

# Adatbázis Rendszerek 2

## Jegyzőkönyv

Formula 1 adatbázis kezelő

Szerző: Kovács Sándor Gergő

Neptunkód: C00KSI

Tárgyfelelős: Dr. Bednarik László

## Az első feladat leírása

Készítsen Java nyelven egy kliens oldali nyilvántartó alkalmazást relációs adatbázis kezelésére.

Az én kliensem témája egy Forma 1 adatbázis kezelő felületének létrehozása volt.

Sémához kapcsolódó elvárások:

- Legalább két táblát vonjon be az adatkezelésbe, melyek között legyen kapcsolat.
- A fő táblák szerkezete legalább 5 mezőből álljon!

Minden adatsorban legyen:

- egyéni azonosító,
- szöveges mező,
- numerikus mező,
- dátum típusú mező,
- a dátum tartalmú mező lehet szöveges típusú, de ez esetben a kódban a tartalom helyességét ellenőrizni kell!

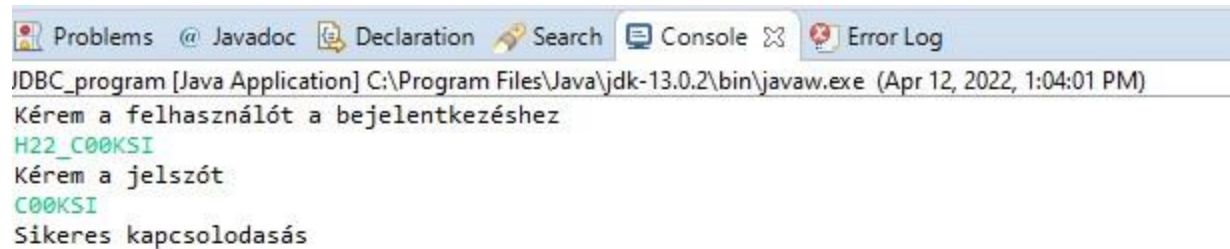
## Folyamat végrehajtása

Első Lépésként létrehoztam a szerverre csatlakozó illetve lecsatlakozó modulokat. A bejelentkezéshez meg kell adni a felhasználó nevet és jelszót különben nem enged be a rendszer.

```
public static void Connect() {  
    try {  
        System.out.println("Kérem a felhasználót a bejelentkezéshez");  
        String felhasznalo = sc.next().trim();  
        if(felhasznalo.equals("H22_C00KSI")) {  
            System.out.println("Kérem a jelszót");  
            String password = sc.next().trim();  
            if(password.equals("C00KSI")){  
                conn = DriverManager.getConnection(url,user,pwd);  
                System.out.println("Sikeres kapcsolódás\n");  
            }else { System.out.println("A jelszó nem megfelelő\n\n"); Connect();}  
        }else { System.out.println("A felhasználó nem megfelelő nem megfelelő\n\n"); Connect();}  
    }catch (Exception e) {  
        System.err.println(e.getMessage());  
    }  
}
```

Sikeres kapcsolódás után megjelenik a program főmenüje:

A baloldali kulcsszó beírása után lehet tovább végrehajtani az utasításokat.



```
Problems @ Javadoc Declaration Search Console Error Log
JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 1:04:01 PM)
Kérem a felhasználót a bejelentkezéshez
H22_C00KSI
Kérem a jelszót
C00KSI
Sikeres kapcsolódás
```

Főmenü!

