

Adatbázis rendszerek II.

Első beadandó feladat

Lénárt Zsófia Eszter
BV9CU6

Gyakorlatvezető: Dr. Kovács László
Gyakorlat időpontja: hétfő 10:00
Képzés: BSc nappali

KISGYŐR, 2023

A feladat leírása:

Készítsen Java nyelven egy kliens oldali nyilvántartó alkalmazást relációs adatbázis kezelésére. A kapcsolódó adatbázis séma az Adatbázis I tárgy keretében kidolgozott sémával egyezik meg. Az adatbázis mérete min 5 tábla legyen kapcsolatokkal. A feladat a kliens oldali JDBC alapú Java program megtervezése és megvalósítása.

A feladat kötelező lépései:

- bejelentkezési modul
- adatok (új rekordok) felvitele (min. 2 táblára)
- adatok lementése helyi TXT állományba
- adatok lekérdezése (szűrés egy és több mező szerint, egy és kapcsolt táblákra)
- A JDBC-ben megvalósítandó elemek:
 - normál parancsok használata
 - paraméterezett kétfázisú parancsok használata
 - kurzor kezelés
 - metaadatok lekérdezése
 - tranzakció kezelés

Az ezen kívüli egyedi ötletek plusz pontot jelentenek.

A megoldást egy PDF dokumentumban kell megadni.

A dokumentum tartalma:

- fedőlap
- feladat leírása
- relációs modell
- Java kód
- program kezelő felületről snapshot

A feladat leírás a GitHub rendszeren keresztül is átadható.

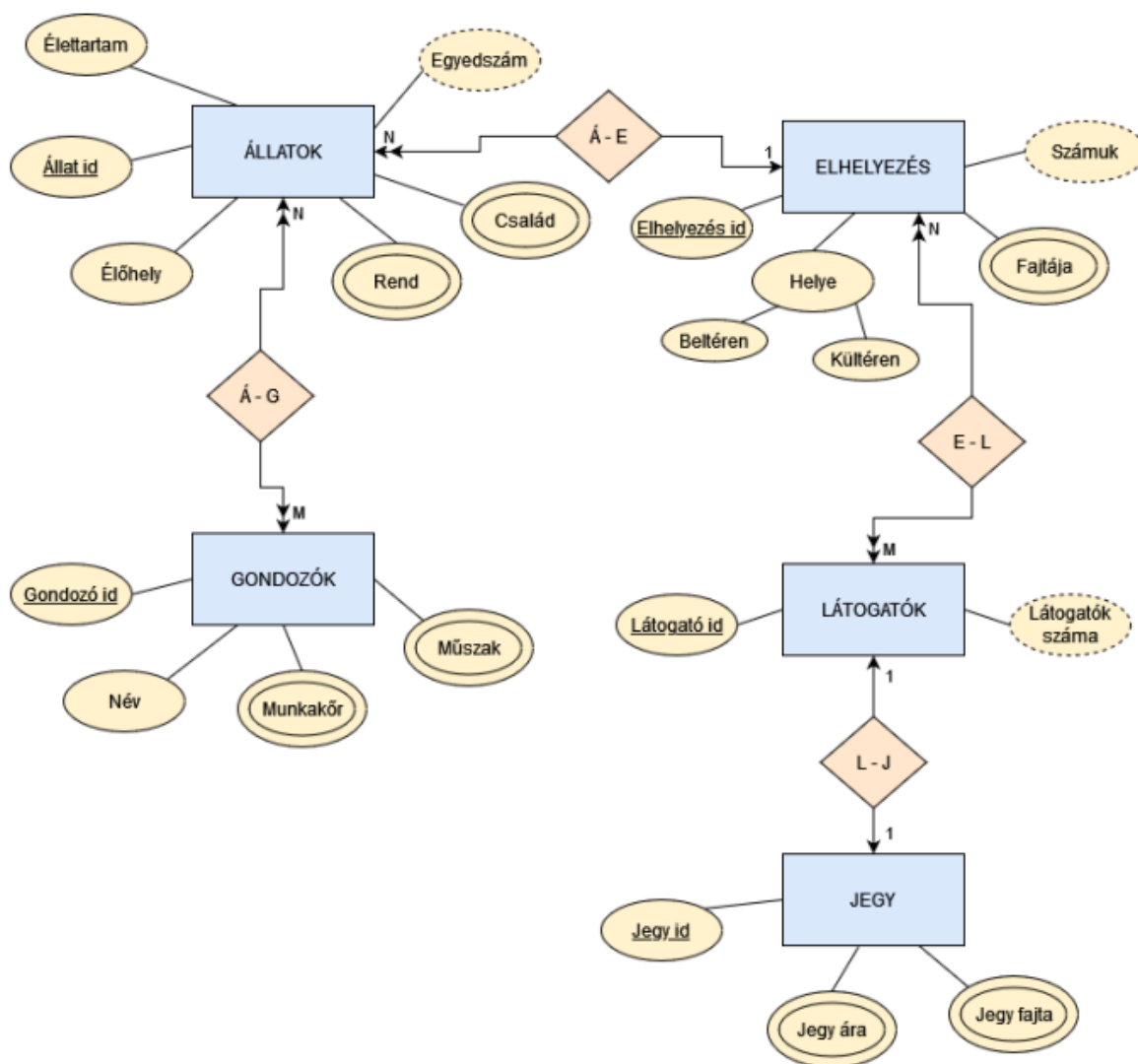
Határidő: 2023. április 15

Az adatbázis, amely alapjául szolgált a feladat megoldásához, az Adatbázis Rendszerek I. tantárgyhoz készített féléves beadandó részeként került létrehozásra.

