**CODEFIRST DATABASE INITIALIZER**

**CreateDatabaselfNotExists** : This is a default database initializer. It creates a database only if does not exists **DropCreateDatabaseWhenModelChanges** : This class drops and re-creates a database only if there is difference between model classes and table schemas

**DropCreateDatabaseAlways** : This will always drops and creates a new database.

**MigrateDatabaseToLatestVersion** : This will updates the database schema when model changes. but without losing existing data.

**Student.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using System.ComponentModel.DataAnnotations;

namespace Codefirst1

{

class Student

{

[Key]

//scalar properties

public int Rollno { get; set; }

public string name { get; set; }

public bool isGraduated { get; set; }

public string Contactno { get; set; }

public int age { get; set; }

public int ProgrammeId { get; set; }

[ForeignKey("ProgrammeId")]

public virtual Programme Programme { get; set; }

public virtual StudentAddress StudentAddress { get; set; }

}

}

Program.cs

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 = 778;

p.Title = "BE";

p.Duration = 5;

p.Fees = 50000;

db.Programmes.Add(p);

db.SaveChanges();

Console.WriteLine("db created");

}

catch(Exception ob)

{

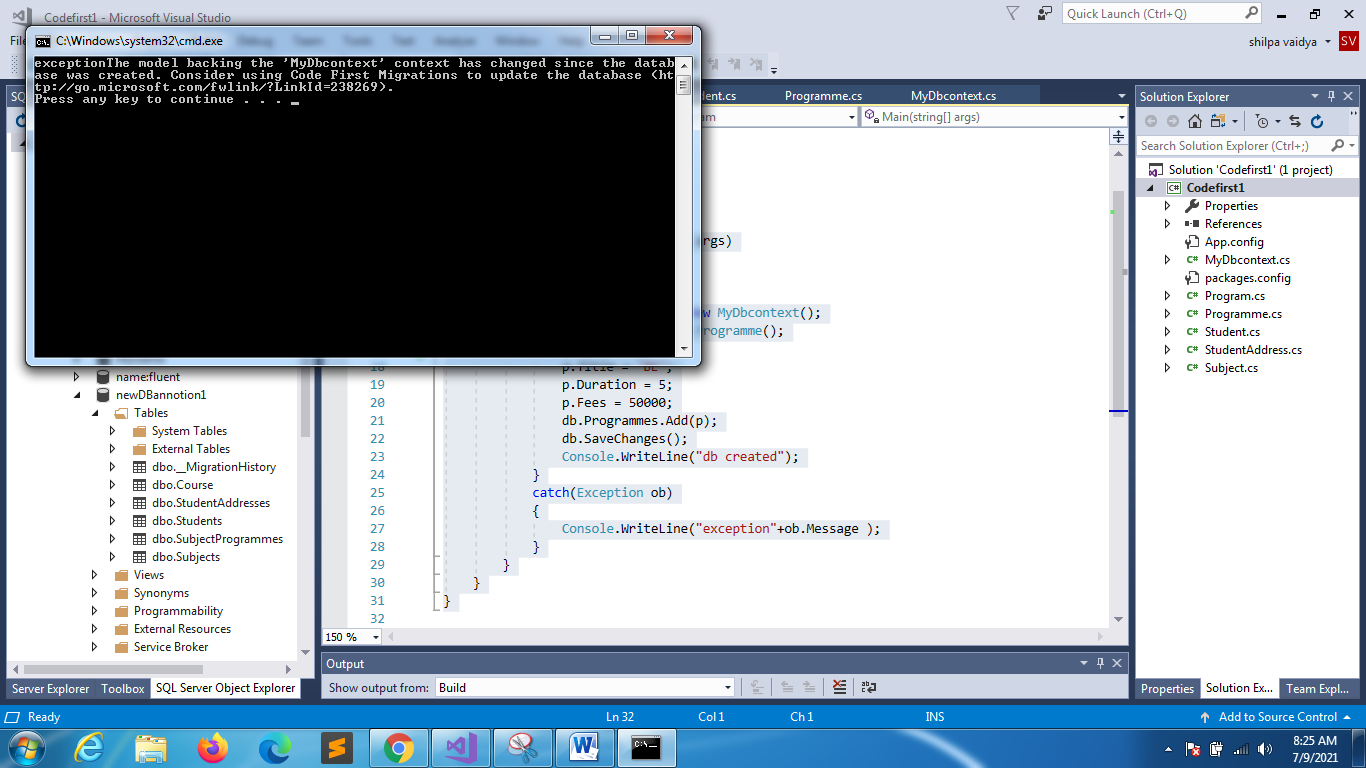
Console.WriteLine("exception"+ob.Message );

