

## Entity Framework Database

אני אפרט במדריך זה איך ליצור מחלקות שייצרו אוטומטית את הטבלאות בבסיס הנתונים שלכם. ולנהל את הבסיס נתונים שלכם בעזרת תכנות מונחה עצמים

ראשית יש להתקין את החבילה של entity framework , באופן הבא, נכנסים ל:

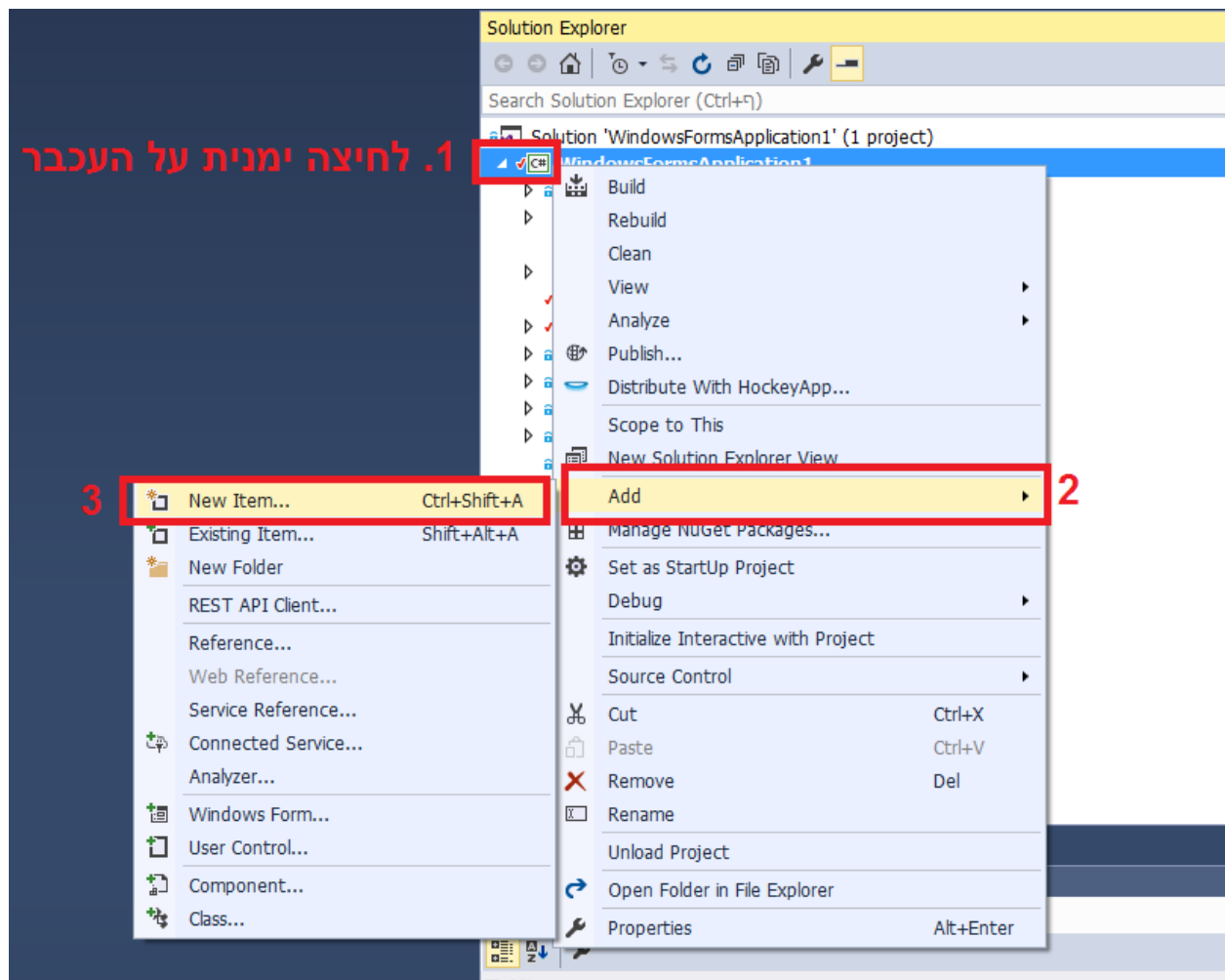
Tools>NuGet Package Manager> Package Manager Console

ואז כותבים שמה:

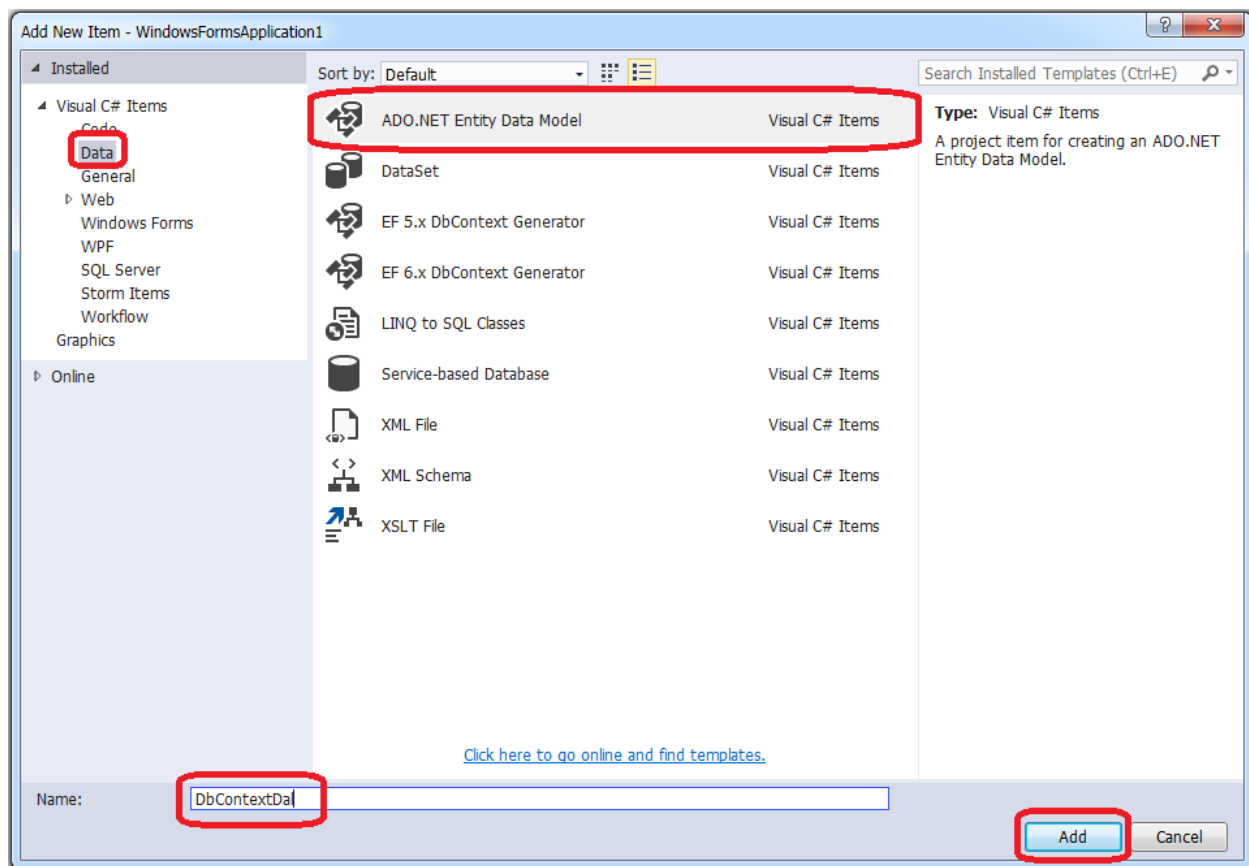
```
PM> Install-Package EntityFramework
```

