**AUTOMATIC MIGRATION**

• This will allows to process database migration automatically.

• The automated migrations can be implemented by executing the enable-migrations command in the Package Manager Console.

• Run the following command enable-migrations -EnableAutomaticMigration:$true

• The next step is to set the database initializer in the context class to **MigrateDatabaseToLatestVersion**

ADD TRAINER.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Codefirst1

{

class Trainer

{

public int TrainerId { get; set; }

public string Trainername { get; set; }

public string Email { get; set; }

}

}

DBCONTEXT

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.Entity;

using System.Data;

namespace Codefirst1

{

class MyDbcontext:DbContext

{

public MyDbcontext():base("newDBannotion1")

{

Database.SetInitializer<MyDbcontext>(new DropCreateDatabaseAlways<MyDbcontext>());

}

public virtual DbSet<Programme> Programmes { get; set; }

public virtual DbSet<Student> Students { get; set; }

public virtual DbSet<StudentAddress> StudentAddresses { get; set; }

public virtual DbSet<Subject> Subjects { get; set; }

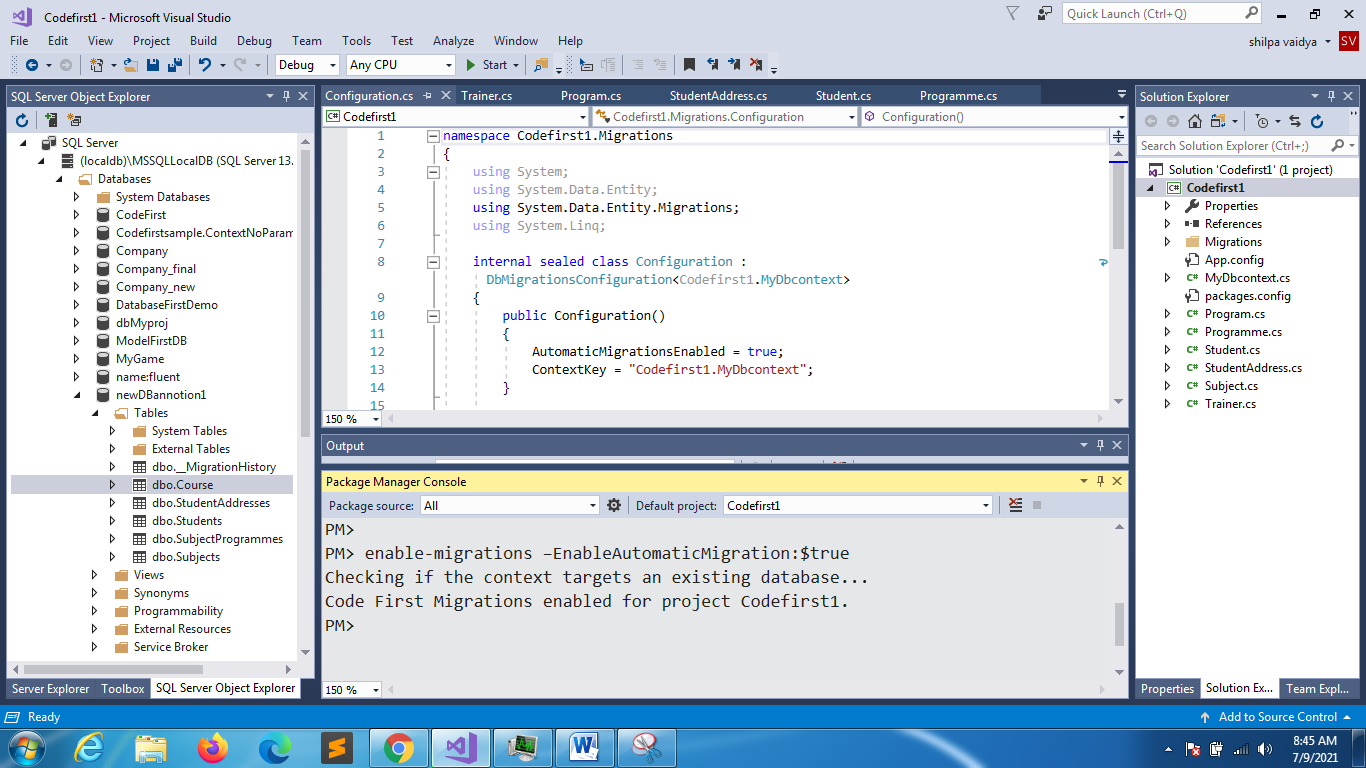
public virtual DbSet<Trainer> Trainers { get; set; }

