev>

    <telefonszam>06704569873</telefonszam>

    <weblap>www.joetterem.com</weblap>

    <rendelesadatok>

      <rendeles idref="2"/>

    </rendelesadatok>

  </etterem>

  <etterem id="3">

    <nev>Volt étterem</nev>

    <telefonszam>06704659873</telefonszam>

    <weblap>www.voltetterem.com</weblap>

    <rendelesadatok>

      <rendeles idref="3"/>

    </rendelesadatok>

  </etterem>

</ettermek>

<!--vevők gyerek elem -->

<vevok>

  <!--gyerek elem id tulajdonsággal ez az elsődleges kulcs-->

  <vevo id="1">
```

<!--nev gyerek elem -->

<nev>Bence</nev>

<!--többértékű gyerek elem ami tartalmaz 2 másik gyerek elemet-->

<fizetes>

<!--készpénz gyerek elem -->

<keszpenz>10000</keszpenz>

<!--kártya gyerek elem -->

<kartya>0</kartya>

</fizetes>

<!--telefonszám gyerek elem -->

<telefonszam>06703333456</telefonszam>

<!--lakcímek gyerek elem hivakozással a lakcímek gyerek elemre-->

<lakcimek>

<lakcim idref="1"/>

</lakcimek>

<!--gyerek elem egy másik gyerek elemre való hivatkozásra-->

<rendelesadatok>

<rendeles idref="1"/>

</rendelesadatok>

</vevo>

<!--ebből is van még 2 ehez hasonló gyerek elem-->

<vevo id="2">

<nev>Zsófi</nev>

<fizetes>

<keszpenz>500</keszpenz>

<kartya>0</kartya>

</fizetes>

<telefonszam>06706343353</telefonszam>

<lakcimek>

<lakcim idref="2"/>

</lakcimek>

<rendelesadatok>

```
        <rendeles idref="2"/>
    </rendelesadatok>
</vevo>

<vevo id="3">
    <nev>Sándor</nev>

    <fizetes>
        <keszpenz>1500</keszpenz>
        <kartya>0</kartya>
    </fizetes>

    <telefonszam>06706343353</telefonszam>

    <lakcimek>
        <lakcim idref="3"/>
    </lakcimek>
    <rendelesadatok>
        <rendeles idref="3"/>
    </rendelesadatok>
</vevo>

</vevok>

<!--lakcimek gyerek elem -->

<lakcimek>
    <!--lakcim gyerek elem elsődleges kulcsal ellátva -->
    <lakcim id="1">
        <!--város gyerek elem -->
        <varos>Sátor</varos>

        <!--irányítószám gyerek elem -->
        <iranyitoszam>3970</iranyitoszam>

        <!--utca gyerek elem -->
        <utca>Paloc</utca>

        <!--házszám gyerek elem -->
        <hazszam>2</hazszam>
```


</lakcim>

<!--a lakcimekből is készítettem 3elemt hogy legyen minden vevőnek legyen egyedi lakcime -->

<lakcim id="2">

<varos>Sátor</varos>

<iranyitoszam>3970</iranyitoszam>

<utca>Lolka</utca>

<hazszam>4</hazszam>

</lakcim>

<lakcim id="3">

<varos>Sátor</varos>

<iranyitoszam>3970</iranyitoszam>

<utca>Lolka</utca>

<hazszam>4</hazszam>

</lakcim>

</lakcimek>

<!--rendelesadatok gyerek elem -->

<rendelesadatok>

<!--rendeles gyerek elem -->

<rendeles id="1">

<!--érkezés gyerek elem -->

<erkezes>

<!--tól-ig gyerek elemek -->

<tol>2022.06.08.18:40</tol>

<ig>2022.06.08.19:30</ig>

</erkezes>

<!--dátu, gyerek elem -->

<datum>2022.06.08.18:00</datum>

<!--étterem gyerek elem -->

<ettermek>

<!--étterem gyerek elem az étterem re való hitvatkozással -->

```
        <etterem idref="1"/>
    </ettermek>
</rendeles>

<!--a rendelésekből is készítettem 3elemt hogy legyen minden vevőnek legyen egyedi
rendelése -->
```

```

    <rendeles id="2">
        <erkezes>
            <tol>2022.07.09.18:40</tol>
            <ig>2022.07.09.19:30</ig>
        </erkezes>
        <datum>2022.07.09.18:00</datum>
        <ettermek>
            <etterem idref="2"/>
        </ettermek>
    </rendeles>
```

