

# Jegyzőkönyv

Adatkezelés XML környezetben

Étterem rendelési folyamat

Készítette: Kércsi Bence

Neptunkód: ILVIYV

Dátum: 2022.11.19

## Tartalomjegyzék:

1. A feladat leírása
  - a. *A feladat témája*
  - b. *Egyed leírás*
  - c. *Kapcsolat leírás*
2. *Az ER modell*
3. *Az XDM modell*
4. *Az XML dokumentum*
5. *Az XMLSchema*
6. DOM Read
  - a. *DOM read (simple)*
  - b. *DOM read (enhanced)*
7. *DOM modify*
8. *DOM Query*

### ***A feladat témája:***

Én a féléves feladatom témájának az online étel rendelést választottam. Akár a Foodpandánál is hasonlóan működhet, ugyanis teljes mértékben nem látok a rendszerbe bele, illetve ez nyilván valóan egy „lebutított” verziója, ami ettől függetlenül még életképes. Az alap koncepció az az, hogy van egy vevőnk. A vevő kiválaszt a weblapon egy éttermet, amiből a számára szimpatikus ételeket megrendeli. Amint a megrendelés elment, a rendeléshez tartozik egy tranzakció, ez ugye az, hogy a vevő kifizeti a rendelést. A rendelésben szereplő ételek közben megrendelésre kerülnek, majd, ha készek, akkor a futár szállítja ki azt. Ez magas szinten az elképzelés.

### ***Egyed leírás:***

A vevőről a legfontosabb információk vannak tárolva, azaz az ID-ja, amit az oldalra kapna a regisztrálás során, a neve, az email címe, a címe, amire a kiszállítás is történik, illetve a telefonszáma, amiből lehet több is a fiókhoz kapcsolva.

A rendelést a beérkezés során szintén ellátjuk egy ID-val, ami az ő egyedi azonosítója, csakúgy, mint a vevőnél. Ezen kívül van egy étterem ID-ja is, ami az étteremnek mutat majd az egyedi azonosítójára azért, hogy az éttermet egyértelműen be lehessen azonosítani. A rendelésnek egy szintén nagyon fontos része, a státusz, hogy milyen fázisban is jár az, kiszállítás alatt van-e, vagy még csak megrendelve (de fizetve nem) vagy aktív, azaz készítés alatt stb. A rendeléshez szintén fontos még a leadási ideje, amikor a rendelésre az igény befutott, illetve a szállítási ideje, ami a rendelés leadásakor kiválasztott időpont, amikorra kéri ugye a vevő a rendelést, de ez egyben számított érték is lehet, az pedig úgy lenne lehetséges, hogy a leadási időhöz hozzáadjuk a legnagyobb elkészítési idővel rendelkező étel idejét majd egy szállítási idő becslést is.

A tranzakciónak szintén egy ID-ja van, egyébként meg egy banki utalás adataival rendelkezik, tehát ki volt a kedvezményező, ki a kedvezményezett, mi szerepel a közleményben és a legfontosabb, mi az összeg, amit ki kell fizetni.

A futárról kevesebb adatot tárolunk, neki is van egy azonosítója, egy neve, illetve nála a legfontosabb adatok az az, hogy mivel közlekedik és mi a pontos tartózkodási helye.

A következő elemünk az étel, amit a nevről lehet felismerni. Az ételnél, amiket fontosnak találtam az az ára természetesen, hogy milyen allergiát kiváltó alapanyagok jelenhetnek meg benne és az elkészítési idő, ami a már megbeszéltek kalkulációhoz szükséges, illetve a vevőt is érdekelheti, mert leszűrhet belőle információkat (akár azt is, hogy pizza esetén mirelit-e, előre fagyasztott tésztás, avagy sem).

Végül, az étteremnek nem meglepő módon szintén van egy egyedi azonosítója, ami ez esetben is az ID névre hallgat. Az étteremről a nevét tároljuk még ugyanis az elég fontos. Ezekén kívül az étteremnek a helyzete, ami még különösen fontos, ugyanis ez is szükséges a kiszállítási idő becslésének kalkulálásához, vagy esetleg egy étterem lánc esetében erről tudnák beazonosítani a vásárlókat, hogy melyikről is van szó. Az

éttermeknek még az értékelésüket tárolom el, hogy segítse az étterem választását a vevőnek.

### ***Kapcsolat leírás:***

Vevő-Rendelés: Egy vevőhöz több rendelés is lehet beregisztrálva, akár aktív akár kiszállított, vagy bármely más státuszban, akár azonos státuszban is szerepelhet több rendelés, ezzel szemben egy rendelés csak egy fiókhoz tartozik. Ezekből következik, hogy a két egyed között **egy-több** kapcsolat van.

Rendelés-Tranzakció: Egy rendeléshez egy összeg tartozik, amit a vevőnek ki kell fizetni, amit egy tranzakció keretein belül várjuk tőle, ugyan így, egy tranzakció egy rendelés fizetésére alkalmas, ezek miatt itt **egy-egy** kapcsolat található meg. Érdekeség, hogy ez egy kötelező kapcsolat, tehát mindenképp kell egy tranzakció, ami tartozik a rendeléshez és fordítva is.

