Jegyzőkönyv

Adatkezelés XML környezetben

Féléves feladat

Készítette:Kércsi Bence Neptunkód: ILVIYV

Dátum:2022.11.19

***Féléves 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.

***Alacsonyabb szinten (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 nevéről 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 megbeszélt 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. Ezeken 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 vásárlók, hogy melyikről is van szó. Az éttermeknek még az értékelésüket tárolom el, hogy segítse az étterem választását a vevőnek.

***Kapcsolatok az egyedek között:***

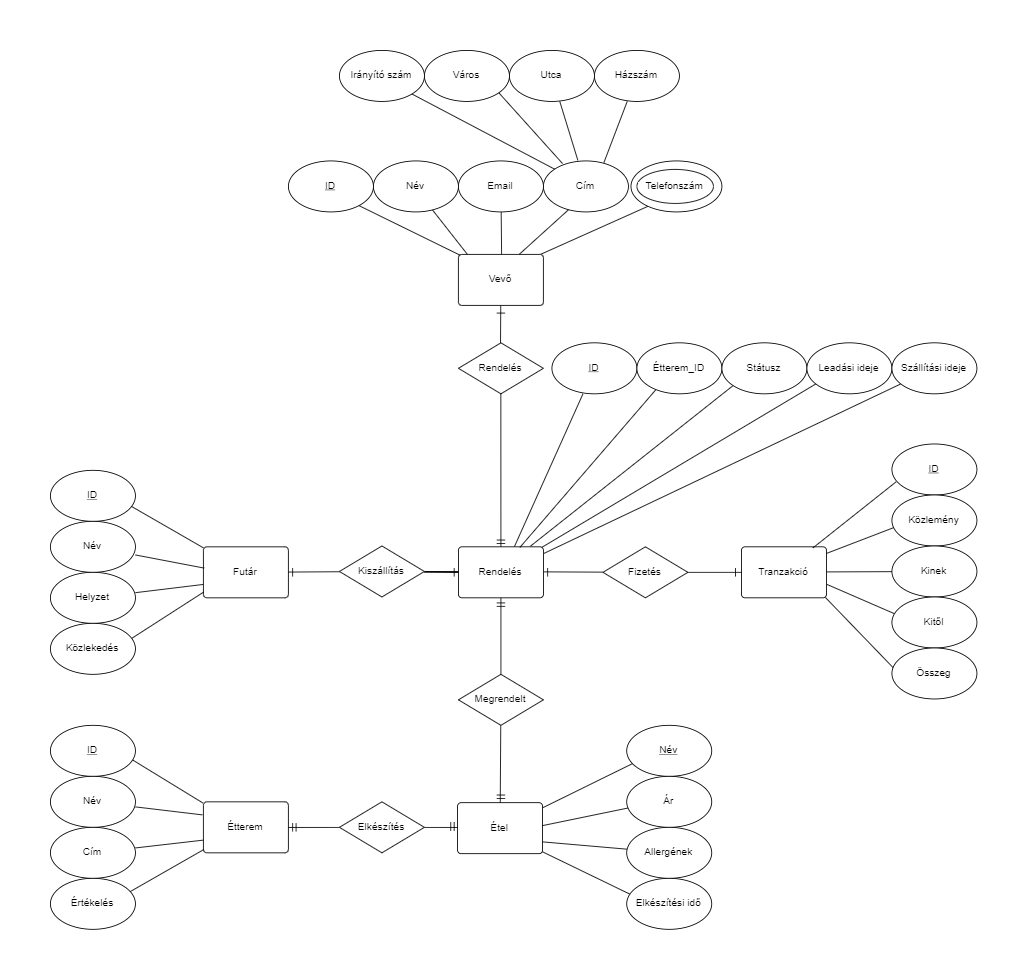
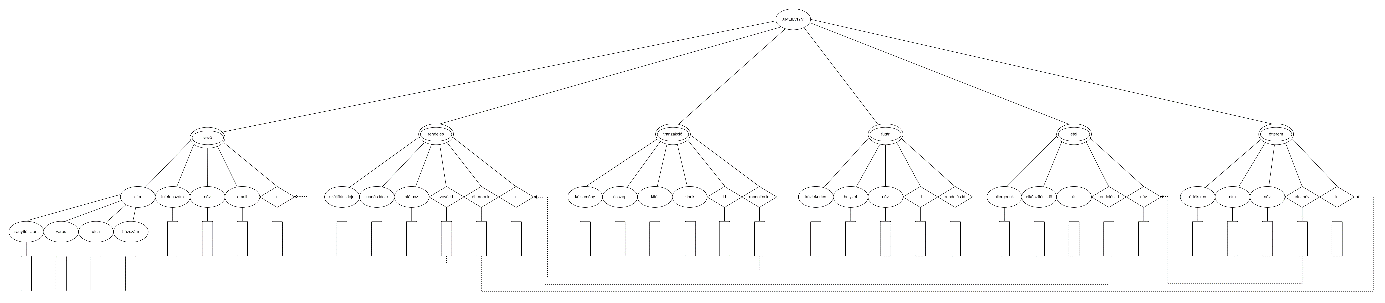
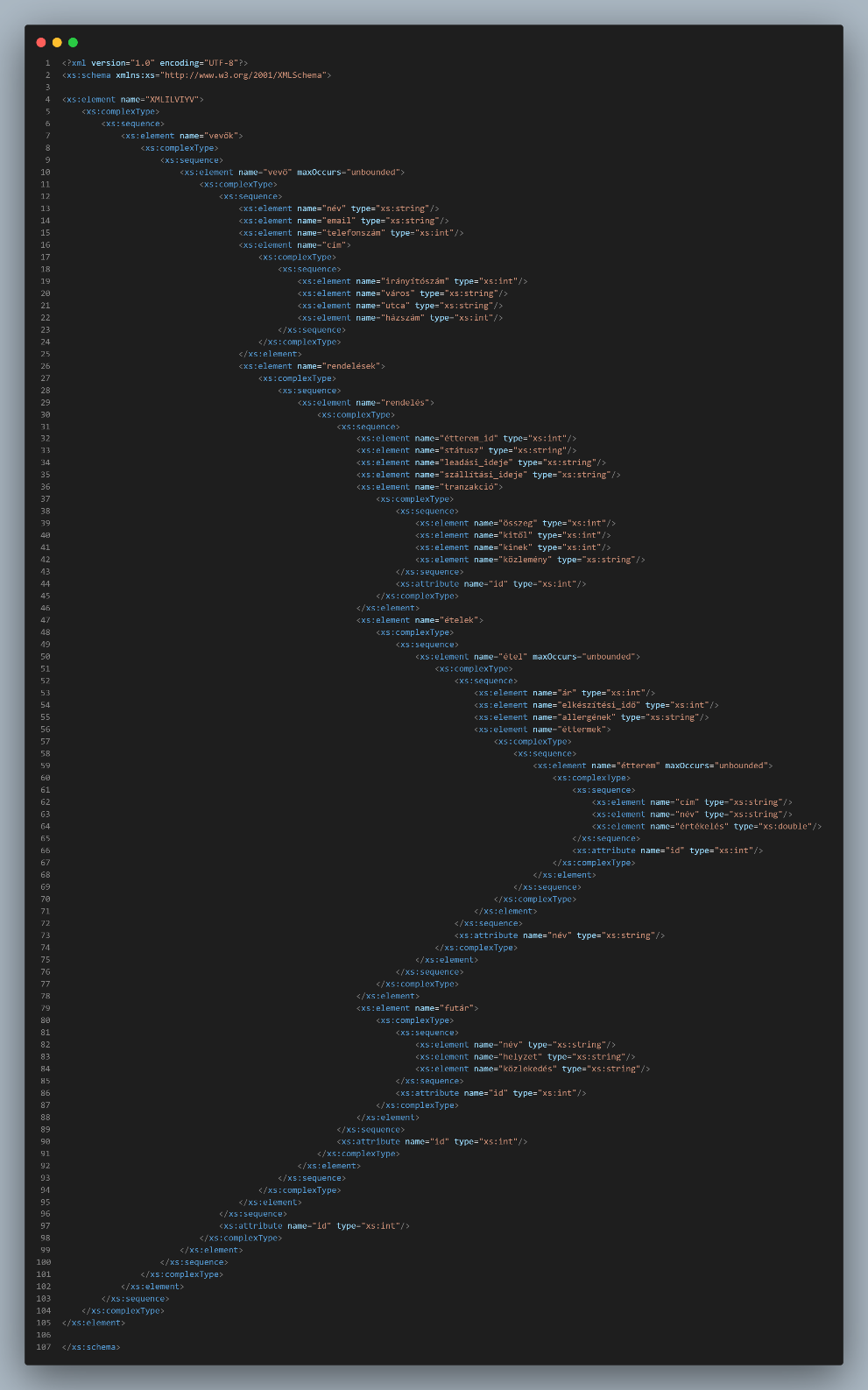
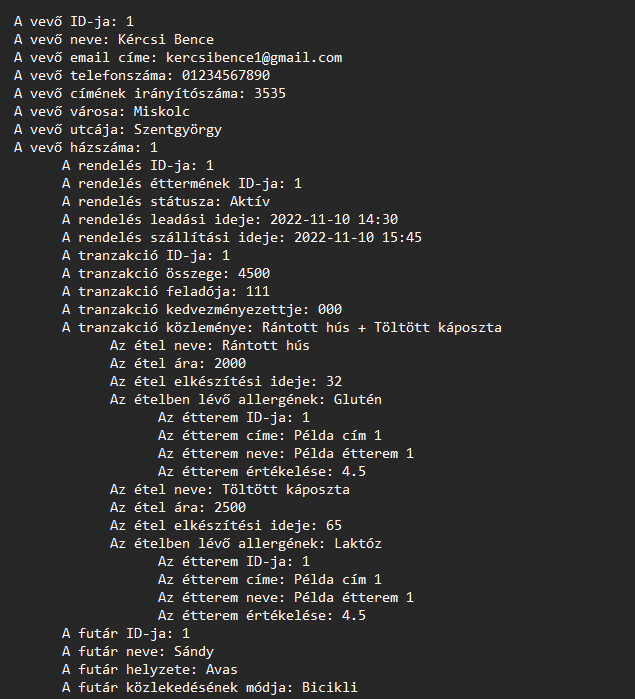
Vevő-Rendelés: Egy vevőhöz több rendelés is lehet beregisztrálva, akár aktív akár kiszállított, vagy bármely más státuszban, akár azonos státuszban is szerepelhet több rendelés, ezzel szemben egy rendelés csak egy fiókhoz tartozik. Ezekből következik, hogy a két egyed között **egy-több** kapcsolat van.

