Adatkezelés XML-ben  
  
BSc  
Féléves feladat  
2023

Készítette:  
Szendrei Gábor  
Programtervezõ Informatikus  
V9ZK10

Tartalomjegyzék

[1a) A feladat témája 3](#_Toc152504789)

[A feladat ER modellje: 4](#_Toc152504790)

[Az Egyedek közötti kapcsolatok 5](#_Toc152504791)

[1b) Az ER-modell konvertálása XDM modellre 5](#_Toc152504792)

[1c) XML Dokumentum készítése 6](#_Toc152504793)

[XML dokumentum alapján XMLSchema készítése 10](#_Toc152504794)

[XML, XSD validálás 15](#_Toc152504795)

[Adatolvasás 16](#_Toc152504796)

[Adatírás 19](#_Toc152504797)

[Adatmódosítás 21](#_Toc152504798)

[Adatlekérdezés 22](#_Toc152504799)

# 1a) A feladat témája

A beadanóm témája egy olyan adatbázis, amely autos cégeket és hozzújuk kapcsolódó egyedeket tartja nyilván.

* **Autóscég**

ceg\_kod – Elsődleges kulcs

cegnev – Cég nevét tartalmazza

helyrajziSzam – Cég elhelyezkedését tartalmazza

dolgozokSzama – Tárolja a dolgozók számát.

* **Vásárlók**

jog\_kod – Elsődleges kulcs

email\_cim – Vásárló email címe

cim – Összetett tulajdonság (Orszag, isz, varos, uHsz)

Telefonszam – Többértékű tulajdonság

* **Autós adatok**

forgalmi\_kod – Elsődleges kulcs

ar – Mennyibe kerül az eladásra váró auto

kmOra – Mennyi kilómérert tettek már meg az autóval

statusz – Szöveges típús amibe leírást lehet adni, pl. felújított

* **Autós típús**

tipus\_kod – Elsődleges kulcs

gyartasiEv – Mikor gyártották az adott autót

marka – Az autó márkáját tartalmazza

nev – Az auto teljes nevét tartalmazza

* **Számlázás**

szamlakod– Elsődleges kulcs

szamlaDatum – Mikor állították ki a számlát

végösszeg – adókkal mindennel együtt mennyit fizettek

jogsiszam – A vásárló azonosítására szolgál

# A feladat ER modellje:

A diagram of a network

Description automatically generated with medium confidence

# Az Egyedek közötti kapcsolatok

Autós cég – Autó adatok: 1:n Egy autó több cégnél is megtud fordulni.

Autó adatok – Autó típús 1:1 Egy auto adataihoz egy típús tartozhat csak.

Autós cég – Vásárló n:m Több vásárló több céghez tartozhat.

Vásárló – Számlázás Egy vásárlóhoz több számla is tartozhat.

# 1b) Az ER-modell konvertálása XDM modellre

XDM modellnél háromféle jelölést alkalmazhatunk. Ezek az ellipszis, a rombusz, illetve a téglalap. Az ellipszis jelöli az elemeket minden egyedből elem lesz, ezen felül a tulajdonságokból is. A rombusz jelöli az attribútumokat, amelyek a kulcs tulajdonságokból keletkeznek. A téglalap jelöli a szöveget, amely majd az XML dokumentumban fog megjelenni. Azoknak az elemeknek, amelyek többször is előfordulhatnak, a jelölése dupla ellipszissel történik. Az idegenkulcsok és a kulcsok közötti kapcsolatot szaggatott vonalas nyíllal jelöljük.

A feladat XDM modellje:

A black and white drawing of a structure

Description automatically generated

# 1c) XML Dokumentum készítése

Az XDM modell alapján az XML dokumentumot úgy készítettem el, hogy először is a gyökér elementtel kezdtem, ami az AutosCegERV9ZK10 volt. A gyermek elemeiből 3-3 példányt hoztam létre, ezeknek az elemeknek az attribútumai közé tartoznak a kulcsok, illetve idegenkulcsok is, mindezek után ezeknek az elemeknek létrehoztam a többi gyermek elementet is. XML dokumentum forráskódja

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer code

Description automatically generated

# XML dokumentum alapján XMLSchema készítése

 Az ERV9ZK10.xsd séma file leírja mindazon megkötéseket, amelyeknek az XML dokumentumnak meg kell felelnie. Itt definiálunk minden típust, amit az XML file-ban használni szeretnénk, valamint az adatbázis kapcsolatait xs:unique és xs:keyref bejegyzésekkel hozom létre. . Az XML Schémám meghatározza az adatokat, mint például a cég nevét, vásárlók címét, telefonszámokat, amelyeket egy telefonszamTipus típus korlátozza, hogy csak bizonyos formátumú (4 szám – 3 szám – 4 szám) fogadja el. Továbbá létezik egy emailCimTipus is, amely szabályozza az emailcím formátumát, amit elfogad. Komplex típusokat is definiáltam, például a vasarloTipus, amiben egy komplex típus definiálja a cím formátumát, autosCegTipus, autosAdatokTipus, autosTipusTipus, vasarlasTipus, szamlazasTipus. Az adatbázis integritásának megőrzése érdekében elsődleges (PK) illetve idegen kulcsok (FK) meghatározására került sor. Az XML séma így biztosítja, hogy az adatok szerkezete és kapcsolatai érvényesek és következetesek legyenek.

