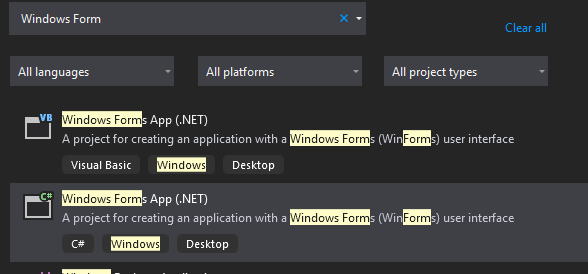
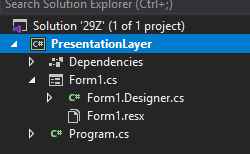
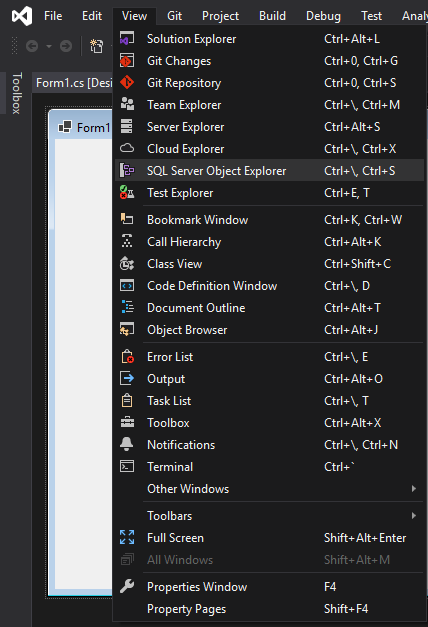
Kreiranje novog projekta

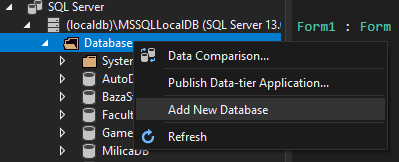
Biramo Windows Forms App (.Net) koji automatski kreira Prezentacijski sloj, potrebno ga je samo preimenovati u **PresentationLayer** i ostaviti ga za kasnije.



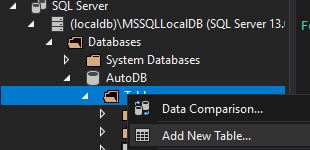


Nakon toga kreiramo bazu podataka sa kojom cemo raditi, za to nam treba SQL Server Object Explorer koji se nalazi sa leve strane, ukoliko ga nema idemo na View = > SQL Server Object…

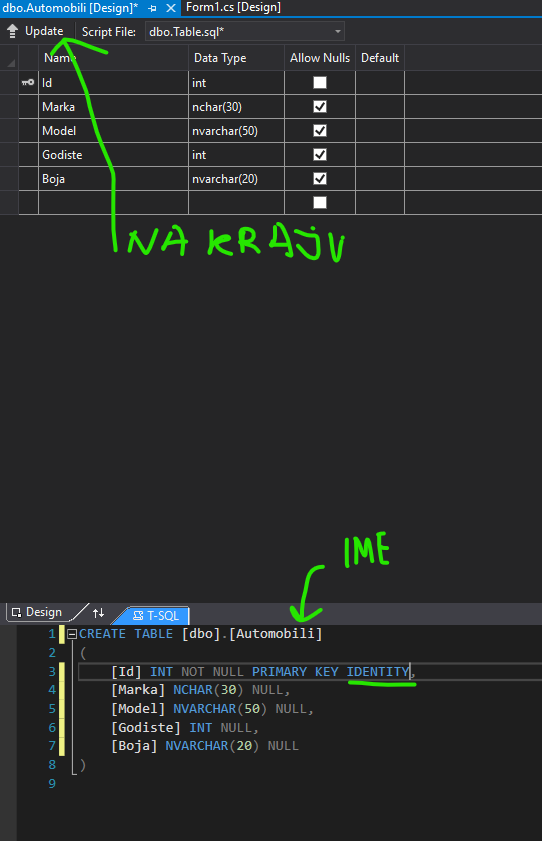


Idemo na sledecu putanju i dodajemo nasu bazu ovako : 

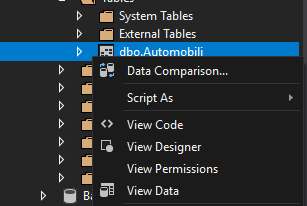
Nakon sto smo napravili bazu potrebno je da napravimo tabelu unutar nase baze :

