

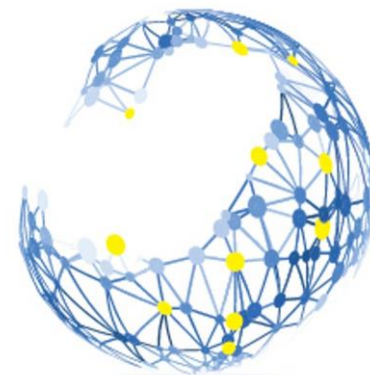
תכנות באינטרנט

ADO.Net

גלעד מרקמן

קריית החינוך פארק המדע,

נס ציונה



קריית החינוך
השש שנתית
פארק המדע
עמיד של חדשנות



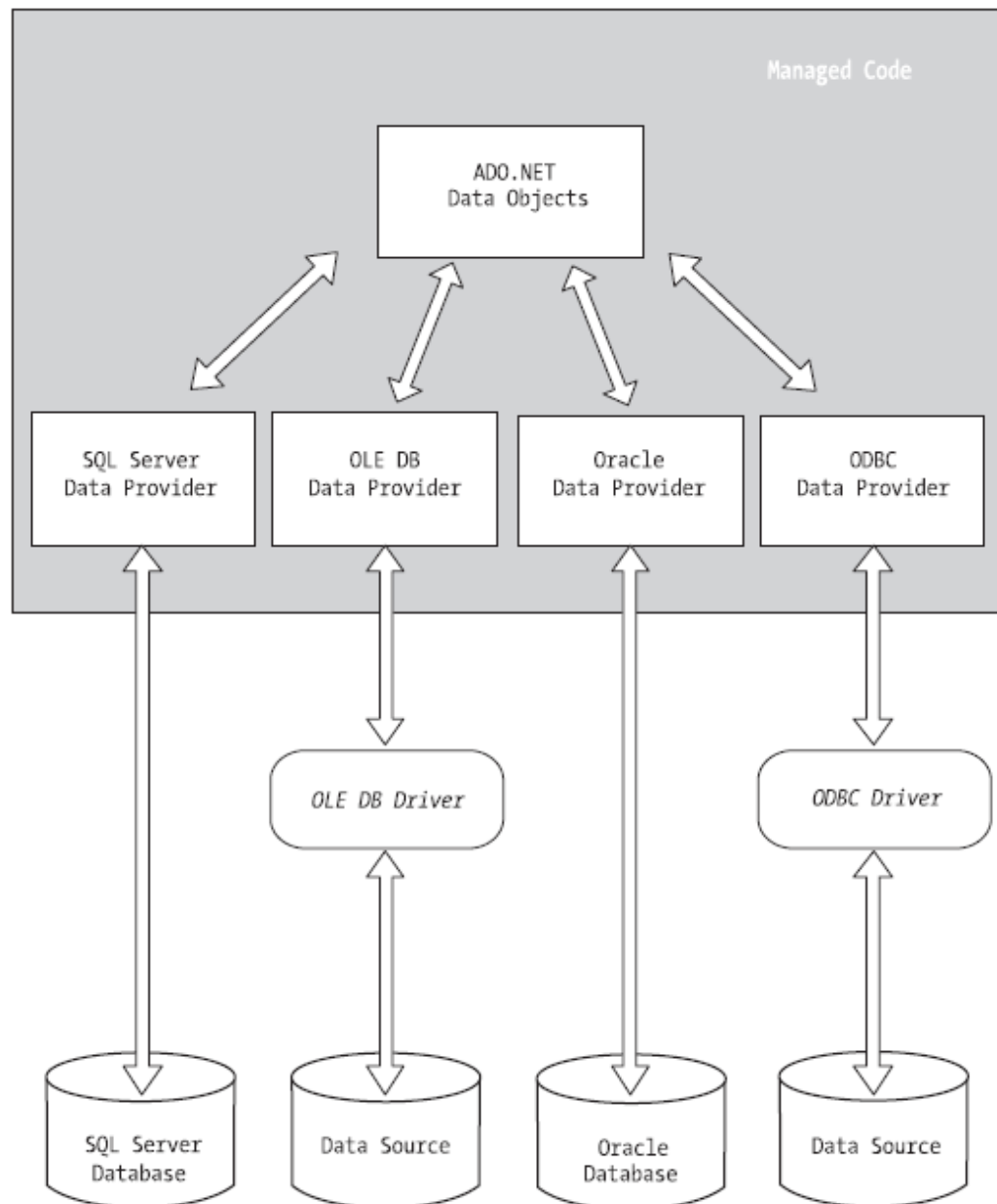
- יישום רשת טיפוסי מורכב משלושה מחשבים לפחות:
- שרת נתונים – בו נשמר בסיס הנתונים.
- שרת יישומים – בו נשמרים דפי ה ASP.Net
- מחשב הלקוח – עליו מורץ הדפדפן.