A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

# XML, XSD validálás

Az XML (eXtensible Markup Language) és az XML séma (XSD) validálása kiemelkedő fontosságú adataink integritásának és konzisztenciájának biztosítása szempontjából. Az XML egy strukturált adatleíró nyelv, amely lehetővé teszi adatok hierarchikus és szemantikus reprezentációját. Az XSD pedig meghatározza az XML dokumentum struktúráját és tartalmát.

A screenshot of a computer

Description automatically generated

# Adatolvasás

A kód egy Java alapú XML feldolgozó program, amely a DOM (Document Object Model) parserét használja. A DOM parser a teljes XML dokumentumot memóriába tölti, ami gyors hozzáférést biztosít az elemekhez, de nagyobb dokumentumok esetén jelentős memóriaigényt jelenthet. A program beolvassa az XML fájlt, normalizálja azt, és különböző függvények segítségével feldolgozza az XML elemeket, melyek az 'AutosCeg', 'AutosAdatok', AutosTipus,'Vasarlo', Szamlazas. Minden elemcsoport feldolgozása külön függvényben történik, ami javítja a kód olvashatóságát és karbantarthatóságát. Hibakezelés is implementálva van a fájlbeolvasás és parse-lás során.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated A screen shot of a computer program

Description automatically generated A computer code with colorful text

Description automatically generated

# Adatírás

A kód fő funkciója, hogy beolvassa és kiírja az XML tartalmakat a konzolra és fájlba is. A {main metódusban az XMLV9ZK10 fájlt olvassa be a megadott útvonalról, használva a DocumentBuilderFactory és DocumentBuilder osztályokat az XML struktúra elemzéséhez. Ezután normalizálja a dokumentumot a normalize metódussal, ami segít a DOM fa struktúrájának rendezésében.

A screen shot of a computer program

Description automatically generated A screen shot of a computer program

Description automatically generated A computer screen with colorful text

Description automatically generated

# Adatmódosítás

Ez a Java program, a DOMModifyV9ZK10, egy XML fájlt olvas be és módosítja azt a DOM(Document Object Model) API segítségével. Az XML fájl, amely egy állatkert adatait tartalmazza, egy előre meghatározott útvonalon található, és a DocumentBuilder osztály segítségévelparse-oljuk. A program két módosítást hajt végre, az első ilyen, hogy a vásárló első elemének város attribútumát Budapest értékre állítja. A második módosítás az autós adatok első elemének árát módosítja 600000-re. A módosítások után a program egy Transformer segítségével visszaalakítja és kiírja a módosított DOM-ot XML formátumban. Az XML kiírás során a Transformer beállításai biztosítják a formázott, olvasható kimenetet, az OutputKeys.INDENT beállítás segítségével.A screen shot of a computer program

Description automatically generated A screen shot of a computer program

Description automatically generated

# Adatlekérdezés

A program a Java Parser-t használja XML fájlunk feldolgozására, ami lehetővé teszi a XML elemek olvasását és manipulálását egy objektumorientált módon. A kód, az XML fájlt, a File objektumon keresztül tölti be, biztosítva ezzel a fájl elérését és kezelését. A DocumentBuilderFactory és DocumentBuilder osztályok használata a fájl DOM reprezentációjának létrehozásához szükséges, ami egy strukturált, fa-szerű modellt biztosít az XML adatok számára. A program minden egyes lekérdezést egy for ciklus segítségével hajt végre, ahol a getElementsByTagName metódus segítségével specifikus XML elemeket keres. Az elemek feldolgozása során a Node és Element interfészeket használja, amelyek lehetővé teszik az egyes elemek attribútumainak és tartalmának elérését. A lekérdezések eredményét egy StringBuilder objektumba gyűjti, amely hatékonyan kezeli a nagy mennyiségű stringek összefűzését. A kód, az összegyűjtött adatokat XML-szerű formátumban állítja elő.A screen shot of a computer program

Description automatically generated A screen shot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated A computer screen shot of many colorful text

Description automatically generated A screen shot of a computer code

Description automatically generated A screen shot of a computer program

Description automatically generated A computer screen shot of a program code

Description automatically generated A screen shot of a computer program

Description automatically generated

Miskolc,  
2023