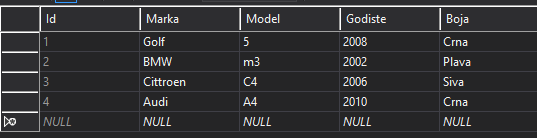


U tabeli menjamo ime, dodajemo odgovarajuce tipove, i kada zavrsimo sa tabelom idemo na **UPDATE** (gore levo)

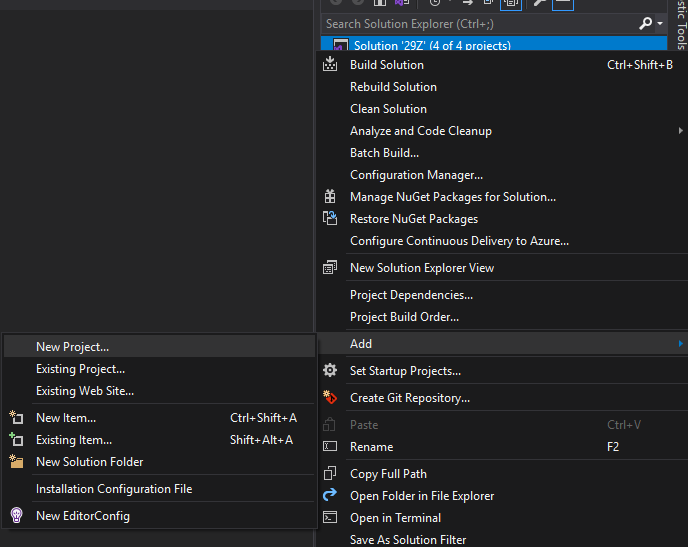


Kada napravimo tabelu potrebno je da unesemo podatke u istu, to radimo tako sto idemo desni klik na tabelu => View Data;



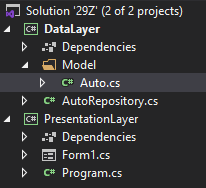


Kad se napravi baza I tabela pravi se **DataLayer (desni klik na projekat, add New Project – Class Library)** i unutar njega se pravi **folder Model** u kom se nalazi klasa koja odgovara tabeli u bazi.



**NA SVE KLASE DODATI \*\*\*PUBLIC**

Klasa **Class1.cs** generisana sa DataLayer bibliotekom se preimenujeu **Ime\_KlaseRepository**



U Model -> ImeKlase.cs se prave tipovi podataka koji odgovaraju tipovima u bazi, pored njih treba napraviti i :

* Gettere i settere
* Konstruktor klase
* toString metodu ( public override toString() )

KLASA AUTO

using System;

using System.Collections.Generic;

using System.Text;

namespace DataLayer.Model

{

public class Auto

{

private string Marka;

private string Model;

private int Godiste;

private string Boja;

public Auto(string marka, string model, int godiste, string boja) **- KONSTRUKTOR**

{

Marka = marka;

Model = model;

Godiste = godiste;

Boja = boja;

}

public string boja – **GET I SET METODA RAZLICIT NAZIV**

{

get { return Boja; }

set { Boja = value; }

}

public string marka

{

get { return Marka; }

set { Marka = value; }

}

public string model

{

get { return Model; }

set { Model = value; }

}

public int godiste

{

get { return Godiste; }

set { Godiste = value; }

}

public override string ToString() **– TO STRING METODA**

{

return $"{Marka} {Model} {Godiste} {Boja}";