Válaszd ki az alábbi lehetőségeket

Default - Feltölti az eredeti táblákat, adatokat és kötéseket

DBTorles - Törli az adatbázist

AdatFelvisz - Adatfelvitel a megadott táblába

Adattorles - Törli a pilótát az adatbázisból

Lekérdez - Dinamikus lekérdezés a pilótákról

SorLekérdez - Dinamikus lekérdezés az adott tábla egy adott soráról

Ellenorzes - Megadja egy tábla össze adatát

Kurzor - Módosítja egy csapat részvételének mennyiségét

Lekapcs - Kilép a programból

**Default:** A programban statikusan fel van véve 3 tábla (ebből 2 össze kötve egymással) és ezek a táblák statikusan fel is vannak töltve adatokkal. Ezzel a kulcsszóval lehet vissza állítani ezeket a rekordokat a mezőkben. (Fontos hogy előtte a DBTorlest vegre kell hajtani).

```
public static void StatikusTablaLetrehozas() {
    String sqlp_versenyzok="create table versenyzok(rajtszam number(2) primary key, nev char(20) not null, futamok number(3) check(futamok>0), gyozelmek number(3), debutalas date)";
    String sqlp_palyak="create table palyak (id number(3) primary key, helyszin char(80) not null, orszag char(40), futamnapja date, palyahossz number(4))";
    String sqlp_csapat="create table csapat (csapatnev char(23) primary key, futamokszama number(4) NOT NULL, gyozelmek number(3), podiumok number(3), bajnoksagok number(2), debutalas date)";

    if(conn!=null) {
        try {
            s=conn.createStatement();
            s.executeUpdate(sqlp_csapat);
            System.out.println("Csapat tábla létrehozva\n");
            s.executeUpdate(sqlp_versenyzok);
            System.out.println("Versenyzők tábla létrehozva\n");
            s.executeUpdate(sqlp_palyak);
            System.out.println("Pályák tábla létrehozva\n");
            s.close();
        }catch(Exception ex) {
            System.err.println(ex.getMessage());
        }
    }
}
```

```

public static void StatikusAdatfelvetel_1() {
    if(conn != null) {
        String sqlp[] = {
            "insert into palyak values(1,'Bahrain International Circuit, Sakhir','Bahrein',to_date('2022.03.20', 'yyyy.mm.dd'), 5412)",
            "insert into palyak values(2,'Saudi Arabia Jeddah Corniche Circuit, Jeddah','Szaúd-arábia',to_date('2022.03.27', 'yyyy.mm.dd'), 3257)",
            "insert into palyak values(3,'Australia Albert Park Circuit, Melbourne','Ausztrália', to_date('2022.04.10', 'yyyy.mm.dd'), 4120)",
            "insert into palyak values(4,'Italy Imola Circuit, Imola','Olaszország', to_date('2022.04.24', 'yyyy.mm.dd'), 3895)",
            "insert into palyak values(5,'United States Miami International Autodrome, Miami Gardens','Egyesült Államok', to_date('2022.05.08', 'yyyy.mm.dd'), 3147)",
            "insert into palyak values(6,'Spain Circuit de Barcelona-Catalunya, Montmeló','Spanyolország', to_date('2022.05.22', 'yyyy.mm.dd'), 3965)",
            "insert into palyak values(7,'Monaco Circuit de Monaco, Monaco','Monaco', to_date('2022.05.29', 'yyyy.mm.dd'), 2485)",
            "insert into palyak values(8,'Azerbaijan Baku City Circuit, Baku','Azerbajdzsán', to_date('2022.06.12', 'yyyy.mm.dd'), 4712)",
            "insert into palyak values(9,'Canada Circuit Gilles Villeneuve, Montréal','Kanada', to_date('2022.06.19', 'yyyy.mm.dd'), 5121)",
            "insert into palyak values(10,'United Kingdom Silverstone Circuit, Silverstone','Nagy-Britannia', to_date('2022.07.03', 'yyyy.mm.dd'), 4632)",
            "insert into palyak values(11,'Austria Red Bull Ring, Spielberg','Ausztria', to_date('2022.07.10', 'yyyy.mm.dd'), 3749)",
            "insert into palyak values(12,'France Circuit Paul Ricard, Le Castellet','Franciaország', to_date('2022.07.24', 'yyyy.mm.dd'), 4008)",
            "insert into palyak values(13,'Hungary Hungaroring, Mogyoród','Magyarország', to_date('2022.07.31', 'yyyy.mm.dd'), 4381)",
            "insert into palyak values(14,'Belgium Circuit de Spa-Francorchamps, Stavelot','Belgium', to_date('2022.08.28', 'yyyy.mm.dd'), 7004)",
            "insert into palyak values(15,'Netherlands Circuit Zandvoort, Zandvoort','Hollandia', to_date('2022.09.09', 'yyyy.mm.dd'), 3896)",
            "insert into palyak values(16,'Italy Monza Circuit, Monza','Olaszország', to_date('2022.09.11', 'yyyy.mm.dd'), 4294)",
            "insert into palyak values(17,'Singapore Marina Bay Street Circuit, Singapore','Szingapúr', to_date('2022.10.02', 'yyyy.mm.dd'), 3986)",
            "insert into palyak values(18,'Japan Suzuka International Racing Course, Suzuka','Japán', to_date('2022.10.09', 'yyyy.mm.dd'), 3745)",
            "insert into palyak values(19,'United States Circuit of the Americas, Austin, Texas','Egyesült Államok', to_date('2022.10.23', 'yyyy.mm.dd'), 4208)",
            "insert into palyak values(20,'Mexico Autódromo Hermanos Rodríguez, Mexico City','Mexikó', to_date('2022.10.30', 'yyyy.mm.dd'), 4398)",
            "insert into palyak values(21,'Brazil Interlagos Circuit, São Paulo','Brazília', to_date('2022.11.13', 'yyyy.mm.dd'), 3625)",
            "insert into palyak values(22,'United Arab Emirates Yas Marina Circuit, Abu Dhabi','Abu-Dzabi', to_date('2022.11.20', 'yyyy.mm.dd'), 3958)",
        };

        for(int i = 0; i < sqlp.length; i++) {
            try {
                s=conn.createStatement();
                s.executeUpdate(sqlp[i]);
                System.out.println("Pálya felvéve\n");
                s.close();
            } catch (Exception ex) {
                System.err.println(ex.getMessage());
            }
        }
    }
}

public static void StatikusAdatfelvetel_3() {
    if(conn != null) {
        String sqlp[] = {
            "insert into csapat values('Mercedes',251,124,265,8,to_date('1954.07.04', 'yyyy.mm.dd'))",
            "insert into csapat values('Red Bull',328,76,207,4,to_date('2005.03.06', 'yyyy.mm.dd'))",
            "insert into csapat values('Ferrari',1034,238,777,16,to_date('1950.05.21', 'yyyy.mm.dd'))",
            "insert into csapat values('McLaren',908,183,493,8,to_date('1966.05.22', 'yyyy.mm.dd'))",
            "insert into csapat values('Alpine',24,1,2,0,to_date('2021.03.28', 'yyyy.mm.dd'))",
            "insert into csapat values('AlphaTauri',41,1,2,0,to_date('2020.07.05', 'yyyy.mm.dd'))",
            "insert into csapat values('Aston Martin',30,0,1,0,to_date('1959.05.31', 'yyyy.mm.dd'))",
            "insert into csapat values('Williams',772,114,313,9,to_date('1977.05.08', 'yyyy.mm.dd'))",
            "insert into csapat values('Alfa Romeo',172,10,26,0,to_date('1950.05.13', 'yyyy.mm.dd'))",
            "insert into csapat values('Haas',124,0,0,0,to_date('2016.03.20', 'yyyy.mm.dd'))",
        };

        for(int i = 0; i < sqlp.length; i++) {
            try {
                s=conn.createStatement();
                s.executeUpdate(sqlp[i]);
                System.out.println("csapat felvéve\n");
                s.close();
            } catch (Exception ex) {
                System.err.println(ex.getMessage());
            }
        }
    }
}
}

