

# **Adatbázis rendszerek BSc**

3. Gyak.

2022.02.23.

**Készítette:**

Kerekes Krisztofer

Mérnökinformatikus

TRNA8A

Miskolc, 2022

## Bevezetés

Létrehoztam egy osztályt DatabaseConnection névvel, aminek példányosításkor kell megadni az adatbázishoz való kapcsolódáshoz szükséges adatokat. Ebben az osztályban hoztam létre különböző metódusokat és az adott DatabaseConnection példánnyal meghívva ezeket lehet lefuttatni az sql parancsokat.

**1. feladat** - Hozza létre Java programból az autó és a tulajdonos táblát, statikus SQL parancsvégrehajtással!

```
5 references
class DatabaseConnection
{
    private string server;
    private string database;
    private string userid;
    private string password;
    private MySqlConnection conn;
    private string connectionString;
    1 reference
    public DatabaseConnection(string server, string database, string userid, string password)
    {
        this.server = server;
        this.database = database;
        this.userid = userid;
        this.password = password;
        this.connectionString = $"server={this.server};database={this.database};uid={this.userid};pwd={this.password}";
    }
    7 references
    private void CreateConnection()
    {
        try
        {
            this.conn = new MySqlConnection(this.connectionString);
            this.conn.Open();
            Console.WriteLine("Sikeres kapcsolódás!");
        }
        catch (Exception e)
        {
            Console.WriteLine("Az adatbázishoz való kapcsolódás megghiúsult!");
            Console.WriteLine(e.Message);
        }
    }
    5 references
}
```

```
5 references
private void CloseConnection()
{
    try
    {
        this.conn.Close();
    }
    catch (Exception e)
    {
        Console.WriteLine("A kapcsolabontás sikertelen!");
        Console.WriteLine(e.Message);
    }
}
2 references
public void CreateTable(string sql)
{
    this.CreateConnection();
    try
    {
        MySqlCommand CreateTable = new MySqlCommand(sql, this.conn);
        CreateTable.ExecuteNonQuery();
        Console.WriteLine("Sikeres tábla létrehozás!");
    }
    catch (Exception e)
    {
        Console.WriteLine("A tábla létrehozás sikertelen!");
        Console.WriteLine(e.Message);
    }
    this.CloseConnection();
}
}
```

Metódus meghívása:

```
DatabaseConnection Connection = new DatabaseConnection("localhost", "test", "root", "");
Connection.CreateTable(@"
    CREATE TABLE auto(
        rsz char(6) PRIMARY KEY,
        tipus char(10) NOT NULL,
        szin char(10) DEFAULT 'feher',
        evjarat int(4),
        ar int(8) CHECK(ar>0)
    );
");
Connection.CreateTable(@"
    CREATE TABLE tulaj(
        id int(3) PRIMARY KEY,
        nev char(20) NOT NULL,
        cim char(20),
        szuldatum date
    );
");
```

**2. feladat** - Módosítsa az autó tábla szerkezetét úgy, hogy az autó tulajdonosa is eltárolható legyen!

```
public void AlterTable(string sql)
{
    this.CreateConnection();
    try
    {
        MySqlCommand CreateTable = new MySqlCommand(sql, this.conn);
        CreateTable.ExecuteNonQuery();
        Console.WriteLine("Sikeres tábla módosítás!");
    }
    catch (Exception e)
    {
        Console.WriteLine("A tábla módosítása sikertelen!");
        Console.WriteLine(e.Message);
    }
    this.CloseConnection();
}
```

A metódus meghívása:

```
Connection.AlterTable("ALTER TABLE auto ADD tulaj_id int(3) references tulaj.id;");
```

**3. feladat** - Vigyen fel rekordokat statikus SQL parancsvégrehajtással!

```
5 references
public void InsertInto(string sql)
{
    this.CreateConnection();
    try
    {
        MySqlCommand cmd = new MySqlCommand(sql, this.conn);
        MySqlDataReader dataReader = cmd.ExecuteReader();
        Console.WriteLine("Sikeres feltöltés!");
    }
    catch (Exception e)
    {
        Console.WriteLine("Feltöltés sikertelen!");
        Console.WriteLine(e.Message);
    }
    this.CloseConnection();
}
```

A metódus meghívása:

```
Connection.InsertInto(@"
    INSERT INTO tulaj VALUES(
        1,
        'Tóth Máté',
        'Miskolc',
        '1980.05.12'
    );
");
```

**3.1. feladat** - Hozzon létre egy osztályt az autó példányoknak, melynek adattagjai az autó tábla mezői. Írja meg az osztály konstruktorát és get metódusait, melyek az adattagok értékét adják vissza.