```

    <rendeles id="3">
        <erkezes>
            <tol>2022.06.09.18:40</tol>
            <ig>2022.06.09.19:30</ig>
        </erkezes>
        <datum>2022.06.09.18:00</datum>
        <ettermek>
            <etterem idref="3"/>
        </ettermek>
    </rendeles>
```

```
</rendelesadatok>
```

```
<!--étlapok gyerek elem -->
```

```
<etlapok>
```

```
    <!--étlap gyerek elem elsődleges kulcsal ellátva-->
```

```
    <etlap id="1">
```

```
        <!--név gyerek elem -->
```

<nev>Pizza</nev>

<!--darab gyerek elem -->

<darab>5</darab>

<!--összeg gyerek elem -->

<osszeg>10000</osszeg>

<!--gyerek elem egy másik gyerek elemre való hivatkozásra—a rendelés adatokra>

<rendelesadatok>

<rendeles idref="1"/>

</rendelesadatok>

</etlap>

<!--az étlapokból is készítettem 3elemt hogy legyen minden vevőnek legyen egyedi rendelése -->

<etlap id="2">

<nev>Hamburger</nev>

<darab>1</darab>

<osszeg>500</osszeg>

<rendelesadatok>

<rendeles idref="2"/>

</rendelesadatok>

</etlap>

<etlap id="3">

<nev>Gyros</nev>

<darab>1</darab>

<osszeg>1500</osszeg>

<rendelesadatok>

<rendeles idref="3"/>

</rendelesadatok>

</etlap>

</etlapok>

</adatok>

1.4 AZ XML dokumentum alapján XMLSchema készítése:

Az XML alapján egy XMLSchemát készítem ami az xml szerkezetét fejezi ki .Ez egy másik fájlkiterjesztés használ ami az XSD .Mint az XMLben itt is az egyes 1 verzió számot használtam és utf-8 karakter kódolást .Ebben a feladatban saját típusokat is hoztam létre .Ugyan úgy mint az xml dokumentumban is van gyökér elem és gyerekelemek is de viszont itt már meg tudjuk adni hogy milyen típusú elemeket tárolhatunk az adott xml tagekben ezek között fel lehetőséget adni elsődleges kulcsok és idegen kulcsok. Minden elemnek van egy elfordulási korlátja ezt alulról felülről be lehet állítani ebben a példában 0-korlátlan számra állítottam.

```

<!--verzió szám karakterkódolás-->
<?xml version="1.1" encoding="UTF-8"?>
<xs:schema targetNamespace="ivljqo" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!--összetett típus elem ami a lakcimTípus névre halgat-->
  <xs:complexType name="lakcimTípus">
    <!--a gyerekelemeket a szekvencia tagek között kell megadni -->
    <!--nyitótag-->
    <xs:sequence>
      <!--gyerek elemek-->
      <xs:element type="xs:string" name="varos"/>
      <xs:element type="xs:string" name="iranyitoszam"/>
      <xs:element type="xs:string" name="utca"/>
      <xs:element type="xs:string" name="hazszam"/>
    <!--zárotag-->
    </xs:sequence>
  <!--zárotag-->
  </xs:complexType>
  <!--összetett típus elem ami a lakcimTípus névre halgat nyitó tag-->
  <xs:complexType name="lakcimekTípus">
    <xs:sequence>
      <!--meghatározzuk hányszor fordulhat elő egy elem-->
      <xs:element type="lakcimTípus" name="lakcim" maxOccurs="unbounded" minOccurs="0"/>
    <!--zárotagek-->
    </xs:sequence>
  </xs:complexType>
  <!--komlex típus ami több fajta típus is felvehet egyben nyitó tag-->
  <xs:complexType name="vevo" mixed="true">
    <xs:sequence>
      <xs:element name="nev" />
      <xs:element name="telefonszam" />
    <!--hivatkozási elem a saját típusunkra-->

```

