**JEGYZŐKÖNYV**

**Adatkezelés XML környezetben**

Féléves feladat

Busz ágazat logisztikai rendszerezése

**Készítette**: Szepesi Glória Virág

**Neptunkód**: W0E0RY

**Dátum**: 2024. 11. 19.

**Tartalomjegyzék**

1. A projekt áttekintése ........................................................................................................ 3

1a) ER modell megtervezése az adatbázishoz ................................................................. 5

1b) Az adatbázis XDM modellre történő átalakítása ....................................................... 7

1c) XML dokumentum létrehozása az XDM modell alapján ........................................... 8

1d) XML Schema készítése az XML dokumentumból .................................................... 20

2. Megoldandó feladatok .................................................................................................... 24

2a) Adatok olvasása ......................................................................................................... 24

2b) Adatok lekérdezése ......................................................................................................... 26

2c) Adatok módosítása ......................................................................................................... 28

2d) Adatok kiirtásához………………………………..................................................................... 30

1. A feladat leírása

Feladatleírás

A következő feladat célja egy autóbusz társaság adatainak adminisztrációja XML dokumentumok segítségével. A projekt keretében különböző lépések során szükséges megtervezni az adatbázis ER modelljét, átalakítani XDM modellé, majd az XDM modell alapján XML dokumentumot létrehozni. Végül az XML dokumentum alapján XMLSchema-t készítünk, és egy DOM program segítségével adminisztráljuk az adatokat.

## Ezen lépések:

1. **ER modell tervezése**: Az autóbusz társaság adatainak modellezése entitások és kapcsolataik segítségével, például buszok, sofőrök, szervizesek stb. meghatározása és az entitások közötti kapcsolatok definiálása.
2. **Az adatbázis XDM modellre történő átalakítása**: Az ER modell átalakítása XML Data Model (XDM) formátumra. Példa az XDM modell átalakítására:

<BusServiceSystem>

<Szerviz>

<SzID></SzID>

<JavitóSzám></JavitóSzám>

<Név></Név>

</Szerviz>

<Busz>

<BID></BID>

<Rendszam></Rendszam>

<Tipus></Tipus>

<max\_hely></max\_hely>

<Szerviz>

<SzID></SzID>

</Szerviz>

</Busz>

…

</BusServiceSystem>

1. **XML dokumentum létrehozása az XDM modell alapján**: Az XDM modell alapján egy XML dokumentum létrehozása. Például az XMLNeptunkod.xml fájl tartalma a következőképpen nézhet ki:

<Palyaudvar xsi:noNameSpaceSchemaLocation="./XMLSchemaHR6121.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<Buszok>

<Busz Id="001">

<Rendszam>KWV-965</Rendszam>

<Tipus>Mercedes</Tipus>

<max\_hely>52</max\_hely>

</Busz>

<!-- További buszok -->

</Buszok>

<Soforok>

<Sofor SoforId="101">

<Jegykiadoszam>H37625</Jegykiadoszam>

<Nev>Ragó Lili</Nev>

<Busz Id\_FK="007"></Busz>

</Sofor>

<!-- További sofőrök -->

</Soforok>

…

</Palyaudvar>

1. **XMLSchema készítése az XML dokumentumból**: Az XML dokumentum alapján létrehozott séma az adatok szerkezetének és érvényességének ellenőrzésére szolgál.
2. **DOM program készítése az XML dokumentum adatai adminisztrálása alapján**: Egy Java program, amely betölti az XMLNeptunkod.xml fájlt, kiírja annak tartalmát fa struktúra formában a konzolra, és egy új fájlba XMLW0E0RY.xml néven menti.