Adatbázis leírása:

Az adatbázisom témája egy **állatkert**, a benne lévő állatokkal, kifutókkal, az ott dolgozó gondozókkal, a látogatókkal és általuk megvásárolt jegyekkel.

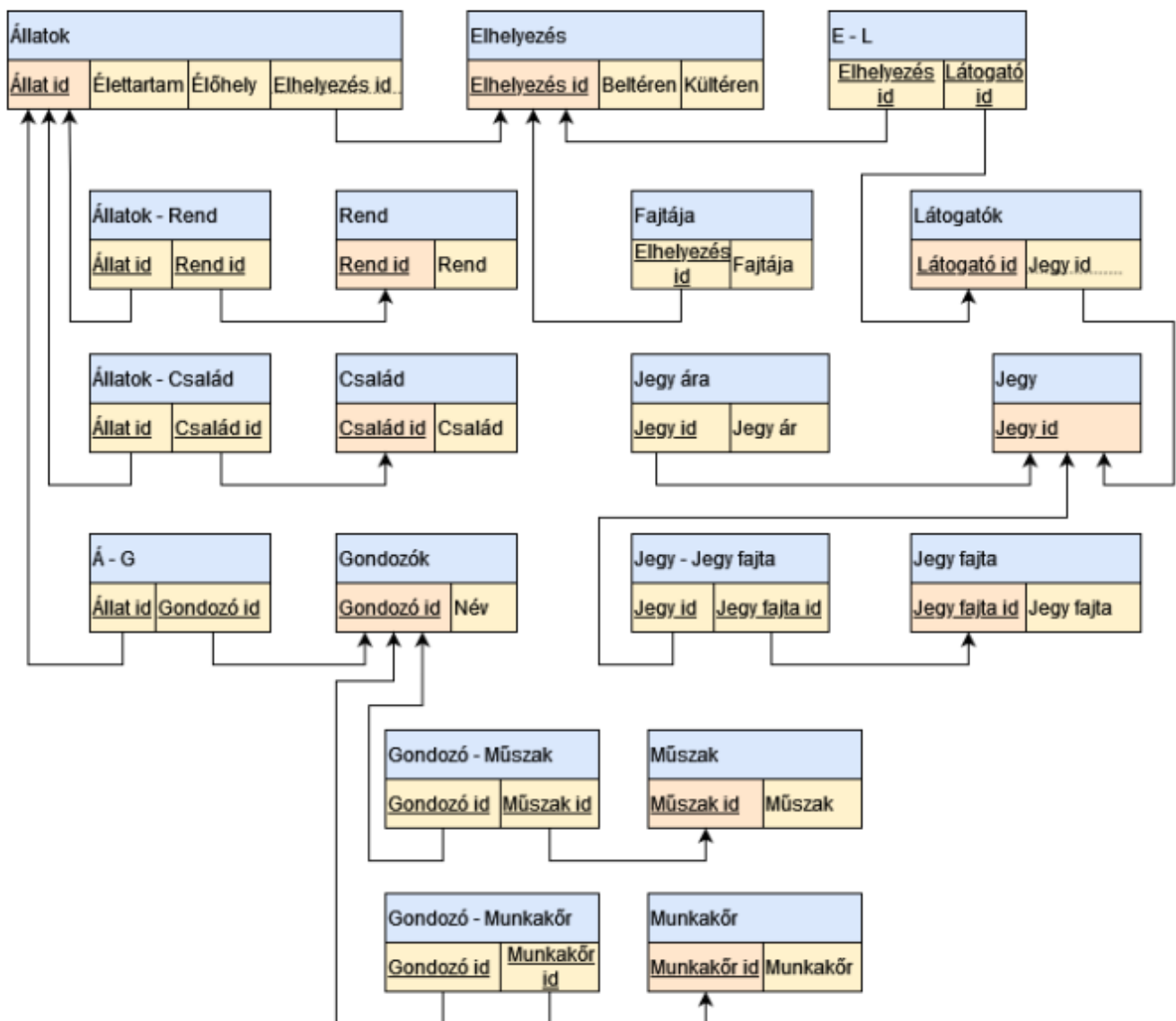
Az adatbázis ER modellje:



Mezők:

- Minden id mező a tábla elsődleges kulcsa, amely egész típusú, növekvő érték lesz
- Értelme szerint kapnak értéket a táblák, vagyis például, az Élettartam és a Jegy ár mező csak numerikus értéket vehet fel; a Jegy fajta mező egyedi értékeket vehet fel; a Műszak, Munkakör, Rend és Család, csak meghatározott értékeken belül vehet fel szöveges értéket (azaz például van x db műszak, y db munkakör, z db család stb.)
- A relációs modellből a származtatott tulajdonság kimarad, de képlettel számolható.
- A relációs modellből az összetett tulajdonságok neve szintén kimarad.