}

}

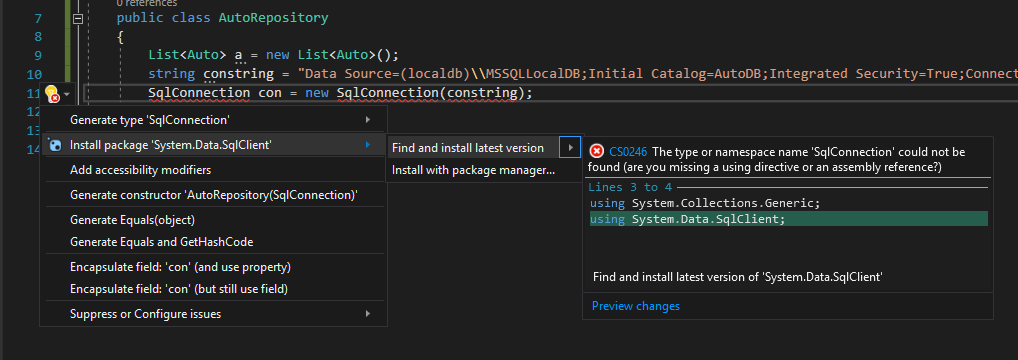
}

**ImeKlaseRepository**

Pri kreiranju liste

Konekcioni string se nalazi kad kliknemo na **BAZU** i dole desno ispod properties nadjemo **connection string**

Nakon toga pravimo SQL konekciju sa bazom i instaliramo System.Data.SqlClient



**METODA ZA CITANJE IZ BAZE**

public List<**Auto**> GetAutos()

{

List**<Auto>** **list** = new List<**Auto**>();

string cons = "**Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=AutoDB;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False**"; *- konekcioni string sa bazom*

SqlConnection con = new SqlConnection(cons);

con.Open();

string command = "Select \* FROM **Automobili**"; *- umesto Automobili ime tabele*

SqlCommand com = new SqlCommand(command, con);

SqlDataReader dr = com.ExecuteReader();

while (dr.Read())

{

list.Add(new Auto(**dr.GetString(1), dr.GetString(2), dr.GetInt32(3), dr.GetString(4)**)); *- dr.get + tip kolone u tabeli I u zagradi njen redni broj (ID je 0)*

}

return **list**;

}

**METODA ZA INSERT**

public int Insert**Auto**(**Auto a**)

{

int result;

string cons = "**Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=AutoDB;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False**";

using (SqlConnection con = new SqlConnection(cons))

{

con.Open();

string command = "Insert Into Automobili (**Marka,Model,Godiste,Boja) VALUES (@Marka,@Model,@Godiste,@Boja**)";

SqlCommand com = new SqlCommand(command, con);

com.Parameters.AddWithValue("**@Marka", a.marka**);

com.Parameters.AddWithValue("**@Model", a.model**);

com.Parameters.AddWithValue("**@Godiste", a.godiste**);

com.Parameters.AddWithValue("**@Boja", a.boja**);

result = com.ExecuteNonQuery();

}

return result;

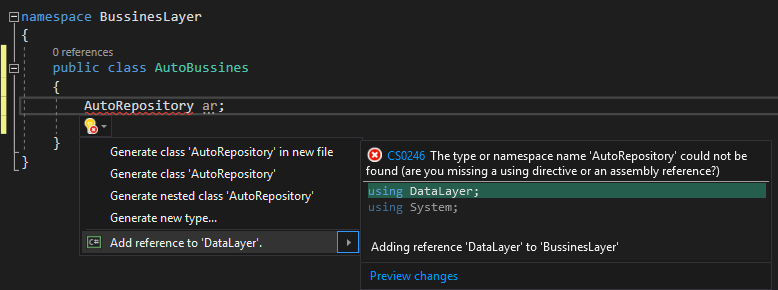
}

**Kada se zavrsi sa DataLayerom pravi se BusinessLayer koji ima u sebi jednu klasu koja se naziva : *Klasa*Bussines I ona sadrzi metode iz *Klasa*Repository.**

**BusinessLayer je CLASS LIBRARY**

**Unutar ove klase se pravi objekat tipa *Klasa*Repository sa konstruktorom klase Business**

**Pri kreiranju objekta potrebno je dodati referencu na DataLayer**

****

***Klasa*Libary :**

namespace BussinesLayer

{

public class AutoBussines

{

AutoRepository ar;

public AutoBussines() -/

{ -/ KREIRANJE OBJEKTA PREKO KONSTRUKTORA

this.ar = new AutoRepository(); -/ REPOSITORY

} -/

public List<Auto> GetAutos() \*/

{ \*/ POZIVANJE METODE ZA UCITAVANJE

return ar.GetAutos(); \*/ PODATAKA

} \*/

public bool InsertAuto(Auto a) // METODA ZA UNOS OBJEKATA U LISTU

{ // POMOCU METODE Insert*Klasa IZ*

int result = ar.InsertAuto(a); // *REPOSITORY-A*

if (result != 0) // METODA VRACA TRUE AKO JE USPESAN UNOS

return true;

else

return false;