}

}

}

}



**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 DropCreateDatabaseIfModelChanges<MyDbcontext>());

}

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

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

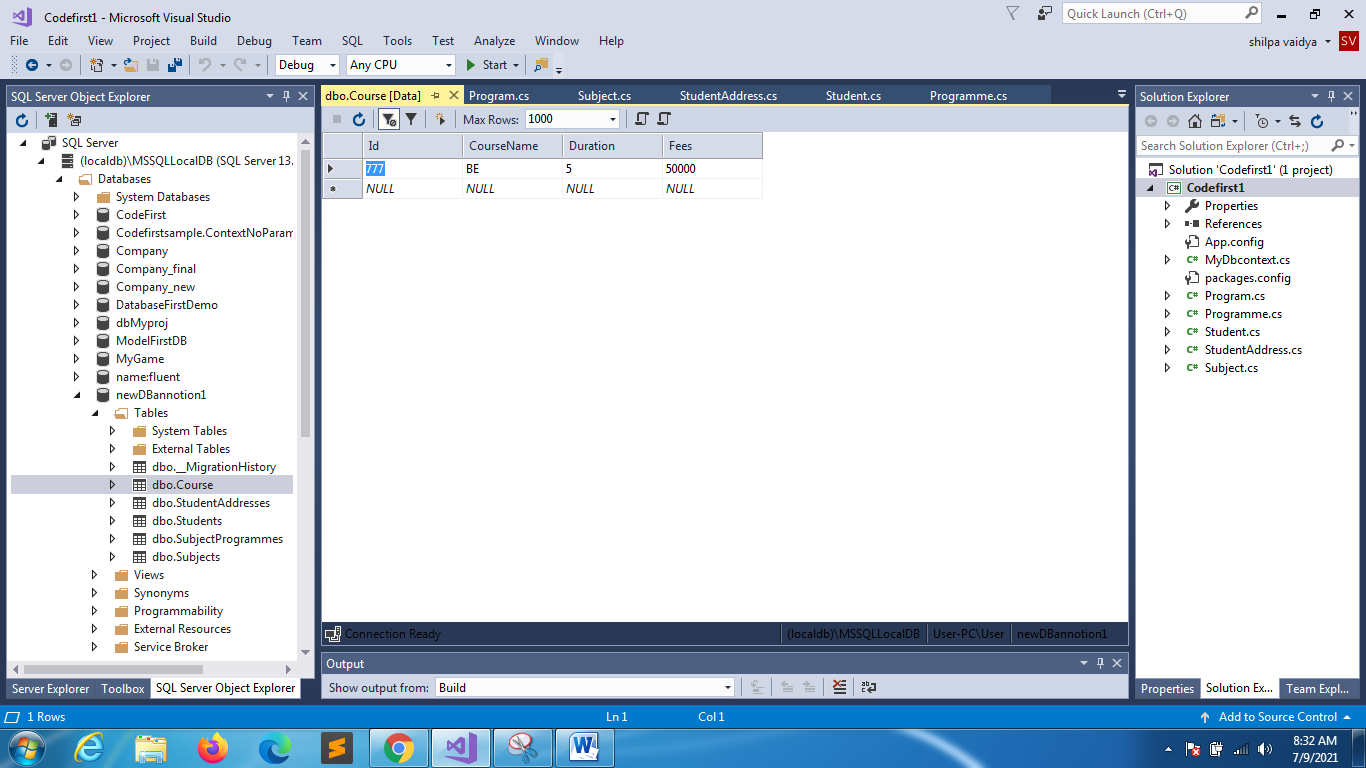
public virtual DbSet<StudentAddress> StudentAddresses { get; set; }