עכשיו לאחר ההתקנה נוכל להתחיל בהסברים:

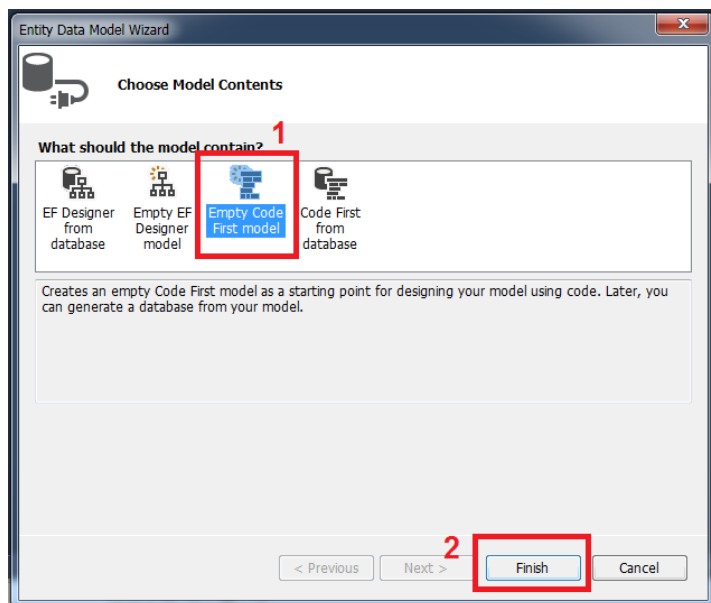
1. נתחיל מהקובץ EF שצריך ליצור, יש לבצע את השלבים הבאים:



2. תבחרו את הקובץ ADO.NET Entity Data Model, ותתנו שם לקובץ שלכם, שם לב זהו הקובץ דרכו אתה ניגש לבסיס הנתונים, נהוג לכנות אותו ב Context (יש כאלה שמכנית אותו ב-Dal בשפת ASP.NET). אני אכנה אותו בשם : DbContextDal



3. לסיום תבחר אתה האפשרות השלישית Empty.



פה נוסף את כל  
הרשימות של  
האובייקטים שנרצה  
לחפש ב-DB.

```
User.cs* - X
WindowsFormsApplication1
WindowsFormsApplication1.User
_UserID

1 namespace WindowsFormsApplication1
2 {
3     //using System.ComponentModel.DataAnnotations;
4     using System.ComponentModel.DataAnnotations.Schema;
5
6     public class User
7     {
8         private int _UserID;
9         [Key, DatabaseGenerated(DatabaseGeneratedOption.None)]
10        [Column(Order = 1)]
11        using System.ComponentModel.DataAnnotations;
12        System.ComponentModel.DataAnnotations.Key
13        Generate type
14        Change 'Key' to 'ForeignKey'.
15        [Required]
16        [RegularExpression("Secretary|Admin")]
17        public string permission { get; set; }
18
19        public User() { password = "0000"; }
20
21        public User(int id, string _permission)
22        {
23            ID = id;
24            password = "0000";
25            permission = _permission;
26        }
27    }
28 }
```

צריך לייבא את המחלקות האלה

תוספו קוד של DataAnnotation כלשהו ואז תייבאו אותו מהמנורה ליד

CS0246 The type or namespace name 'KeyAttribute' could not be found (are you missing a using directive or an assembly reference?)

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations;

Preview changes

אם הגדרתם בנאי כלשהו, תמיד תשאירו בנאי דיפולטיבי כלשהו במחלקה!

## 5.א שאלתם מה זה DataAnnotation ?

DataAnnotation זהו אמצעי לביטויים ריגוריים על מנת לייצג אילוצי קיום לעמודות שלכם בDatabase אמנה כמה דוגמאות פה: (כל משתנה זהו עמודה בDatabase שלכם!!)

[Key] - הגדרת מפתח ראשי במשתנה שמייצג עמודה בDatabase

[Required] - זהו הצהרה על זה שהמשתנה הזה לא יוכל להיות NULL, כלומר זהו שדה חובה!

יש עוד כאלה כמובן.. אך בינתיים זה מספיק בשביל המדריך הזה.

5.ב לאחר שהגדרתם את כל המשתנים והסימונים שלכם, עליכם ליצור את DbSet, שהוא בעצם המערך של האובייקטים שלכם שדרכו אתם מקבלים מידע מהבסיס נתונים או מעדכנים את הבסיס נתונים. (נראה בהמשך את השימוש, בינתיים תתנו לו את סוג המחלקה שלכם, ושם של האובייקט שלכם ברבים).

```

DbContextDal.cs  User.cs*
WindowsFormsApplication1  WindowsFormsApplication1.DbContextDal  students

1 namespace WindowsFormsApplication1
2 {
3     using System.Data.Entity;
4
5     public class DbContextDal : DbContext
6     {
7         public DbContextDal() : base("BdContextDal_ConnectionString_appConfig")
8         {
9             Configuration.LazyLoadingEnabled = false;
10        }
11
12        תוסיפו את השורה הבאה במחלקה של הcontext שלכם
13        public virtual DbSet<User> users { get; set; }
14        שימו לב, דרך המשתנה users אנחנו נגשים לטבלה שלנו בDB, ומקבלים את כל הרשומות שמה כ-List בחזרה, נראה שימוש בזה מאוחר יותר
15    }
16 }

```

## 6. יש לעדכן את ה connectionString של ה EF הזה.

יש להיכנס לקובץ app.config שבצד ימין בפרויקט (נמצא שמה באופן אוטומטי, יש לבצע שינויים ב- connectionString עצמו לפי הפרויקט שלכם.).

```

userStudent.cs*  User.cs  DbContextDal.cs*  App.config*
1 <?xml version="1.0" encoding="utf-8"?>
2 <configuration>
3   <configSections>
4     <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
5     <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0, Culture=
6   </configSections>
7   <startup>
8     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2" />
9   </startup>
10  <entityFramework>
11    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
12    <providers>
13      <provider invariantName="System.Data.SqlClient" type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
14    </providers>
15  </entityFramework>
16
17  <connectionStrings>
18    <add name="BdContextDal_ConnectionString_appConfig"
19      connectionString="Data Source=HADRIEL\SQLEXPRESS;Initial Catalog=project10;Integrated Security=True;App=EntityFramework;"
20      providerName="System.Data.SqlClient" />
21  </connectionStrings>
22
23 </configuration>
24
25

```

אתה יוצר אחד לכל קונץ DbContextDal שאתה יוצר, שזה בעצם לכל קונץ שמקושר עם DB אחד, בפרויקט שלנו, ובדרך כלל אנחנו נשתמש בבסיס נתונים אחד.