Az adatbázis relációs modellje:



A táblákat és a táblák feltöltését cmd-ben valósítottam meg:

```
C:\Windows\System32\cmd.exe X Settings X + v
C:\sql3>sqlite3
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open allattest.db
sqlite> CREATE TABLE elhelyezes(elhelyezes_id int primary key, belteren number(1), kulteren number(1), szamuk int);
sqlite> .tables
elhelyezes
sqlite> CREATE TABLE jegy(jegy_id int primary key);
sqlite> CREATE TABLE latogatok(latogato_id int primary key, jegy_id int unique, foreign key(jegy_id) references jegy(jegy_id));
sqlite> CREATE TABLE allatok(allat_id int primary key, nev varchar(20), elohely varchar(20), elettartam int, egyedszam int, elhelyezes_id int, foreign key(elhelyezes_id) references elhelyezes(elhelyezes_id));
sqlite> CREATE TABLE gondozok(gondozo_id int primary key, nev varchar(20));
sqlite> CREATE TABLE A-G(allat_id int, gondozo_id int, primary key(allat_id, gondozo_id), foreign key(allat_id) references allatok(allat_id), foreign key(gondozo_id) references gondozok(gondozo_id));
Parse error: near "-": syntax error
CREATE TABLE A-G(allat_id int, gondozo_id int, primary key(allat_id, gondozo_id)
----- error here
sqlite> CREATE TABLE A-G(allat_id int, gondozo_id int, primary key(allat_id, gondozo_id), foreign key(allat_id) references allatok(allat_id), foreign key(gondozo_id) references gondozok(gondozo_id));
sqlite> CREATE TABLE E-L(elhelyezes_id int, latogato_id int, primary key(elhelyezes_id, latogato_id), foreign key(elhelyezes_id) references elhelyezes(elhelyezes_id), foreign key(latogato_id) references latogatok(latogato_id));
sqlite> .tables
A_G      allatok      gondozok      latogatok
E_L      elhelyezes  jegy
sqlite> insert into elhelyezes values (1, '1', '0', '8');
sqlite> insert into elhelyezes values (2, '0', '1', '11');
sqlite> insert into allatok values(1, 'tigris', 'azsia', '20', 2);
Parse error: table allatok has 6 columns but 5 values were supplied
sqlite> insert into allatok values(2, 'leopard', 'azsia', '17', 2); insert into allatok values(3, 'szurikata', 'afrika', '12', 2); insert into allatok values(4, 'damszarvas', 'europa', '25', 2); insert into allatok values(5, 'barnamedve', 'eurazsia', '40', 2); insert into allatok values(6, 'emu', 'ausztralia', '20', 2); insert into allatok values(7, 'selyemmajom', 'del-amerika', '12', 1); insert into allatok values(8, 'hollo', 'europa', '40', 1); insert into allatok values(9, 'guanako', 'del-amerika', '30', 2); insert into allatok values(10, 'oroszlán', 'afrika', '20', 2); insert into allatok values(11, 'lama', 'del-amerika', '20', 2);
Parse error: table allatok has 6 columns but 5 values were supplied
sqlite> DROP TABLE allatok
...> ;
sqlite> CREATE TABLE allatok(allat_id int primary key, nev varchar(20), elohely varchar(20), elettartam int, egyedszam int, elhelyezes_id int, foreign key(elhelyezes_id) references elhelyezes(elhelyezes_id));
sqlite> DROP TABLE allatok;
sqlite> CREATE TABLE allatok(allat_id int primary key, nev varchar(20), elohely varchar(20), elettartam int, egyedszam int, foreign key(elhelyezes_id) references elhelyezes(elhelyezes_id));
Parse error: unknown column 'elhelyezes_id' in foreign key definition
sqlite> CREATE TABLE allatok(allat_id int primary key, nev varchar(20), elohely varchar(20), elettartam int, egyedszam int, elhelyezes_id int, foreign key(elhelyezes_id) references elhelyezes(elhelyezes_id));
sqlite> DROP TABLE allatok
...> ;
sqlite> .tables
A_G      E_L      elhelyezes  gondozok  jegy      latogatok
sqlite> CREATE TABLE allatok(allat_id int primary key, nev varchar(20), elohely varchar(20), elettartam int, egyedszam int, elhelyezes_id int, foreign key(elhelyezes_id) references elhelyezes(elhelyezes_id));
sqlite> insert into allatok values(1, 'tigris', 'azsia', '20', 2, 2);
sqlite> insert into allatok values(2, 'leopard', 'azsia', '17', 2, 2);
sqlite> insert into allatok values(3, 'szurikata', 'afrika', '12', 2, 2);
sqlite> insert into allatok values(4, 'damszarvas', 'europa', '25', 2, 2);
sqlite> insert into allatok values(5, 'barnamedve', 'eurazsia', '40', 2, 2);
sqlite> insert into allatok values(6, 'emu', 'ausztralia', '20', 2, 2);
sqlite> insert into allatok values(7, 'selyemmajom', 'del-amerika', '12', 1, 1);
sqlite> insert into allatok values(8, 'hollo', 'europa', '40', 1, 1);
sqlite> insert into allatok values(9, 'guanako', 'del-amerika', '30', 2, 2);
sqlite> insert into allatok values(10, 'oroszlán', 'afrika', '20', 2, 2);
sqlite> insert into allatok values(11, 'lama', 'del-amerika', '20', 2, 2);
sqlite> insert into gondozok values(1, 'Kiss Istvan'); insert into gondozok values(2, 'Enekes Zoltan'); insert into gondozok values(3, 'Mezei Virag'); insert into gondozok values(4, 'Nagy Mate'); insert into gondozok values(5, 'Pasztor Miklos'); insert into gondozok values(6, 'Tamas Gyula'); insert into gondozok values(7, 'Magyar Bernadett'); insert into gondozok values(8, 'Sandor Szilvia');
sqlite> insert into jegy values(1);
sqlite> insert into jegy values(2);
sqlite> insert into jegy values(3);
sqlite> insert into jegy values(4);
sqlite> insert into jegy values(5);
sqlite> insert into jegy values(6);
sqlite> insert into jegy values(7);
sqlite> insert into latogatok values(1, 1);
sqlite> insert into latogatok values(2, 2);
sqlite> insert into latogatok values(3, 3);
sqlite> insert into latogatok values(4, 4);
sqlite> insert into latogatok values(5, 5);
sqlite> insert into latogatok values(6, 6);
sqlite> insert into latogatok values(7, 7);
sqlite> insert into A-G values (1, 4);
sqlite> insert into A-G values (2, 3);
sqlite> insert into A-G values (3, 8);
sqlite> insert into A-G values (4, 3);
sqlite> insert into A-G values (5, 1);
sqlite> insert into A-G values (6, 5);
sqlite> insert into A-G values (7, 6);
sqlite> insert into A-G values (8, 8);
sqlite> insert into A-G values (9, 7);
sqlite> insert into A-G values (10, 1);
sqlite> insert into A-G values (11, 2);
sqlite> insert into E-L values (1, 1);
sqlite> insert into E-L values (2, 1);
sqlite> insert into E-L values (1, 2);
sqlite> insert into E-L values (2, 2);
sqlite> insert into E-L values (1, 3);
sqlite> insert into E-L values (2, 3);
sqlite> insert into E-L values (1, 4);
sqlite> insert into E-L values (2, 4);
sqlite> insert into E-L values (1, 5);
sqlite> insert into E-L values (2, 5);
sqlite> insert into E-L values (1, 6);
sqlite> insert into E-L values (2, 6);
sqlite> insert into E-L values (1, 7);
sqlite> insert into E-L values (2, 7);
```