איור 3-4

מודל של יישום רשת המשתמש במסד נתונים

מודל ADO.Net



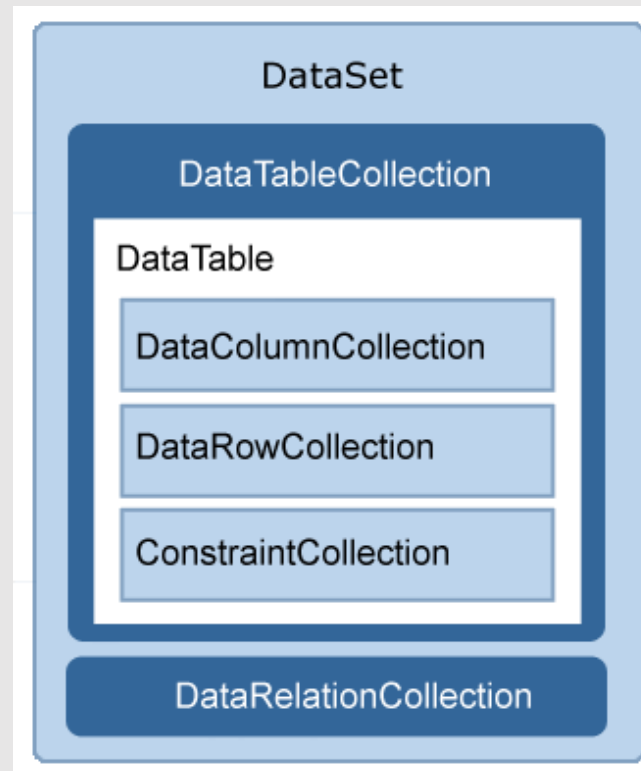
- Active Data Object - ADO.Net.
- טכנולוגיה של Microsoft שמאפשרת גישה למסד נתונים, ללא תלות בסוג מסד הנתונים עימו עובדים.
- לכל סוג מסד נתונים קיים Driver או Provider (מתווכ) ייחודי משלו:
 - *SQL Server provider;*
 - *OLE DB provider*
 - *Oracle provider*
 - *ODBC provider*
 - *Others...*

• עצמים לחיבור לבסיס הנתונים:

	SQL Server Data Provider	OLE DB Data Provider	Oracle Data Provider	ODBC Data Provider
Connection	SqlConnection	OleDbConnection	OracleConnection	OdbcConnection
Command	SqlCommand	OleDbCommand	OracleCommand	OdbcCommand
DataReader	SqlDataReader	OleDbDataReader	OracleDataReader	OdbcDataReader
DataAdapter	SqlDataAdapter	OleDbDataAdapter	OracleDataAdapter	OdbcDataAdapter

• עצמים לשמירה וניהול הנתונים ביישום

- DataSet
- DataTable
- DataRow
- DataRelation (*)



DataSet

אופן הגישה לאיבר הזה	האיבר בטבלה שאליו ניגשים
<code>ds.Tables[0].Rows[3]</code>	השורה הרביעית (שמספרה הסידורי 3) בטבלה הראשונה (שמספרה הסידורי 0)
<code>ds.Tables[2].Rows[0]</code>	השורה הראשונה (שמספרה הסידורי 0) בטבלה השלישית (שמספרה הסידורי 2)
<code>ds.Tables["tblUsers"].Rows[0]</code>	השורה הראשונה בטבלה ששמה <code>tblUsers</code>
<code>ds.Tables[0].Rows[3][4]</code>	העמודה החמישית (שמספרה הסידורי 4) בשורה הרביעית (מספר 3) שבטבלה הראשונה (מספר 0).
<code>ds.Tables[0].Rows[2]["userName"]</code>	עמודה ששמה <code>userName</code> בשורה השלישית שבטבלה הראשונה
<code>ds.Tables["tblUsers"].Rows[10] ["userAddress"].ToString()</code>	הערך הנמצא בעמודה <code>userAddress</code> בשורה 11 בטבלה ששמה <code>tblUsers</code>