```



```

public static void StatikusAdatfelvetel_2() {
    if(conn != null) {

        String sqlp[] = {
            "insert into versenyzok values(44,'Lewis Hamilton',290,103,to_date('2007.03.18', 'yyyy.mm.dd'),'Mercedes')",
            "insert into versenyzok values(63,' George Russell',62,0,to_date('2019.03.17', 'yyyy.mm.dd'),'Mercedes')",
            "insert into versenyzok values(1,'Max Verstappen',143,20,to_date('2015.03.15', 'yyyy.mm.dd'),'Red Bull')",
            "insert into versenyzok values(11,'Sergio Pérez',219,2,to_date('2011.03.27', 'yyyy.mm.dd'),'Red Bull')",
            "insert into versenyzok values(16,'Charles Leclerc',83,3,to_date('2018.03.25', 'yyyy.mm.dd'),'Ferrari')",
            "insert into versenyzok values(55,'Carlos Sainz Jr.',143,0,to_date('2015.03.15', 'yyyy.mm.dd'),'Ferrari')",
            "insert into versenyzok values(3,'Daniel Ricciardo',212,8,to_date('2011.07.11', 'yyyy.mm.dd'),'McLaren')",
            "insert into versenyzok values(4,'Lando Norris',62,0,to_date('2019.03.17', 'yyyy.mm.dd'),'McLaren')",
            "insert into versenyzok values(14,'Fernando Alonso',338,32,to_date('2001.03.04', 'yyyy.mm.dd'),'Alpine')",
            "insert into versenyzok values(31,'Esteban Ocon',91,1,to_date('2016.08.28', 'yyyy.mm.dd'),'Alpine')",
            "insert into versenyzok values(10,'Pierre Gasly',88,1,to_date('2017.10.01', 'yyyy.mm.dd'),'AlphaTauri')",
            "insert into versenyzok values(22,'Cunoda Júki',24,0,to_date('2021.03.28', 'yyyy.mm.dd'),'AlphaTauri')",
            "insert into versenyzok values(5,'Sebastian Vettel',280,53,to_date('2007.06.17', 'yyyy.mm.dd'),'Aston Martin')",
            "insert into versenyzok values(18,'Lance Stroll',103,0,to_date('2017.03.27', 'yyyy.mm.dd'),'Aston Martin')",
            "insert into versenyzok values(6,'Nicholas Latifi',41,0,to_date('2020.07.05', 'yyyy.mm.dd'),'Williams')",
            "insert into versenyzok values(23,'Alexander Albon',40,0,to_date('2019.03.19', 'yyyy.mm.dd'),'Williams')",
            "insert into versenyzok values(24,'Csou Kuan-jü',2,0,to_date('2022.03.20', 'yyyy.mm.dd'),'Alfa Romeo')",
            "insert into versenyzok values(77,'Valtteri Bottas',181,10,to_date('2013.03.17', 'yyyy.mm.dd'),'Alfa Romeo')",
            "insert into versenyzok values(47,'Mick Schumacher',24,0,to_date('2021.03.28', 'yyyy.mm.dd'),'Haas')",
            "insert into versenyzok values(20,'Kevin Magnussen',122,0,to_date('2014.03.16', 'yyyy.mm.dd'),'Haas')",

        };

        for(int i = 0; i < sqlp.length; i++) {
            try {
                s=conn.createStatement();
                s.executeUpdate(sqlp[i]);
                System.out.println("Versenyző felvéve\n");
                s.close();
            } catch (Exception ex) {
                System.err.println(ex.getMessage());
            }
        }

        System.out.println("\nMinden adat bekerült az adatbázisba!\n\n");
        Menu();
    }
}

public static void StatikusTablaModosiTas() {
    if(conn != null) {}
    try {
        String sqlp="alter table versenyzok add(csapatnev references csapat)";
        s=conn.createStatement();
        s.executeUpdate(sqlp);
        System.out.println("Versenyzők táblához csapatnév hozzáadva!\n");
        s.close();
    } catch (Exception ex) {
        System.err.println(ex.getMessage());
    }
}
}

```

Főmenü!

Válaszd ki az alábbi lehetőségeket

Default - Feltölti az eredeti táblákat, adatokat és kötéseket

DBTorles - Törli az adatbázist

AdatFelvisz - Adatfelvitel a megadott táblába

Adattorles - Törli a pilótát az adatbázisból

Lekerdez - Dinamikus lekérdezés a pilótákról

SorLekerdez - Dinamikus lekérdezés az adott tábla egy adott soráról

Ellenorzes - Megadja egy tábla össze adatát

Kurzor - Módosítja egy csapat részvételének mennyiségét

Lekapcs - Kilép a programból

Default

{csapat tábla létrehozva

Versenyzők tábla létrehozva

Pályák tábla létrehozva

Versenyzők táblához csapatnév hozzáadva!

Pálya felvéve

Pálya felvéve



```
JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 1:04:01 PM)

Versenyző felvéve
Versenyző felvéve
Versenyző felvéve
Versenyző felvéve
Versenyző felvéve
Versenyző felvéve
Versenyző felvéve

Minden adat bekerült az adatbázisba!
```