שם connection string שאנחנו נשתמש בו

את ה connection string שאתם מקבלים מהשרת שלכם, תעתיקו לפה, רק תקפידו להשאיר את APP ב"App=EntityFramework;"

7. יש להעתיק את namen מהשלב הקודם בתג של add לתוך basen בקובץ Daln

```

userStudent.cs  User.cs  DbContextDal.cs
WindowsFormsApplication1  WindowsFormsApplication1.DbContextDal
1 namespace WindowsFormsApplication1
2 {
3     using System.Data.Entity;
4
5     public class DbContextDal : DbContext
6     {
7         public DbContextDal() : base("BdContextDal_ConnectionString_appConfig")
8         {
9             Configuration.LazyLoadingEnabled = false;
10        }
11
12        public virtual DbSet<userStudent> users { get; set; }
13
14
15

```

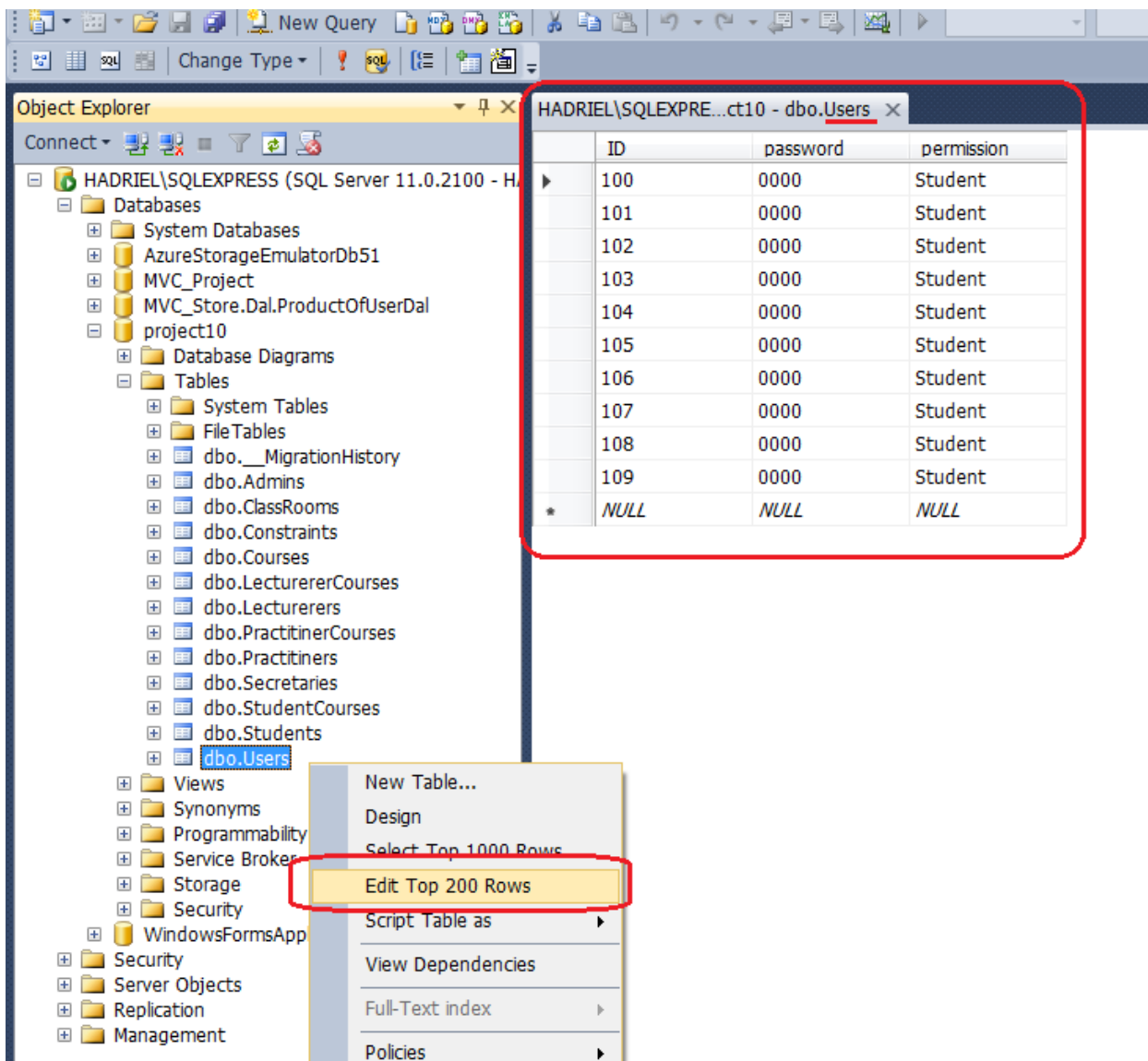
8. כעת מה שנשאר זה לפתוח אובייקט חדש מהנתונים שבנינו ולשמור בבסיס נתונים, בצורה הבאה:

```

Form_AddStudent.cs  DbContextDal.cs  User.cs*
WindowsFormsApplication1  WindowsFormsApplication1.Form_AddStudent  Add_User_For_Studer
25 private void btn_AddStudent_Click(object sender, EventArgs e)
124
125 private bool filledAll()
139 private void updateButtons()
151
152 private bool Add_User_For_Student(int id, string password)
153 {
154     try
155     {
156         // יצירת אובייקט של משתמש , בשביל להכניס אותו לבסיס נתונים שלנו
157         User newStudentUser = new User(id, password, "Student");
158
159         // יצירת קשר עם הבסיס נתונים שלנו בעזרת המשתנה
160         // ברגע שאנחנו יוצרים אובייקט כזה, כל הבסיס נתונים שלנו נוצר אוטומטי!!
161         DbContextDal dal = new DbContextDal();
162
163         // הוספת האובייקט שלנו לתוך הרשימה של משתמשים המופיע באובייקט dal
164         dal.users.Add(newStudentUser);
165
166         // לאחר שסיימנו לערוך את הבסיס נתונים שלנו , יש לבצע שמירת המצב הזה
167         dal.SaveChanges();
168
169         // במידה והגענו לפה, הפעולה שלנו הצליחה, והמשתמש מופיע כעת בבסיס הנתונים
170         MessageBox.Show("new User create for student with id: " + id + " , Added!");
171
172         return true;
173
174     } // במידה והפעולה לא הצליחה מסיבות כלשהם, אז יזרקו חריגות
175     catch (DbEntityValidationException ex)
176     {
177         //MessageBox.Show("Db Entity Validation Exception");
178         string str = "" + ex.Source + " : " + ex.GetType() + "\n-----"
179             + "Message: " + ex.Message + "\n\nExplain & solution:\nThe problem is that you
180         foreach (var entityValidationErrors in ex.EntityValidationErrors)
181         {

```

9. לאחר הפעלה של התוכנית, ניתן לראות שנוצר לנו טבלה חדשה בבסיס נתונים שלנו עם הנתונים ששמרנו.