Aktiválja a Windowst
Aktiválja a Windows rendszert a Gépházban.

```
C:\Windows\System32\cmd.exe X Settings X + v
sqlite> insert into allatok values(4, 'damszarvas', 'europa', '25', 2, 2);
sqlite> insert into allatok values(5, 'barnamedve', 'eurazsia', '40', 2, 2);
sqlite> insert into allatok values(6, 'emu', 'ausztralia', '20', 2, 2);
sqlite> insert into allatok values(7, 'selyemmajom', 'del-amerika', '12', 1, 1);
sqlite> insert into allatok values(8, 'hollo', 'europa', '40', 1, 1);
sqlite> insert into allatok values(9, 'guanako', 'del-amerika', '30', 2, 2);
sqlite> insert into allatok values(10, 'oroszlán', 'afrika', '20', 2, 2);
sqlite> insert into allatok values(11, 'lama', 'del-amerika', '20', 2, 2);
sqlite> insert into gondozok values(1, 'Kiss Istvan'); insert into gondozok values(2, 'Enekes Zoltan'); insert into gondozok values(3, 'Mezei Virag'); insert into gondozok values(4, 'Nagy Mate'); insert into gondozok values(5, 'Pasztor Miklos'); insert into gondozok values(6, 'Tamas Gyula'); insert into gondozok values(7, 'Magyar Bernadett'); insert into gondozok values(8, 'Sandor Szilvia');
sqlite> insert into jegy values(1);
sqlite> insert into jegy values(2);
sqlite> insert into jegy values(3);
sqlite> insert into jegy values(4);
sqlite> insert into jegy values(5);
sqlite> insert into jegy values(6);
sqlite> insert into jegy values(7);
sqlite> insert into latogatok values(1, 1);
sqlite> insert into latogatok values(2, 2);
sqlite> insert into latogatok values(3, 3);
sqlite> insert into latogatok values(4, 4);
sqlite> insert into latogatok values(5, 5);
sqlite> insert into latogatok values(6, 6);
sqlite> insert into latogatok values(7, 7);
sqlite> insert into A-G values (1, 4);
sqlite> insert into A-G values (2, 3);
sqlite> insert into A-G values (3, 8);
sqlite> insert into A-G values (4, 3);
sqlite> insert into A-G values (5, 1);
sqlite> insert into A-G values (6, 5);
sqlite> insert into A-G values (7, 6);
sqlite> insert into A-G values (8, 8);
sqlite> insert into A-G values (9, 7);
sqlite> insert into A-G values (10, 1);
sqlite> insert into A-G values (11, 2);
sqlite> insert into E-L values (1, 1);
sqlite> insert into E-L values (2, 1);
sqlite> insert into E-L values (1, 2);
sqlite> insert into E-L values (2, 2);
sqlite> insert into E-L values (1, 3);
sqlite> insert into E-L values (2, 3);
sqlite> insert into E-L values (1, 4);
sqlite> insert into E-L values (2, 4);
sqlite> insert into E-L values (1, 5);
sqlite> insert into E-L values (2, 5);
sqlite> insert into E-L values (1, 6);
sqlite> insert into E-L values (2, 6);
sqlite> insert into E-L values (1, 7);
sqlite> insert into E-L values (2, 7);
```