}

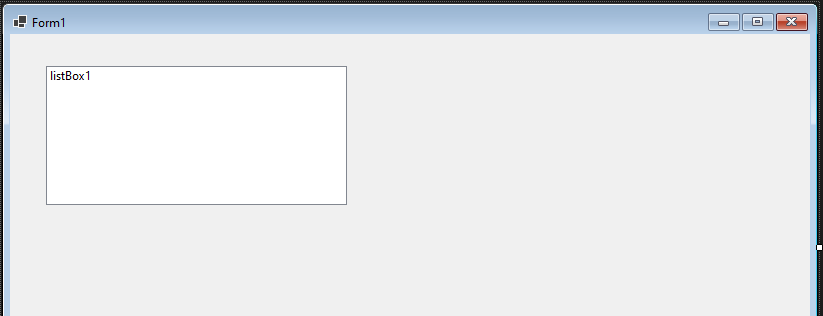
}

}

**PRESENTATION LAYER**

Kada se zavrsi sa Data i Business Layer-om prelazi se na prvi kreirani layer to jest Presentation Layer koji u sebi ima Form1.cs tj. Windows Formu.

Potrebno je dodati referencu na Data i Business Layere.

Da bi ispisali podatke iz tabele moramo imati listbox koji stavljamo u formu. ****

Posle toga dvoklik na prazno da bi usli u Form1\_Load

**KOD ZA ISPIS U LISTBOX**

private void Form1\_Load(object sender, EventArgs e)

{

List<*Auto*> *list* = new List<*Auto*>(); \* Pravimo listu koja ima tip nase

*AutoBussines ab* = new *Auto*Bussines(); \* kreirane klase, i objekat tipa

*list* = *ab*.Get*Autos*(); \* *Klasa*Business koji nam omogucava

foreach (*Auto a* in *list*) \* da preko njega ucitamo podatke u

{ \* listu, zatim

listBox1.Items.Add(*a*.ToString()) \* petlja foreach(*ImeKlase neko*

} \* *slovo* **in** *imeliste*)

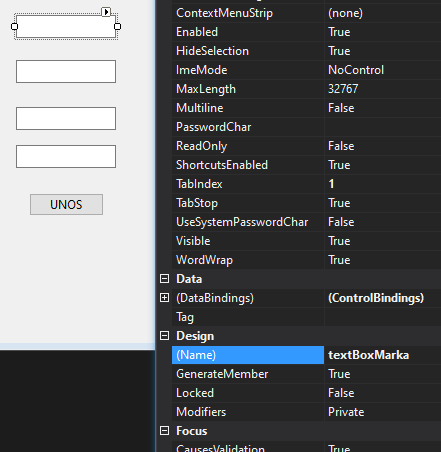
} \* unutar items.add umesto slovo a stavljamo to neko slovo iz foreach petlje

**Unos objekta preko forme**

Prvo dodajemo textboxove za sve podatke, u ovom slucaju to su Model, Marka, Godiste i Boja.

Pored textboxova treba dodati button na ciji ce se klik pozvati funkcija za unos objekta u listu.

textBoxovima I Buttonu dati ime koje asocira na njegovu svthu preko properties taba.



**Unos objekta u listu**

\*\*\* prvo kliknemo na button dva puta da bi napravili button1\_Click event

private void button1\_Click(object sender, EventArgs e)

{

Auto a = new Auto(textBoxMarka.Text,textBox2.Text, \*UZIMANJE VREDNOSTI IZ Convert.ToInt32(textBox3.Text), textBox4.Text); TEXTBOXOVA I KREIRANJE

AutoBussines ab = new AutoBussines(); OBJEKTA

if (ab.InsertAuto(a) == true) \*POZIVANJE FUNKCIJE ZA UNOS

MessageBox.Show("Valid"); PREKO OBJEKTA KLASE

else *Klasa*Business

MessageBox.Show("Invalid"); \* UKOLIKO JE UNOS USPESAN

listBox1.Items.Clear(); METODA VRACA TRUE

Form1\_Load(null,null); AKO JE TRUE ONDA OBAVESTAVAMO

} PORUKOM I RADIMO UPDATE FORME POMOCU **Form1\_Load(null,null);**