Rendelés-Futár: Nem szeretnénk túldolgoztatni és leharcolni a futárjainkat, azt sem szeretnénk, hogy a rendelésekben elveszenek emiatt a Foodpandának egy olyan utópiáját valósítjuk meg, ahol egy futár egyszerre csak egy rendelés kiszállításáért felelős. A rendelés lehet nagy mértékű, ezekre felkészültek a futárjaink, így egy rendeléssel mindenképp megbirkózik egy futár, emiatt itt is **egy-egy** kapcsolat alakul ki.

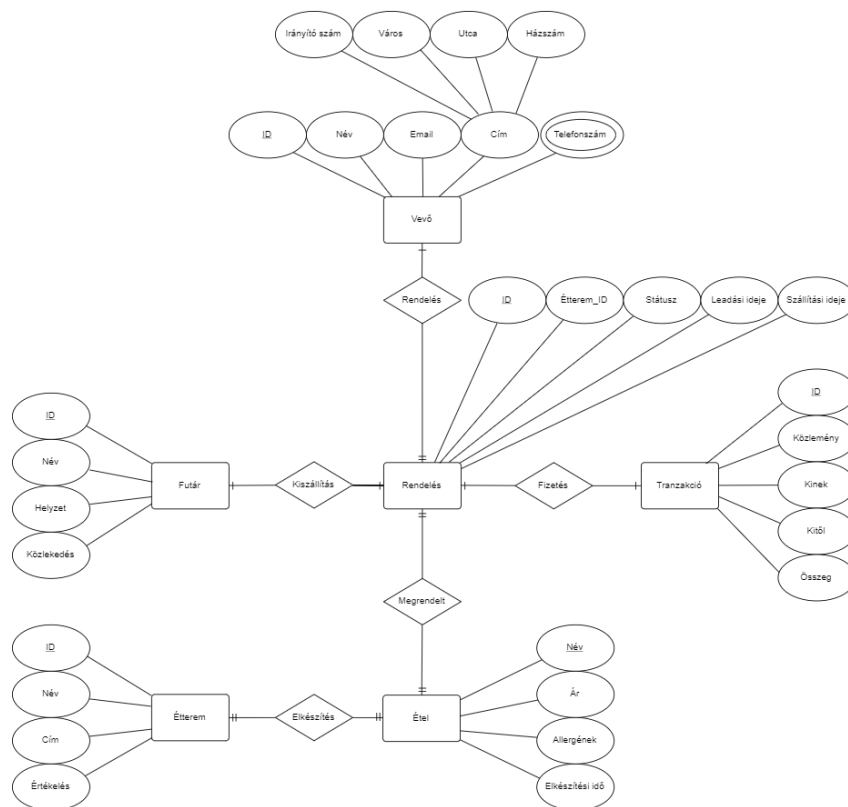
Rendelés-Étel: Szomorú is lenne, ha egyszerre csak egy étel lenne rendelhető, emiatt természetesen egy rendelés keretein belül több ételt is rendelhetünk (bár egy rendelés csak egy étteremből lehetséges így ez korlátozza), viszont, ha egy rántott húst talál valaki, azt nem lehet egyértelműen rendeléshez kötni ugyanis fordítva is igaz, hogy egy adott ételt több vevő is várhatja otthon. Az említettek miatt a kapcsolat jelen esetben **több-több**.

Étel-Étterem: Ez az előzőhöz hasonló szituáció lesz, ugyanis az éttermek közötti versengés eredménye részben az, hogy ugyan azt a terméket is próbálják egyre jobban elkészíteni, hogy ők nyerjék meg a vevők szívét. Ebben a mondatban benne is volt a kulcs, hogy egy étel több étteremnél is megjelenhet. Viszont egy étterem biztos nem maradna hosszú távon versenyképes, ha csak egy terméke lenne, így az éttermek is törekednek minél bővebb étlappal szolgálni, ezért egy étteremben is lehet több étel. Így mint elején lelőttem a poént, ez is érzékelhetően **több-több** kapcsolat lesz.

### ***Az ER modell:***

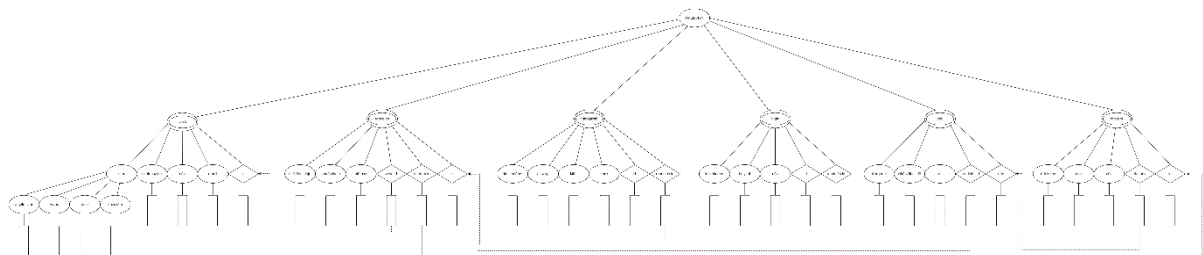
Az ER modell megtalálható a Github-on, a forráskód az ERILVIYV.txt-ben, ahol egy GitMind cloud-ra mutató link van, de PDF és PNG formátumban is megtalálható az ER modell. A Kicsinyített verzióját ide is beilleszttem. Az egyedekről, kapcsolataikról és attribútumaikról feljebb beszéltem már, ezért itt inkább a diagrammot emelem ki.