Aktiválja a Windowst
Aktiválja a Windows rendszert a Gépházban.

```

C:\Windows\System32\cmd.e  X  +  v
sqlite> DROP TABLE allat_gondozo_kapcsolat
...> ;
sqlite> .tables
E_L      allatok      elhelyezes  gondozok    jegy      latogatok
sqlite> CREATE TABLE allat_gondozo_kapcsolat (
(x1...> allat_id int,
(x1...> gondozo_id int,
(x1...> foreign key(allat_id) references allatok(allat_id),
(x1...> foreign key(gondozo_id) references gondozok(gondozo_id)
(x1...> );
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (1, 1);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (2, 1);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (3, 2);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (4, 3);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (5, 3);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (6, 4);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (7, 8);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (8, 5);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (9, 5);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (10, 6);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (11, 7);
sqlite> .quit

C:\sql3>sqlite3
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open allatkert.db
sqlite> DROP TABLE allat_gondozo_kapcsolat
...> ;
sqlite> CREATE TABLE allat_gondozo_kapcsolat(allat_id int, gondozo_id int, primary key(allat_id, gondozo_id), foreign key(allat_id) references allatok(allat_id), foreign key(gondozo_id) references gondozok(gondozo_id));
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (1, 1);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (2, 1);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (3, 2);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (4, 3);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (4, 3);
Runtime error: UNIQUE constraint failed: allat_gondozo_kapcsolat.allat_id, allat_gondozo_kapcsolat.gondozo_id (19)
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (6, 4);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (5, 3);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (7, 8);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (8, 5);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (9, 5);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (10, 6);
sqlite> INSERT INTO allat_gondozo_kapcsolat VALUES (11, 7);
sqlite> .quit

```

JAVA – kód részekre bontva:

A teljes kódot külön csatoltam a pdf mellé.

Az **import** szó a Java nyelvben arra szolgál, hogy az egyik fájlban definiált osztályokat, interfészeket vagy más típusokat más fájlokban is felhasználhassuk. Az import segítségével

lehetővé válik, hogy az egyik fájlban definiált típusokat más fájlokban használjuk anélkül, hogy a típus nevét minden alkalommal ki kellene írunk teljesen. Az importált típusokat vagy a teljes csomagnevet, vagy csak a típusnevet adhatjuk meg.

```
package sqlite;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;
import java.util.Scanner;
import java.io.PrintWriter;
import java.io.IOException;
```

Ez a kód egy privát adattagot definiál a **Connect** típusú **conn** névvel. Az **Connect** típus egy adatbázis-kapcsolatot reprezentáló objektum, ami valószínűleg a programban több helyen is használatos lesz. A privát adattag azt jelenti, hogy csak a main függvényen belül és az osztályon belül lehet hivatkozni a **conn** változóra, és nem lehet kívülről módosítani. Ez a privátság szabályozásának egyik módja, amely lehetővé teszi az osztály implementációjának részleteinek elrejtését a külvilág előtt, és így a programot biztonságosabbá és rugalmasabbá teszi.

```
public class allatkert {

    private Connection conn;
```

Kapcsolat:

Először a **kapcsolatot** hoztam létre, külön függvényben, amit a main függvényben meghívtam.

```
//Csatlakoztatás
public Connection connect(String dbFileName) throws SQLException {
    String url = "jdbc:sqlite:C:\\sql3\\" + dbFileName;
    conn = DriverManager.getConnection(url);
    //System.out.println("Sikeres csatlakozas");
    return conn;
}
```

Beléptetés:

Ezután a **beléptetés** következett:

```
//Beleptetes
public void belepes(String felhasznalo, String jelszo) throws SQLException {
    Connection conn = this.connect("belepes.db");
    String parancs = "SELECT * FROM user WHERE felhasznalo = ? AND jelszo = ?";
    try (var pstmt = conn.prepareStatement(parancs)) {
        pstmt.setString(1, felhasznalo);
        pstmt.setString(2, jelszo);
        ResultSet rs = pstmt.executeQuery();
        if(!rs.next()) {
            throw new SQLException("Helytelen felhasznalo vagy jelszo.");
        } else {
            String felhasznaloSzerep = rs.getString("szerep");
            System.out.println("Udv, " + felhasznalo + "!");
            System.out.println("A te szerepkorod: " + felhasznaloSzerep);
            rs.close();
        }
    } catch (SQLException ex) {
        throw ex;
    }
}
```

Main függvényben:

```
public static void main(String[] args) {

    try(Scanner in = new Scanner(System.in)) {
        allatkert prog = new allatkert();

        prog.connect("belepes.db");

        System.out.println("Felhasznamev:");
        String felhasznalo = in.nextLine();
        System.out.println("Jelszo:");
        String jelszo = in.nextLine();

        try {
            //Beleptetes
            prog.belepes(felhasznalo, jelszo);
            System.out.println("Sikeres bejelentkezés!");
        }
    }
}
```


Hibás felhasználónév vagy jelszó esetén:

```
<terminated> allatkert [Java Application] C:\Program
Felhasznalonev:
lenartzse
Jelszo:
Jelszol23
Helytelen felhasznalo vagy jelszo.
```

Helyes felhasználónév és jelszó esetén:

```
Felhasznalonev:
lenartzse
Jelszo:
jelszol23
Udv, lenartzse!
A te szerepkorod: admin
Sikeres bejelentkezés!
```

Menü rendszer:

Ez után a **menüt** csináltam meg, hogy a felhasználó könnyebben eligazodjon a programban, jobb átláthatóság miatt:

```
Fo menu:
1. Uj rekord felvitele ide: allatok tabla
2. Uj rekord felvitele ide: gondozok tabla
3. Lekerdezések
4. Kilepes
```

```
int option = 0;
while (option !=4 ) {
    //Menu létrehozás
    System.out.println("\nFo menu:");
    System.out.println("1. Uj rekord felvitele ide: allatok tabla");
    System.out.println("2. Uj rekord felvitele ide: gondozok tabla");
    System.out.println("3. Lekerdezések");
    System.out.println("4. Kilepes");
```

1. Menü pont: Új rekordok felvitele az **allatok** táblába:

Main függvényben:

```
switch (option) {
//Új rekord felvitele ide: allatok tabla
case 1:
    System.out.println("Allat neve:");
    String Nev = in.nextLine();
    System.out.println("Allat elohelye:");
    String Elohely = in.nextLine();
    System.out.println("Allat elettartama:");
    int Elettartam = Integer.parseInt(in.nextLine());
    System.out.println("Allat egyedszama:");
    int Egyedszam = Integer.parseInt(in.nextLine());
    System.out.println("Allat elhelyezese:");
    int Elhelyezes = Integer.parseInt(in.nextLine());
    prog.add_new_allatok(Nev, Elohely, Elettartam, Egyedszam, Elhelyezes);
    System.out.println("Uj allat rekord beillesztve!");
    break;
```

```
//Új rekord felvitele ide: allatok tabla
public void add_new_allatok(String nev, String elohely, int elettartam, int egyedszam, int elhelyezes_id)
{
    int allat_id = 0;
    try
    {
        var stmt = this.conn.createStatement();
        String CommandText = "SELECT max(allat_id) FROM allatok";
        ResultSet rs = stmt.executeQuery(CommandText);
        if (rs.next()) {
            allat_id = rs.getInt(1);
        }

        System.out.println("allat_id:"+allat_id);
        CommandText = " INSERT INTO allatok VALUES (?, ?, ?, ?, ?, ?)";
        var pstmt = this.conn.prepareStatement(CommandText);
        pstmt.setInt(1, allat_id+1);
        pstmt.setString(2,nev);
        pstmt.setString(3,elohely);
        pstmt.setInt(4,elettartam);
        pstmt.setInt(5,egyedszam);
        pstmt.setInt(6,elhelyezes_id);

        pstmt.execute();

        String allatok = "allat_id: " + allat_id + " nev: " + nev + " elohely: " + elohely + " elettartam: "
            + elettartam + " egyedszam: " + egyedszam + " elhelyezes_id: " + elhelyezes_id + "\n";
        kiiras(allatok, "allatok_kiiras.txt");
    }
    catch(Exception ex) { ex.printStackTrace(); }
}
```

```

Fo menu:
1. Uj rekord felvitele ide: allatok tabla
2. Uj rekord felvitele ide: gondozok tabla
3. Lekerdezések
4. Kilepes
1
Allat neve:
Vizilo
Allat elohelye:
afrika
Allat elettartama:
35
Allat egyedszama:
3
Allat elhelyezese:
2
|allat_id:13
Uj allat rekord beillesztve!

