

Adatbázis rendszerek II.

Első beadandó feladat

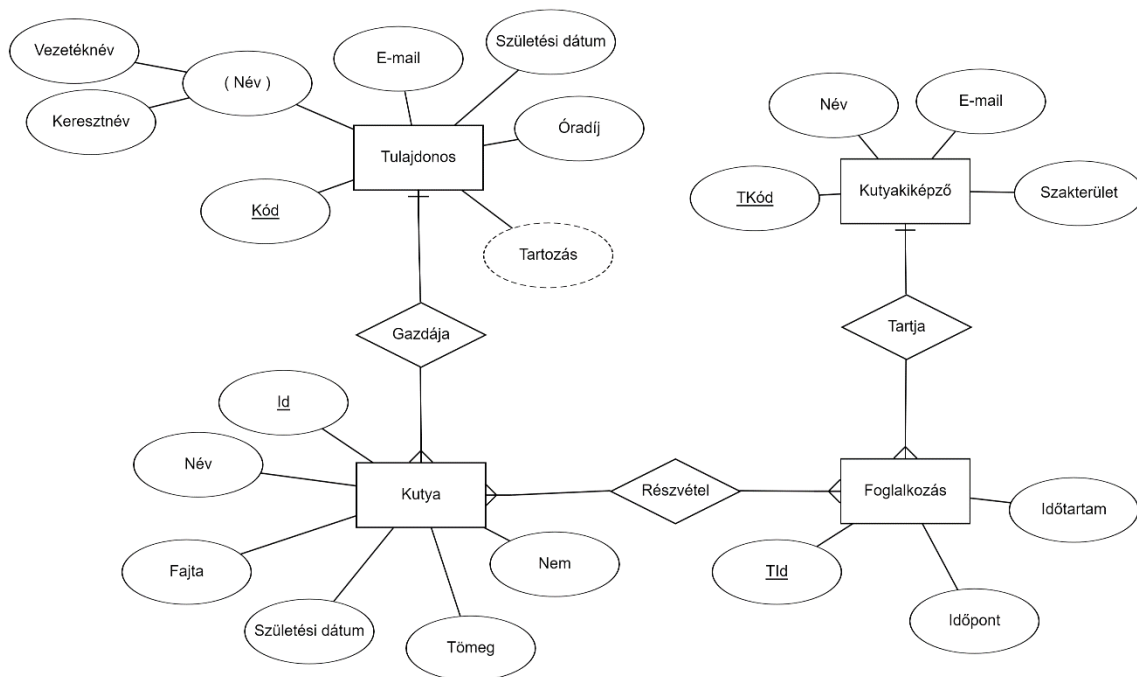
Palencsár Enikő
YD11NL

Gyakorlatvezető: dr. Kovács László
Gyakorlat időpontja: Hétfő 10:00
Képzés: BSc nappali

A feladat leírása:

Programom egy kutyaiskolával kapcsolatos SQLite 3 adatbázis kezelését végzi, az adatbázis szemantikai modelljét az érthetőség kedvéért magyar nyelven állítottam össze, a Java alkalmazásom felhasználói felületének nyelve is magyar, a mögöttes tábla- és mezőnevek, továbbá a programváltozók és metódusok nevei ugyanakkor angol nyelvűek (már ránézni is rossz lenne egy `getKutya()` vagy `setKutya()` metódusra).

A programom Java nyelven íródott, az adatbáziskezelést JDBC-vel végzi, felhasználói felülete grafikus.



A kezelt sémáról:

Az adatbázisban 6 tábla található, közülük 5 vesz részt az adatkezelésben, míg a **USERS** táblának csupán a felhasználók azonosításában van szerepe.

A táblák és mezőik:

OWNERS – Kutyatulajdonosok

- **Code**: integer azonosító, automatikusan növekvő értékekkel
- **Name**: a tulajdonos teljes neve (nem volt ok a keresztnév- és vezetéknév szétválasztására)
- **Email**: a tulajdonos e-mail címe
- **Date_of_birth**: a tulajdonos születési dátuma, maradhat üresen
- **Hourly_fee**: a tulajdonossal kötött szerződésben szereplő óradíj, mely a tulajdonos valamennyi kutyájára vonatkozik

Az óradíj a gazdi anyagi helyzetétől, lakhelyétől, beiratkozott kutyái számától függ, de például a közeli Mancs Állatmenhely kutyáit térítésmentesen fogadják.

