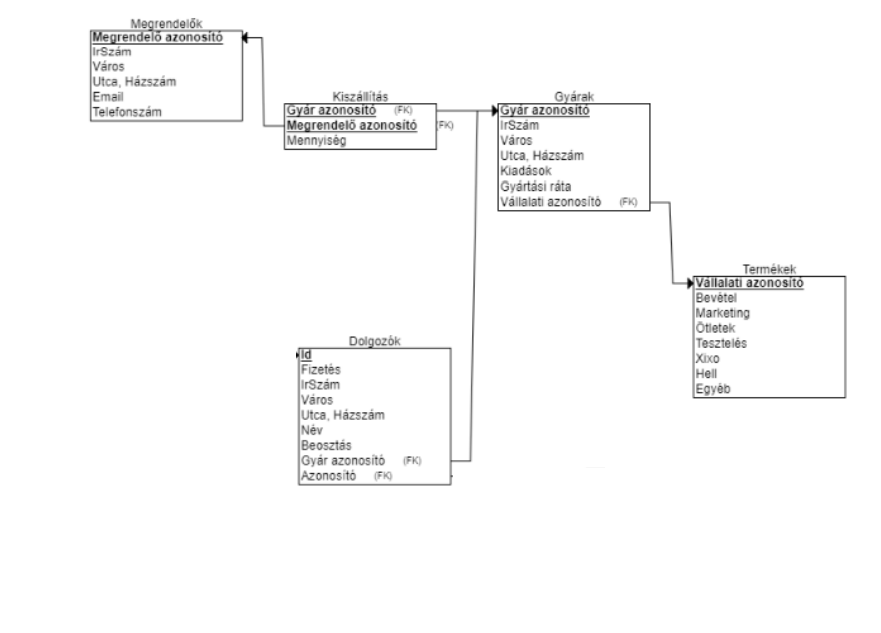
Csicsely Gábor

NE-C7H5VB

Adatbázis Rendszerek 2

1. Beadandó

Adatbázis relációs modelje:



A táblák létrehozása cmd-ben:

Create TABLE Megrendelok(Megrendeloazonosito PRIMARY KEY, Iranyitoszam NUMBER(4), Varos VARCHAR(30), Utca VARCHAR(30), Email VARCHAR(40), Telefonszam NUMBER(11));

CREATE TABLE Kiszallitas(

Megrendeloazonosito INT,

Mennyiseg NUMBER(7),

Gyarazonosito INT,

FOREIGN KEY (Megrendeloazonosito) REFERENCES Megrendelok(Megrendeloazonosito),

FOREIGN KEY (Gyarazonosito) REFERENCES Gyarak(Gyarazonosito);

CREATE TABLE Dolgozok (

ID INTL,

Fizetes NUMBER(10),

Irszam NUMBER(4),

Varos VARCHAR(30),

Utca VARCHAR(30),

Nev VARCHAR(30),

Beosztas VARCHAR(30),

Gyarazonosito INT,

FOREIGN KEY (Gyarazonosito) REFERENCES Gyarak(Gyarazonosito),

PRIMARY KEY (ID);

Táblák létrehozása Eclipse-ben

Eclipsen belül is egyszerüen lehet adatokat felvinni, táblákat létrehozni.

Ha felállítottuk a kapcsolatot az adatbázisunkkal egy Statement-et létrehozva egy Stringet létrehozva, beírjuk az SQL kódot, amit később execute-olhatunk.