**DBTorles:** Törli az összes táblát az adatbázisból

```
public static void StatikusTablaTorles() {
    if(conn != null) {}
    try {

        String sqlp_palyak="DROP TABLE palyak";
        String sqlp_csapat="DROP TABLE csapat";
        String sqlp_versenyzok="DROP TABLE versenyzok";

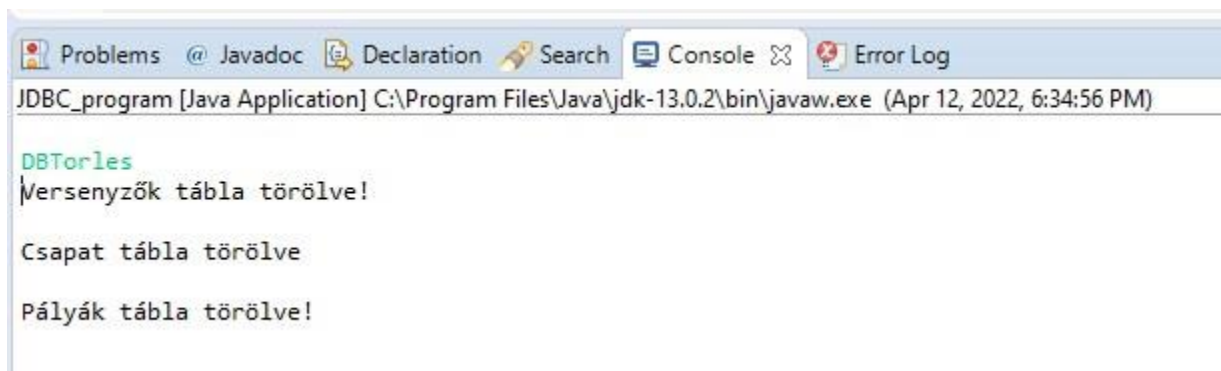
        s=conn.createStatement();

        s.executeUpdate(sqlp_versenyzok);
        System.out.println("Versenyzők tábla törölve!\n");

        s.executeUpdate(sqlp_csapat);
        System.out.println("Csapat tábla törölve\n");

        s.executeUpdate(sqlp_palyak);
        System.out.println("Pályák tábla törölve!\n");

        s.close();
        Menu();
    }catch(Exception ex) {
        System.err.println(ex.getMessage());
    }
}
```



**Adatfelvisz:** Az általunk megadott táblába vihetünk fel rekordokat.

```
public static void DinamikusAdatfelvetel() {
    if(conn != null) {
        System.out.println("Add meg a táblát, ahova szeretnél új adatokat berakni:");
        String tabla = sc.next().trim();

        if(tabla.equals("palyak")) {

            System.out.println("Kérem a pálya azonosítóját: ");
            int id= sc.nextInt();
            System.out.println("Mi a pálya neve?: ");
            String helyszin = sc.next().trim();
            System.out.println("Melyik országban található a pálya?: ");
            String orszag = sc.next().trim();
            System.out.println("Melyik napon lesz a futam?: ");
            String futamnapja = sc.next().trim();
            System.out.println("Milyen hosszú a pálya?: ");
            int palyahossz = sc.nextInt();

            String sqlp="insert into palyak values('"+id+"','"+helyszin+"','"+orszag+"', to_date('"+futamnapja+"', 'yyyy.mm.dd'), '"+palyahossz+"')";
            System.out.println(sqlp);
            try {

                s=conn.createStatement();
                s.executeUpdate(sqlp);

                System.out.println("Pálya felvéve\n");
            }catch(Exception ex) {
                System.err.println(ex.getMessage());
            }
        }

        if(tabla.equals("versenyzok")) {

            System.out.println("Kérem a rajtszámát: ");
            int rajtszam= sc.nextInt();
            System.out.println("Hogy hívják a pilótát?: ");
            String nev = sc.next().trim();
            System.out.println("Mennyi futamot teljesített már (nem lehet nulla)?: ");
            int futamok = sc.nextInt();
            System.out.println("Hány futamot nyert meg?: ");
            int gyozelmek = sc.nextInt();
            System.out.println("Mikor debütált?: ");
            String debutalas = sc.next().trim();
            System.out.println("Melyik csapatban versenyzik?: ");
            String csapatnev = sc.next().trim();

            String sqlp="insert into versenyzok values('"+rajtszam+"', '"+nev+"', '"+futamok+"', '"+gyozelmek+"', to_date('"+debutalas+"', 'yyyy.mm.dd'), '"+csapatnev+"')";

            System.out.println(sqlp);
            try {

                s=conn.createStatement();
                s.executeUpdate(sqlp);

                System.out.println("Versenyző felvéve\n");
            }catch(Exception ex) {
                System.err.println(ex.getMessage());
            }
        }
    }
}
```



```

        if(tabla.equals("csapat")) {

            System.out.println("Kérem a csapat nevét: ");
            String csapatnev = sc.next().trim();
            System.out.println("Hány futamon vettek részt?: ");
            int futamokszama = sc.nextInt();
            System.out.println("Hány futamot nyertek meg?: ");
            int gyozelmek = sc.nextInt();
            System.out.println("Hányszor álltak a csapat versenyzői a pódiumon?: ");
            int podiumok = sc.nextInt();
            System.out.println("Hányszor nyert a csapat világ bajnoki címet?: ");
            int bajnoksagok = sc.nextInt();
            System.out.println("Mikor debütált a csapat?: ");
            String debutalas = sc.next().trim();

            String sqlp="insert into csapat values('"+csapatnev+"', '"+futamokszama+"', '"+gyozelmek+"', '"+podiumok+"', '"+bajnoksagok+"', to_date('"+debutalas+"', 'yyyy.mm.dd'))";
            System.out.println(sqlp);
            try {

                s=conn.createStatement();
                s.executeUpdate(sqlp);

                System.out.println("Csapat felvéve\n");
            }catch(Exception ex) {
                System.err.println(ex.getMessage());
            }
        }
    }
}

```

```

JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 1:04:01 PM)

AdatFelvisz
Add meg a táblát, ahova szeretnél új adatokat berakni:
versenyzok
Kérem a rajtszámát:
45
Hogy hívják a pilótát?:
Jack
Mennyi futamot teljesített már (nem lehet nulla) ?:
100
Hány futamot nyert meg?:
4
Mikor debütált?:
2012.02.28
Melyik csapatban versenyzik?:
Mercedes
insert into versenyok values(45,' Jack', 100, 4, to_date('2012.02.28', 'yyyy.mm.dd'), 'Mercedes')
Versenyző felvéve