- `using System.Data` – Data Objects
- `using System.Data.SqlClient` – ADO.Net Objects
- `using System.Data.OracleClient`
- `using System.Data.OleDb`
- `using System.Data.Odbc`

- **מקושרת (connected)** – היישום מתחבר ישירות למסד הנתונים ומבצע את הפעולות כשהוא מחובר למסד הנתונים.
- **לא מקושרת (disconnected)** – היישום מתחבר למסד הנתונים, מאחזר את המידע, שומר אצלו ומתנתק. עיבוד הנתונים נעשה על עותק מקומי. בסיום העיבוד היישום מתחבר שוב למסד הנתונים ומעדכן אותו.

עבודה בשיטה
הלא המקושרת
disconnected

- התחברות למסד הנתונים

```
SqlConnection con = new SqlConnection(connectionString);
```

- בניית פקודת SQL

```
SqlCommand cmd = new SqlCommand("SQL String", con);
```

- בניית DataAdapter

```
SqlDataAdapter adapter = new SqlDataAdapter(cmd);
```

- בניית DataSet לאיכסון נתונים מקומי

```
DataSet ds = new DataSet();
```

- טעינת הנתונים ממסד הנתונים ל DataSet

```
adapter.Fill(ds, "users");
```

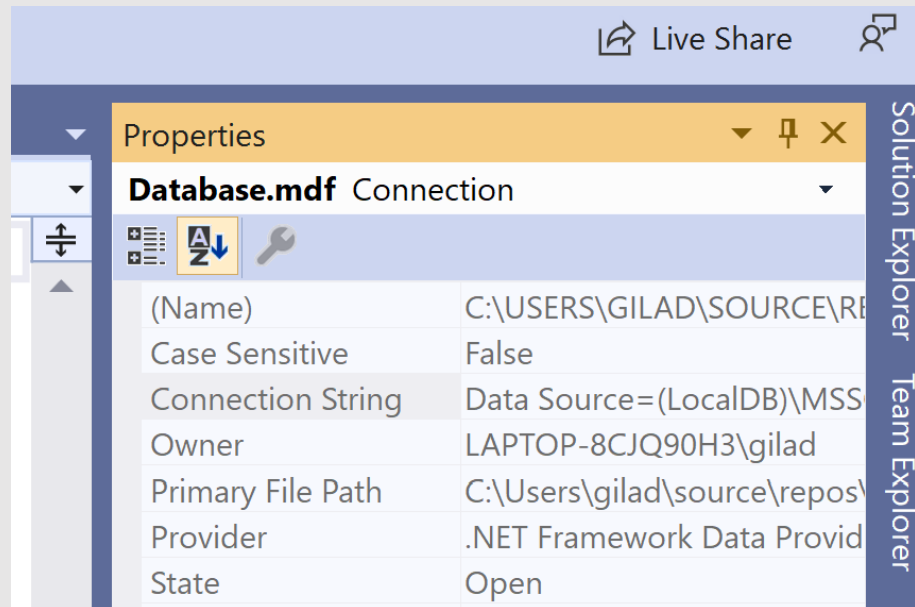
- עדכון הנתונים ב DataSet

-

- עדכון בסיס הנתונים על פי ה DataSet המעודכן

```
SqlCommandBuilder builder = new SqlCommandBuilder(adapter);  
adapter.UpdateCommand= builder.GetDeleteCommand();  
adapter.UpdateCommand = builder.GetUpdateCommand();  
adapter.UpdateCommand = builder.GetInsertCommand();  
adapter.Update(ds, "users");
```

Connection String



- מחרוזת ההתקשרות נוצרת אוטומטית בעת יצירת בסיס הנתונים.
- אנו מעתיקים אותה מהגדרות (Properties) של בסיס הנתונים.
- מומלץ להשתמש בקישור יחסי.