```

2. Menü pont: Új rekordok felvitele az **gondozok** táblába:

```

//Uj rekord felvitele ide: gondozok tabla
public void add_new_gondozok(String nev)
{
    int gondozo_id = 0;
    try
    {
        var stmt = this.conn.createStatement();
        String CommandText = "SELECT max(gondozo_id) FROM gondozok";
        ResultSet rs = stmt.executeQuery(CommandText);
        if (rs.next()) {
            gondozo_id = rs.getInt(1);
        }

        System.out.println("gondozo_id:"+gondozo_id+1);
        CommandText = " INSERT INTO gondozok VALUES (?,?)";
        var pstmt = this.conn.prepareStatement(CommandText);
        pstmt.setInt(1, gondozo_id+1);
        pstmt.setString(2,nev);

        pstmt.execute();

        String gondozok = "gondozo_id: " + gondozo_id + " nev: " + nev + "\n";
        kiiras(gondozok, "gondozok_kiiras.txt");
    }
    catch(Exception ex) { ex.printStackTrace(); }
}

```

Main függvényben:

```
//Uj rekord felvitele ide: gondozok tabla
case 2:
    System.out.println("Gondozo neve:");
    String GNev = in.nextLine();
    prog.add_new_gondozok(GNev);
    System.out.println("Uj gondozo rekord beillesztve!");
    break;
```

Lekérdezések:

3. Menü pont: Lekérdezések:

Első lekérdezés: Írja ki azon állatok adatait, amelyek életkora nagyobb, mint 'x'! Az 'x'-t a felhasználó adja meg a lekérdezés meghívásakor a programkódban, a main függvényben.

```
//Írja ki azon állatok adatait amelyek életkora nagyobb, mint x
public void lekerdezes_elettartam (int szam) {

    try {
        String parancs = "SELECT * FROM allatok WHERE elettartam > ?";
        PreparedStatement pstmt = conn.prepareStatement(parancs);
        pstmt.setInt(1, szam);
        ResultSet rs = pstmt.executeQuery();
        ResultSetMetaData rsmd = rs.getMetaData();

        for(int i=1; i <= rsmd.getColumnCount(); i++) {
            String oszlopNev = rsmd.getColumnName(i);
            String oszlopTipus = rsmd.getColumnTypeName(i);
            System.out.println("Oszlop " + i + ": " + oszlopNev + " (" + oszlopTipus + ")");
        }
        while (rs.next()) {
            int allat_id = rs.getInt("allat_id");
            String nev = rs.getString("nev");
            String elohely = rs.getString("elohely");
            int elettartam = rs.getInt("elettartam");
            int egyedszam = rs.getInt("egyedszam");
            int elhelyezes_id = rs.getInt("elhelyezes_id");
            System.out.println(allat_id + " " + nev + " " + elohely + " " + elettartam + " " + egyedszam + " " + elhelyezes_id);
        }
        rs.close();
        pstmt.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

```

Oszlop 1: allat_id (INT)
Oszlop 2: nev (VARCHAR)
Oszlop 3: elohely (VARCHAR)
Oszlop 4: elettartam (INT)
Oszlop 5: egyedszam (INT)
Oszlop 6: elhelyezes_id (INT)
4 damszarvas europa 25 7 2
5 barnamedve eurazsia 40 2 2
8 hollo europa 40 4 1
9 guanako del-amerika 30 3 2
12 Lo europa 22 4 2

```

Második lekérdezés: Írja ki azon állatok nevét, élőhelyét és egyedszámát, amelyek kültéri karámban vannak tartva! (Kültéri ketrec id-je '2').

```

//Azon allatok neve, elohelye, egyedszamanak kiirasa akik kulteren vannak

```

```

public void lekerdezés_elohely() {

```

```

    try {

```

```

        String query = "SELECT nev, elohely, egyedszam FROM allatok WHERE elhelyezes_id = 2";

```

```

        Statement stmt = conn.createStatement();

```

```

        ResultSet rs = stmt.executeQuery(query);

```

```

        while (rs.next()) {

```

```

            String nev = rs.getString("nev");

```

```

            String elohely = rs.getString("elohely");

```

```

            int egyedszam = rs.getInt("egyedszam");

```

```

            System.out.println("Nev: " + nev + ", Elohely: " + elohely + ", Egyedszam: " + egyedszam);

```

```

        }

```

```

    } catch (SQLException e) {

```

```

        e.printStackTrace();

```

```

    }

```

```

}

```

```

Nev: tigris, Elohely: azsia, Egyedszam: 4

```

```

Nev: leopard, Elohely: azsia, Egyedszam: 2

```

```

Nev: szurikata, Elohely: afrika, Egyedszam: 8

```

```

Nev: damszarvas, Elohely: europa, Egyedszam: 7

```

```

Nev: barnamedve, Elohely: eurazsia, Egyedszam: 2

```

```

Nev: emu, Elohely: ausztralia, Egyedszam: 4

```

```

Nev: guanako, Elohely: del-amerika, Egyedszam: 3

```

```

Nev: oroszlan, Elohely: afrika, Egyedszam: 3

```

```

Nev: lama, Elohely: del-amerika, Egyedszam: 6

```

```

Nev: Lo, Elohely: europa, Egyedszam: 4

```

Harmadik lekérdezés: Lekérdezzük, hogy melyik állatot ki gondozza, felhasználva az állatok és a gondozók tábla közötti kapcsolatot!