```

**Adattörles:** Törli a megadott táblából a rekordokat. Minden tábla primary key rekordját kell megadni a teljes sor törléséhez.

```

JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 1:04:01 PM)

Lekapcs - Kilép a programból

Adattörles
Melyik táblából töröljük ki a sorokat? (versenyzok, csapat vagy palyak)
versenyzok
Melyik pilótát töröljük az adatbázisból?:
45
45 rajtszámú pilóta törölve

```

```

public static void DinamikusAdattorles() {
    System.out.println("Melyik táblából töröljük ki a sorokat? (versenyzok, csapat vagy palyak)");
    String torles = sc.next().trim();

    if(torles.equals("versenyzok")) {
        System.out.println("Melyik pilótát töröljük az adatbázisból?: ");
        String rajtszam = sc.next();
        String user = null;
        String sqlp = "delete from versenyzok where rajtszam=?";
        if (conn != null) {
            try {
                ps = conn.prepareStatement(sqlp);
                ps.setString(1, rajtszam);
                ps.executeUpdate();
                ps.close();
                System.out.println(rajtszam + " rajtszámú pilóta törölve\n");
            } catch (Exception ex) {
                System.err.println(ex.getMessage());
                DinamikusAdattorles();
            }
        }
    } Menu();

    if(torles.equals("palyak")) {
        System.out.println("Melyik pályát töröljük az adatbázisból?: (Alap esetben az azonosító száma: 1-22)");
        String id = sc.next();
        String user = null;
        String sqlp = "delete from palyak where id=?";
        if (conn != null) {
            try {
                ps = conn.prepareStatement(sqlp);
                ps.setString(1, id);
                ps.executeUpdate();
                ps.close();
                System.out.println(id + " azonosítójú pálya törölve\n");
            } catch (Exception ex) {
                System.err.println(ex.getMessage());
                DinamikusAdattorles();
            }
        }
    } Menu();

    if(torles.equals("csapat")) {
        System.out.println("Melyik csapatot töröljük az adatbázisból? (A csapat kitörlése a versenyzők táblára is hatni fog) : ");
        String csapatnev = sc.next();
        String user = null;
        String sqlp2 = "delete from csapat where csapatnev='"+csapatnev+"'";
        String sqlp1 = "delete from versenyzok where csapatnev='"+csapatnev+"'";
        if (conn != null) {
            try {
                /* ps = conn.prepareStatement(sqlp);
                ps.setString(1, csapatnev);
                ps.executeUpdate(); */

                s=conn.createStatement();
                s.executeUpdate(sqlp2);
                s.executeUpdate(sqlp1);
                ps.close();
                System.out.println(csapatnev + " nevű csapat törölve\n");
            } catch (Exception ex) {
                System.err.println(ex.getMessage());
                DinamikusAdattorles();
            }
        }
    } Menu();
}

```

**Lekerdez:** Az általunk kiválasztott táblából kiválaszthatunk egy mezőt és megjelenik minden adata.

```
Problems @ Javadoc Declaration Search Console Error Log
JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 6:51:10 PM)

Lekerdez
Adja meg melyik táblából szeretne lekérdezni: (versenyzok, csapat vagy palyak)
csapat
Adja meg melyik mezőt szeretné lekérdezni: (csapatnev, futamokszama, gyozelmek, podiumok, bajnoksagok vagy debutalas
csapatnev
select csapatnev from csapat
Adatok: Alfa Romeo
Adatok: AlphaTauri
Adatok: Alpine
Adatok: Aston Martin
Adatok: Ferrari
Adatok: Haas
Adatok: McLaren
Adatok: Mercedes
Adatok: Red Bull
Adatok: Williams

public static void DinamikusLekerdez() {
    System.out.println("Adja meg melyik táblából szeretne lekérdezni: (versenyzok, csapat vagy palyak) ");
    String tabla = sc.next().trim();

    if(tabla.equals("versenyzok")) {
        System.out.println("Adja meg melyik mezőt szeretné lekérdezni: (rajtszam, nev, futamok, gyozelmek, debutalas vagy csapatnev");
        String mezo = sc.next().trim();

        String sqlp = "select "+mezo+" from versenyok";
        if(conn != null) {
            try {
                s=conn.createStatement();
                s.executeQuery(sqlp);
                rs=s.getResultSet();
                while(rs.next()) {

                    if(mezo.equals("rajtszam")) {String nev = rs.getString("rajtszam");
                        System.out.println("Adatok: "+nev);}
                    if(mezo.equals("nev")) {String nev = rs.getString("nev");
                        System.out.println("Adatok: "+nev);}
                    if(mezo.equals("futamok")) {String nev = rs.getString("futamok");
                        System.out.println("Adatok: "+nev);}
                    if(mezo.equals("gyozelmek")) {String nev = rs.getString("gyozelmek");
                        System.out.println("Adatok: "+nev);}
                    if(mezo.equals("debutalas")) {String nev = rs.getString("debutalas");
                        System.out.println("Adatok: "+nev);}
                    if(mezo.equals("csapatnev")) {String nev = rs.getString("csapatnev");
                        System.out.println("Adatok: "+nev);}

                }
                rs.close();
            }catch(Exception ex) {
                System.out.println(ex.getMessage());
            }
        }
    }Menu();}
```

```

if(tabla.equals("palyak")) {
    System.out.println("Adja meg melyik mezőt szeretné lekérdezni: (id, helyszin, orszag, futamnapja, palyahossz);");
    String mezo = sc.next().trim();

    String sqlp = "select "+mezo+" from palyak";
    if(conn != null) {
        try {
            s=conn.createStatement();
            s.executeQuery(sqlp);
            rs=s.getResultSet();
            while(rs.next()) {
                if(mezo.equals("id")) {String nev = rs.getString("id");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("helyszin")) {String nev = rs.getString("helyszin");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("orszag")) {String nev = rs.getString("orszag");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("futamnapja")) {String nev = rs.getString("futamnapja");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("palyahossz")) {String nev = rs.getString("palyahossz");
                System.out.println("Adatok: "+nev);}
            }
            rs.close();
        }catch(Exception ex) {
            System.out.println(ex.getMessage());
        }
    }Menu();}

if(tabla.equals("csapat")) {
    System.out.println("Adja meg melyik mezőt szeretné lekérdezni: (csapatnev, futamokszama, gyozelmek, podiumok, bajnoksagok vagy debutalas");
    String mezo = sc.next().trim();

    String sqlp = "select "+mezo+" from csapat";
    System.out.println(sqlp);
    if(conn != null) {
        try {
            s=conn.createStatement();
            s.executeQuery(sqlp);
            rs=s.getResultSet();
            while(rs.next()) {
                if(mezo.equals("csapatnev")) {String nev = rs.getString("csapatnev");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("futamokszama")) {String nev = rs.getString("futamokszama");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("gyozelmek")) {String nev = rs.getString("gyozelmek");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("podiumok")) {String nev = rs.getString("podiumok");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("bajnoksagok")) {String nev = rs.getString("bajnoksagok");
                System.out.println("Adatok: "+nev);}
                if(mezo.equals("debutalas")) {String nev = rs.getString("debutalas");
                System.out.println("Adatok: "+nev);}
            }
            rs.close();
        }catch(Exception ex) {
            System.out.println(ex.getMessage());
        }
    }Menu();}
}