DOGS – Kutyák

- **Id**: integer azonosító, automatikusan növekvő értékekkel

- Name: a kutya neve
- Breed: a kutya fajtája, maradhat üresen
- Date_of_birth: a kutya születési dátuma, maradhat üresen
- Weight: a kutya tömege kilogrammban, lebegőpontos szám
- Gender: a kutya neme
 - K: kan
 - Sz: szuka
 - IK, ISz: ivartatalított kan, szuka
- Owner_id: a tulajdonos azonosítója

TRAINERS – Kutyakiképzők

- Tcode: integer azonosító, automatikusan növekvő értékekkel
- Name: a kiképző neve
- Email: a kiképző e-mail címe
- Expertise: a kiképző szakterülete

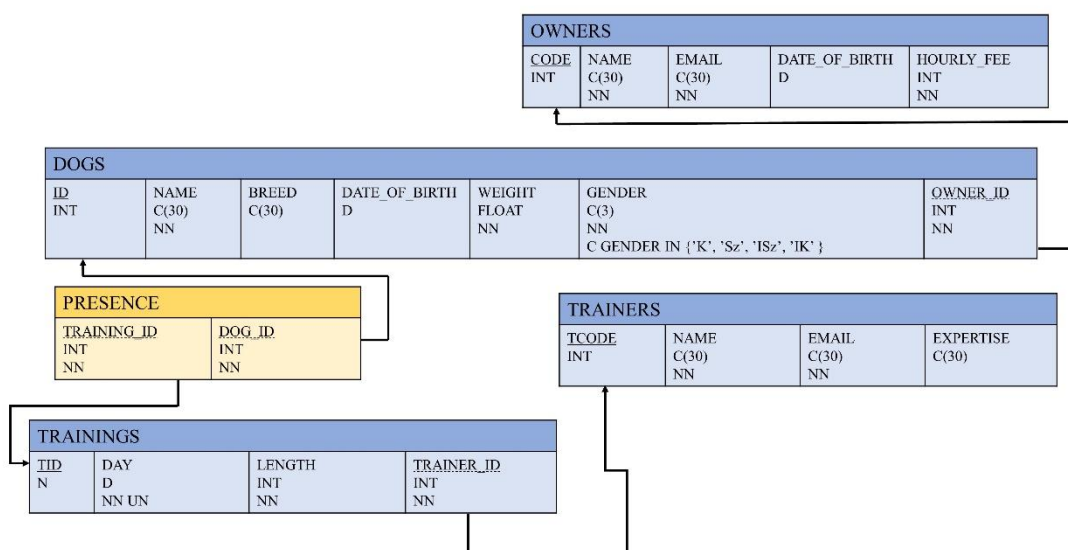
TRAININGS – Foglalkozások

- Tid: integer azonosító, automatikusan növekvő értékekkel
- Day: a foglalkozás dátuma, nem szerepelhet kétszer ugyanaz (valamennyi foglalkozás délután 5 órakor kezdődik)
- Length: a foglalkozás időtartama órában
- Trainer_id: a foglalkozást tartó kiképző azonosítója

PRESENCE – Részvétel (kapcsolótábla)

- Training_id: foglalkozás azonosító
- Dog_id: foglalkozáson résztvevő kutya azonosítója

Relációs séma:



A kezelt sémát létrehozó és a táblákat feltöltő SQL parancsok:

A felsorolt parancsok megtalálhatók a Githubra feltöltött *schema.txt* állományban másolható formában, de a *dogschoool.db* fájl is feltöltöttem, ebben elvileg már létezik a kezelt séma.

```
CREATE TABLE OWNERS(  
    CODE INTEGER PRIMARY KEY AUTOINCREMENT,  
    NAME CHAR(30) NOT NULL,  
    EMAIL CHAR(30) NOT NULL,  
    DATE_OF_BIRTH DATE,  
    HOURLY_FEE INTEGER NOT NULL);  
CREATE TABLE DOGS(  
    ID INTEGER PRIMARY KEY AUTOINCREMENT,  
    NAME CHAR(30) NOT NULL,  
    BREED CHAR(30),  
    DATE_OF_BIRTH DATE,  
    WEIGHT FLOAT NOT NULL,  
    GENDER CHAR(3) NOT NULL,  
    OWNER_ID INTEGER REFERENCES OWNERS(CODE) NOT NULL);  
CREATE TABLE TRAINERS(  
    TCODE INTEGER PRIMARY KEY AUTOINCREMENT,  
    NAME CHAR(30) NOT NULL,  
    EMAIL CHAR(30) NOT NULL,  
    EXPERTISE CHAR(30));  
CREATE TABLE TRAININGS(  
    TID INTEGER PRIMARY KEY AUTOINCREMENT,  
    DAY DATE NOT NULL UNIQUE,  
    LENGTH INTEGER NOT NULL,  
    TRAINER_ID INTEGER NOT NULL REFERENCES TRAINERS(TCODE));  
CREATE TABLE PRESENCE(  
    TRAINING_ID INTEGER NOT NULL REFERENCES TRAININGS(TID),  
    DOG_ID INTEGER NOT NULL REFERENCES DOGS(ID),  
    PRIMARY KEY(TRAINING_ID, DOG_ID));  
CREATE TABLE USERS(  
    USERNAME CHAR(30) PRIMARY KEY,  
    PASSWORD CHAR(30) NOT NULL);  
  
INSERT INTO USERS VALUES('proba', 'proba123');  
  
INSERT INTO OWNERS VALUES(1, 'Nagy Béla', 'nagybela@gmail.com', '1990-10-21', 5000);  
INSERT INTO OWNERS VALUES(2, 'Mancs Menhely', 'mancs@gmail.com', null, 0);  
INSERT INTO OWNERS VALUES(3, 'Kovács Cecília', 'kovacs23@citromail.hu', '1980-07-11',  
4000);  
INSERT INTO OWNERS VALUES(4, 'Kerepesi Fanni', 'kfanni2001@gmail.com', '2001-02-04',  
3000);  
INSERT INTO OWNERS VALUES(5, 'Csengey Imre', 'csimre@gmail.com', '1987-11-10', 5000);  
  
INSERT INTO DOGS VALUES(10, 'Muci', 'pudli', '2021-03-04', 15.7, 'ISz', 1);  
INSERT INTO DOGS VALUES(11, 'Vakarc', 'csivava', '2020-09-16', 6.4, 'IK', 3);  
INSERT INTO DOGS VALUES(12, 'Rex', 'német juhász', null, 34.7, 'IK', 2);  
INSERT INTO DOGS VALUES(13, 'Mancs', null, null, 25.5, 'K', 2);
```

```
INSERT INTO DOGS VALUES(14, 'Tappancs', 'golden retriever', '2022-09-30', 20.9, 'Sz', 2);
INSERT INTO DOGS VALUES(15, 'Frakk', 'magyar vizsla', '2022-03-01', 28.6, 'IK', 4);
INSERT INTO DOGS VALUES(16, 'Lukrécia', 'yorkshire terrier', '2022-06-18', 3.1, 'IK', 4);
INSERT INTO DOGS VALUES(17, 'Szerénke', 'tacsókó', null, 6.8, 'ISz', 4);
```

```
INSERT INTO TRAINERS VALUES(1000, 'Eb Zebulon', 'zeb@email.hu', 'kutyapszichológia');
INSERT INTO TRAINERS VALUES(1001, 'Chi Va Va', 'chvv@gmail.com', 'engedelmesség');
INSERT INTO TRAINERS VALUES(1002, 'Uga Tóbiás', 'tobiasuga@gmail.com', 'agility');
```

```
INSERT INTO TRAININGS VALUES(100, '2023-01-20', 1, 1000);
INSERT INTO TRAININGS VALUES(101, '2023-02-05', 2, 1001);
INSERT INTO TRAININGS VALUES(102, '2023-03-02', 2, 1001);
INSERT INTO TRAININGS VALUES(103, '2023-03-15', 1, 1002);
```

```
INSERT INTO PRESENCE VALUES(100, 12);
INSERT INTO PRESENCE VALUES(100, 13);
INSERT INTO PRESENCE VALUES(100, 14);
INSERT INTO PRESENCE VALUES(101, 10);
INSERT INTO PRESENCE VALUES(101, 15);
INSERT INTO PRESENCE VALUES(101, 16);
INSERT INTO PRESENCE VALUES(101, 17);
INSERT INTO PRESENCE VALUES(102, 10);
INSERT INTO PRESENCE VALUES(102, 14);
INSERT INTO PRESENCE VALUES(103, 15);
INSERT INTO PRESENCE VALUES(103, 16);
INSERT INTO PRESENCE VALUES(103, 17);
```