Igyekeztem egy sorba, rendezetten elkészíteni a diagrammot.



### Az XDM modell:

Az XDM modell szintén megtalálható a githubom-on, hasonlóan az XDMILVIYV tartalmazza a forrásra mutató linket, és erről is elkészült egy PNG és PDF verzió is. Az XDM-nél minden egyed a root elemből származik le, ezért az összes egy sorban szerepel a root gyerekeként, ők dupla elipszisben, ugyanis több is előfordulhat belőlük. Itt az ER-hez hasonlóan elipszisben szerepelnek az attribútumok, viszont az egyedi kulcsok (primary/foreign key) rombusz alakzatban. Ezeknek mind van egy field része, ami tartalmazná az adatot, ez a téglalap alatta. Amint megvoltam ezekkel, már csak az hiányzott, hogy a kulcsokat beállítsam egymásra, akire mutatnak.



### Az XML dokumentum:

Az XML dokumentum elkészítése ennyi segéd modell mellett már gyerekjáték volt, csak a kreativitásom tette próbára az értékek kitalálásánál. Itt sok nehézségbe nem ütköztem. Van egy root az XMLILVIYV, aztán elkezdtem felépíteni az XML dokumentumot a modellek alapján. Az XML megtalálható a github-on XMLILVIYV.xml néven, illetve a róla készült kép is.