```



SorLekerdez: Az általunk kiválasztott táblából megadunk egy sor és annak minden elemét kiírja  
(A kívánt adat primary key-t kell megadni)

```
Problems @ Javadoc Declaration Search Console Error Log
JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 6:51:10 PM)

SorLekerdez
Add meg a táblát, amelynek szeretnéd az adatait látni? (versenyzok, csapat vagy palyak):
versenyzok
Add meg a pilóta azonsítóját
16
Rajtszam      Nev      Csapat      Futamok      Gyozelem      Debutalas
-----
16      Charles Leclerc      Ferrari      83      3      2018-03-25 00:00:00.0

public static void DinamikusAdatLekerdez2() {
    System.out.println("Add meg a táblát, amelynek szeretnéd az adatait látni? (versenyzok, csapat vagy palyak):");
    String tabla = sc.next().trim();

    if(tabla.equals("versenyzok")) {
        System.out.println("Add meg a pilóta azonsítóját");
        String adat = sc.next().trim();
        if(conn != null) {
            String sqlp="select * from versenyok where rajtszam="+adat;
            System.out.println("Rajtszam      Nev      Csapat      Futamok      Gyozelem      Debutalas");
            System.out.println("-----");
            try {
                s=conn.createStatement();
                s.executeQuery(sqlp);
                rs=s.getResultSet();
                while(rs.next()) {
                    int rajtszam = rs.getInt("rajtszam");
                    String nev=rs.getString("nev");
                    String csapatnev = rs.getString("csapatnev");
                    int futamok = rs.getInt("futamok");
                    int gyozelmek = rs.getInt("gyozelmek");
                    String debutalas = rs.getString("debutalas");

                    System.out.println(rajtszam+"      "+nev+"      "+csapatnev+"      "+futamok+"      "+gyozelmek+"      "+debutalas);
                }
                rs.close();
            } catch (Exception ex) {
                System.out.println(ex.getMessage());
                DinamikusAdatLekerdez2();
            }
            Menu();
        }
    } else if(tabla.equals("palyak")) {
        System.out.println("Add meg a pálya azonsítóját");
        String adat = sc.next().trim();
        if(conn != null) {
            String sqlp="select * from palyak where id="+adat;
            System.out.println("ID, Helyszin,      Ország,      Futamnapja      Palyahossz");
            System.out.println("-----");
            try {
                s=conn.createStatement();
                s.executeQuery(sqlp);
                rs=s.getResultSet();
                while(rs.next()) {
                    int id=rs.getInt("id");
                    String helyszin = rs.getString("helyszin");
                    String orszag = rs.getString("orszag");
                    String futamnapja = rs.getString("futamnapja");
                    int palyahossz=rs.getInt("palyahossz");
                    System.out.println(id+"      "+helyszin+"      "+orszag+"      "+futamnapja+"      "+palyahossz);
                }
                rs.close();
            } catch (Exception ex) {
                System.out.println(ex.getMessage());
                DinamikusAdatLekerdez2();
            }
            Menu();
        }
    }
}
```



```

    }else if(tabla.equals("csapat")){
        System.out.println("Add meg a csapat nevét");
        String adat = sc.next().trim();
        if(conn != null) {
            String sqlp="select * from csapat where csapatnev='"+adat+"'";
            System.out.println("Csapatnev, FutamokSzama, Gyozelmek, Podiumok, Bajnoki Címek, Debutálás");
            System.out.println("-----");
            try {
                s=conn.createStatement();
                s.executeQuery(sqlp);
                rs=s.getResultSet();
                while(rs.next()) {
                    String csapatNev=rs.getString("csapatNev");
                    int futamokSzama = rs.getInt("futamokszama");
                    int gyozelmek = rs.getInt("gyozelmek");
                    int podiumok = rs.getInt("podiumok");
                    int bajnoksagok = rs.getInt("bajnoksagok");
                    String debutalas = rs.getString("debutalas");

                    System.out.println(csapatNev+" "+futamokSzama+" "+gyozelmek+" "+podiumok+" "+bajnoksagok+" "+debutalas);
                }
            }catch(Exception ex) {
                System.out.println(ex.getMessage());
                DinamikusAdatLekezerdes2();
            }
        }
    }
}
}
}

```

## Ellenorzes: Lekérdezi a megadott tábla összes adatát

Problems
@ Javadoc
Declaration
Search
Console
Error Log

JDBC\_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 6:51:10 PM)