}

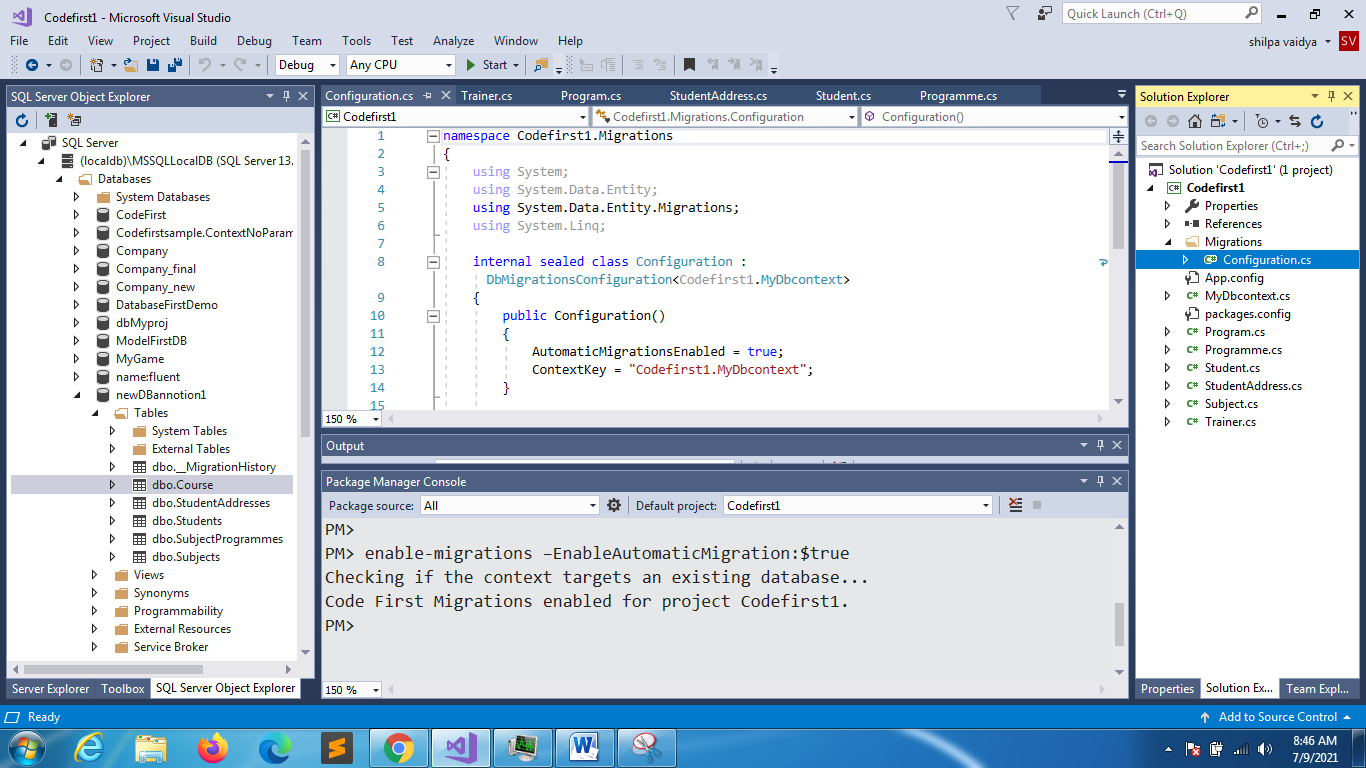
}

PACKAGE MANAGER

enable-migrations –EnableAutomaticMigration:$true



Configuration class will be created



**MyDbcontext**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.Entity;

using System.Data;

namespace Codefirst1

{

class MyDbcontext:DbContext

{

public MyDbcontext():base("newDBannotion1")

{

Database.SetInitializer<MyDbcontext>(new MigrateDatabaseToLatestVersion<MyDbcontext,Codefirst1.Migrations.Configuration>());

}

public virtual DbSet<Programme> Programmes { get; set; }

public virtual DbSet<Student> Students { get; set; }

public virtual DbSet<StudentAddress> StudentAddresses { get; set; }

public virtual DbSet<Subject> Subjects { get; set; }

public virtual DbSet<Trainer> Trainers { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Codefirst1