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

```
<XMLILVIYV>
<!-- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:namespaceSchemaLocation="XSDILVIYV.xsd" -->
<vevő id="1">
  <név>Kéresi Bence</név>
  <email>keresibence@gmail.com</email>
  <telefonszám>01234567890</telefonszám>
  <cím>
    <irányítószám>3535</irányítószám>
    <város>Miskolc</város>
    <utca>Szentgyörgy</utca>
    <házzszám>1</házzszám>
  </cím>
  <rendelések>
    <rendelés id="1">
      <étterem_id>1</étterem_id>
      <státusz>Aktív</státusz>
      <leadási_ideje>2022-11-10 14:30</leadási_ideje>
      <szállítási_ideje>2022-11-10 15:45</szállítási_ideje>
      <tranzakció id="1">
        <összeg>4500</összeg>
        <kezői>11</kezői>
        <keinek>000</keinek>
        <közzlemény>Rántott hús + Töltött káposzta</közzlemény>
      </tranzakció>
      <ételek>
        <étel név="Rántott hús">
          <ár>2000</ár>
          <elkészítési_id>32</elkészítési_id>
          <allergének>Glutén</allergének>
        </éttermek>
        <étterem id="1">
          <cím>Példa cím 1</cím>
          <név>Példa étterem 1</név>
          <értékelés>4.5</értékelés>
        </étterem>
      </éttermek>
    </étel>
    <étel név="Töltött káposzta">
      <ár>2500</ár>
      <elkészítési_id>65</elkészítési_id>
      <allergének>Laktóz</allergének>
    </éttermek>
    <étterem id="1">
      <cím>Példa cím 1</cím>
      <név>Példa étterem 1</név>
      <értékelés>4.5</értékelés>
    </étterem>
  </éttermek>
</étel>
</ételek>
<futár id="1">
  <név>Sándor</név>
  <helyzet>Avas</helyzet>
  <közzlekedés>Bicikli</közzlekedés>
</futár>
</rendelés>
</rendelések>
</vevő>
<vevő id="2">
  <név>Glonczi Bence</név>
  <email>glonozibence@gmail.com</email>
  <telefonszám>01234567891</telefonszám>
  <cím>
    <irányítószám>3535</irányítószám>
    <város>Miskolc</város>
    <utca>Szentgyörgy</utca>
    <házzszám>2</házzszám>
  </cím>
  <rendelések>
    <rendelés id="2">
      <étterem_id>1</étterem_id>
      <státusz>Kiszállítás alatt</státusz>
      <leadási_ideje>2022-11-10 12:20</leadási_ideje>
      <szállítási_ideje>2022-11-10 16:35</szállítási_ideje>
      <tranzakció id="2">
        <összeg>3500</összeg>
        <kezői>222</kezői>
        <keinek>000</keinek>
        <közzlemény>Rántott hús + Borsó leves</közzlemény>
      </tranzakció>
      <ételek>
        <étel név="Rántott hús">
          <ár>2000</ár>
          <elkészítési_id>32</elkészítési_id>
          <allergének>Glutén</allergének>
        </éttermek>
        <étterem id="1">
          <cím>Példa cím 1</cím>
          <név>Példa étterem 1</név>
          <értékelés>4.5</értékelés>
        </étterem>
      </éttermek>
    </étel>
  </rendelések>
</vevő>
```

```
</étterem>
</éttermek>
</étel>
<étel név="Borsó loves">
  <ár>1500</ár>
  <elkészítési_id>65</elkészítési_id>
  <allergének>Laktóz</allergének>
  <éttermek>
    <étterem_id="1">
      <cím>Példa cím 1</cím>
      <név>Példa étterem 1</név>
      <értékelés>4.5</értékelés>
    </étterem>
    <étterem_id="3">
      <cím>Példa cím 3</cím>
      <név>Példa étterem 3</név>
      <értékelés>4.1</értékelés>
    </étterem>
  </éttermek>
</étel>
</ételek>
<futár_id="2">
  <név>Sándor</név>
  <helyzet>Tapolca</helyzet>
  <közlekedés>Roller</közlekedés>
</futár>
</rendelés>
</rendelések>
</vevő>
<vevő_id="3">
  <név>Kopasz János</név>
  <email>kopaszjanos@gmail.com</email>
  <telefonszám>01234567892</telefonszám>
  <cím>
    <irányítószám>3535</irányítószám>
    <város>Miskolc</város>
    <utca>Arany János</utca>
    <házzsám>1</házzsám>
  </cím>
  <rendelések>
    <rendelés_id="3">
      <étterem_id>2</étterem_id>
      <státusz>Aktív</státusz>
      <leadási_ideje>2022-11-10 12:20</leadási_ideje>
      <szállítási_ideje>2022-11-11 11:45</szállítási_ideje>
      <tranzakció_id="3">
        <összeg>3500</összeg>
        <kezői>333</kezői>
        <keinek>000</keinek>
        <közlemény>Hamburger</közlemény>
      </tranzakció>
    </éttelek>
    <étel név="Hamburger">
      <ár>2500</ár>
      <elkészítési_id>15</elkészítési_id>
      <allergének>Glutén, laktóz</allergének>
      <éttermek>
        <étterem_id="1">
          <cím>Példa cím 1</cím>
          <név>Példa étterem 1</név>
          <értékelés>4.5</értékelés>
        </étterem>
        <étterem_id="2">
          <cím>Példa cím 2</cím>
          <név>Példa étterem 2</név>
          <értékelés>3.8</értékelés>
        </étterem>
        <étterem_id="4">
          <cím>Példa cím 4</cím>
          <név>Példa étterem 4</név>
          <értékelés>3.3</értékelés>
        </étterem>
        <étterem_id="5">
          <cím>Példa cím 5</cím>
          <név>Példa étterem 5</név>
          <értékelés>2.4</értékelés>
        </étterem>
      </éttermek>
    </étel>
  </ételek>
  <futár_id="3">
    <név>Shándor</név>
    <helyzet>Egyetem</helyzet>
    <közlekedés>Görkovi</közlekedés>
  </futár>
</rendelés>
</rendelések>
</vevő>
<vevő_id="4">