Java kód: <https://github.com/enikop/AdatbázisRendszerek2>

JDK verzió: 17

A program funkciói:

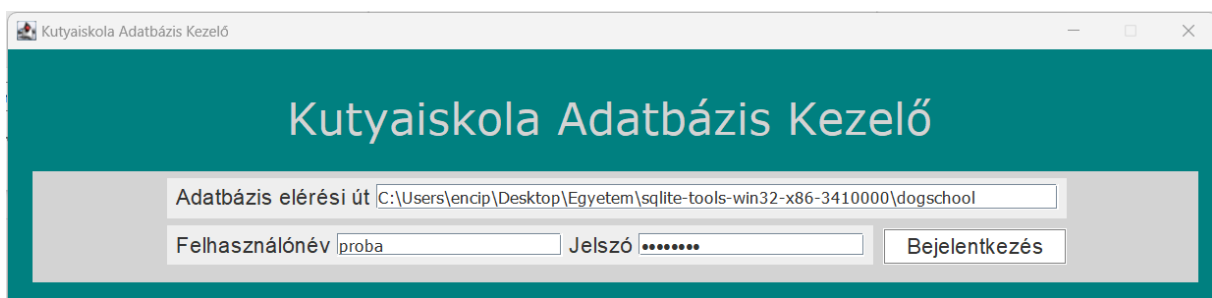
- Adatbáziskezeléssel foglalkozó osztály: **jdbc.DataBaseHandler**
- Konstruktora ResultSetet dolgoz fel, abból csinál táblát, így szintén ide kapcsolódik: **gui.DataBaseTableModel**
- A sémaellenőrzéshez felhasznált osztályok a **schemacheck** csomagban találhatók

A feladat kötelező lépései:

1. Bejelentkezési modul

Belépéshez szükséges adatok:

- az adatbázist tartalmazó fájl teljes elérési útja
- felhasználónév: proba
- jelszó: proba123



Bejelentkezéskor a felhasználónév és jelszó ellenőrzése az adatbázison keresztül történik, ezt a DataBaseHandler osztály *login* metódusa tartalmazza.

Az adatbáziskapcsolat felvételekor új SQLiteConfig létrehozásával értem el, hogy az SQLite-ra rá legyen kényszerítve az idegen kulcsok ellenőrzése adatfelvitelkor és törléskor.

```
//specialis konfiguráció készítése, hogy az sqlite elvegye a foreign key ellenőrzéseket
Class.forName("org.sqlite.JDBC");
SQLiteConfig config = new SQLiteConfig();
config.enforceForeignKeys(true);
connection = DriverManager.getConnection(connectionString, config.toProperties());
```

2. Adatok (új rekordok) felvitele (min. 2 táblára)

Bejelentkezést követően az első szintű menüben az „Adatfelvitel” opciót kell választani, második szinten pedig azt a táblát, amibe adatot szeretnénk felvinni.

Az idegen kulcs mezők lehetséges értékei (például foglalkozás felvitelénél az azt tartó kiképző azonosítója) dinamikusan vannak kilistázva a létező idegenkulcs értékek közül (*getValidIds* metódus), így minimalizálva a felhasználói hibák lehetőségét. A foglalkozások felvételekor lehetőség van az azokon résztvevő kutyákat is egyből felsorolni, azaz a PRESENCE táblába is rekordokat felvinni egyúttal. Egy JList objektummal oldottam meg, hogy a ctrl billentyűt

használva egyszerre több kutya azonosító is kijelölhető legyen a felsoroltak közül (a képen látható).

Az adatfelvitelt megvalósító metódusok a DataBaseHandler osztályban:

- insertDog
- insertOwner
- insertTrainer
- insertTraining
- insertPresence

Kutyaiskola Adatbázis Kezelő

Adatfelvitel OK

Foglalkozás OK

Foglalkozás adatainak felvétele (több résztvevő választása: ctrl)

Dátum (yyyy-MM-dd) 2023-04-05

Időtartam (óra) 2

Kiképző azonosító 1001

Résztevő kutyák azonosítói

10
11
12
13
14
15

OK

3. Adatok lementése helyi TXT állományba

A `DataBaseHandler` `printResultSetToFile`, `saveTableToTxt` és `saveDBToTxt` metódusai segítségével, az első szintű menüben a „Mentés txt-be” lehetőséget kiválasztva az adatbázis tábláit a táblanevekkel megegyező nevű .txt fájlokba lehet kimenteni a projekt gyökérkönyvtárába. Minden tábla kiírásra kerül, kivéve természetesen a `USERS` táblát. A kiírás táblázatszerű formázással történik.