Az Az adatbázis egyedei, amelyeket a korábbi XML dokumentumban meghatároztunk, a következőképpen néznek ki:

1. **Buszok**

* Id
* Rendszam
* Tipus
* max\_hely

1. **Soforok**

* SoforId
* Jegykiadoszam
* Nev
* Buszok Id\_FK

1. **Szervizesek**

* SzervizId
* Javitoszam
* Nev
* Buszok Id\_FK

1. **Eszkozok**

* EszkozId
* Termekszam
* Gyarto
* Ev
* SzervizId\_FK

1. **Utvonalak**

* UtvonalId
* KezdAll
* VegAll
* Km

1. **Halad:**

* UtvonalId\_FK
* Buszok ID\_FK
* Menetido

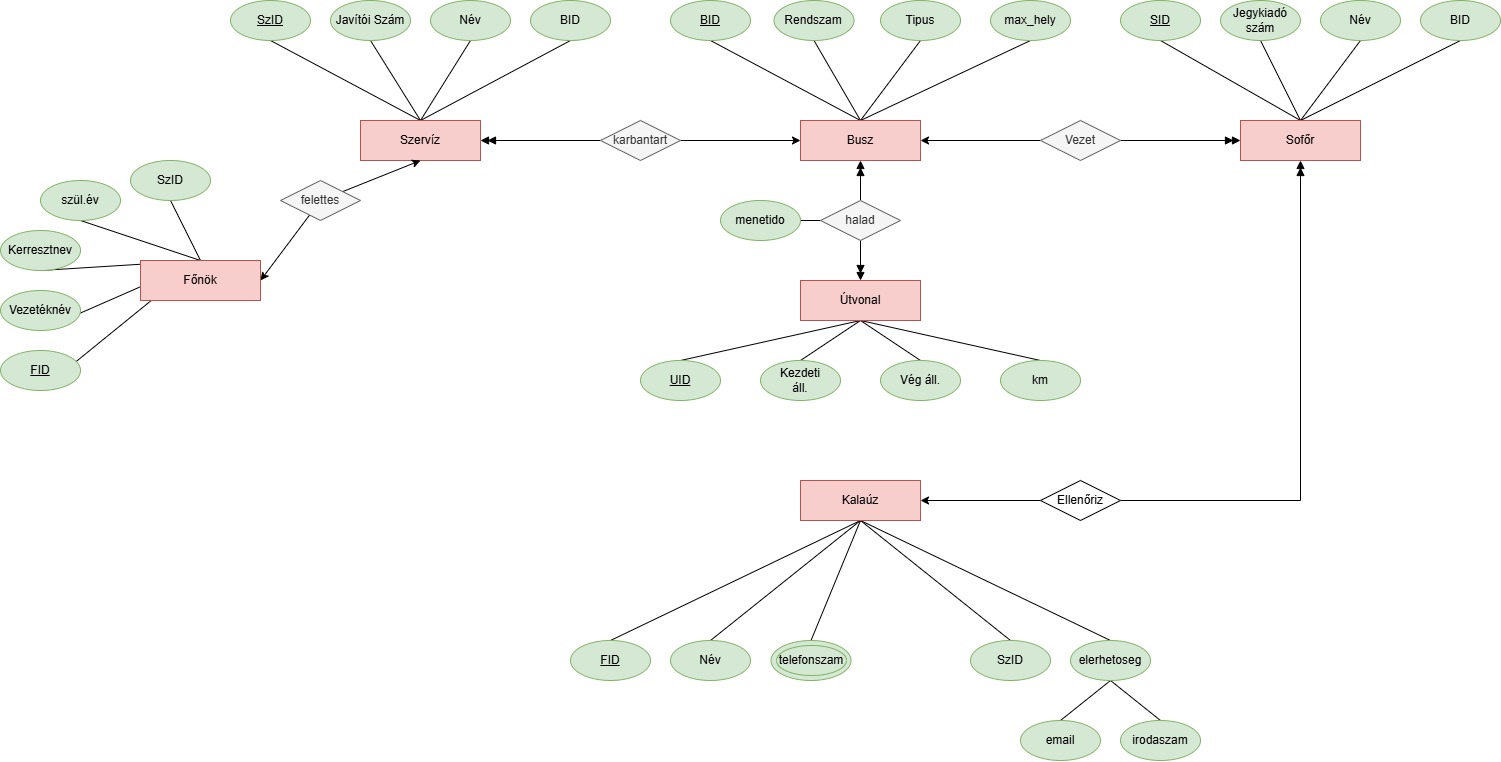
1. **Kalauzok**

* KalauzId
* Nev
* SoforId\_FK
* SzervizId\_FK
* Elerhetoseg (Irodaszam, Email)
* Telefonszam

