

# Adatbázis rendszerek 2 féléves feladat egyéni JDBC feladat, Király Péter, Y559X9, Dr. Bednarik László

## Szerzo tábla létrehozása:

```
dbm.Reg();
```

```
String sqlp= " CREATE TABLE Szerzo (id number primary  
key, nev char(50), szuletes date, kor number, konyvek number)";
```

```
dbm.CommandExec(sqlp);
```

## Konyv tábla létrehozása:

```
dbm.Reg();
```

```
String sqlp= " CREATE TABLE Konyv (id number primary  
key, cim char(100), megjelenes date, szerzo number, nyelv char(15), FOREIGN KEY(szerzo)  
REFERENCES Szerzo(id))";
```

```
dbm.CommandExec(sqlp);
```

## bejelentkezési modul:

Menü

=====

0. Kilépés

1. Konyv

2. Szerzo

Add meg a választott menü számát:

1

|

Konyv

=====

0. Kilépés

1. Adatok felvitele

2. Adatok lekérdezése (szűrés több mező egy tábla)

3. Adatok lekérdezése (szűrés egy mező kapcsolt táblák)

4. Adatok módosítása

5. Kijelölt adatok törlése

Add meg a választott menü számát:

## Adatok kiírása:

```
public void ReadAllDataKonyv () {  
    String cim="", megjelenes="", nyelv="", x="\t";  
    int id=0, szerzo=0;  
    String sqlp= "SELECT id, cim, megjelenes, szerzo, nyelv FROM Konyv";  
    Connection conn= Connect();  
    try {  
        Statement s = conn.createStatement();
```

```

        ResultSet rs = s.executeQuery(sqlp);
        while(rs.next()) {
            id = rs.getInt("id");
            cim = rs.getString("cim");
            megjelenes = rs.getString("megjelenes");
            szerzo = rs.getInt("szerzo");
            nyelv = rs.getString("nyelv");
            SM(id+x+cim+x+megjelenes+x+szerzo+x+nyelv);
        }
        rs.close();
    } catch (SQLException e) {
        SM("ReadAllData: "+e.getMessage());
    }
    Disconnect(conn);
}

public void ReadAllDataSzerzo() {
    String nev="", szuletes="", x="\t";
    int id=0, kor=0, konyvek=0;
    String sqlp= "SELECT id, nev, szuletes, kor, konyvek FROM Szerzo";
    Connection conn= Connect();
    try {
        Statement s = conn.createStatement();
        ResultSet rs = s.executeQuery(sqlp);
        while(rs.next()) {
            id = rs.getInt("id");
            nev = rs.getString("nev");
            szuletes = rs.getString("szuletes");
            kor = rs.getInt("kor");

```

```

        konyvek = rs.getInt("konyvek");

        SM(id+x+nev+x+szuletes+x+kor+x+konyvek);

    }

    rs.close();

} catch (SQLException e) {

    SM("ReadAllData: "+e.getMessage());

}

Disconnect(conn);

}

```

## Adatok felvitele:

```

public void InsertKonyv(String id, String cim, String megjelenes, String szerzo, String nyelv)
{

    Connection conn = Connect();

    String sqlp = "INSERT INTO Konyv Values '"+id+"', '"+cim+"',
    '"+megjelenes+"', '"+szerzo+"', '"+nyelv+"'";

    try {

        Statement s =conn.createStatement();

        s.execute(sqlp);

        SM("Insert OK");

    } catch (SQLException e) {

        SM("JDBC insert: "+e.getMessage());

    }

    Disconnect(conn);

}

```

```

public void InsertSzerzo(String id, String nev, String szuletes, String kor, String
konyvek) {

    Connection conn = Connect();

    String sqlp = "INSERT INTO Szerzo Values '"+id+"', '"+nev+"', '"+szuletes+"',
    '"+kor+"', '"+konyvek+"'";

```

```

    try {
        Statement s =conn.createStatement();

        s.execute(sqlp);

        SM("Insert OK");
    } catch (SQLException e) {

        SM("JDBC insert: "+e.getMessage());

    }

    Disconnect(conn);
}

```

Adatok felvitele:

```

adja meg az id-t:
8
adja meg a címét:
kutyus
adja meg a megjelenési dátumot(YYYY.MM.DD):
2000.09.12
adja meg az szerzo id-jét:
2
adja meg a nyelvet:
Román
Insert OK

```

1	Promises to Keep: On Life and Politics	2008.09.30	2	Magyar
2	A Don bukása	2022.02.05	1	Angol
8	kutyus	2000.09.12	2	Román

## Adatok lekérdezése (szűrés több mező szerint, egy táblára):

```

public void UniList(String table, String condition, String row1, String row2) {

    String sqlp = "Select "+row1+", "+row2+" FROM "+table+" WHERE
"+condition;

    Connection conn = Connect();

    SM(sqlp);

    try {

        Statement s = conn.createStatement();