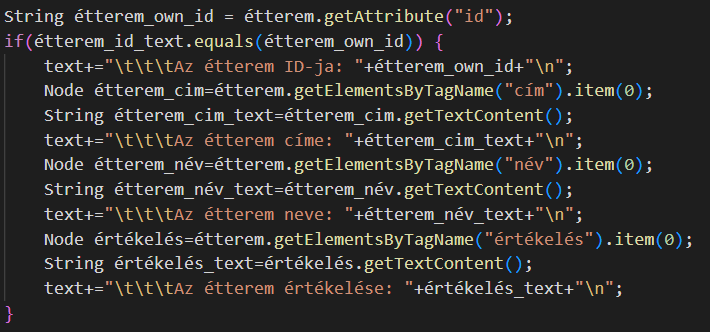
Rendelés-Tranzakció: Egy rendeléshez egy összeg tartozik, amit a vevőnek ki kell fizetni, amit egy tranzakció keretein belül várjuk tőle, ugyan így, egy tranzakció egy rendelés fizetésére alkalmas, ezek miatt itt **egy-egy** kapcsolat található meg. Érdekesség, hogy ez egy kötelező kapcsolat, tehát mindenképp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

1. feladat
   1. Az ER modell megtalálható a githubon, a forráskód az ERILVIYV.txtben, ahol egy GitMind cloudra mutató link van, de PDF és PNG formátumban is megtalálható az ER modell. A Kicsinyített verzióját ide is beillesztem. Az egyedekről, kapcsolataikról és attribútumaikról feljebb beszéltem már, ezért itt inkább a diagrammot emelem ki. Igyekeztem egy sorba, rendezetten elkészíteni a diagrammot.   
      
   2. Az XDM modell szintén megtalálható a githubomon, 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.   
      
   3. 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 githubon XMLILVIYV.xml néven, illetve a róla készült kép is.  
       
   4. Az XMLSchema Szintén megtalálható githubon. A róla készült kép szint úgy. A schemába semmi különlegességet nem alkalmaztam, alapvető leírást végzek az xmlről, azért döntöttem így, mert sokszor a minimalista hozzáállás kevesebb hibához vezethet. Igazából csak az elementek leírását végeztem el, hogy milyen típusnak kell lenniük és pár helyen jeleztem a fordítónak, hogy több is előfordulhat belőle. A validálást külön programmal végeztem, de sikeres volt.   
      
2. A dom programok elkészítése:
   1. Az adatolvasás igazából pár elemnek az ismételt használata volt csak, az xml szerkezetét kellett a kódban is felépíteni. Az xmlt strukturáltan (tabulátorokkal betolva) jeleníti meg a kimeneten, illetve kiírja fileba 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):  
        
      A kimenet (részlet):   
      

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:   


1. 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 xml filet hoz létre, amiben a módosított értékek szerepelnek. A program:  
    
2. Az adatlekérdezés 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:  
     
   A program XPATH lekérdezéseket csinál. A végeredmény (részlet):   
   