Az adatbázis ER modelljének terve:

Az ER modellben nagyon fontos az átláthatóság. Ezt igyekeztem szem előtt tartani. Tanár úr is mindig kiemeli, hogy az összekötő elemek ne lógjanak egymásba, valamint, hogy legyen szintenként elhelyezve minden.

A modell tartalmaz minimum 5 egyedet, többféle kapcsolatot, fellelhető benne például 1:1 valamint M:N kapcsolat is. Az egyedek többféle tulajdonsággal rendelkeznek, van normál, id-azaz a kulcs- valamint többértékű is.



## Az Egyedekről részletesebben:

**A Busz egyed:**

Ez az egyed egy központibb egyed. Itt találhatóak meg az buszok alap adatai. az ID az egyed elsődleges kulcsa. Emellett megadható még a busz rendszáma, a gyárója ez string típusú tulajdonság mindkettő. Jellemezhetjük még a buszt a férőhelyével, hogy hány utast szállíthat egyszerre, ami integer típusú.

**A Szerviz egyed:**

A szervíz egyed a buszokhoz tartozó szerviz adatait tárolja. Rendelkezik egy „k\_id” azonosítóval, ami itt az elsődleges kulcs. Van még neve, mint string adat, egy azonosító ami integer tipusú. Van még itt egy Foreign key ami a buszhoz kapcsolja az egyedet.

**A Főnök egyed:**

A Főnök egyed egy 1:1 kapcsolattal kapcsolódik a szervízhez, minden szervíznek/ ott dolgozónak van egy felettese. Ezekhez tartozik egy elsődleges kulcs (id) egy vezetéknév és keresztnév ami szöveges adatként szerepel. Mindenkihez tartozik egy születési dátum, valamint összeköttetés miatt egy idegenkulcsként itt is fellelhető egy szerviz\_id.

**A Sofőr egyed:**

Ebben az egyedben találjuk meg a buszokat vezető emberek adatait. Találunk egy SID – a sofőrök azonosítója- itt ez z elsődleges kulcs. A Vezetők egyértelműen rendelkeznek névvel ami szöveges bevitelt vár, egy- egy jegykiadó számmal ami integer vár. A megjelenő idegen kulcs pedig a buszok egyedi azonosítói.

**Az útvonal egyed:**

Ez az egyed tartalmazza a sofőrök által levezetett útvonalat. Az id jelölésű rész itt is az elsődleges kulcsot jelenti, Van kezdet- és végállomás, ahova szöveges adatként tudjuk beírni a településeket/ megállókat. Tárol még az egyed egy kilométer számot, ami a két állomás közötti távolságot jelenti.

**A Kalauz egyed:**

Végül de nem utolsó sorban a Kalauz egyed. Ez az egyed, aki ellenőrzi a buszokat, utasokat, valamint a sofőröket. Van egy ID, ami az elsődleges kulcs. Rendelkezik egy névvel, ami szöveges bevitelt igényel. Találunk még egy összetett tulajdonságot, ami az elérhetőség. Ezen belül van a szöveges típusú email cím, valamint egy irodaszám, amit számokkal tudunk megadni.   
Eltárol még egy telefonszám tulajdonságot, ami akár több adatot is tud rögzíteni tehát ez többértékű tulajdonság. Csatlakozik még ehhez két idegen kulcs.