  <név>Szabó Renáta</név>
  <email>szaborenata@gmail.com</email>
  <telefonszám>01234567865</telefonszám>
  <cím>
```

```
<irányítószám>3529</irányítószám>
<város>Miskolc</város>
<utca>Arany János tér</utca>
<házszám>12</házszám>
</cím>
<rendelések>
  <rendelés_id="4">
    <étterem_id>2</étterem_id>
    <státusz>Aktív</státusz>
    <leadási_id>2022-11-10 12:19</leadási_id>
    <szállítási_id>2022-11-11 11:46</szállítási_id>
    <tranzakció_id="4">
      <összeg>3500</összeg>
      <keiő>444</keiő>
      <keinek>000</keinek>
      <közzemény>Hamburger</közzemény>
    </tranzakció>
    <ételek>
      <étel_név="Hamburger">
        <ár>2500</ár>
        <elkészítési_id>15</elkészítési_id>
        <allergének>Glutén, laktóz</allergének>
      <éttermek>
        <étterem_id="1">
          <cím>Példa cím 1</cím>
          <név>Példa étterem 1</név>
          <értékelés>4.5</értékelés>
        </étterem>
        <étterem_id="2">
          <cím>Példa cím 2</cím>
          <név>Példa étterem 2</név>
          <értékelés>3.8</értékelés>
        </étterem>
        <étterem_id="4">
          <cím>Példa cím 4</cím>
          <név>Példa étterem 4</név>
          <értékelés>3.3</értékelés>
        </étterem>
        <étterem_id="5">
          <cím>Példa cím 5</cím>
          <név>Példa étterem 5</név>
          <értékelés>2.4</értékelés>
        </étterem>
      </éttermek>
    </étel>
  </ételek>
  <futár_id="4">
    <név>Józsa</név>
    <helyzet>Tapolca</helyzet>
    <közzekedés>Gyalog</közzekedés>
  </futár>
</rendelés>
</rendelések>
</veő_id="5">
  <név>Kérsi Tibor</név>
  <email>kérsitibor@gmail.com</email>
  <telefonszám>01234567823</telefonszám>
  <cím>
    <irányítószám>3860</irányítószám>
    <város>Eöcs</város>
    <utca>Jókai út</utca>
    <házszám>13</házszám>
  </cím>
  <rendelések>
    <rendelés_id="5">
      <étterem_id>3</étterem_id>
      <státusz>Aktív</státusz>
      <leadási_id>2022-11-11 12:19</leadási_id>
      <szállítási_id>2022-11-12 11:46</szállítási_id>
      <tranzakció_id="5">
        <összeg>1500</összeg>
        <keiő>555</keiő>
        <keinek>000</keinek>
        <közzemény>Borsó leves</közzemény>
      </tranzakció>
      <ételek>
        <étel_név="Borsó leves">
          <ár>1500</ár>
          <elkészítési_id>65</elkészítési_id>
          <allergének>Laktóz</allergének>
        <éttermek>
          <étterem_id="1">
            <cím>Példa cím 1</cím>
            <név>Példa étterem 1</név>
            <értékelés>4.5</értékelés>
          </étterem>
          <étterem_id="3">
            <cím>Példa cím 3</cím>
            <név>Példa étterem 3</név>
            <értékelés>4.1</értékelés>
          </étterem>
        </éttermek>
      </étel>
    </rendelés>
  </ételek>
</rendelések>
```





```

<xs:complexType>
  <xs:sequence>
    <xs:element name="étterem" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="cím" type="xs:string"/>
          <xs:element name="név" type="xs:string"/>
          <xs:element name="értékelés" type="xs:double"/>
        </xs:sequence>
        <xs:attribute ref="id"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<xs:sequence>
  <xs:element name="név" type="xs:string"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="futár">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="név" type="xs:string"/>
      <xs:element name="helyzet" type="xs:string"/>
      <xs:element name="közlekedés" type="xs:string"/>
    </xs:sequence>
    <xs:attribute ref="id"/>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute ref="id"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute ref="id"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute ref="id"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:unique name="vevő_id">
  <xs:selector xpath="vevő"/>
  <xs:field xpath="@id"/>
</xs:unique>
<xs:unique name="rendelés_id">
  <xs:selector xpath="rendelés"/>
  <xs:field xpath="@id"/>
</xs:unique>
<xs:unique name="tranzakció_id">
  <xs:selector xpath="tranzakció"/>
  <xs:field xpath="@id"/>
</xs:unique>
<xs:unique name="futár_id">
  <xs:selector xpath="futár"/>
  <xs:field xpath="@id"/>
</xs:unique>
<xs:unique name="étel_név">
  <xs:selector xpath="étel"/>
  <xs:field xpath="@név"/>
</xs:unique>
<xs:unique name="étterem_id">
  <xs:selector xpath="étterem"/>
  <xs:field xpath="@id"/>
</xs:unique>

<xs:keyref ref="vevő_id" name="Rendelés_vevő">
  <xs:selector xpath="rendelés"/>
  <xs:field xpath="@vevő_id"/>
</xs:keyref>
<xs:keyref ref="rendelés_id" name="Tranzakció_rendelés">
  <xs:selector xpath="tranzakció"/>
  <xs:field xpath="@id"/>
</xs:keyref>
<xs:keyref ref="rendelés_id" name="Futár_rendelés">
  <xs:selector xpath="futár"/>
  <xs:field xpath="@id"/>
</xs:keyref>
<xs:keyref ref="rendelés_id" name="Étel_rendelés">
  <xs:selector xpath="étel"/>
  <xs:field xpath="@id"/>
</xs:keyref>
<xs:keyref ref="étel_név" name="Étterem_étel">
  <xs:selector xpath="étterem"/>
  <xs:field xpath="@étel_név"/>
</xs:keyref>