```

    <xs:element type="fizetesTipus" name="fizetes" />
    <xs:element type="lakcimekTipus" name="lakcim" />
    <xs:element type="rendelesadatokTipus" name="rendeles" />
  </xs:sequence>
<!-- idegen kulcsokra való hivatkozás -->
  <xs:attribute type="xs:string" name="idref" />
  <xs:attribute type="xs:string" name="id" />
</xs:complexType>
<!-- összetett típus elem ami a vevokTipus névre halgat -->
<xs:complexType name="vevokTipus">
  <xs:sequence>
    <!-- meghatározzuk hogy hányszor fordulhat elő egy vevő típus -->
    <xs:element type="vevotipus" name="vevo" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<!-- összetett típus elem ami a rendelesadatokTipus névre halgat -->
<xs:complexType name="rendelesadatokTipus">
  <xs:sequence>
    <xs:element type="erkezesTipus" name="erkezes"/>
    <xs:element type="xs:string" name="datum"/>
    <xs:element type="etteremekTipus" name="etterem"/>
  </xs:sequence>
  <xs:attribute type="xs:string" name="id"/>
  <xs:attribute type="xs:string" name="idref" />
</xs:complexType>
<!-- összetett típus elem ami a rendelesadatokTipus névre halgat -->
<xs:complexType name="rendelesadatokTipus">
  <xs:sequence>
    <xs:element type="rendelesTipus" name="rendeles"/>
  </xs:sequence>
</xs:complexType>
<!-- összetett típus elem ami a etteremTipus névre halgat -->

```

<!--több fajta tipust is felvehet-->

<xs:complexType name="etteremTipus" mixed="true">

<xs:sequence>

<!--elemek és elem hivatkozások-->

<xs:element type="xs:string" name="nev"/>

<xs:element type="fizetesTipus" name="fizetes"/>

<xs:element type="xs:string" name="telefonszam"/>

<xs:element type="lakcimekTipus" name="lakcim"/>

<xs:element type="rendelesadatokTipus" name="rendeles"/>

<xs:element type="xs:string" name="weblap"/>

</xs:sequence>

<xs:attribute type="xs:string" name="idref" />

<xs:attribute type="xs:string" name="id" />

</xs:complexType>

<!--összetett típus elem ami a etteremTipus névre halgat-->

<xs:complexType name="ettermekTipus">

<xs:sequence>

<!--meghatározzuk hogy hányszor fordulhat elő egy etterm típus-->

<xs:element type="etteremTipus" name="etterem" maxOccurs="unbounded" minOccurs="0"/>

</xs:sequence>

</xs:complexType>

<!--összetett típus elem ami a etlapTipus névre halgat-->

<xs:complexType name="etlapTipus">

<xs:sequence>

<!--elemek meg adása hivatkozás egy másik elemre-->

<xs:element name="nev" />

<xs:element name="darab" />

<xs:element name="osszeg" />

<xs:element type="rendelesadatokTipus" name="date" />

</xs:sequence>

<!--kulcsokra való hivatkozása-->

<xs:attribute type="xs:string" name="idref" />

```

    <xs:attribute type="xs:string" name="id" />
  </xs:complexType>
<!--összetett típus elem ami a etlapokTípus névre halgat-->
<xs:complexType name="etlapokTípus">
  <xs:sequence>
    <xs:element type="etlapTípus" name="etlap" maxOccurs="unbounded" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="erkezesTípus">
  <xs:sequence>
    <xs:element name="tol" />
    <xs:element name="ig" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="fizetesTípus">
  <xs:sequence>
    <xs:element name="keszpenz" />
    <xs:element name="kartya" />
  </xs:sequence>
</xs:complexType>
<!--adatok gyökér elem elemi egyedi típusok-->
<xs:element name="adatok">
<!--komplex típus benne a saját típusok-->
<xs:complexType>
  <xs:sequence>
    <xs:element type="lakcimekTípus" name="lakcimek"/>
    <xs:element type="vevokTípus" name="vevok"/>
    <xs:element type="rendelesadatokTípus" name="rendeles"/>
    <xs:element type="ettermekTípus" name="ettermek"/>
    <xs:element type="etlapokTípus" name="etlapok"/>

```