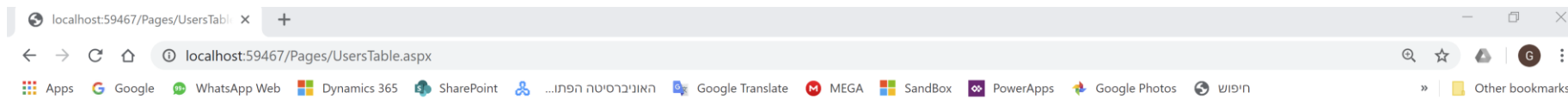
```
string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;  
AttachDbFilename=C:\Users\gilad\source\repos\DBWeb\DBWeb\App_Data\Database.mdf;  
Integrated Security=True";
```

```
string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;  
AttachDbFilename=|DataDirectory|\Database.mdf;Integrated Security=True";
```

- הפעולות adapter.Fill (ds) או adapter.Update (ds) מחייבת פתיחה של החיבור וסגירתו בסיום הפעולה.
- הפתיחה והסגירה נעשית באופן אוטמטי ואנחנו לא צריכים לתת פקודה ייחודית.
- אם בכל זאת מוסיפים פקודת פתיחה אז חובה גם להוסיף פקודת סגירה:

```
con.Open();  
adapter.Fill(ds);  
con.Close();
```

דוגמה הדפסת טבלת המשתמשים



Wellcom Gilad



Enter Text to search name:

Filter

userId	firstName	lastName	userName	password	admin	birthday	city
1	Gilad	Markman	Gilad	1968	True	25/01/1968 00:00:00	Rehovot
2	Menny	Abudi	Menny	1234	True	01/01/1980 00:00:00	Ramat Gan
3	Yoav	Markman	Yoav	abcd1234	False	03/03/2003 00:00:00	Rehovot
8	Haim	Cohen	HaCo1	Haim1	False	31/12/2020 00:00:00	Rehovot
10	Gal	Levy	GaLe	g1123	False	25/01/2020 00:00:00	Rehovot
12	Orly	Markman	OM888	a1b2	False		Rehovot
16	Lihl	Markman	Lihl	1234	True	26/03/2003 00:00:00	Rehovot
17	Haim	Ziv	Haim	hmz123	True	01/01/1956 00:00:00	Rehovot
21	Noa	Bitton	Noa	1111	False	01/01/1900 00:00:00	

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!(bool)Session["Admin"])...

    if (!IsPostBack)
    {
        string SQLStr = "SELECT * FROM tblUsers";
        DataSet ds = RetrieveUsersTable(SQLStr);
        DataTable dt = ds.Tables[0];
        string table = BuildUsersTable(dt);
        tableDiv.InnerHtml = table;
    }
}
```



```
public DataSet RetrieveUsersTable(string SQLStr)
{
    // connect to DataBase
    string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFile=
SqlConnection con = new SqlConnection(connectionString);

    // Build SQL Query
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = SQLStr;
    cmd.Connection = con;
    //SqlCommand cmd = new SqlCommand(SQLStr, con);

    // Build DataAdapter
    SqlDataAdapter ad = new SqlDataAdapter(cmd);

    // Build DataSet to store the data
    DataSet ds = new DataSet();

    // Get Data form DataBase into the DataSet
    //con.Open();
    ad.Fill(ds, "users");
    //con.Close();

    return ds;
}
```

טעינת נתונים ל DataSet

בניית הטבלה ב HTML

2 references

```
public string BuildUsersTable (DataTable dt)
{
    string str = "<table class='usersTable' align='center'>";
    str += "<tr>";
    foreach (DataColumn column in dt.Columns)
    {
        str += "<td>" + column.ColumnName + "</td>";
    }