```

```

<xs:keyref refer="étterem_id" name="Rendelés_étterem">
  <xs:selector xpath="rendelés" />
  <xs:field xpath="@étterem_id" />
</xs:keyref>
</xs:element>
</xs:schema>

```

## DOM read (simple):

Az adatolvasás igazából pár elemnek az ismételt használata volt csak, az XML szerkezetét kellett a kódban is felépíteni. Az XML-t strukturáltan (tabulátorokkal betolva) jeleníti meg a kimeneten, illetve kiírja file-ba is a végeredményt. A kód azért nem lett kommentelve (mint a többi sem), mert elég magát magyarázóra sikerült, így rögtön lehet látni a kódba, hogy mit szerettem volna és miért. A változókat is próbáltam elég beszédesre választani. Maga a kód az olvasásról (az első típusú):

```

package hu.dom.parse.ILVIYV;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

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

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

public class DomReadILVIYV {

    public static void main(String argv[]) throws SAXException,
        IOException, ParserConfigurationException {

        File xmlFile = new File("E:\\Gît\\ILVIYV_XML\\gyakr\\Feleves beadando\\Program\\XMLILVIYV.xml");
        String text="";

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();
        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        NodeList vevList = doc.getElementsByTagName("vev");
        for (int i = 0; i < vevList.getLength(); i++) {

            Node vevNode = vevList.item(i);

            if (vevNode.getNodeType() == Node.ELEMENT_NODE) {
                Element vev = (Element) vevNode;

                String vev_id = vev.getAttribute("id");
                text+="A vev ID-ja: "+vev_id+"\n";
                Node nev = vev.getElementsByTagName("nev").item(0);
                String nev_text = nev.getTextContent();
                text+="A vev neve: "+nev_text+"\n";
                Node email = vev.getElementsByTagName("email").item(0);
                String email_text = email.getTextContent();
                text+="A vev email címe: "+email_text+"\n";
                Node telefonszám = vev.getElementsByTagName("telefonszám").item(0);
                String telefonszám_text = telefonszám.getTextContent();
                text+="A vev telefonszáma: "+telefonszám_text+"\n";

                Node címNode=vev.getElementsByTagName("cím").item(0);
                Element cím = (Element) címNode;
                Node irányítószám = cím.getElementsByTagName("irányítószám").item(0);
                String irányítószám_text = irányítószám.getTextContent();
                text+="A vev címének irányítószáma: "+irányítószám_text+"\n";
                Node város = cím.getElementsByTagName("város").item(0);
                String város_text = város.getTextContent();
                text+="A vev városa: "+város_text+"\n";
                Node utca = cím.getElementsByTagName("utca").item(0);
                String utca_text = utca.getTextContent();
                text+="A vev utódja: "+utca_text+"\n";
                Node házzszám = cím.getElementsByTagName("házzszám").item(0);
                String házzszám_text = házzszám.getTextContent();
                text+="A vev házzsáma: "+házzszám_text+"\n";

                NodeList rendelésekNodeList = vev.getElementsByTagName("rendelések");
                Node rendelésekNode=rendelésekNodeList.item(0);
                Element rendelésekElement=(Element) rendelésekNode;
                NodeList rendelésNodeList=rendelésekElement.getElementsByTagName("rendelés");
                for (int j = 0; j < rendelésNodeList.getLength(); j++) {

```

```

Node rendelésNode = rendelésNodeList.item(j);
Element rendelés = (Element) rendelésNode;

String rendelés_id = rendelés.getAttribute("id");
text+="\tA rendelés ID-ja: "+rendelés_id+"\n";
Node étterem_id = rendelés.getElementsByTagName("étterem_id").item(0);
String étterem_id_text = étterem_id.getTextContent();
text+="\tA rendelés éttermének ID-ja: "+étterem_id_text+"\n";
Node státusz = rendelés.getElementsByTagName("státusz").item(0);
String státusz_text = státusz.getTextContent();
text+="\tA rendelés státusza: "+státusz_text+"\n";
Node leadási_ideje = rendelés.getElementsByTagName("leadási_ideje").item(0);
String leadási_ideje_text = leadási_ideje.getTextContent();
text+="\tA rendelés leadási ideje: "+leadási_ideje_text+"\n";
Node szállítási_ideje = rendelés.getElementsByTagName("szállítási_ideje").item(0);
String szállítási_ideje_text = szállítási_ideje.getTextContent();
text+="\tA rendelés szállítási ideje: "+szállítási_ideje_text+"\n";

Node tranzakcióNode=rendelés.getElementsByTagName("tranzakció").item(0);
Element tranzakció=(Element) tranzakcióNode;
String tranzakció_id=tranzakció.getAttribute("id");
text+="\tA tranzakció ID-ja: "+tranzakció_id+"\n";
Node összeg=tranzakció.getElementsByTagName("összeg").item(0);
String összeg_text=összeg.getTextContent();
text+="\tA tranzakció összege: "+összeg_text+"\n";
Node kitétele=tranzakció.getElementsByTagName("kitétele").item(0);
String kitétele_text=kitétele.getTextContent();
text+="\tA tranzakció feladója: "+kitétele_text+"\n";
Node kineke=tranzakció.getElementsByTagName("kineke").item(0);
String kineke_text=kineke.getTextContent();
text+="\tA tranzakció kedvezményezettje: "+kineke_text+"\n";
Node közlemény=tranzakció.getElementsByTagName("közlemény").item(0);
String közlemény_text=közlemény.getTextContent();
text+="\tA tranzakció közleménye: "+közlemény_text+"\n";

NodeList ételkekNodeList = rendelés.getElementsByTagName("ételke");
Node ételkekNode=ételkekNodeList.item(0);
Element ételkeElement=(Element) ételkekNode;
NodeList ételNodeList=ételkeElement.getElementsByTagName("étel");
for (int k = 0; k < ételNodeList.getLength(); k++) {

    Node ételNode = ételNodeList.item(k);
    Element étel = (Element) ételNode;

    String étel_név = étel.getAttribute("név");
    text+="\t\tAz étel neve: "+étel_név+"\n";
    Node ár=étel.getElementsByTagName("ár").item(0);
    String ár_text=ár.getTextContent();
    text+="\t\tAz étel ára: "+ár_text+"\n";
    Node elkészítési_idő=étel.getElementsByTagName("elkészítési_idő").item(0);
    String elkészítési_idő_text=elkészítési_idő.getTextContent();
    text+="\t\tAz étel elkészítési ideje: "+elkészítési_idő_text+"\n";
    Node allergének=étel.getElementsByTagName("allergének").item(0);
    String allergének_text=allergének.getTextContent();
    text+="\t\tAz ételben lévő allergének: "+allergének_text+"\n";

    NodeList éttermekNodeList = étel.getElementsByTagName("éttermek");
    Node éttermekNode=éttermekNodeList.item(0);
    Element éttermElement=(Element) éttermekNode;
    NodeList étteremNodeList=éttermElement.getElementsByTagName("étterem");
    for (int l = 0; l < étteremNodeList.getLength(); l++) {

        Node étteremNode = étteremNodeList.item(l);
        Element étterem = (Element) étteremNode;

        String étterem_own_id = étterem.getAttribute("id");
        text+="\t\t\tAz étterem ID-ja: "+étterem_own_id+"\n";
        Node étterem_cim=étterem.getElementsByTagName("cim").item(0);
        String étterem_cim_text=étterem_cim.getTextContent();
        text+="\t\t\tAz étterem címe: "+étterem_cim_text+"\n";
        Node étterem_név=étterem.getElementsByTagName("név").item(0);
        String étterem_név_text=étterem_név.getTextContent();
        text+="\t\t\tAz étterem neve: "+étterem_név_text+"\n";
        Node értékelés=étterem.getElementsByTagName("értékelés").item(0);
        String értékelés_text=értékelés.getTextContent();
        text+="\t\t\tAz étterem értékelése: "+értékelés_text+"\n";

    }

    Node futárNode=rendelés.getElementsByTagName("futár").item(0);
    Element futár=(Element) futárNode;
    String futár_id=futár.getAttribute("id");
    text+="\tA futár ID-ja: "+futár_id+"\n";
    Node futár_név=futár.getElementsByTagName("név").item(0);
    String futár_text=futár_név.getTextContent();
    text+="\tA futár neve: "+futár_text+"\n";
    Node helyzet=futár.getElementsByTagName("helyzet").item(0);
    String helyzet_text=helyzet.getTextContent();
    text+="\tA futár helyzete: "+helyzet_text+"\n";
    Node közlekedés=futár.getElementsByTagName("közlekedés").item(0);
    String közlekedés_text=közlekedés.getTextContent();
    text+="\tA futár közlekedésének módja: "+közlekedés_text+"\n";

}