{

class Program

{

static void Main(string[] args)

{

try

{

MyDbcontext db = new MyDbcontext();

Programme p = new Programme();

p.Id = 105;

p.Title = "BCOM";

p.Duration = 5;

p.Fees = 10000;

db.Programmes.Add(p);

db.SaveChanges();

Console.WriteLine("db created");

}

catch(Exception ob)

{

Console.WriteLine("exception"+ob.Message );

}

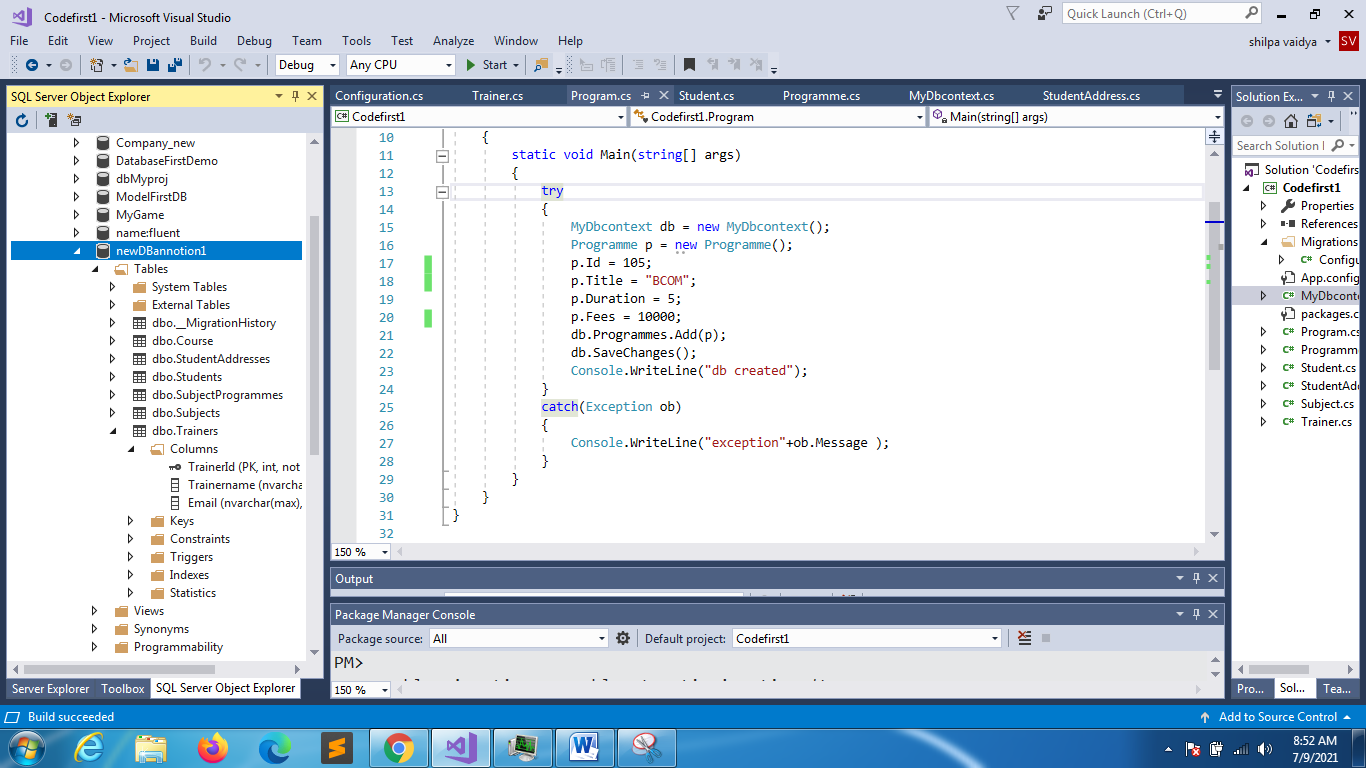
}

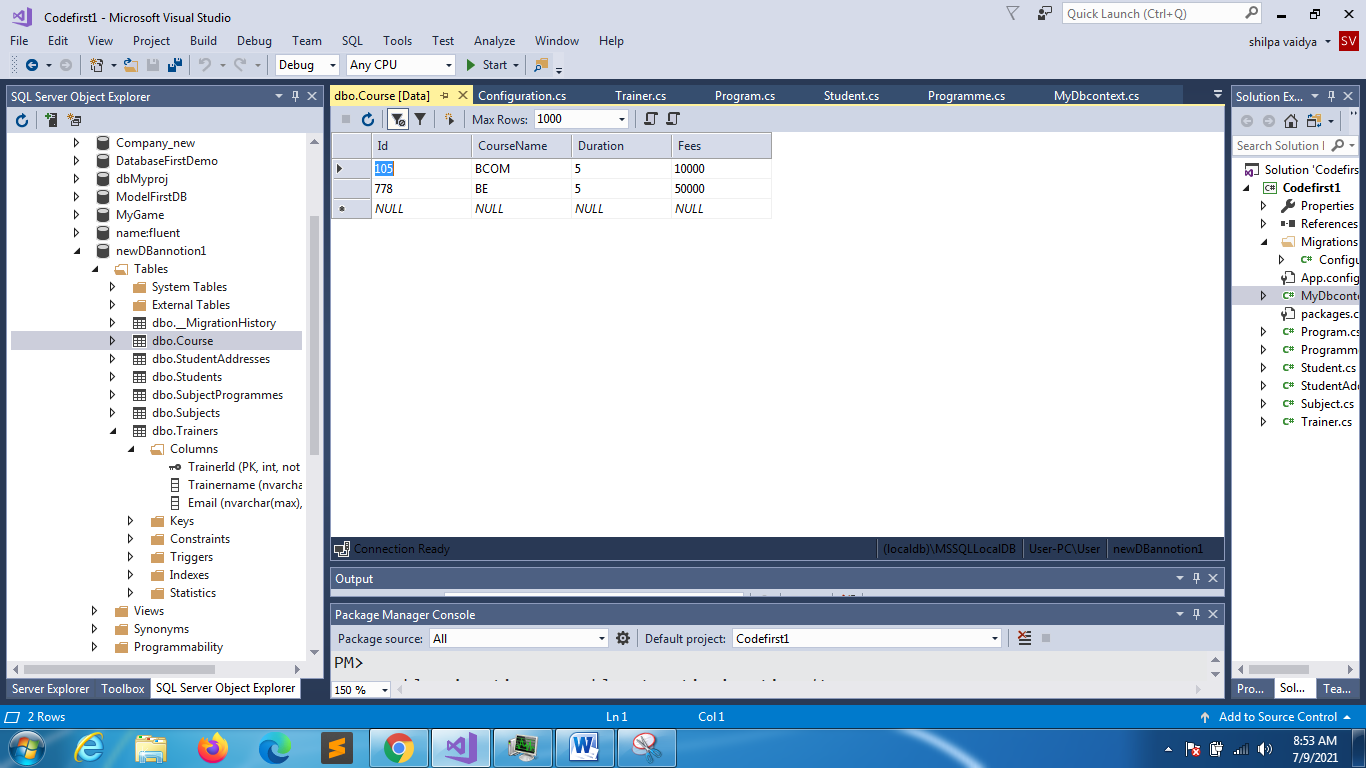
}

}

TRAINER TABLE CREATED

ALSO DATA WAS NOT LOST





PROGRAMME

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace Codefirst1

{

[Table("Course")]

class Programme

{

[DatabaseGenerated(DatabaseGeneratedOption.None)]

//scalar properties

public int Id { get; set; }//primary key

//not null

[Required]

[Column("CourseName",TypeName ="varchar")]

[MaxLength(50)]

public string Title { get; set; }

public int Duration { get; set; }

public float Fees { get; set; }

public int hours { get; set; }

//navigation

public virtual ICollection<Student> Students { get; set; }

public virtual ICollection<Subject> Subjects { get; set; }

}

}

namespace Codefirst1.Migrations

{

using System;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

internal sealed class Configuration : DbMigrationsConfiguration<Codefirst1.MyDbcontext>

{

public Configuration()

{

AutomaticMigrationsEnabled = true;

AutomaticMigrationDataLossAllowed = true;

ContextKey = "Codefirst1.MyDbcontext";

}

protected override void Seed(Codefirst1.MyDbcontext context)

{

// This method will be called after migrating to the latest version.

// You can use the DbSet<T>.AddOrUpdate() helper extension method

// to avoid creating duplicate seed data.

}

}

}