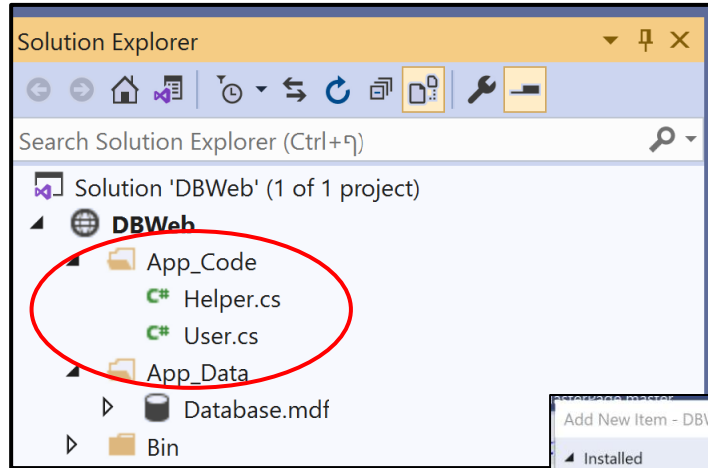
    foreach (DataRow row in dt.Rows)
    {
        str += "<tr>";
        foreach (DataColumn column in dt.Columns)
        {
            str += "<td>" + row[column] + "</td>";
        }
        str += "</tr>";
    }
    str += "</tr>";
    str += "</Table>";
    return str;
}
```

- יצירת הטבלה נעשית על ידי בניית תגית HTML של טבלה הכוללת את כל הנתונים.
- הלולאה הראשונה – בונה את הכותרת.
- הלולאה השנייה בונה את תוכן הטבלה.

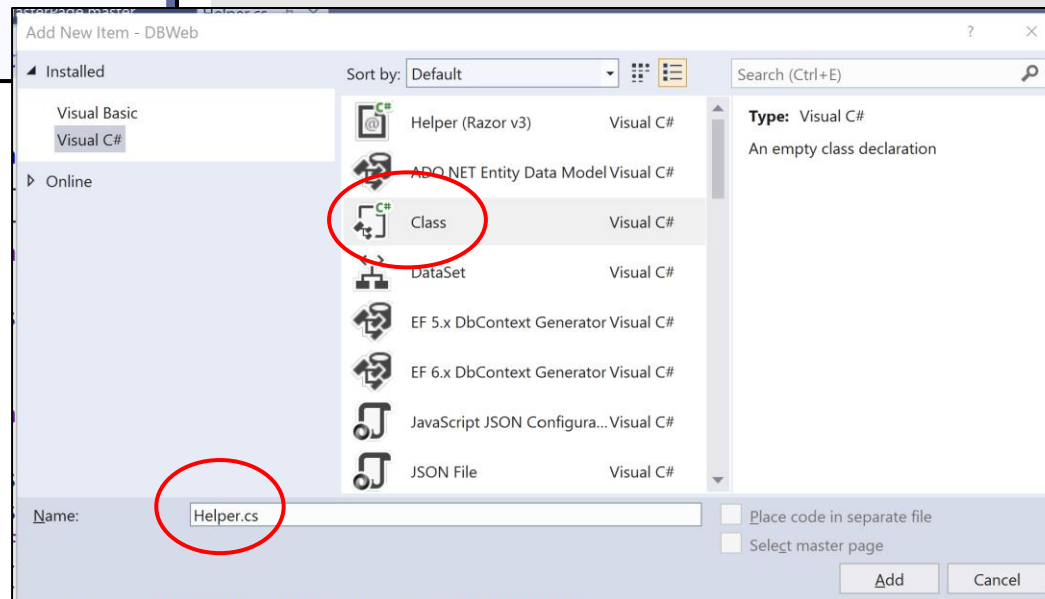
מחלקה סטטית

Helper

- על מנת לחסוך בכתיבת קוד אנו נקים מחלקה סטטית שתקרא Helper, ובה נכתוב את הפעולות שמתקשרות על בסיס הנתונים שלנו.
- אנו נשתמש בפעולות שמוגדרות במחלקה זו בכל הדפים שלנו בפרויקט.
- המחלקה הינה מחלקה Static כך שניתן להפעיל את הפעולות ללא צורך ביצירת אובייקט מיוחד לצורך כך.
- המחלקה גם תכלול קבועים המשמשים אותנו להתקשרות עם בסיס הנתונים, כמו ה- connection String.
- בנוסף נקים מחלקה נוספת שתכונה User התואמת את השדות בבסיס הנתונים שלנו, ובתוך אובייקט כזה נשמור את הנתונים שהתקבלו מבסיס הנתונים.



- את המחלקה ניצור בספריה מיוחדת שתקרא App_Code בספריה הראשית של הפרויקט.
- יצירת מחלקה נעשית באמצעות Add Item.