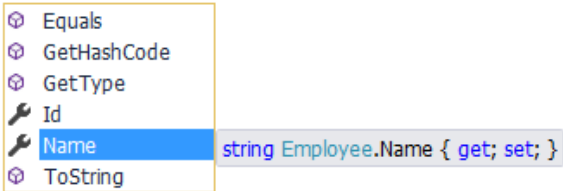
עכשיו אראה איך ניתן להשתמש בEntity Framework , כדי להשתמש בנתונים שבבסיס נתונים בעזרת תכנות Object Oriented

ניתן גם כן לקבל רשימה של כל האובייקטים כך: (ולעבור עליהם בלולאה ולחפש לפי ערכים ספציפיים)

```
EmployeeDal dal = new EmployeeDal();

List<Employee> employeeList = dal.employees.ToList<Employee>();

for (int i = 0; i < employeeList.Count(); i++)
{
    employeeList.ElementAt(i).;
}
```



10. ניתן גם לקבל אובייקט ספציפי על פי מפתח חיפוש, כך:

```
String id = textBox_userName.Text.ToString();

EmployeeDal dal = new EmployeeDal();

Employee emp = dal.employees.FirstOrDefault(x => x.Id.Equals(Convert.ToInt32(id)));
```

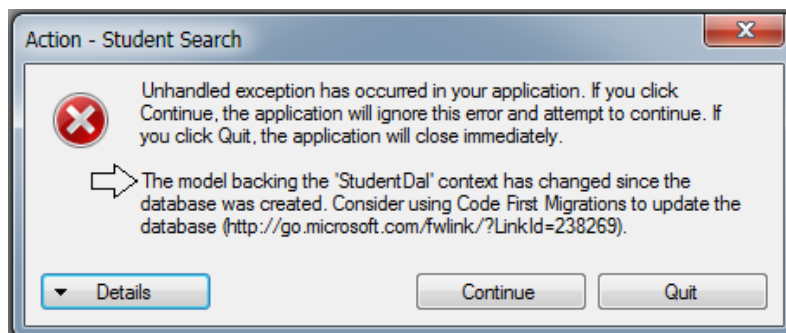
11. ניתן גם כן לקבל רשימה של האובייקטים על פי סינון של שאילתת SQL , כך:

```
EmployeeDal dal = new EmployeeDal();

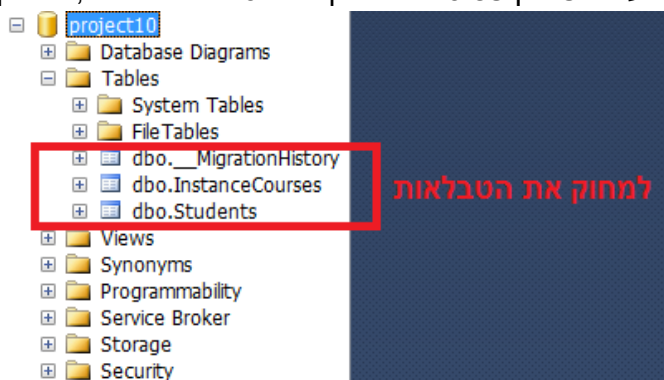
String name = textBox_userName.Text.ToString();

List<Employee> employeesList = (from x in dal.employees
                                where x.Name.Contains(name)
                                select x).ToList<Employee>();
```

12. אז עד עכשיו דיברנו על זה שכאשר אנחנו פותחים אובייקט מסוג dal , אז הוא יוצר קשר עם databasen ומגדיר שמה טבלאות בעצמו על פי המשתנים , ואם אנחנו רוצים להגדיר אילוצים , עושים זאת בעזרת data annotation אבל מה קורה אחרי זה? כשאנחנו כתבנו כבר קוד , והרצנו ונוצרו הטבלאות, אבל עכשיו אנחנו רואים שצריך לשנות את הקוד (וזה אמור לשנות את הקוד בDB , אנחנו נקבל את השגיאה הבאה):

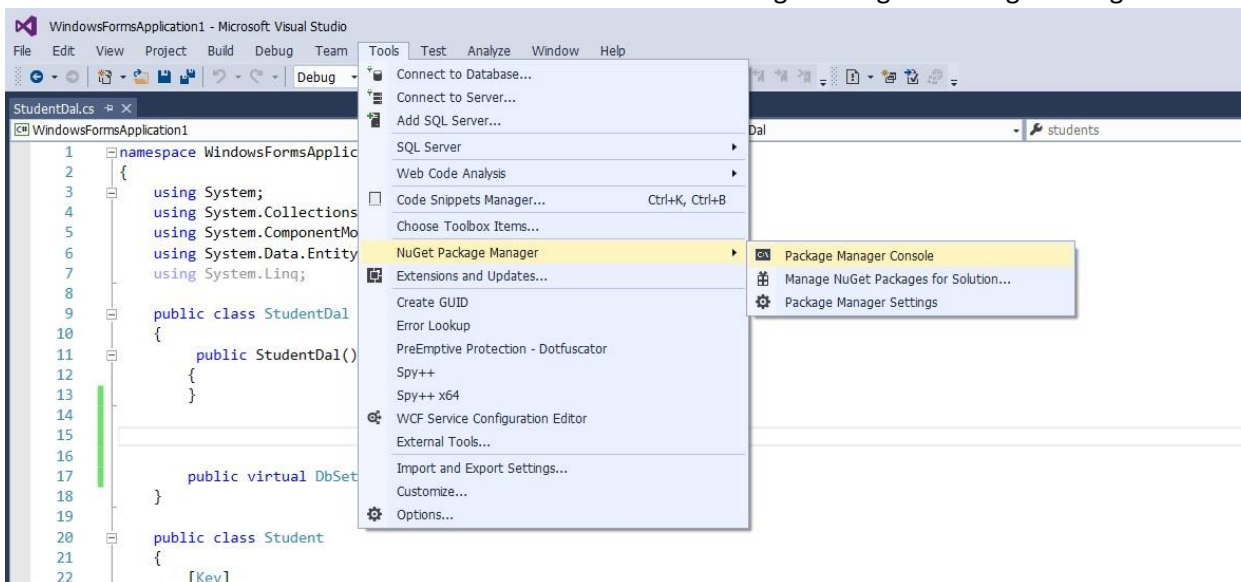


אז כדי להתמודד עם הבעיה בפתרון פשוט זה למחוק את הטבלאות מהDB , ולהריץ מחדש :