```

```

        text+="\n";
    }
}
FileWriter readResultFile;

try
{
    readResultFile = new FileWriter("E:\\Git\\LVIV_XMLGyak\\Feleves beadando\\Program\\ReadResult.txt");

    BufferedWriter readResult = new BufferedWriter(readResultFile);
    readResult.write(text);

    readResult.close();
    System.out.println("Written successfully");
}
catch (IOException except)
{
    except.printStackTrace();
}

    System.out.print(text);
}
}

```

A

kimenet

(részlet):

```

A vevő ID-ja: 1
A vevő neve: Kércsi Bence
A vevő email címe: kercsibence1@gmail.com
A vevő telefonszáma: 01234567890
A vevő címének irányítószáma: 3535
A vevő városa: Miskolc
A vevő utcája: Szentgyörgy
A vevő házzsáma: 1
  A rendelés ID-ja: 1
    A rendelés éttermének ID-ja: 1
    A rendelés státusza: Aktív
    A rendelés leadási ideje: 2022-11-10 14:30
    A rendelés szállítási ideje: 2022-11-10 15:45
    A tranzakció ID-ja: 1
    A tranzakció összege: 4500
    A tranzakció feladója: 111
    A tranzakció kedvezményezettje: 000
    A tranzakció közleménye: Rántott hús + Töltött káposzta
      Az étel neve: Rántott hús
      Az étel ára: 2000
      Az étel elkészítési ideje: 32
      Az ételben lévő allergének: Glutén
        Az étterem ID-ja: 1
        Az étterem címe: Példa cím 1
        Az étterem neve: Példa étterem 1
        Az étterem értékelése: 4.5
      Az étel neve: Töltött káposzta
      Az étel ára: 2500
      Az étel elkészítési ideje: 65
      Az ételben lévő allergének: Laktóz
        Az étterem ID-ja: 1
        Az étterem címe: Példa cím 1
        Az étterem neve: Példa étterem 1
        Az étterem értékelése: 4.5
    A futár ID-ja: 1
    A futár neve: Sándy
    A futár helyzete: Ávas
    A futár közlekedésének módja: Bicikli

```

***DOM read (enhanced):***

Az olvasásnak a második fajtája: Abban különbözik, hogy ez nem csak az XML-t írja ki, ugyan abban a formában, amiben beolvassa, hanem az éttermeket leszűkíti. Az éttermek alatt több étterem is szerepelhet, ugyanis egy ételt több is készítheti, és azt, hogy melyik étterem a „szerencsés” a rendelésben lévő étterem ID alapján azonosítjuk be. Ez az olvasás és kiírás ebben különbözik, hogy ez csak azt az éttermet fogja kiírni

számunkra, amelyik a ténylegesen kiválasztott. A kódban a különbség:

```
String étterem_own_id = étterem.getAttribute("id");
if(étterem_id_text.equals(étterem_own_id)) {
    text+="\t\t\tAz étterem ID-ja: "+étterem_own_id+"\n";
    Node étterem_cim=étterem.getElementsByTagName("cím").item(0);
    String étterem_cim_text=étterem_cim.getTextContent();
    text+="\t\t\tAz étterem címe: "+étterem_cim_text+"\n";
    Node étterem_név=étterem.getElementsByTagName("név").item(0);
    String étterem_név_text=étterem_név.getTextContent();
    text+="\t\t\tAz étterem neve: "+étterem_név_text+"\n";
    Node értékelés=étterem.getElementsByTagName("értékelés").item(0);
    String értékelés_text=értékelés.getTextContent();
    text+="\t\t\tAz étterem értékelése: "+értékelés_text+"\n";
}
```