Adattagok:

```
class Auto
{
    private string rsz;
    private string tipus;
    private string szin;
    private int evjarat;
    private int ar;
    private int tulaj_id;

    5 references
    public string Rsz { get => rsz; set => rsz = value; }
    5 references
    public string Tipus { get => tipus; set => tipus = value; }
    5 references
    public string Szin { get => szin; set => szin = value; }
    5 references
    public int Evjarat { get => evjarat; set => evjarat = value; }
    5 references
    public int Ar { get => ar; set => ar = value; }
    5 references
    public int Tulaj_id { get => tulaj_id; set => tulaj_id = value; }
```

Konstruktorok:

```
0 references
public Auto(string rsz, string tipus, string szin, int evjarat, int ar, int tulaj_id)
{
    this.Rsz = rsz;
    this.Tipus = tipus;
    this.Szin = szin;
    this.Evjarat = evjarat;
    this.Ar = ar;
    this.Tulaj_id = tulaj_id;
}
1 reference
public Auto()
{
    Console.WriteLine("Add meg a rendszámot: ");
    this.Rsz = Console.ReadLine();
    Console.WriteLine("Add meg a típust: ");
    this.Tipus = Console.ReadLine();
    Console.WriteLine("Add meg a színt: ");
    this.Szin = Console.ReadLine();
    Console.WriteLine("Add meg az évjáratot: ");
    this.Evjarat = int.Parse(Console.ReadLine());
    Console.WriteLine("Add meg az árat: ");
    this.Ar = int.Parse(Console.ReadLine());
    Console.WriteLine("Add meg a tulajdonos azonosítóját: ");
    this.Tulaj_id = int.Parse(Console.ReadLine());
}
1 reference
```

```

public Auto(MySqlDataReader dataReader)
{
    try
    {
        this.Rsz = Convert.ToString(dataReader[0]);
        this.Tipus = Convert.ToString(dataReader[1]);
        this.Szin = Convert.ToString(dataReader[2]);
        this.Evjarat = int.Parse(Convert.ToString(dataReader[3]));
        this.Ar = int.Parse(Convert.ToString(dataReader[4]));
        this.Tulaj_id = int.Parse(Convert.ToString(dataReader[5]));
    }
    catch (Exception e)
    {
        Console.WriteLine("Sikertelen lekérdezés!");
        Console.WriteLine(e.Message);
    }
}

```

**4. feladat** - Vigyen fel autó rekordokat dinamikus SQL parancsvégrehajtással!

```

Connection.InsertInto(@"
    INSERT INTO auto VALUES(
        'aaa111',
        'opel',
        'piros',
        2014,
        1650000,
        1
    );
");
Connection.InsertInto(@"
    INSERT INTO auto VALUES(
        'bbb222',
        'mazda',
        null,
        2016,
        2800000,
        1
    );
");
Connection.InsertInto(@"
    INSERT INTO auto (rsz, tipus, evjarat, ar) VALUES(
        'ccc333',
        'ford',
        2009,
        15000000
    );
");

```

**5. feladat** - Törölje egy megadott rendszámú autót.

A következő metódus az Auto osztályban van definiálva és osztályszintű.

```
1 reference
public static void DeleteByRsz(DatabaseConnection Connection, string rsz)
{
    Connection.DeleteFrom($"DELETE FROM auto WHERE rsz='{rsz}';");
    Console.WriteLine($"A {rsz} rendszámú autó törölve.");
}
```

A DatabaseConnetion példányt át kell adni, hogy hozzáférhessen az adatbázishoz.

A metódus meghívása:

```
Auto.DeleteByRsz(Connection, "ddd444");
```

**6. feladat** - DQL utasítások végrehajtása

Select() metódus a DatabaseConnection osztályban:

```
1 reference
public MySqlDataReader Select(string sql)
{
    this.CreateConnection();
    MySqlCommand cmd = new MySqlCommand(sql, this.conn);
    this.CreateConnection();
    return cmd.ExecuteReader();
}
5 references
```

Autók adatainak kiírása:

```
List<Auto> autok = new List<Auto>();
MySqlDataReader dataReader = Connection.Select("SELECT * FROM auto;");
while (dataReader.Read())
{
    autok.Add(new Auto(dataReader));
}
foreach (var item in autok)
{
    Console.WriteLine(item.ToString());
}
```