```
1 reference
public static class Helper1
{
    0 references
    public Helper1()
    {
        //
        // TODO: Add constructor logic here
        //
    }
}
```

- לאחר שיוצר הקובץ יש להוסיף לכתרת המחלקה את המילה השמורה static, ולמחוק את הבנאי שנוצר אוטומטית.

- בתחילת המחלקה נוסיף שמות של קבועים (const) שימשו אותנו בכל המחלקה:
 - DBName – שם בסיס המידע שיצרנו.
 - tblName – שם הטבלה של המשתמשים ביצרנו.
 - connectionString – הגדרת מחרוזת ההתקשרות עם בסיס הנתונים.
- המחרוזת הוגדרה באמצעות נתיב יחסי (|DataDirectory|), וזאת בהנחה ששמרנו את בסיס הנתונים בספרייה שנקראת App_Data בשורש הפרויקט.

5 references

```
public static class Helper
{
    public const string DBName = "Database.mdf";
    public const string tblName = "tblUsers";
    public const string connectionString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\\"
        + DBName + ";Integrated Security=True";
}
```

1 reference

```
public static DataSet RetrieveTable(string SQLStr)
{
    // connect to DataBase
    SqlConnection con = new SqlConnection(conString);

    // Build SQL Query
    SqlCommand cmd = new SqlCommand(SQLStr, con);

    // Build DataAdapter
    SqlDataAdapter ad = new SqlDataAdapter(cmd);

    // Build DataSet to store the data
    DataSet ds = new DataSet();

    // Get Data form DataBase into the DataSet
    ad.Fill(ds, tblName);

    return ds;
}
```

- חתימת הפעולה חייבת לכלול את המילה static.
- הפעולה מקבלת משתנה מסוג מחרוזת הכולל פקודת SELECT בשם SQL.
- שם הטבלה ב dataSet יהיה כשם הקבוע tblName שהגדרנו בראש המחלקה.
- הפעולה מחזירה אובייקט מסוג DataSet עם הטבלה שהתקבלה מבסיס הנתונים.

- חתימת הפעולה היא static.
- הפעולה מקבלת אובייקט מסוג DataTable שהתקבל מבסיס הנתונים באמצעות הפעולה RetrieveTable.
- הפעולה מחזירה מחרוזת HTML עם טבלת המשתנים.

0 references

```
public static string BuildSimpleUsersTable(DataTable dt)
{
    string str = "<table class='usersTable' align='center'>";
    str += "<tr>";
    foreach (DataColumn column in dt.Columns) ...

    foreach (DataRow row in dt.Rows) ...
    str += "</tr>";
    str += "</Table>";
    return str;
}
```

- כאמור, כדי להשתמש בפעולות הכתובות במחלקה סטטית אין צורך להגדיר אובייקט של המחלקה עם המילה השמורה `.new`.
- הפניה למחלקה מתבצעת עם שם המחלקה, נקודה ולאחריה שם הפעולה.
- השתמשנו הן בפעולה `RetrieveTable`, והן בפעולה `BuildSimpleUserTable`.

```
if (!IsPostBack)
{
    string SQLStr = "SELECT * FROM " + Helper.tblName;
    DataSet ds = Helper.RetrieveTable(SQLStr);
    DataTable dt = ds.Tables[Helper.tblName];
    string table = Helper.BuildSimpleUsersTable(dt);
    tableDiv.InnerHtml = table;
}
```

```
6 references
public class User
{
    public int userId;
    public string userName;
    public string password;
    public string firstName;
    public string lastName;
    public DateTime birthday;
    public string city;
    public bool Admin;
    2 references
    public User()
    {
        userId = -1;
        userName = "";
        password = "";
        firstName = "";
        lastName = "";
        birthday = DateTime.Today;
        city = "";
        Admin = false;
    }
}
```

- המחלקה User כוללת מאפיינים התואמים את טבלת המשתמשים בבסיס הנתונים שלנו.
- כמו כן, כוללת המחלקה בנאי פשוט.
- המחלקה אינה מסוג static, ולכן כדי להשתמש בה נצטרך ליצור אובייקט באמצעות המילה new.
- מחלקה זו תשמש אותנו כדי להעביר ולקבל נתונים לפעולות הסטטיות שלנו.

- מהו ADO.NET
- מודל ADO.Net ובסיסי נתונים שונים (NameSpaces).
- עצמים לשמירה וניהול של נתונים DataSet.
- עצמים לחיבור לבסיס הנתונים.
- שתי שיטות עבודה (מקושרת ולא מקושרת).
- שלבי עבודה בשיטה הלא מקושרת.
- מחרוזת ההתקשרות Connection String
- דוגמה – הדפסת טבלה.
- מחלקה סטטית Helper.