Ellenorzes  
Add meg a táblát, amelynek szeretné az adatait látni? (versenyzo, csapat vagy palyak):  
palyak

ID, Helyszín,	Ország,	Futamnapja	Palyahossz
1 Bahrain International Circuit, Sakhir	Bahrein	2022-03-20 00:00:00.0	5412
2 Saudi Arabia Jeddah Corniche Circuit, Jeddah	Szaúd-arábia	2022-03-27 00:00:00.0	3257
3 Australia Albert Park Circuit, Melbourne	Ausztrália	2022-04-10 00:00:00.0	4120
4 Italy Imola Circuit, Imola	Olaszország	2022-04-24 00:00:00.0	3895
5 United States Miami International Autodrome, Miami Gardens	Egyesült Államok	2022-05-08 00:00:00.0	3147
6 Spain Circuit de Barcelona-Catalunya, Montmeló	Spanyolország	2022-05-22 00:00:00.0	3965
7 Monaco Circuit de Monaco, Monaco	Monaco	2022-05-29 00:00:00.0	2485
8 Azerbaijan Baku City Circuit, Baku	Azerbajdzsán	2022-06-12 00:00:00.0	4712
9 Canada Circuit Gilles Villeneuve, Montréal	Kanada	2022-06-19 00:00:00.0	5121
10 United Kingdom Silverstone Circuit, Silverstone	Nagy-Britannia	2022-07-03 00:00:00.0	4632
11 Austria Red Bull Ring, Spielberg	Ausztria	2022-07-10 00:00:00.0	3749
12 France Circuit Paul Ricard, Le Castellet	Franciaország	2022-07-24 00:00:00.0	4008
13 Hungary Hungaroring, Mogyoród	Magyarország	2022-07-31 00:00:00.0	4381
14 Belgium Circuit de Spa-Francorchamps, Stavelot	Belgium	2022-08-28 00:00:00.0	7004
15 Netherlands Circuit Zandvoort, Zandvoort	Hollandia	2022-09-09 00:00:00.0	3896
16 Italy Monza Circuit, Monza	Olaszország	2022-09-11 00:00:00.0	4294
17 Singapore Marina Bay Street Circuit, Singapore	Szingapúr	2022-10-02 00:00:00.0	3986
18 Japan Suzuka International Racing Course, Suzuka	Japán	2022-10-09 00:00:00.0	3745
19 United States Circuit of the Americas, Austin, Texas	Egyesült Államok	2022-10-23 00:00:00.0	4208
20 Mexico Autódromo Hermanos Rodríguez, Mexico City	Mexikó	2022-10-30 00:00:00.0	4398
21 Brazil Interlagos Circuit, São Paulo	Brazília	2022-11-13 00:00:00.0	3625
22 United Arab Emirates Yas Marina Circuit, Abu Dhabi	Abu-Dzabi	2022-11-20 00:00:00.0	3958

```

public static void StatikusLekezerdes() {
    System.out.println("Add meg a táblát, amelynek szeretné az adatait látni? (versenyzo, csapat vagy palyak):");
    String tabla = sc.next().trim();
    if(tabla.equals("versenyzo")) {
        if(conn != null) {
            String sqlp="select * from versenyzo";
            System.out.println("Rajtszam      Nev      Csapat      Futamok      Gyozelem      Debutalas");
            System.out.println("-----");
            try {
                s=conn.createStatement();
                s.executeQuery(sqlp);
                rs=s.getResultSet();
                while(rs.next()) {
                    int rajtszam = rs.getInt("rajtszam");
                    String nev=rs.getString("nev");
                    String csapatnev = rs.getString("csapatnev");
                    int futamok = rs.getInt("futamok");
                    int gyozelmek = rs.getInt("gyozelmek");
                    String debutalas = rs.getString("debutalas");

                    System.out.println(rajtszam+" "+nev+" "+csapatnev+" "+futamok+" "+gyozelmek+" "+debutalas);
                }
            }catch(Exception ex) {
                System.out.println(ex.getMessage());
            }
        }
    }
}

```

```

} else if (tabla.equals("palyak")) {
    if (conn != null) {
        String sqlp = "select * from palyak";
        System.out.println("ID, Helyszín, Ország, Futamnapja, Palyahossz");
        System.out.println("-----");
        try {
            s = conn.createStatement();
            s.executeQuery(sqlp);
            rs = s.getResultSet();
            while (rs.next()) {
                int id = rs.getInt("id");
                String helyszin = rs.getString("helyszin");
                String orszag = rs.getString("orszag");
                String futamnapja = rs.getString("futamnapja");
                int palyahossz = rs.getInt("palyahossz");
                System.out.println(id + " " + helyszin + " " + orszag + " " + futamnapja + " " + palyahossz);
            }
            rs.close();
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
        Menu();
    }
}

} else if (tabla.equals("csapat")) {
    if (conn != null) {
        String sqlp = "select * from csapat";
        System.out.println("Csapatnev, FutamokSzama, Gyozelmek, Podiumok, Bajnoki Cimek, Debutalas");
        System.out.println("-----");
        try {
            s = conn.createStatement();
            s.executeQuery(sqlp);
            rs = s.getResultSet();
            while (rs.next()) {
                String csapatnev = rs.getString("csapatnev");
                int futamokszama = rs.getInt("futamokszama");
                int gyozelmek = rs.getInt("gyozelmek");
                int podiumok = rs.getInt("podiumok");
                int bajnoksagok = rs.getInt("bajnoksagok");
                String debutalas = rs.getString("debutalas");
                System.out.println(csapatnev + " " + futamokszama + " " + gyozelmek + " " + podiumok + " " + bajnoksagok + " " + debutalas);
            }
            rs.close();
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
        Menu();
    }
} else {
    System.out.println("Nem megfelelo tablalt adott meg. Kérem próbálja újra");
    StatikusLekerdzes();
}
}
}