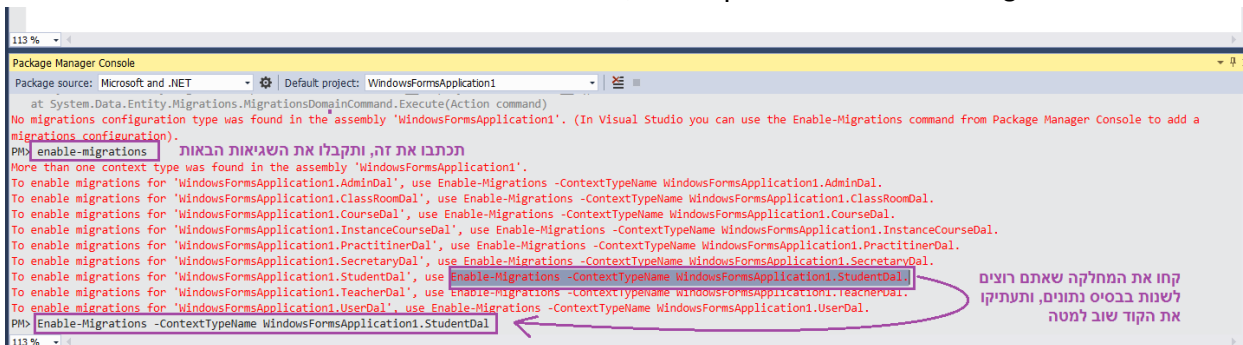
אבל זה קצת בעיה , מה כל פעם שנרצה לשנות איזה טיפוס של שדה או להוסיף שדה חדש למחלקה או למחוק אחד מהמחלקה נצטרך לעשות את כל זה??

אז הפתרון הבא הוא כזה:  
יש להתקין את migration בפרויקט שלנו, נתחיל בזה שיש לגשת ל:  
Tools>NuGet Package Manager> Package Manager Console





במידה ויש לכם כמה connection string בפרויקט שלכם, אז יפתח לכם ההערה הבאה:



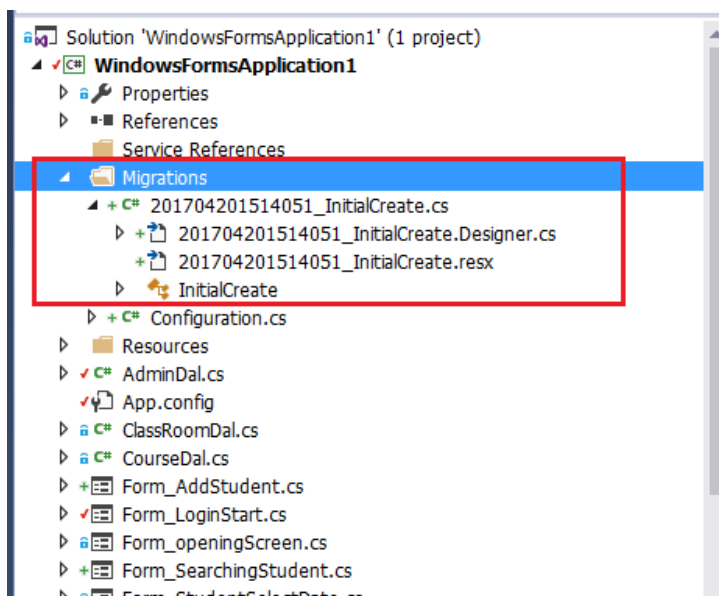
ואז תבחרו את המחלקה שאתם רוצים לשנות בבסיס נתונים, ותרשמו

Enable-Migrations -ContextTypeName <name of the project>.<name of the class dal>  
דוגמה:

Enable-Migrations -ContextTypeName WindowsFormsApplication1.StudentDal

או שפשוט תעתיקו את ההערה שאתם מקבלים, איפה שסימנתי בתמונה לעיל.

יותקן לכם תיקייה של migrations בפרויקט :



הקובץ שמה הוא מראה את המחלקות שנוצרו בהם שינויים במשפיעים על הבסיס נתונים, ואת מה שצריך לעדכן בבסיס נתונים, הוא מאוד סלחני, כלומר אם צריך להוסיף עמודה (משתנה) או למחוק עמודה, הוא מוחק אותם ומשאיר את שאר הנתונים שהיו לנו בבסיס נתונים .

```

201704201514051_InitialCreate.cs* Configuration.cs StudentDal.cs
WindowsFormsApplication1 WindowsFormsApplication1.Migrations.InitialCreate

1 namespace WindowsFormsApplication1.Migrations
2 {
3     using System;
4     using System.Data.Entity.Migrations;
5     public partial class InitialCreate : DbMigration
6     {
7         public override void Up()
8         {
9             CreateTable(
10                 "dbo.Students",
11                 c => new
12                 {
13                     Id = c.Int(nullable: false, identity: true),
14                     Name = c.String(nullable: false),
15                     BirthDate = c.DateTime(nullable: false),
16                     Age = c.Int(nullable: false),
17                     Department = c.String(),
18                     sex = c.String(),
19                     Phone = c.String(maxLength: 12),
20                     Study_year = c.Int(nullable: false),
21                     Study_semester = c.String(maxLength: 1),
22                     GPA = c.Single(nullable: false),
23                 })
24             .PrimaryKey(t => t.Id);
25
26             CreateTable( // בגלל שיש לי רשימה של קורסים אז הוא יוצר גם את הטבלה של המחלקה הזאת
27                 "dbo.InstanceCourses",
28                 c => new
29                 {
30                     Id = c.Int(nullable: false, identity: true),
31                     year = c.Int(nullable: false),
32                     Study_semester = c.Int(nullable: false),
33                     syllabus = c.String(nullable: false, maxLength: 4000),
34                     Student_Id = c.Int(),
35                 })
36             .PrimaryKey(t => t.Id)
37             .ForeignKey("dbo.Students", t => t.Student_Id)
38             .Index(t => t.Student_Id);
39         }
40
41         public override void Down()
42         {
43             DropForeignKey("dbo.InstanceCourses", "Student_Id", "dbo.Students");
44             DropIndex("dbo.InstanceCourses", new[] { "Student_Id" });
45             DropTable("dbo.InstanceCourses");
46             DropTable("dbo.Students");
47         }
48     }
49 }