### *DOM modify:*

Az adatmódosítás esetén nekem a futár nevére esett. Tehát ha ezt futtatjuk, akkor a konzolra ki fogja a futár nevét írni és kéri, hogy írjunk be egy új értéket a számára. Ha ezt megtesszük, kéri a következő futárra is. Ez addig megy, még van futár, ez jelen esetben 3 egyed nálam. A módosítás során egy másik MXL file-t hoz létre, amiben a módosított értékek szerepelnek. A program:

```
package hu.domparsa.ILVIYV;

import java.io.File;
import java.io.IOException;
import java.util.Scanner;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

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

public class DomModifyILVIYV {

    public static void main(String[] args) throws SAXException, IOException, ParserConfigurationException, TransformerConfigurationException, TransformerException {

        File xmlFile = new File("E:\\Git\\ILVIYV_XMLGyak\\Feleves beadando\\Program\\XMLILVIYV.xml");
        File xmlOutputFile = new File("E:\\Git\\ILVIYV_XMLGyak\\Feleves beadando\\Program\\XMLILVIYVModifyResult.xml");

        Scanner in = new Scanner(System.in);

        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder dBuilder = factory.newDocumentBuilder();

        Document doc = dBuilder.parse(xmlFile);

        doc.getDocumentElement().normalize();

        NodeList nList = doc.getElementsByTagName("futár");

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

            Node nNode = nList.item(i);
```

```

        Element elem = (Element) nNode;

        Node node1 = elem.getElementsByTagName("név").item(0);
        String nev = node1.getTextContent();

        System.out.println("A futár jelenlegi neve: " + nev);
        System.out.println("Adja meg a futár új nevét: ");
        String newname = in.next();
        node1.setTextContent(newname);
    }

    in.close();

    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();

    DOMSource source = new DOMSource(doc);

    StreamResult result = new StreamResult(xmlOutputFile);

    transformer.transform(source, result);
}
}

```

## DOM Query:

Az adat lekérdezés esetében 5 kifejezést értékel ki a program, és mindig a rendelés id-t mutatja, amelyikre igaz lett az állítás. Ez is megjeleníti a kimeneten is az eredményt, illetve ő is hoz létre egy TXT-t.

A

kód:

```

package hu.dom.parse.LVIV;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
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;

public class DomQueryLVIV {

    static String text="";
    public static void main(String[] args) {

        String expression1="//rendelés";
        XPathQuery(expression1,"Az összes rendelés");
        String expression2="//tranzakció[@id=\'2\']";
        XPathQuery(expression2,"A rendelés aminek tranzakció id-ja 2");
        String expression3="//tranzakció[összeg>4000]";
        XPathQuery(expression3,"A 4000-nél drágább rendelések");
        String expression4="//rendelés[státusz=\'Aktív\']";
        XPathQuery(expression4,"Aktív státuszú rendelések");
        String expression5="//futár[nev=\'Sándy\']";
        XPathQuery(expression5,"Azok a rendelések amiket Sándy szállít ki");

        FileWriter queryResultFile;

        try
        {
            queryResultFile = new FileWriter("E:\\Git\\LVIV_XML\\gyak\\Feleves beadando\\Program\\QueryResult.txt");

            BufferedWriter queryResultWrite = new BufferedWriter(queryResultFile);
            queryResultWrite.write(text);

            queryResultWrite.close();
            System.out.println("Written successfully");
        }
        catch (IOException except)
        {
            except.printStackTrace();
        }

        System.out.print(text);

    }

    public static void XPathQuery(String expression, String kérdés) {
        text += "\n";
        try {

```

```

File xmlFile=new File("E:\\Git\\ILVIYV_XMLGyak\\Feleves beadando\\Program\\XMLILVIYV.xml");
DocumentBuilderFactory documentBuilderFactory=DocumentBuilderFactory.newInstance();
DocumentBuilder documentBuilder=documentBuilderFactory.newDocumentBuilder();

Document document=documentBuilder.parse(xmlFile);

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

NodeList nodeList=(NodeList) xPath.compile(expression).evaluate(document, XPathConstants.NODESET);

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

    if(node.getNodeType()==Node.ELEMENT_NODE && node.getNodeName().equals("rendelés")){(
        Element element=(Element) node;
        text+=kérdés+"\n";
        text+="Rendelés ID: "+element.getAttribute("id")+"\n";
        text+="-----"+ "\n";
    }
}
}catch(Exception e) {

}
}
}

```

A program XPATH lekérdezéseket csinál. A végeredmény (részlet):

```

Az összes rendelés
Rendelés ID: 1
-----

Az összes rendelés
Rendelés ID: 2
-----

Az összes rendelés
Rendelés ID: 3
-----

A rendelés aminek tranzakció ID-ja 2
Rendelés ID: 2
-----

```