Az adatbevitelre, az executeUpdatet tudjuk használni

függvények: uj\_termekek\_tabla() uj\_gyarak\_tabla()

Connection conn = this.Connect("bead.db");

Statement stmt = conn.createStatement();

String par = "CREATE TABLE Gyarak (Gyarazonosito INT, Iranyitoszam NUMBER(4), Varos VARCHAR(30), Utca VARCHAR(30), Kiadasok NUMBER(10), Gyartasirata NUMBER(11), Vallalatiazonosito INT, PRIMARY KEY (Gyarazonosito), FOREIGN KEY (Vallalatiazonosito) REFERENCES Termekek(Vallalatiazonosito))";

stmt.execute(par);

stmt.executeUpdate("INSERT INTO Gyarak VALUES (11, 3535, 'Miskolc', 'Gőz út', 100, 10, 1)");

stmt.executeUpdate("INSERT INTO Gyarak VALUES (22, 3522, 'Miskolc', 'Kandó Kálmán útca', 200, 30, 2)");

stmt.executeUpdate("INSERT INTO Gyarak VALUES (33, 2234, 'Budapest', 'Nagy útca', 300, 20, 3)");

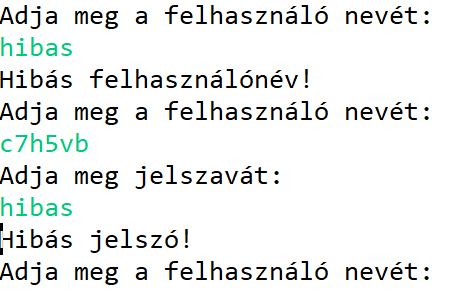
I

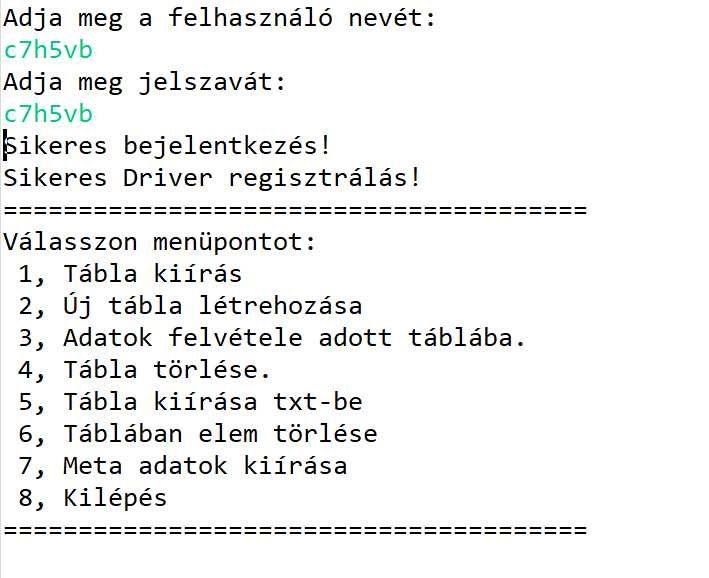
Bejelentkezési modul:

A bejelentkezési modul egy egyszerű while ciklus, amiben 2 egymásba ágazó if-van. Addig kéri újra a felhasználó nevet/jelszót, amíg a felhasználó sikeresen meg nem adja.

Ameddig ez nem történik meg nem jön létre kapcsolat az adatbázissal.

A függvény: bejelentkez();