```

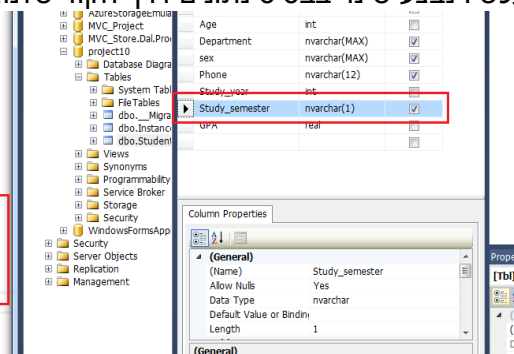
עכשיו נבצע שינוי בבסיס נתונים דרך הקוד שלנו:

```

42 [RegularExpression("05[0-9]-[0-9]$3-[0-9]$4")]
43 [MinLength(10)]
44 [MaxLength(12)]
45 public string Phone { get; set; }
46
47 [Range(1, 4)]
48 public int Study_year { get; set; }
49
50 [RegularExpression("[A-C]")]
51 [MinLength(1)]
52 [MaxLength(5)]//Instead of the previous one: [MaxLength(1)]
53 public string Study_semester { get; set; }
54
55 [Range(0, 100)]
56 public float GPA { get; set; }

```

השינוי



למשל שינוי את הגודל של המחזור מ 1 (char) למחזור בגודל 5  
 נוסף את הקוד הבא ל PM שלנו : add-migration <name that description the changes>

## הדריאל בנג'ו

```

10 AlterColumn("dbo.Students", "sex", c => c.String(maxLength: 6));
11 AlterColumn("dbo.Students", "Name", c => c.String(nullable: false));
12
Package Manager Console
Package source: Microsoft and .NET Default project: WindowsFormsApplication1
existing database. To use an automatic migration instead, delete the Migrations folder and re-run Enable-Migrations specifying the
-EnableAutomaticMigrations parameter.
Code First Migrations enabled for project WindowsFormsApplication1.
PM> add-migration AddStudy_semesterSize
Scaffolding migration 'AddStudy_semesterSize'.
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is used to calculate
the changes to your model when you scaffold the next migration. If you make additional changes to your model that you want to
include in this migration, then you can re-scaffold it by running 'Add-Migration AddStudy_semesterSize' again.
PM>

```

ונוצר לנו קובץ חדש בתיקייה של migrations המציין את השינוי :

```

1 namespace WindowsFormsApplication1.Migrations
2 {
3     using System;
4     using System.Data.Entity.Migrations;
5
6     public partial class AddStudy_semesterSize : DbMigration
7     {
8         public override void Up()
9         {
10             AlterColumn("dbo.Students", "Name", c => c.String());
11             AlterColumn("dbo.Students", "sex", c => c.String(maxLength: 6));
12             AlterColumn("dbo.Students", "Study_semester", c => c.String(maxLength: 5));
13         }
14
15         public override void Down()
16         {
17             AlterColumn("dbo.Students", "Study_semester", c => c.String(maxLength: 1));
18             AlterColumn("dbo.Students", "sex", c => c.String());
19             AlterColumn("dbo.Students", "Name", c => c.String(nullable: false));
20         }
21     }
22 }
23

```

השלב האחרון זה לעדכן את הבסיס נתונים שלנו:

Update-database : PM הבא

```

10 AlterColumn("dbo.Students", "Name", c => c.String(nullable: false));
11
Package Manager Console
Package source: Microsoft and .NET Default project: WindowsFormsApplication1
Scaffolding migration 'AddStudy_semesterSize'.
The Designer Code for this migration file includes a snapshot of your current Code First model. This snapshot is used to calculate
the changes to your model when you scaffold the next migration. If you make additional changes to your model that you want to
include in this migration, then you can re-scaffold it by running 'Add-Migration AddStudy_semesterSize' again.
PM> Update-database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [201704201605393_AddStudy_semesterSize].
Applying explicit migration: 201704201605393_AddStudy_semesterSize.
Running Seed method.
PM>

```

נוכל ללכת לשרת שלנו ולבצע שמה ריענון על DB , ולראות שהשינוי עבר בהצלחה.

פשוט וקל!

MVC_Projct	Department	nvarchar(MAX)	<input checked="" type="checkbox"/>
MVC_Store.Dal.Proj	sex	nvarchar(6)	<input checked="" type="checkbox"/>
project10	Phone	nvarchar(12)	<input checked="" type="checkbox"/>
Database Diagram	Study_year	int	<input type="checkbox"/>
Tables	Study_semester	nvarchar(5)	<input checked="" type="checkbox"/>
System Table	GPA	real	<input type="checkbox"/>
File Tables			
dbo.__Migra			
dbo.Instance			
dbo.Student			
Views			
Synonyms			
Assemblies			

לאחר העדכון של migration אתם יכולים למחוק את אותם קבצים שנמצאים בתיקייה של Migration אם אתם רוצים

13. אז אם מדברים על שינויים זה המקום לדבר על data annotation :

אז הנה כמה דוגמאות שימושיות : הערות: הגדרת טיפוסים:

C#	איך זה נראה בבסיס נתונים	Allow Null	הערה חשובה, EF יוצר בעצמו את keyid של הטבלה, ולכן אם אתם רוצים ליצור מפתח ייחודי לאובייקט שלכם, תצרו אותו בצורה הבאה:
Int	int	No	[key] עבור המפתח של הבסיס – Public int id נתונים שמתעדכן אוטומטי ואין לכם אפשרות לערוך או לגעת בו.  [required] [Index(IsUniqu = true)] עבור המפתח – Public int IdStudent הייחודי של האובייקט שלכם
Int?	int	Yes	
double	float	No	
double?	float	Yes	
float	real	No	
float?	real	Yes	
String	nvarchar(max)	Yes	
Char	EF לא תומך ב-char צריך להשתמש ב-string עם אילוץ בגודל של 1.	-	

```

Form_AddStudent.cs  DbContextDal.cs  User.cs*  X
WindowsFormsApplication1  WindowsFormsApplication1.User  _UserID

1  namespace WindowsFormsApplication1
2  {
3      using System.ComponentModel.DataAnnotations;
4      using System.ComponentModel.DataAnnotations.Schema;
5
6      public class User
7      {
8          // הגדרה של מפתח, בצורה שלא ימולא אוטומטית על ידי השרת שלנו
9          private int _UserID;
10         [Key, DatabaseGenerated(DatabaseGeneratedOption.None)]
11         public int ID
12         {
13             get { return _UserID; }
14             set
15             {
16                 if (value != _UserID)
17                 {
18                     _UserID = value;
19                 }
20             }
21         }
22         // דרך לא נכונה זה להגדיר באופן הבא:
23         // [Key] // לא יהיה ניתן לשינוי
24         // public int id { get; set; }
25
26
27         [Required] // אומר שזהו שדה חובה, כלומר לא מאפשר שיהיה בעמודה/בשדה
28         [StringLength(25)]
29         public string password { get; set; }
30
31         [Required]
32         // אפשר להזין רק את אחת מהמחרוזות האלה
33         [RegularExpression("Secretary|Admin|Lecturer|Practitioner|Student|StudentSchedule")]
34         public string permission { get; set; }
35
36         // בנאי דיפולטיבי - חובה!!!
37         public User() { password = "0000"; }
38
39         // בנאים אחרים
40         public User(int id, string _permission)
41     }