```

```

        ResultSet rs = s.executeQuery(sqlp);

        while(rs.next()) {

            String data1 = rs.getString(row1);

            String data2 = rs.getString(row2);

            SM(data1 + " | " + data2);

        }

        rs.close();

    } catch (SQLException e) {

        SM(e.getMessage());

    }

    Disconnect(conn);

}

```

Add meg a választott menü számát:

2

Adja meg a tábla nevét:

Konyv

Adja meg az egyik mező nevét:

Cim

Adja meg a másik mező nevét:

ID

Adja meg a feltételt:

Id > 0

Select Cim, ID FROM Konyv WHERE Id > 0

Promises to Keep: On Life and Politics | 1

A Don bukása | 2

kutyus | 8

## Adatok lekérdezése (szűrés egy mező szerint, kapcsolt táblákra):

```

public void Grouped_Aggregated_Value() {

    String nev="", x="\t";

    int db=0;

```

```

Connection conn = Connect();

String sqlp = "SELECT Szerzo.nev, COUNT(Konyv.id) AS konyv_darabszam
"

        + "FROM Konyv "
        + "LEFT JOIN Szerzo ON Konyv.szerzo = Szerzo.id "
        + "GROUP BY Szerzo.nev";

try {
    Statement s = conn.createStatement();
    ResultSet rs = s.executeQuery(sqlp);
    while(rs.next()) {
        nev = rs.getString("nev");
        db = rs.getInt("konyv_darabszam");
        SM(nev+x+db);
    }
    rs.close();
} catch (SQLException e) {
    SM("JDBC UpdateData: "+e.getMessage());
}
Disconnect(conn);
}

```

Add meg a választott menü számát:

3  
Joe Biden 1

Mikuska József 1

## Adatok módosítása:

```

public void UpdateDataKonyv(String id, String nyelv) {
    Connection conn = Connect();

    String sqlp = "UPDATE Konyv SET nyelv = '"+nyelv+"' WHERE id =
    '"+id+"'";

```

```

        try {
            Statement s = conn.createStatement();

            int db = s.executeUpdate(sqlp);

            if (db==0) SM("A megadott id-jű könyv nem létezik, nem történt
változás");

            else

                SM("A megadott id-jű könyv új nyelve:"+nyelv);
        } catch (SQLException e) {
            SM("JDBC UpdateData: "+e.getMessage());
        }
        Disconnect(conn);
    }

    public void UpdateDataSzerzo(String id, String nev) {
        Connection conn = Connect();

        String sqlp = "UPDATE Szerzo SET nev = '"+nev+"' WHERE id = '"+id+"'";

        try {
            Statement s = conn.createStatement();

            int db = s.executeUpdate(sqlp);

            if (db==0) SM("A megadott id-jű szerző nem létezik, nem történt
változás");

            else

                SM("A megadott id-jű szerző új neve:"+nev);
        } catch (SQLException e) {
            SM("JDBC UpdateData: "+e.getMessage());
        }
        Disconnect(conn);
    }
}

```

Az adatok felvitelénél láthattunk, hogy a Don bukása című könyv nyelve eddig angol



volt, ezt most módosítjuk magyarra.

Add meg a választott menü számát:

4

Adja meg a könyv id-jét:

2

Adja meg a könyv új nyelvét:

Magyar

A megadott id-jű könyv új nyelve:Magyar

1	Promises to Keep: On Life and Politics	2008.09.30	2	Magyar
2	A Don bukása	2022.02.05	1	Magyar

## Adatok törlése és létrehozás kétfázisú parancsokal:

```
public void InsertWithPS(Konyv[] konyv) {  
    Connection conn = Connect();  
    int db = 0;  
    String sqlp = "Insert INTO Konyv Values(?, ?, ?, ?, ?)";  
    try {  
        PreparedStatement ps = conn.prepareStatement(sqlp);  
        for (int i = 0; i < konyv.length; i++) {  
            ps.setInt(1, konyv[i].getId());  
            ps.setString(2, konyv[i].getCim());  
            ps.setString(3, konyv[i].getMegjelenes());  
            ps.setInt(4, konyv[i].getSzerzo());  
            ps.setString(5, konyv[i].getNyelv());  
            ps.execute();  
            db++;  
        }  
        SM(db+" darab rekord beszúrva");  
    } catch (SQLException e) {SM(e.getMessage());}  
    Disconnect(conn);  
}
```

```

public void DeleteWithPS(Konyv[] konyv) {
    Connection conn = Connect();
    String sqlp = "DELETE FROM Konyv WHERE id = ?";
    try {
        PreparedStatement ps = conn.prepareStatement(sqlp);
        for (int i = 0; i<konyv.length; i++) {
            ps.setInt(1, konyv[i].getId());
            ps.execute();
        }
        SM("rekord törölve");
    } catch (SQLException e) {SM(e.getMessage());}
    Disconnect(conn);
}