CODE	NAME	EMAIL	DATE_OF_BIRTH	HOURLY_FEE
1	Nagy Béla	nagybela@gmail.com	1990-10-21	5000
2	Mancs Menhely	mancs@gmail.com	nincs megadva	0
3	Kovács Cecília	kovacs23@citromail.hu	1980-07-11	4000
4	Kerepesi Fanni	kfanni2001@gmail.com	2001-02-04	3000
5	Csengey Imre	csimre@gmail.com	1987-11-10	5000

4. Adatok lekérdezése (szűrés egy és több mező szerint, egy és kapcsolt táblákra)

Az első szintű menüben a „Lekérdezés” lehetőséget kiválasztva többféle lekérdezésre is lehetőségünk nyílik. Ezeket a DataBaseHandler alábbi metódusai végzik el:

- listAll (adott tábla valamennyi rekordja, az összes tábla lekérdezhető)
- listSumToPay (kapcsolt táblák – 4 db, paraméter nélküli)
- listOwnersByDogNumber (kapcsolt táblák – 2 db, paraméter nélküli)
- listDogsByGenderHeavierThan (egy tábla, két paraméter)
- listTrainingDetails (kapcsolt táblák – 3 db, egy paraméter)
- listDogsOwnedBy (kapcsolt táblák – 2 db, egy paraméter)
- listDogsYoungerThan (egy tábla, egy paraméter)

Kutyaiskola Adatbázis Kezelő

Lekérdezés OK

Adott kutya részvétele foglalkozásokon OK

Foglalkozás

Részvétel

Gazdik tartozásai

Gazdik kutyaainak száma

Adott kutya részvétele foglalkozásokon

Adott nemű nehéz kutyák

Adott gazdi kutyái

X hónapnál fiatalabb kutyák

Kutyaiskola Adatbázis Kezelő

Lekérdezés OK

Adott nemű nehéz kutyák OK

Adott nemű, adott tömegnél nehezebb kutyák listázása

Kutya neve K - Kan

Minimum tömeg (kg)

A felsorolt metódusok által létrehozott ResultSeteket a gui.DataBaseTableModel osztály konstruktora dolgozza fel és veszi fel egy tábla modellbe, melyet aztán JTable-ként egy felugró ablakban meg is jelenít a program.

Gazdik tartozásai OK

Lekérdezés eredménye

CODE	NAME	AMOUNT_DUE	TOTAL_HOURS	PERHOUR
5	Csengey Imre	0	0	5000
1	Nagy Béla	20000	4	5000
2	Mancs Menhely	0	5	0
3	Kovács Cecília	0	0	4000
4	Kerepesi Fanni	27000	9	3000

A JDBC-ben megvalósítandó elemek:

Normál parancsok használata:

- listAll
- listSumToPay
- listOwnersByDogNumber

Paraméterezett kétfázisú parancsok használata:

- Valamennyi insert-, delete- és update- metódus: felhasználói input alapúak és gyakran használatosak
- Azok a select parancsok, melyekhez felhasználói paramétereket kértem be (*listDogsByGenderHeavierThan*, *listTrainingDetails*, *listDogsOwnedBy*, *listDogsYoungerThan*): biztonsági okokból, SQLInjectiont elkerülendő
- Az insert-, list- és a delete- metódusokban használt prepared statementek a DataBaseHandler osztály adatai
 - o az osztály konstruktorában hívódik meg a *prepareStatements* metódus, melyben az összes ilyen parancs előkészítése lezajlik
 - o a *disconnect* metódusban a kapcsolat lezárása előtt ezeket is lezárjuk

Kurzor kezelés:

- Minden lekérdezésnél

Metaadatok lekérdezése:

- *queryMetaData* metódus a DataBaseHandler osztályban
 - o lekérdezi és Stringként visszaadja az adatbázis táblaneveit, a táblákban a kulcsmezőket, a mezők neveit, típusait, és hogy lehetnek-e üresek
 - o a menüben a „Metaadat összesítő” lehetőséget kiválasztva kiíródnak a séma adatai