```

```

Student.cs*
WindowsFormsApplication1
WindowsFormsApplication1.Student
a_StudentID

9
10 //Table("tblStudents") // Student: שם אחר מ: Student
11 public class Student
12 {
13     private int _StudentID;
14     [Key, DatabaseGenerated(DatabaseGeneratedOption.None)]
15     [ForeignKey("user")] // תיקון על ההגדרה באינטרנט
16     // בגלל שהתכונה הזאת ממוקמת מעל הת.ז. של הסטודנט שהוא גם מפתח ראשי, הוא יהיה גם מפתח זר
17     [Column(Order = 0)]
18     public int ID { get; set; }
19     public User user { get; set; } // זר את המפתח זר
20
21
22     public string Name { get; set; }
23
24     [Required]
25     [RegularExpression("Female|Male")] // ערכים ספציפיים בלבד
26     [MinLength(4)] // גודל מילה מינמלית בת 4 תווים
27     [MaxLength(6)] // גודל מילה מקסימלית בת 6 תווים
28     public string Gender { get; set; }
29
30     // הגדרת טיפוס המשתנה, למשל הגדרנו אותו פה לתאריך
31     [DataType(DataType.DateTime)]
32     public DateTime? BirthDate { get; set; }
33
34     [Range(0, 120)] // מספרים מותרים למשתנה הזה
35     private int _age = 18; // הגדרת ערך דיפולטיבי
36     [DefaultValue(18)]
37     public int Age
38     {
39         get { return _age; }
40         set { _age = value; }
41     }
42
43     public string Department { get; set; }
44
45     [Range(1, 4)]
46     [Required]
47     public int? Study_year { get; set; }
48
49     [RegularExpression("[A-C]")] // הגדרת של התווים המורשים לשדה הזה
50     [MinLength(1)] // ! char אפשרות להגדיר פה
51     [MaxLength(1)] // ! char אפשרות להגדיר פה
52     [Required]
53     public string Study_semester { get; set; }
54
55

```

## 14. הערות נפוצות ב- Entity Framework

System.Data.Entity.Infrastructure.DbUpdateException: .... The duplicate key value is (111)...  
 כלומר ניסתם להכניס לטבלה ערך שכבר key שלו קיים בבסיס נתונים, וזה פעולה בלתי אפשרית!  
 וכ"ו ..

המלצה שלי, תגדירו כל פונקציה של הכנסת אובייקט לבסיס נתונים בצורה הבאה:

```
private void Add_Student(Student std)
{
    try {
        StudentDal dal = new StudentDal();
        dal.students.Add(std);
        dal.SaveChanges();
        MessageBox.Show("The Student " + std.IdStudent + " Added!");
    }
    catch (DbEntityValidationException ex){
        //MessageBox.Show("Db Entity Validation Exception");
        string str = "" + ex.Source + " : " + ex.GetType() + "\n-----\n" +
            "Message: " + ex.Message + "\n\nExplain & solution:\nThe problem is that you try to insert an object with data that does not fit the limitations of
        foreach (var entityValidationErrors in ex.EntityValidationErrors)
        {
            foreach (var validationError in entityValidationErrors.ValidationErrors)
            {
                str += ("Property: " + validationError.PropertyName + " Error: " + validationError.ErrorMessage + "\n");
            }
        }
        MessageBox.Show(str);
    }
    catch (DbUnexpectedValidationException ex) {
        //MessageBox.Show("Db Unexpected Validation Exception");
        string str = "" + ex.Source + " : " + ex.GetType() + "\n-----\n" +
            "Message: " + ex.Message;
        MessageBox.Show(str);
    }
    catch (DbUpdateException ex) {
        //MessageBox.Show("Db Update Exception");
        string str = "" + ex.Source + " : " + ex.GetType() + "\n-----\n" +
            "Message: " + ex.Message + "\n\nExplain & solution:\nThe problem is that you are trying to insert an object that already exists in the system with th
        MessageBox.Show(str);
    }
    catch (InvalidOperationException ex) {
        //MessageBox.Show("Invalid Operation Exception");
        string str = "" + ex.Source + " : " + ex.GetType() + "\n-----\n" +
            "Message: " + ex.Message + "\n\nExplain & solution:\nThe problem is that the database is formatted differently from what is currently in your class
        MessageBox.Show(str);
    }
    catch (Exception ex) {
        //MessageBox.Show("Exception");
        string str = "" + ex.Source + " : " + ex.GetType() + "\n-----\n" +
            "Message: " + ex.Message;
        MessageBox.Show(str);
    }
}
```

ואז תשתמשו בה ככה בכל מקום. זה יעזור לכם להיפטר מקריסת התוכנית, ומאפשר הצגת פלט ברורה למשתמש על תיאור הבעיה עצמה, ולא סתם מלא מלל לא מובן של תקלות..

15. אז לסיום, נראה איך אפשר לעדכן את הבסיס נתונים בערכים אחרים ממה שקיים בו. ואיך למחוק נתונים.

### עדכון אובייקט:

כתבתי פונקציה שמקבלת מספר תעודת זהות (מספר ייחודי – מפתח ראשי) ומקבלת סיסמה חדשה, והפונקציה מחליפה את הסיסמה הישנה בחדשה.

```
public static void changePasswordUser(int id, string newPassword)
{
    try
    {
        DbContextDal dal = new DbContextDal();

        // פעולת חיפוש משתנה

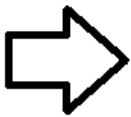
        // יצירת אובייקט משתמש כדי שניקלוט אליו משתמש קיים מהבסיס נתונים
        User user;

        // מציאת משתמש לפי מפתח ראשי
        user = dal.users.Find(id);
        // או

        // מציאת סטודנט לפי סינון, ובחירה של הראשון או דיפולטיבי
        user = dal.users.Where(x => x.ID == id).FirstOrDefault();

        // שינוי נתונים באובייקט
        user.password = newPassword;

        // שמירת מצב השינויים
        dal.SaveChanges();
    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex.ToString());
    }
}
```



ID	password	permission
100	0000	Student
101	0000	Student
102	0000	Student
103	0000	Student
104	0000	Student
105	0000	Student
106	0000	Student