```

A program osztályban:

```

dbm.Reg();

System.out.println("paraméterezett kétfázisú parancsok: ");
Konyv[] konyv = new Konyv[2];
konyv[0] = new Konyv(3, "Laptop táska", "2025.04.18", 1, "Magyar");
konyv[1] = new Konyv(4, "Forma 1 hell yeah", "2024.04.16", 2, "Német");
dbm.InsertWithPS(konyv);
dbm.ReadAllDataKonyv();
dbm.WriteAllDataKonyv();
dbm.DeleteWithPS(konyv);
dbm.ReadAllDataKonyv();
dbm.WriteAllDataKonyv();

```

ezek futási képe:

---

Sikeres driver regisztrálás

paraméterezett kétfázisú parancsok:  
2 darab rekord beszúrva

1	Promises to Keep: On Life and Politics	2008.09.30	2	Magyar
2	A Don bukása	2022.02.05	1	Magyar
3	Laptop táska	2025.04.18	1	Magyar
4	Forma 1 hell yeah	2024.04.16	2	Német

rekord törölve

1	Promises to Keep: On Life and Politics	2008.09.30	2	Magyar
2	A Don bukása	2022.02.05	1	Magyar

## A program extra funkcióként egy txt fájlban mindig frissíti a táblák adatait

```
public void WriteAllDataSzerzo() {  
    String nev = "", szuletes = "", x = "\t\t";  
    int id = 0, kor = 0, konyvek = 0;  
    String sqlp = "SELECT id, nev, szuletes, kor, konyvek FROM Szerzo";  
    Connection conn = Connect();  
    FileWriter writer = null;  
  
    try {  
        writer = new FileWriter("szerzo.txt");  
  
        Statement s = conn.createStatement();  
        ResultSet rs = s.executeQuery(sqlp);  
  
        writer.write("ID\t\tNév\t\tSzületés\t\tKor\t\tKönyvek\n\n");  
    }  
}
```

```

while (rs.next()) {

    id = rs.getInt("id");

    nev = rs.getString("nev");

    szuletes = rs.getString("szuletes");

    kor = rs.getInt("kor");

    konyvek = rs.getInt("konyvek");


    writer.write(id+x+nev+x+szuletes+x+kor+x+konyvek+"\n");

}


rs.close();
} catch (SQLException e) {

    SM("WriteAllDataSzerzo: " + e.getMessage());
} catch (IOException e) {

    SM("Hiba az írásban: " + e.getMessage());
} finally {

    try {

        if (writer != null) {

            writer.close();

        }

    } catch (IOException e) {

        SM("Hiba a bezárásban: " + e.getMessage());

    }

    Disconnect(conn);

}
}

```

```

public void WriteAllDataKonyv() {

    String cim = "", megjelenes = "", nyelv = "", x = "\t\t";

```

```

int id = 0, szerzo = 0;

String sqlp = "SELECT id, cim, megjelenes, szerzo, nyelv FROM Konyv";

Connection conn = Connect();

FileWriter writer = null;

try {

    writer = new FileWriter("konyv.txt");

    Statement s = conn.createStatement();

    ResultSet rs = s.executeQuery(sqlp);

    writer.write("ID\t\tCím\t\tMegjelenés\t\tSzerzőID\t\tNyelv\n\n");

    while (rs.next()) {

        id = rs.getInt("id");

        cim = rs.getString("cim");

        megjelenes = rs.getString("megjelenes");

        szerzo = rs.getInt("szerzo");

        nyelv = rs.getString("nyelv");

        writer.write(id+x+cim+x+megjelenes+x+szerzo+x+nyelv+"\n");

    }

    rs.close();

} catch (SQLException e) {

    SM("WriteAllDataKonyv: " + e.getMessage());

} catch (IOException e) {

    SM("Hiba az írásban: " + e.getMessage());

} finally {

```

```

    try {
        if (writer != null) {
            writer.close();
        }
    } catch (IOException e) {
        SM("Hiba a bezárásban: " + e.getMessage());
    }
    Disconnect(conn);
}
}

```

ID	Név	Születés	Kor	Könyvek
1	<u>Mikuska</u> József	2000.10.15	25	3
2	Joe <u>Biden</u>	1942.11.20	83	2

  

ID	Cím	Megjelenés	<u>SzerzőID</u>	Nyelv
1	<u>Promises to Keep: On</u> Life and <u>Politics</u>	2008.09.30	2	Magyar
2	A Don bukása	2022.02.05	1	Magyar

módosítjuk a programmal:

Add meg a választott menü számát:

4

Adja meg a könyv id-jét:

1

Adja meg a könyv új nyelvét:

Német

A megadott id-jű könyv új nyelve:Német

Ez után módosul a txt fájlban is a nyelve:

ID	Cím	Megjelenés	<u>SzerzőID</u>	Nyelv
1	<u>Promises to Keep: On</u> Life and <u>Politics</u>	2008.09.30	2	Német
2	A Don bukása	2022.02.05	1	Magyar