***Ha hibás a jelszó/felhasználó****:*

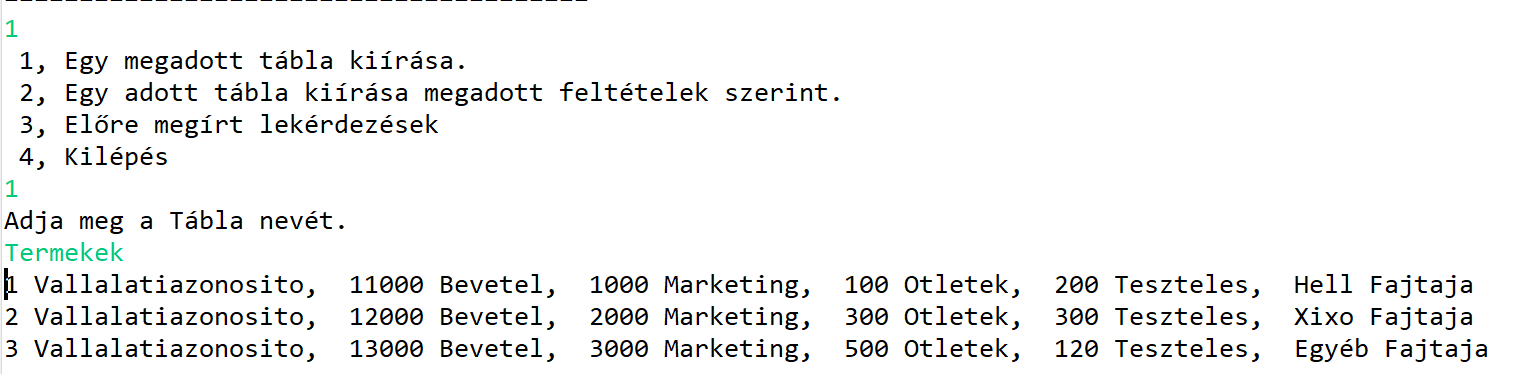
***Ha sikeres a bejelentkezés:***

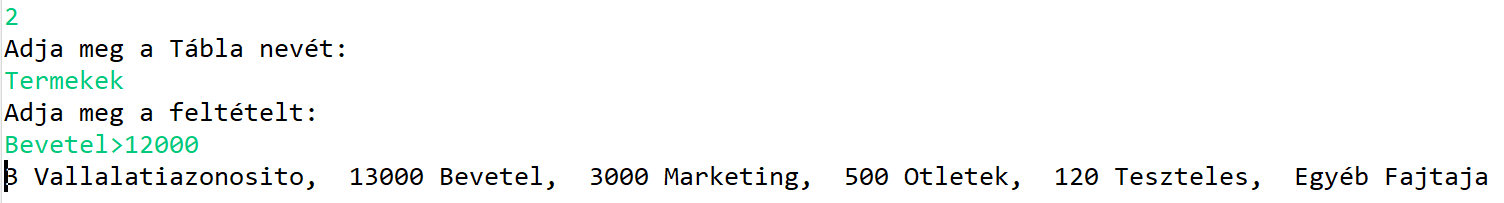
A menü:

A menü egy konzolos switch rendszer, ami addig fog futni, ameddig ki nem lép a felhasználó (nem választja ki a 8, as Kilépés menüpontot).

Ha a felhasználó rossz inputot ad meg, hibaüzenet kap és újra választhat.

Ez a main függvénybe van megoldva.

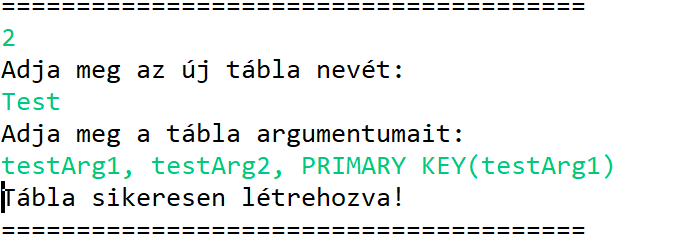
Tábla kiírás:

A felhasználónak itt is lehetősége van választani, hogy egy egész táblát szeretne kiíratni, vagy csak azon elemeit, amelyek megfelelnek az user által megadott feltételnek, vagy 2 előre megírt lekérdezést szeretne kiíratni.

Az adatok kiírásánál az adatok után le van írva az adattag neve is.

A függvények: kiirTabla(); kiirTablaFelt(); eloreMegirtLekerdezesek();

Új tábla létrehozása:



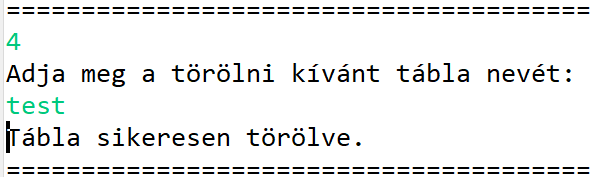
A felhasználó megadja a hozzáadni kívánt tábla nevét, argumentumait.