ID	password	permission
100	1234	Student
101	0000	Student
102	0000	Student
103	0000	Student
104	0000	Student
105	0000	Student
106	0000	Student

```

public static void deleteUser(int id)
{
    try
    {
        // פעולת מחיקה
        DbContextDal dal = new DbContextDal();

        // יצירת אובייקט משתמש כדי שניקלוט אליו משתמש קיים מהבסיס נתונים
        User user;
        Student std;
        // בגלל שסטודנט הוא מפתח זר של משתמש, כלומר אם קיים סטודנט זה רק בתנאי שקיים משתמש כזה
        // !אז אם נמחק את המשתמש קודם, אז הסטודנט לא יעמוד בתנאי הזה
        // ולכן חייבים למחוק את הסטודנט לפני שמוחקים את המשתמש
        // ככה הגדרתי, שהת.ז. של הסטודנט הוא מפתח זר של משתמש

        // מציאת משתמש וסטודנט לפי מפתח ראשי
        user = dal.users.Find(id);
        std = dal.students.Find(id);

        // בדיקה אם מצאנו אובייקטים כלשהם במידה ולא, מסיימים את הפונקציה
        if (std == null || user == null) return;

        // הסרת האובייקט מרשימת האובייקטים שבבסיס נתונים
        dal.students.Remove(std); // קודם מסירים את הסטודנט
        dal.users.Remove(user);   // ורק אז את המשתמש

        // שמירת מצב השינויים
        dal.SaveChanges();
    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex.ToString());
    }
}

```

דוגמה:

```

4 namespace WindowsFormsApplication1
5 {
6     static class Program
7     {
8         /// <summary>
9         /// The main entry point for the application.
10        /// </summary>
11        [STAThread]
12        static void Main()
13        {
14            Application.EnableVisualStyles();
15            Application.SetCompatibleTextRenderingDefault(false);
16
17            // עדכון סיסמה של משתמש 101
18            GeneralFuntion.changePasswordUser(101,"0101");
19            // מחיקת משתמש 108 - כולל הסטודנט שלו
20            GeneralFuntion.deleteUser(108);
21
22            Application.Run(new FormMenu());
23        }
24    }
25 }

```



HADRIEL\SQLEXPRESS...ct10 - dbo.Users			
	ID	password	permission
	100	1234	Student
	101	0000	Student
	102	0000	Student
	103	0000	Student
	104	0000	Student
	105	0000	Student
	106	0000	Student
	107	0000	Student
	108	0000	Student
*	NULL	NULL	NULL

HADRIEL\SQLEXPRESS...0 - dbo.Students					
	ID	Name	BirthDate	Aqe	Department
	100	john	1980-12-12 0...	18	Mechanical er
	101	levan	1980-12-12 0...	18	Structural en
	102	levan	1985-07-01 0...	18	Mechanical er
	103	john	1985-07-01 0...	18	Electrical eng
	104	arthur	1980-01-05 0...	18	Industry and
	105	roey	1980-01-05 0...	18	Structural en
	106	arthur	1980-12-12 0...	18	Software eng
	107	smith	1980-12-12 0...	18	Structural en
	108	gabriel	1985-07-01 0...	18	Software eng
*	NULL	NULL	NULL	NULL	NULL

HADRIEL\SQLEXPRESS...0 - dbo.Students			
	ID	password	permission
	100	1234	Student
	101	0101	Student
	102	0000	Student
	103	0000	Student
	104	0000	Student
	105	0000	Student
	106	0000	Student
	107	0000	Student
*	NULL	NULL	NULL

HADRIEL\SQLEXPRESS...0 - dbo.Students					
	ID	Name	BirthDate	Aqe	Department
	100	john	1980-12-12 0...	18	Mechanical er
	101	levan	1980-12-12 0...	18	Structural en
	102	levan	1985-07-01 0...	18	Mechanical er
	103	john	1985-07-01 0...	18	Electrical eng
	104	arthur	1980-01-05 0...	18	Industry and
	105	roey	1980-01-05 0...	18	Structural en
	106	arthur	1980-12-12 0...	18	Software eng
	107	smith	1980-12-12 0...	18	Structural en
*	NULL	NULL	NULL	NULL	NULL

ניתן לראות שהסטודנט והמשתמש עם מספר תעודת זהות 108 נמחקו מהבסיס נתונים  
(ושהסיסמה של משתמש 101, השתנתה מ0000 ל 0101)

נשאר לנו לראות איך מוסיפים אובייקטים לבסיס נתונים:

```
public static void createUser(int id, string password)
{
    try
    {
        // פעולת הוספה
        DbContextDal dal = new DbContextDal();

        // יצירת אובייקט משתמש וסטודנט כדי להכניס לבסיס נתונים
        User userStudent = new User(id, password, "Student");
        Student std = new Student();
        std.ID = id;
        std.Name = "Hadriel";
        std.Study_semester = "B";
        std.Study_year = 2;
        std.GPA = 0;
        std.user = userStudent;

        // הוספת האובייקט לרשימת האובייקטים שבבסיס נתונים
        dal.users.Add(userStudent); // קודם מוסיפים את המשתמש
        dal.students.Add(std);      // ואחר כך את הסטודנט
        // שם לב, בגלל שהגדרתי שאם יש סטודנט אז הוא חייב להיות מקושר
        // למשתמש כלשהו, אז חובה קודם להוסיף משתמש לבסיס נתונים

        // שמירת מצב השינויים
        dal.SaveChanges();
    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex.ToString());
    }
}
```

HADRIEL\SQLEXPRESS - dbo.Students			
ID	password	permission	
100	1234	Student	
101	0101	Student	
102	0000	Student	
103	0000	Student	
104	0000	Student	
105	0000	Student	
106	0000	Student	
107	0000	Student	
*	NULL	NULL	NULL



HADRIEL\SQLEXPRESS - dbo.Students			
ID	password	permission	
100	1234	Student	
101	0101	Student	
102	0000	Student	
103	0000	Student	
104	0000	Student	
105	0000	Student	
106	0000	Student	
107	0000	Student	
201	abc123	Student	
*	NULL	NULL	NULL

HADRIEL\SQLEXPRESS - dbo.Students					
ID	Name	BirthDate	Age	Depai	
100	john	1980-12-12 0...	18	Mecha	
101	levan	1980-12-12 0...	18	Struct	
102	levan	1985-07-01 0...	18	Mecha	
103	john	1985-07-01 0...	18	Electri	
104	arthur	1980-01-05 0...	18	Indust	
105	roey	1980-01-05 0...	18	Struct	
106	arthur	1980-12-12 0...	18	Softw	
107	smith	1980-12-12 0...	18	Struct	
*	NULL	NULL	NULL	NULL	NULL

HADRIEL\SQLEXPRESS - dbo.Students				
ID	Name	BirthDate	Age	
100	john	1980-12-12 0...	18	
101	levan	1980-12-12 0...	18	
102	levan	1985-07-01 0...	18	
103	john	1985-07-01 0...	18	
104	arthur	1980-01-05 0...	18	
105	roey	1980-01-05 0...	18	
106	arthur	1980-12-12 0...	18	
107	smith	1980-12-12 0...	18	
201	Hadriel	NULL	18	
*	NULL	NULL	NULL	NULL

## בהצלחה

עד לכאן זה המדריך שלי על Entity Framework מקווה שנהניתם, אל תשכחו לפרגן ;)