Metaadat összesítő			OK
Táblanév: DOGS			
FK: ID			
Mezőnév	Mezőtípus	Lehet null (ha nem FK)	

ID	INTEGER	1	
NAME	CHAR	0	
BREED	CHAR	1	
DATE_OF_BIRTH	DATE	1	
WEIGHT	FLOAT	0	
GENDER	CHAR	0	
OWNER_ID	INTEGER	0	
Táblanév: OWNERS			
FK: CODE			
Mezőnév	Mezőtípus	Lehet null (ha nem FK)	

CODE	INTEGER	1	
NAME	CHAR	0	
EMAIL	CHAR	0	
DATE_OF_BIRTH	DATE	1	
HOURLY_FEE	INTEGER	0	
Táblanév: PRESENCE			
FK: DOG_ID TRAINING_ID			
Mezőnév	Mezőtípus	Lehet null (ha nem FK)	

TRAINING_ID	INTEGER	0	

Tranzakció kezelés:

- az *insertTraining* és a *deleteTraining* metódusokban
 - o egyszerre történik adatfelvitel a TRAININGS és a PRESENCE táblákba, amikor foglalkozást veszünk fel, felvehetjük egyben a résztvevőket is
 - o amikor egy foglalkozást törölünk, töröljük az esetlegesen rá vonatkozó részvételi adatokat is
 - o ezek a műveletek egységet alkotnak, vagy mindegyik végrehajtódik, vagy egyik sem, így ezeket tranzakcióként hajtom végre
 - o ha a végrehajtás során kivétel dobódik, akkor rollback-kel visszavonom a változtatásokat, egyébként commit-olok
 - o ezt követően visszatérek AutoCommit módba

```
//Foglalkozas es a resztvevok torlese tranzakciokent
public int deleteTraining(int tid) throws SQLException {
    Statement stmt;
    int count=-1;
    try {
        connection.setAutoCommit(false);
        stmt = connection.createStatement();
        int particNum = stmt.executeUpdate("DELETE FROM PRESENCE WHERE TRAINING_ID = "+tid);
        deleteTrainingsPrepared.setInt(1, tid);
        if (deleteTrainingsPrepared.executeUpdate() == 0)
            throw new SQLException("Nincs ilyen azonosítójú foglalkozás.");
        connection.commit();
        count=particNum;
    } catch(SQLException e) {
        connection.rollback();
        throw new SQLException(e.getMessage());
    } finally {
        connection.setAutoCommit(true);
    }
    return count;
}
```

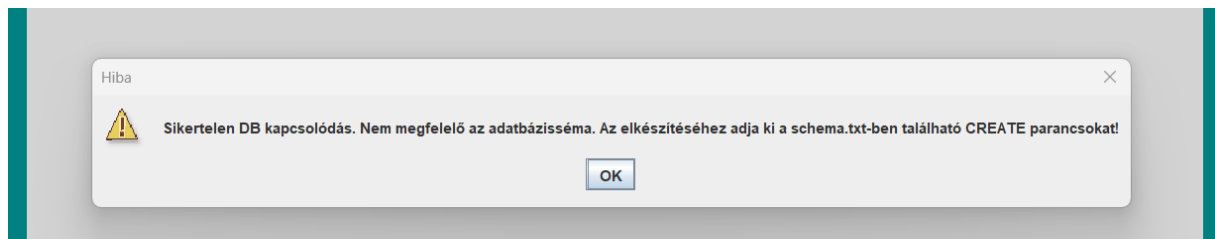
Extra feladatok:

1. GUI felület

A gui csomagban 10 grafikus osztály található, melyek a grafikus felület egy-egy komponensének megjelenítéséért felelősek.

2. Séma ellenőrzés

A programba történő bejelentkezéskor a schemacheck csomag osztályainak segítségével megtörténik az adatbázis séma ellenőrzése. Ha nincs meg az a séma, amit az alkalmazás kezelni tud, akkor helyes felhasználónév és jelszó megadása után sem engedi továbblépni a felhasználót a program.



Ami ellenőrzésre kerül:

- léteznek-e a megadott nevű táblák?

- léteznek-e a táblában a megadott nevű mezők?
 - a mezők típusa megegyezik-e az elvárt típusokkal?
3. UPDATE, DELETE, INSERT mindegyik táblára – mert így kerek-egész a program