```

</xs:sequence>
</xs:complexType>
<!--a kulcs neve-->
<xs:key name="lakcimId">
  <!--az elérési útvonal hogy honnan legyen az elsődleges kulcs-->
  <xs:selector xpath="/lakcimek/lakcim"/>
  <!--amiben tárolódik a kulcs attribútuma-->
  <xs:field xpath="@id" />
  <!--zárotag-->
</xs:key>
  <!--a kulcs neve-->
<xs:key name="vevoId">
  <!--az elérési útvonal hogy honnan legyen az elsődleges kulcs-->
  <xs:selector xpath="/vevok/vevo"/>
  <!--amiben tagolódik a kulcs attribútuma-->
  <xs:field xpath="@id" />
  <!--zárotag-->
</xs:key>
  <!--a kulcs neve-->
<xs:key name="rendelesID">
  <!--az elérési útvonal hogy honnan legyen az elsődleges kulcs-->
  <xs:selector xpath="/rendelesadatok/rendeles"/>
  <!--amiben tagolódik a kulcs attribútuma-->
  <xs:field xpath="@id" />
  <!--zárotag-->
</xs:key>
  <!--a kulcs neve-->
<xs:key name="etteremID">
  <!--az elérési útvonal hogy honnan legyen az elsődleges kulcs-->
  <xs:selector xpath="/ettermek/etterem"/>
  <!--amiben tagolódik a kulcs attribútuma-->
  <xs:field xpath="@id" />

```

```

        <!--zárotag-->
</xs:key>

        <!--a kulcs neve-->
<xs:key name="etlapID">
    <!--az elérési útvonal hogy honnan legyen az elsődleges kulcs-->
    <xs:selector xpath="/etlapok/etlap"/>
    <!--amiben tagolódik a kulcs attribútuma-->
    <xs:field xpath="@id" />
    <!--zárotag-->
</xs:key>

<xs:keyref name="rendelesRef" refer="rendelesId">
    <xs:selector xpath="/ettermek/etterem/rendelesadatok/rendeles"/>
    <xs:field xpath="@idref" />
</xs:keyref>

<xs:keyref name="lakcimRef" refer="lakcimId">
    <xs:selector xpath="/vevok/vevo/lakcimek/lakcim"/>
    <xs:field xpath="@idref" />
</xs:keyref>

<xs:keyref name="ettermekRef" refer="etteremID">
    <xs:selector xpath="/rendelesadatok/rendeles/ettermek/etterem"/>
    <xs:field xpath="@idref" />
</xs:keyref>
</xs:element>
</xs:schema><!--schema zárotagje-->

```

2 Feladat:

Feladat leírása:

A program elkészítéséhez az eclipsset használtam. Létre hoztam a programot az XMLIVLJQO.xml adatainak adminisztrálása alapján. A project neve DOMParseIVLJQO Packeg hu.domparse.IVLJQO Ezen belül létre hoztam három osztályt egyet olvasásra ami a DomReadIVLJQO névre hallgat ebben a részben bekérem az XMLIVLJQO.xml összes adattagját és ki irattam rendezett formában a konzolra .Ezután következett a DomQueryIVLJQO amiben lekérdezéseket hoztam létre az XMLIVLJQO.xml felhasználásával és kiirattam a konzolra az eredményeket

2.1Adatbeolvasás:

Az osztály neve DomReadIVLJQO Ebben a részben minden adatot ki irattam a konzolra ami az xml fájlban található rendezett formában külön részre tagoltan. Szöveget is ki irattam az adtok mellé a könnyebb értelmezhetés érdekében.

Dom program adatkezeléshez a commentek a kódban olvashatóak:

Külön behívtam az összes elemet majd azokat ki irattam rendezett formában a konzolra

Olvasás: a XMLIVLJQO.xml fájlból.

DomReadIVLJQO kód commentel:

```
package hu.domparse.ivljqo;
```

```
import java.io.File;
```

```
import java.util.Iterator;
```

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.xpath.XPath;
```

```
import javax.xml.xpath.XPathConstants;
```

```
import javax.xml.xpath.XPathExpressionException;
```

```
import javax.xml.xpath.XPathFactory;
```

```
import javax.xml.parsers.ParserConfigurationException;
```

```
import org.w3c.dom.Node;
```

```
import org.w3c.dom.NodeList;
```

```
import org.w3c.dom.Document;
```

```
import org.w3c.dom.Element;
```

```
import org.xml.sax.SAXException;
```

```
public class DomReadIVLJQO {
```