## A Kapcsolatok:

A Szerviz és a Főnök kapcsolata:

E kettő egyed közötti kapcsolat egy-egy kapcsolat, mivel egy szervízhez esetleg szerelőhöz egy felettest rendelünk.

A Szervíz és Busz kapcsolata:

Itt a kapcsolat egy a többhöz kapcsolat, ugyanis egy szervíz több műszakilag károsult járművet is tud szervízelni, ugyanez szerelőkre értve nyilván.

A Busz és a Sofőr kapcsolat:

Itt is egy a több kapcsolat áll fenn a két egyed között ugyanis egy járművet nyilván egy személy vezet, de például egy hosszú út során a busz rendelkezhet több sofőrrel is.

A Busz és Útvonal kapcsolat:

Ez a kapcsolat több a többhöz azaz M:N kapcsolat. Egy jármű több útvonalon is közlekedhet, valamint egy útvonalat több busz is bejárhat. Ide köthetünk még egy „menetidő” tulajdonságot.

A Kalauz és a Sofőr kapcsolata:

Itt is egy a többhöz kapcsolat áll fenn, ugyanis egy kalauz több sofőrt is ellenőrizhet, ám egy a volán mögött ülő személyt egyszerre csak egy ember tud ellenőrizni.

## Az adatbázis XDM modellre történő átalakítása

Az Experience Data Model (XDM) egy olyan adatmodell, amelyet az Adobe fejlesztett ki annak érdekében, hogy a vállalkozások jobban integrálhassák, kezelhessék és elemezhesék az ügyféladatokat. Az XDM célja, hogy egységesítsen különböző adatforrásokat és struktúrákat, ezáltal könnyebbé téve az adatok kezelését és felhasználását az ügyfélélmény javítása érdekében.

Az első feladat b, része az XDM lemodellezése volt. Ennek alapjául szolgált az előbb bemutatott ER modell. Figyelembe kell venni a példányosítás folyamatát, hogy átlátható és könnyen lekövethető legyen az ábra. Mindent egy szülő egyedbe vezettem vissza a többszörösitek miatt.

A folyamat során szükséges volt létrehozni egy új osztályt, ami a Halad névre hallgat. Ezt azért kellett létrehozni, mert található az ER modellben egy több a többhöz kapcsolat, ennek az átképzése.

A képen vázlat, rajz, fekete-fehér látható

Automatikusan generált leírás

A legfelül található elem, ami mindent összefog, szintén egy segéd a „pályaudvar”.

# A képen szöveg, képernyőkép, menü látható Automatikusan generált leírásXML dokumentum létrehozása az XDM modell alapján

A következő lépés a megoldás során az XDM modell mintájára lekódolni az XML részt.

A képen szöveg, képernyőkép, menü, tervezés látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, szoftver, Betűtípus látható

Automatikusan generált leírás

# XML Schema készítése az XML dokumentumból

Ebben a résznem minden egyes egyedhez egy egy saját változót csináltam, majd feltöltöttem példákkal.

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, tervezés látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Betűtípus, menü látható

Automatikusan generált leírás

# 2. Feladatrész

## DOM project

Az adatolvasás:

A fájlbeolvasás arról szól, hogy be tudjuk olvasni a koprábban megírt dolgokat. Itt először is egy XML kiterjesztésű fájlt kell beolvasni.

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

## Az adatlekérdezés része:

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

Ezen programrész különböző lekérdezáseket futtat le.

Különböző adatokat dolgoz fel és értékel k a megadott kritériumok alapján.

## Az Adatmódosítás:

Ebben a részben módosítunk elemeket és azok attribútumait tulajdonságaikat valamint a különböző tartalmakat.

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, menü látható

Automatikusan generált leírás

## Az Adatírás:

Itt láthatjuk azokat a program részleteket, amikkel kiíratunk konzolra különböző adatokat.A képen szöveg, képernyőkép látható

Automatikusan generált leírás