```
//Lekerdezzuk hogy melyik allatot ki gondozza
public void lekerdezes_kapcsolat() {
    try {
        Statement stmt = conn.createStatement();
        String sql = "SELECT a.nev AS allat, g.nev AS gondozo " +
            "FROM allatok a JOIN allat_gondozo_kapcsolat ag ON a.allat_id = ag.allat_id " +
            "JOIN gondozok g ON ag.gondozo_id = g.gondozo_id";

        ResultSet rs = stmt.executeQuery(sql);

        while (rs.next()) {
            String allat = rs.getString("allat");
            String gondozo = rs.getString("gondozo");

            System.out.println(allat + " - " + gondozo);
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

```
tigris - Kiss Istvan
leopard - Kiss Istvan
szurikata - Enekes Zoltan
damszarvas - Mezei Virag
emu - Nagy Mate
barnamedve - Mezei Virag
selyemmajom - Sandor Szilvia
hollo - Pasztor Miklos
guanako - Pasztor Miklos
oroszlán - Tamas Gyula
lama - Magyar Bernadett
```

4. Menü pont: Kilépés

Kilépünk a programból. A program futása megáll.

Adatok lementése helyi TXT állományba:

Ehhez létrehoztam a program főkönyvtárában kettő txt formátumú fájlt:

« eclipse-wor... > JDBC_allatok >		↺ ↻	Keresés: JDBC_allatok		
Név	Módosítás dátuma	Típus	Méret		
ve					
folder .settings	2023. 04. 14. 13:52	Fájlmappa			
folder bin	2023. 04. 14. 13:56	Fájlmappa			
folder src	2023. 04. 14. 13:56	Fájlmappa			
file .classpath	2023. 04. 14. 13:54	CLASSPATH fájl	1 KB		
file .project	2023. 04. 14. 13:52	PROJECT fájl	1 KB		
file allatok_kiiras	2023. 04. 14. 22:45	Szöveges dokume...	1 KB		
file gondozok_kiiras	2023. 04. 15. 0:04	Szöveges dokume...	1 KB		
file sqlite-jdbc-3.41.0.0	2023. 04. 14. 13:54	Executable Jar File	12 537 KB		


```
//Kiiras txt fajlba
private void kiiras(String dataname, String filename) {

    try (var writer = new FileWriter(filename, true)) {
        writer.write(dataname);
    } catch (IOException e) {
        e.printStackTrace();
    }

}

public void KI(String s) {
    System.out.println(s+"\n");
}
```

Állatok kiírása:



allatok_kiiras

×

+

Fájl

Szerkesztés

Megtekintés

allat_id: 12 nev: Kutya elohely: europa elettartam: 16 egyedszam: 7 elhelyezes_id: 1

allat_id: 13 nev: Vizilo elohely: afrika elettartam: 35 egyedszam: 3 elhelyezes_id: 2

Az új rekord felvitele az állatok táblába – függvényben:

```
System.out.println("allat_id:"+allat_id);
CommandText =" INSERT INTO allatok VALUES (?, ?, ?, ?, ?, ?)";
var pstmt = this.conn.prepareStatement(CommandText);
pstmt.setInt(1, allat_id+1);
pstmt.setString(2,nev);
pstmt.setString(3,elhely);
pstmt.setInt(4,elettartam);
pstmt.setInt(5,egyedszam);
pstmt.setInt(6,elhelyezes_id);

pstmt.execute();

String allatok = "allat_id: " + allat_id + " nev: " + nev + " elohely: " + elohely + " elettartam: "
                + elettartam + " egyedszam: " + egyedszam + " elhelyezes_id: " + elhelyezes_id + "\n";
kiiras(allatok, "allatok_kiiras.txt");
}
catch(Exception ex) { ex.printStackTrace(); }
```

A gondozók kiírása:



Az új rekord felvitele a gondozók táblába – függvényben:

```
System.out.println("gondozo_id:"+gondozo_id+1);
CommandText =" INSERT INTO gondozok VALUES (?, ?)";
var pstmt = this.conn.prepareStatement(CommandText);
pstmt.setInt(1, gondozo_id+1);
pstmt.setString(2,nev);

pstmt.execute();

String gondozok = "gondozo_id: " + gondozo_id + " nev: " + nev + "\n";
kiiras(gondozok, "gondozok_kiiras.txt");
}
catch(Exception ex) { ex.printStackTrace(); }
```