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

```
        try {
```

```
            //fájl helyének megadása
```

```
            File xml = new File("XMLIVLJQO.xml");
```

```
            //
```

```
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
```

```
            //Dokumentum fá felépítése
```

```
            DocumentBuilder db = dbf.newDocumentBuilder();
```

```

//xml beolvasás
Document doc = db.parse(xml);

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

//Minden adat kiírása
//összeszedi az elemeket a fából

String etterem = "/adatok/ettermek/etterem";
NodeList nlEtterem = (NodeList) xPath.compile(etterem).evaluate( doc,
XPathConstants.NODESET);
String rendeles = "/adatok/rendelesadatok/rendeles";
NodeList nlrendelesa = (NodeList) xPath.compile(rendeles).evaluate( doc,
XPathConstants.NODESET);
String vevo = "/adatok/vevok/vevo";
NodeList nlVevo = (NodeList) xPath.compile(vevo).evaluate( doc,
XPathConstants.NODESET);
String etlap = "/adatok/etlapok/etlap";
NodeList nlEtlap = (NodeList) xPath.compile(etlap).evaluate( doc,
XPathConstants.NODESET);
String lakcim = "/adatok/lakcimek/lakcim";
NodeList nlLakcim = (NodeList) xPath.compile(lakcim).evaluate( doc,
XPathConstants.NODESET);

//lista elemeire ciklus
for (int i = 0; i < nlVevo.getLength(); i++) {
    Node n1 = nlVevo.item(i);

    if(n1.getNodeType() == Node.ELEMENT_NODE) {
        Element el1 = (Element) n1;

        //vevő adatainak kiírása
        System.out.println((i+1) + ". |Vevő neve: " +
el1.getElementsByTagName("nev").item(0).getTextContent() + " |Kartya: " +
el1.getElementsByTagName("kartya").item(0).getTextContent()
+ " |Kézpénz: " +
el1.getElementsByTagName("keszpenz").item(0).getTextContent()+ " |Telefonszáma: " +
el1.getElementsByTagName("telefonszam").item(0).getTextContent());

    }
}

for (int i = 0; i < nlLakcim.getLength(); i++) {
    Node n5 = nlLakcim.item(i);

    if(n5.getNodeType() == Node.ELEMENT_NODE) {
        Element el5 = (Element) n5;

        //vevő adatainak kiírása
        System.out.println((i+1) + ". |Vevő lakcime: " + " |Irsz: "
+el5.getElementsByTagName("iranyitoszam").item(0).getTextContent()+ " |Város: " +
el5.getElementsByTagName("varos").item(0).getTextContent()
+ " |Utca: " +
el5.getElementsByTagName("utca").item(0).getTextContent()+ " |Házzsám: " +
el5.getElementsByTagName("hazzsam").item(0).getTextContent());

    }
}

//étterem adatainak kiírása

```

```

        for (int j = 0; j < nlEtterem.getLength(); j++) {
            Node n2 = nlEtterem.item(j);

            if(n2.getNodeType() == Node.ELEMENT_NODE) {
                Element el2 = (Element) n2;
                System.out.println((j+1) + ". |Étterem neve: " +
el2.getElementsByTagName("nev").item(0).getTextContent()+ " |Weblap: " +
el2.getElementsByTagName("weblap").item(0).getTextContent()+
                " |Telefonszam: " +
el2.getElementsByTagName("telefonszam").item(0).getTextContent());
            }
        }
        for (int i = 0; i < nlrendelesa.getLength(); i++) {
            Node n3 = nlrendelesa.item(i);

            if(n3.getNodeType() == Node.ELEMENT_NODE) {
                Element el3 = (Element) n3;

                //rendelés adatainak kiírása
                System.out.println((i+1) + ". |Rendelés tól: " +
el3.getElementsByTagName("tol").item(0).getTextContent()+ ". |Rendelés ig: " +
el3.getElementsByTagName("ig").item(0).getTextContent());

            }
        }
        for (int i = 0; i < nlEtlap.getLength(); i++) {
            Node n4 = nlEtlap.item(i);

            if(n4.getNodeType() == Node.ELEMENT_NODE) {
                Element el4 = (Element) n4;

                //rendelés adatainak kiírása
                System.out.println((i+1) + ". |Étel név: " +
el4.getElementsByTagName("nev").item(0).getTextContent()+ ". |Étel darab száma: " +
el4.getElementsByTagName("darab").item(0).getTextContent()
                + ". |Összeg: " +
el4.getElementsByTagName("osszeg").item(0).getTextContent());

            }
        }
        //kivételkezelés
    } catch (Exception e) {
        System.out.println("Hiba történt: " +e.getMessage());
    }
}
}
Futási eredmény:

```

```

1. |Vevő neve: Bence |Kartya: 0 |Készpénz: 10000 |Telefonszáma: 06703333456
2. |Vevő neve: Zsófi |Kartya: 0 |Készpénz: 500 |Telefonszáma: 06706343353
3. |Vevő neve: Sándor |Kartya: 0 |Készpénz: 1500 |Telefonszáma: 06706343353
1. |Vevő lakcíme: |Irsz: 3970 |Város: Sátor |Utca: Paloc |Házszám: 2
2. |Vevő lakcíme: |Irsz: 3970 |Város: Sátor |Utca: Lolka |Házszám: 4
3. |Vevő lakcíme: |Irsz: 3970 |Város: Sátor |Utca: Lolka |Házszám: 4
1. |Étterem neve: Egyszer étterem |Weblap: www.egyszeretterem.com |Telefonszam: 06704564653
2. |Étterem neve: Jó étterem |Weblap: www.joetterem.com |Telefonszam: 06704569873
3. |Étterem neve: Volt étterem |Weblap: www.voltetterem.com |Telefonszam: 06704659873
1. |Rendelés tól: 2022.06.08.18:40. |Rendelés ig: 2022.06.08.19:30
2. |Rendelés tól: 2022.07.09.18:40. |Rendelés ig: 2022.07.09.19:30
3. |Rendelés tól: 2022.06.09.18:40. |Rendelés ig: 2022.06.09.19:30
1. |Étel név: Pizza. |Étel darab száma: 5. |Összeg: 10000
2. |Étel név: Hamburger. |Étel darab száma: 1. |Összeg: 500
3. |Étel név: Gyros. |Étel darab száma: 1. |Összeg: 1500

```

2.1 Adatmódosítás:

Ebben a feladatban behívom az XMLIVLJQO.xml fájlt és módosítok egy pár adatot benne és elmentem őket az Update_XMLIVLJQO.xml amit maga a programom hoz létre. A végén egy rövid szöveget irattam ki hogy az XML módosítva lett ez csak egy kis vissza csatolás nekem hogy a program hiba nélkül lefutott.

DomModifyIVLJQO kód commentel:

```
package hu.domparse.ivljqo;
```

```
import java.io.File;
```

```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.transform.OutputKeys;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerFactoryConfigurationError;
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.Node;
import org.w3c.dom.NodeList;
```

```
public class DomModifyIVLJQO {
```

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

```
        try {
```

```
            //fájl helyének megadása
```

```
            String path = "XMLIVLJQO.xml";
```

```
            File xmlFile = new File (path);
```

```
            //
```

```
            DocumentBuilder db;
```

```
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
```

```

        db = dbf.newDocumentBuilder();
        Document doc = db.parse(xmlFile);
        doc.getDocumentElement().normalize();

        // Módosítás
        updateElement(doc);

        // Törlés
        deleteElement(doc);

        // Element hozzáadása
        addElement(doc);
        addElement2(doc);

        // Módosított fájl létrehozása
        writeFile(doc);

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

private static void writeFile(Document doc)
throws TransformerFactoryConfigurationError, TransformerConfigurationException, TransformerException
{
    doc.getDocumentElement().normalize();
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();
    DOMSource ds = new DOMSource(doc);
    StreamResult result = new StreamResult(new File("Updated_XMLIVLJQO.xml"));
    transformer.setOutputProperty(OutputKeys.INDENT, "yes");
    transformer.transform(ds, result);
    System.out.println("XML módosítva");
}

//új elem hozzáadása az étteremhez
private static void addElement(Document doc) {
    NodeList etterem1 = doc.getElementsByTagName("etterem");
    Element etterem = null;

    //minden darabra
    for (int i = 0; i < etterem1.getLength(); i++) {
        etterem = (Element) etterem1.item(i);
        Element id = doc.createElement("kapacitas");
        id.appendChild(doc.createTextNode("500"));
        etterem.appendChild(id);
    }
}

private static void addElement2(Document doc) {
    NodeList vevo = doc.getElementsByTagName("vevo");
    Element vevoad = null;

    //minden darabra
    for (int i = 0; i < vevo.getLength(); i++) {
        vevoad = (Element) vevo.item(i);
        Element id = doc.createElement("kor");
        id.appendChild(doc.createTextNode("22"));
        vevoad.appendChild(id);
    }
}