```

**Kurzor:** Az általunk kiválasztott csapat valamelyik számértékű adatát megduplázhatjuk

```

public static void ModosithatoKurzor() {
    System.out.println("Melyik csapat adatait duplazzuk meg? ");
    String csapatnev = sc.next().trim();
    System.out.println("A " + csapatnev + " csapat melyik adatát duplazzuk? (futamokszama, gyozelmek, podiumok, bajnoksagok) ");
    String adat = sc.next().trim();
    String sqlp = "select " + adat + " from csapat where csapatnev = '" + csapatnev + "'";
    if (conn != null) {
        try {
            s = conn.createStatement(ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_UPDATABLE);
            rs = s.executeQuery(sqlp);
            while (rs.next()) {
                int regiadatok = rs.getInt(adat);
                rs.updateInt(adat, (regiadatok * 2));
                rs.updateRow();
                System.out.printf("Az futamok száma a módosítás után: %d \n", regiadatok * 2);
            }
        } catch (Exception ex) {
            System.err.println(ex.getMessage());
            ModosithatoKurzor();
        }
    }
    Menu();
}
}

```

```
Problems @ Javadoc Declaration Search Console Error Log
JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 6:51:10 PM)
Lekapcs - Kilép a programból

Kurzor
Melyik csapat adatait duplázzuk meg?
Ferrari
A Ferrari csapat melyik adatát duplázzuk? (futamokszama, gyozelmek, podiumok, bajnoksagok)
futamokszama
Az futamok száma a módosítás után: 2068
```

**Lekapcs:** Lekapcsolódunk a szerverről. Lekapcsolódáskor minden adatunk megmarad, így onnan folytathatjuk a munkát ahol abbahagytuk.

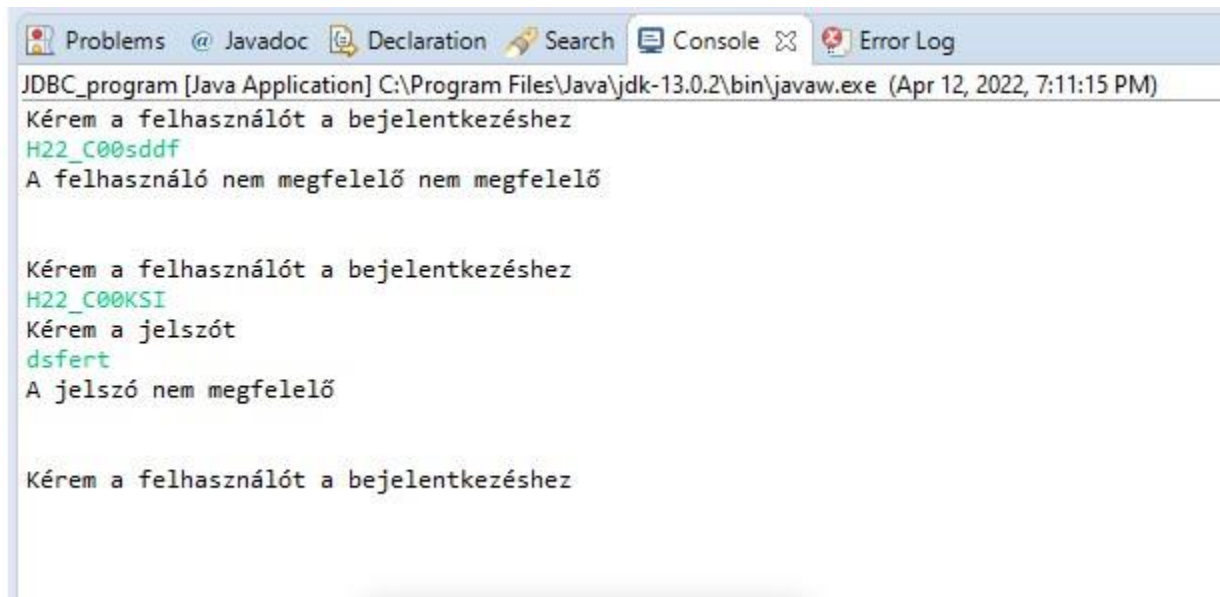
```
public static void Lekapcs() {
    if (conn != null) {
        try {
            conn.close();
            System.out.println("Sikeres lekapcsolodas \n");
        } catch (Exception e) {
            System.err.println(e.getMessage());
        }
    }
}
```

```
Problems @ Javadoc Declaration Search Console Error Log
<terminated> JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 6:51:10 PM - 7:09:39 PM)
Válasz ki az alábbi lehetőségeket

Default - Feltölti az eredeti táblákat, adatokat és kötéseket
DBTorles - Törli az adatbázist
AdatFelvisz - Adatfelvitel a megadott táblába
Adattorles - Törli a rekordot az adatbázisból
Lekerdez - Dinamikus lekérdezés a tábla mezőiről
SorLekerdez - Dinamikus lekérdezés az adott tábla egy adott soráról
Ellenorzes - Megadja egy tábla össze adatát
Kurzor - Módosítja egy csapat részvételének mennyiségét
Lekapcs - Kilép a programból

Lekapcs
Sikeres lekapcsolodas
```

Minden menüpont rendelkezik egy ellenőrző felülettel is, vagyis nem engedi, hogy téves vagy rossz parancsot adjunk meg a programnak



```
Problems  @ Javadoc  Declaration  Search  Console  Error Log
JDBC_program [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 12, 2022, 7:11:15 PM)
Kérem a felhasználót a bejelentkezéshez
H22_C00sddf
A felhasználó nem megfelelő nem megfelelő

Kérem a felhasználót a bejelentkezéshez
H22_C00KSI
Kérem a jelszót
dsfert
A jelszó nem megfelelő

Kérem a felhasználót a bejelentkezéshez
```