Ez egy sima scanner bekérdezés, ahol bekérjük a tábla nevét és az argumentumokat. Majd létrehozunk egy Stringet amiben megírjuk az SQL kódot, majd egy ez előtt létrehozott Statement-el executeoljuk.

String par = "CREATE TABLE " + tNev + "(" + args + ")";

stmt.execute(par);

A függvény: ujTabla()

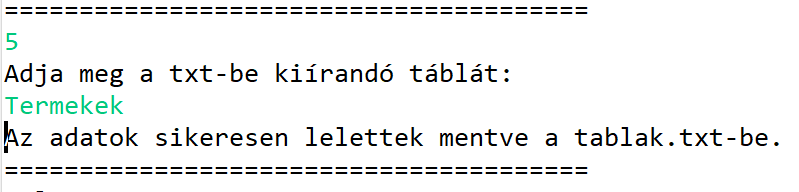
Tábla törlés:

Itt is bekérjük a tábla nevet, majd executeUpdate-et hívunk rá.

stmt.executeUpdate("DROP TABLE " + tNev);

Függvény: tablaTorol()

TXT-be kiírás



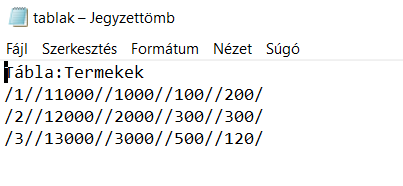
Függvény: TablaFileba()

Bekérjük a felhasználótól a tábla nevét, majd létrehozunk egy resultSet-et ami megszerezzük a tábla adatait.

ResultSet resultSet = stmt.executeQuery("SELECT \* FROM " + tabla);

Létrehozunk egy FileWriter-t és megadjuk a file nevét ahova menteni szeretnénk:

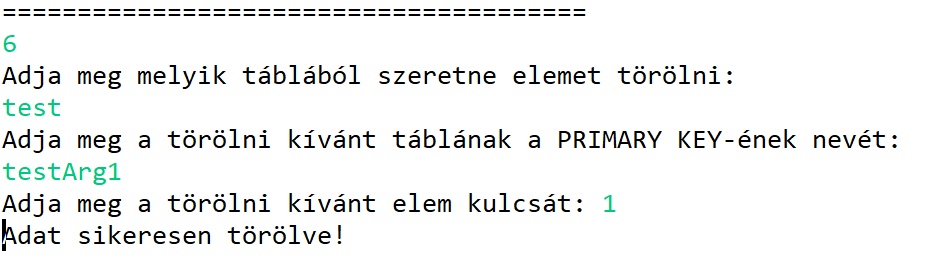
FileWriter fileWriter = **new** FileWriter("tablak.txt");

Majd metadata segítségével megtudjuk hány oszlopos a táblánk, ennek segítségével megtudjuk írni a for ciklusunkat a fájl-ba íráshoz.

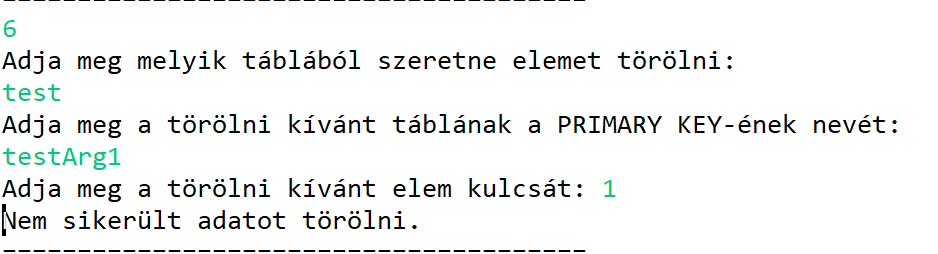
Függvény: TablaFileba()

Elem törlése

Ha sikeres:



Ha sikertelen:



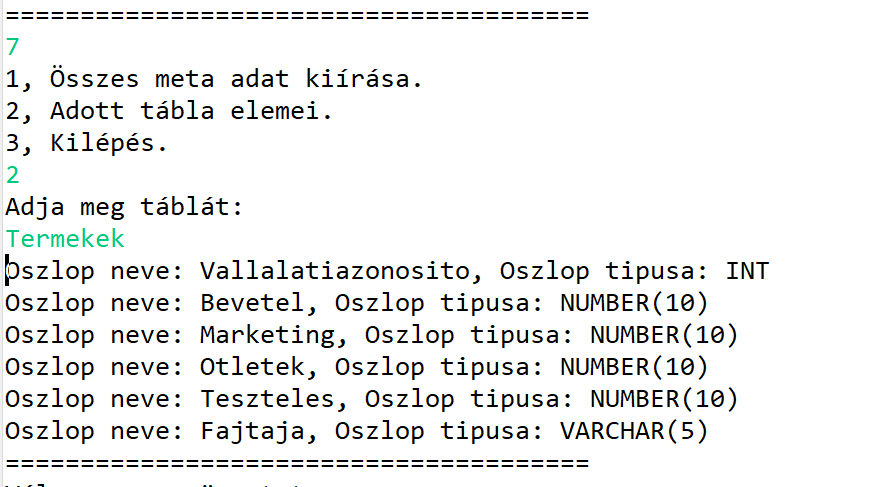
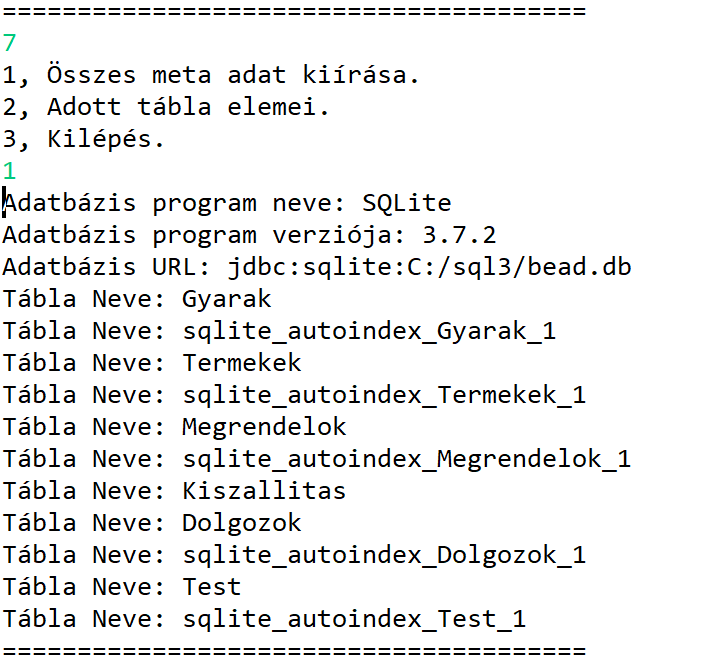
Itt bekérjük a Tábla nevét, a PRIMARY KEY nevét, majd értékét a törölni kívánt elemnek.

A függvény preparedStatement-el lett megoldva.

String sql = "DELETE FROM " + tNev +" WHERE " + kulcs + "=?";

Függvény: elemTorol()

Meta adatok:



Az első menüpont az összes meta adatot megjeleníti, míg a második, csak a kijelölt tábláját.

DatabaseMetaData segítségével tudtuk lekérni ezeket az adatokat

metaData = connection.getMetaData();

Függvény: metaDataKiir() metaDataTabla()

Rekordok módisítása

Függvény: rekordModosit()

Ez a függvény is egy rövid executeUpdate használatával készűlt.

Elsőnek bekéri a táblát, a változtatni kívánt rekordot, majd a változtatás feltételét, ezek alapján futtatja és hajtja végre.