```

```

    }

    //Egy elem törlése
    private static void deleteElement(Document doc) {
        NodeList etterem = doc.getElementsByTagName("fizetes");
        Element fizetes = null;
        //minden elementre
        for (int i = 0; i < etterem.getLength(); i++) {
            fizetes = (Element) etterem.item(i);
            Node kartya = fizetes.getElementsByTagName("kartya").item(0);
            fizetes.removeChild(kartya);
        }
    }

    //Egy ügyfél nevének megváltoztatása (első elem használatával)
    private static void updateElement(Document doc) {
        NodeList etterem = doc.getElementsByTagName("vevo");
        Element vevo = null;
        // minden elementre
        for (int i = 0; i < etterem.getLength(); i++) {
            vevo = (Element) etterem.item(i);
            Node nev = vevo.getElementsByTagName("nev").item(0).getFirstChild();
            //megfelelő ügyfél megkeresése és adat megváltoztatása
            if (vevo.getAttribute("id").contentEquals("1")) {
                nev.setNodeValue("Ádám");
            }
        }
    }
}

```

Futási eredményt egy új xml fájlban jön létre ami az Update_XMLIVLJQO.xml

2.3Adatlekérdezés:

Ebben a részben le kéredezéskét hozok létre az XMLIVLJQO.xml fájlból.

Ezek:

- egy vevő elem ki íratása a konzolra
- teljes étterem gyerek elem ki irtása a konzolra
- összegzem a bevételt
- átlagolom a bevételt
- meg keresem a legnagyobb bevételt

DomQueryIVLJQO kód commentel

```
package hu.dompars.ivljqo;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.util.Scanner;
```

```
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 DomQueryIVLJQO {

    public static void main(String[] args) {

        try {
            //fájl helyének megadása
            File xml = new File("XMLIVLJQO.xml");
            //Dokumentum fá felépítése
            DocumentBuilderFactory dbF = DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder;

            dBuilder = dbF.newDocumentBuilder();
            //xml beolvasás
            Document doc = dBuilder.parse(xml);
            doc.getDocumentElement().normalize();
            //Utvonál
            XPath xPath = XPathFactory.newInstance().newXPath();

            //Egy Vevő Lekérdezés
            System.out.println("-----");
            System.out.println("1.Lekérdezés: Lekérjük azt a vevőt akinek az 1 id-t adtuk");
            String expression = "/adatok/vevok/vevo[@id = '1']";
            NodeList nodeList = (NodeList) xPath.compile(expression).evaluate(
                doc, XPathConstants.NODESET);

            for (int i = 0; i < nodeList.getLength(); i++) {
                Node nNode = nodeList.item(i);
                System.out.println("\nJelenlegi Element : " + nNode.getNodeName());
                //Vevő adataink kiírása
                if (nNode.getNodeType() == Node.ELEMENT_NODE) {
                    Element eElement = (Element) nNode;
                    System.out.println("Vevő id : "
                        + eElement.getAttribute("id"));
                    System.out.println("nev : "
                        + eElement
                            .getElementsByTagName("nev")
                                .item(0)
                                    .getTextContent());
                    System.out.println("keszpenz : "
                        + eElement
                            .getElementsByTagName("keszpenz")
                                .item(0)
                                    .getTextContent());
                    System.out.println("kartya : "
                        + eElement
                            .getElementsByTagName("kartya")
                                .item(0)
                                    .getTextContent());
                    System.out.println("telefonszám : "
                        + eElement

```

```

        .getElementsByTagName("telefonszam")
        .item(0)
        .getTextContent());
    }
}
System.out.println("-----");
System.out.println("2.Lekérdezés: Lekérjük az étterem összes adatát");
//Étterem összes elemének kiírása
String expression2 = "/adatok/ettermek/etterem";
NodeList nodeList2 = (NodeList) XPath.compile(expression2).evaluate(
    doc, XPathConstants.NODESET);

for (int i = 0; i < nodeList2.getLength(); i++) {
    Node nNode2 = nodeList2.item(i);
    System.out.println("\nJelenlegi Element : " + nNode2.getNodeName());
    //Éttrem adataink kiírása
    if (nNode2.getNodeType() == Node.ELEMENT_NODE) {
        Element eElement = (Element) nNode2;
        System.out.println("Etterem id : "
            + eElement.getAttribute("etteremid"));
        System.out.println("nev : "
            + eElement
            .getElementsByTagName("nev")
            .item(0)
            .getTextContent());
        System.out.println("telefonszám : "
            + eElement
            .getElementsByTagName("telefonszam")
            .item(0)
            .getTextContent());
        System.out.println("Weblap : "
            + eElement
            .getElementsByTagName("weblap")
            .item(0)
            .getTextContent());
    }
}

System.out.println("-----");
System.out.println("3.Lekérdezés: Megnézzük hogy mennyi volt a bevétel");
NodeList nlVevo = doc.getElementsByTagName("vevo");
//Kifizettet készpénzek összege
int osszeg=0;
for (int i = 0; i < nlVevo.getLength(); i++) {
    int t[] = new int[nlVevo.getLength()];
    Node n1 = nlVevo.item(i);

    if(n1.getNodeType() == Node.ELEMENT_NODE) {
        Element e1 = (Element) n1;

        System.out.println("\n" + (i+1) + ". vevő készpénz: " +
e1.getElementsByTagName("keszpenz").item(0).getTextContent());

        t[i]=Integer.parseInt(e1.getElementsByTagName("keszpenz").item(0).getTextContent());

        }osszeg=osszeg+t[i];
    }
}

System.out.println("\nKifizette készpénz Összeg: "+osszeg);

```

```

System.out.println("-----");
System.out.println("4.Lekérdezés: Meg nézzük mennyi volt az átlaga a bevételnek");
//Készpénz átlaga
float atl;
int osszeg2=0;
for (int i = 0; i < nlVevo.getLength(); i++) {

    int t[] = new int[nlVevo.getLength()];
    Node n1 = nlVevo.item(i);

    if(n1.getNodeType() == Node.ELEMENT_NODE) {
        Element e1 = (Element) n1;

        t[i]=Integer.parseInt(e1.getElementsByTagName("keszpenz").item(0).getTextContent());
        }osszeg2=osszeg2+t[i];

    }

    atl=osszeg/nlVevo.getLength();

    System.out.println("\nKifizette készpénz Átlaga: "+atl);
    System.out.println("-----");
    System.out.println("5.Lekérdezés: Meg keressük a legnagyobb értéket");
    //Legnagyobb Készpénz
    int max=0;
    for (int i = 0; i < nlVevo.getLength(); i++) {

        int t[] = new int[nlVevo.getLength()];
        Node n1 = nlVevo.item(i);

        if(n1.getNodeType() == Node.ELEMENT_NODE) {
            Element e1 = (Element) n1;

            t[i]=Integer.parseInt(e1.getElementsByTagName("keszpenz").item(0).getTextContent());
            }

            if(max<t[i]) {
                max=t[i];
            }

        }

        System.out.println("\nKifizette készpénz Max: "+max);

    } catch (ParserConfigurationException e) {
        e.printStackTrace();
    } catch (SAXException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (XPathExpressionException e) {
        e.printStackTrace();
    }
}
}
:

```

Futási eredmény:

1.Lekérdezés: Lekérjük azt a vevőt akinek az 1 id-t adtuk

Jelenlegi Element :vevo
Vevő id : 1
nev : Bence
készpénz : 10000
kartya : 0
telefonszám : 06703333456

2.Lekérdezés: Lekérjük az étterem összes adatát

Jelenlegi Element :etterem
Etterem id :
nev : Egyszer étterem
telefonszám : 06704564653
Weblap : www.egyszeretterem.com

Jelenlegi Element :etterem
Etterem id :
nev : Jó étterem
telefonszám : 06704569873
Weblap : www.joetterem.com

Jelenlegi Element :etterem
Etterem id :
nev : Volt étterem
telefonszám : 06704659873
Weblap : www.voltetterem.com

3.Lekérdezés: Megnézzük hogy mennyi volt a bevétel

1. vevő készpénz: 10000

2. vevő készpénz: 500

3. vevő készpénz: 1500

Kifizette készpénz Összeg: 12000

4.Lekérdezés: Meg nézzük mennyi volt az átlaga a bevételnek

Kifizette készpénz Átlaga: 4000.0

5.Lekérdezés: Meg keressük a legnagyobb értéket

Kifizette készpénz Max: 10000