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

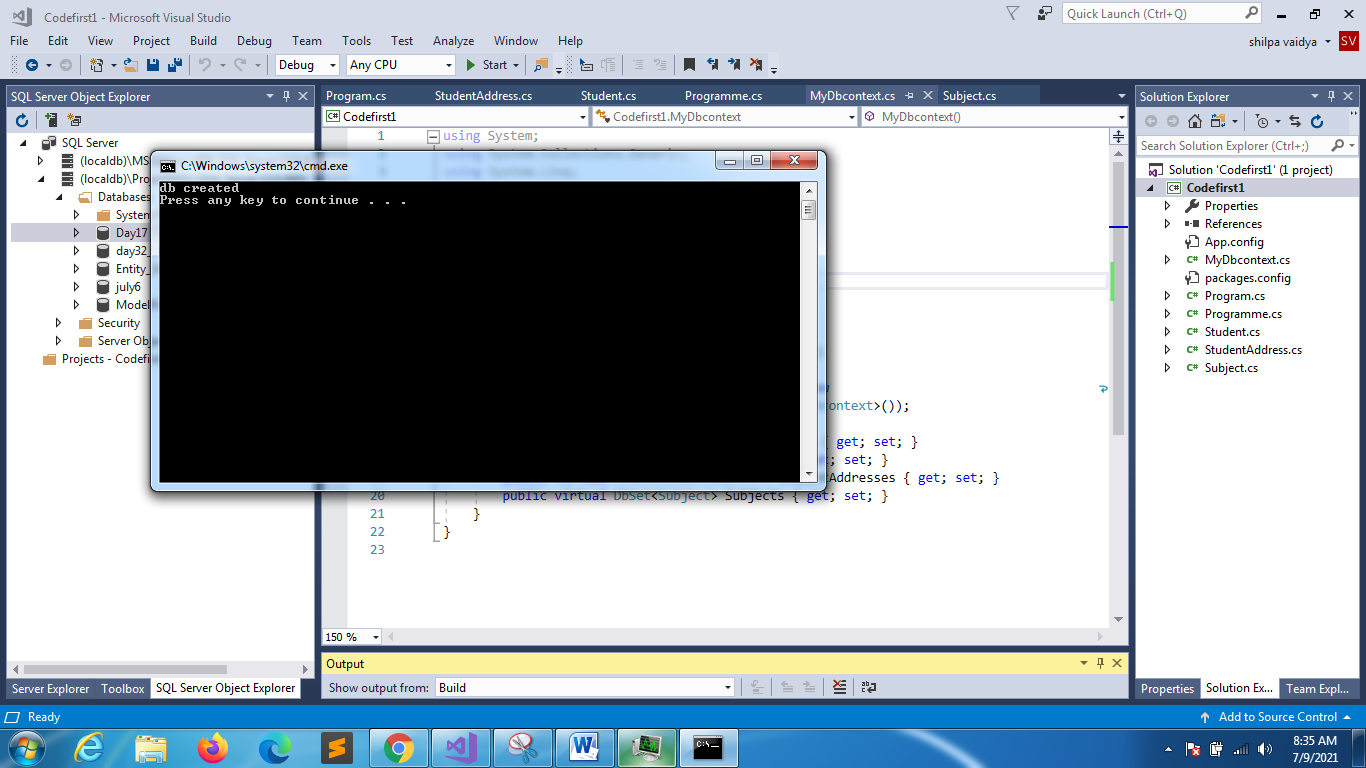
}

}

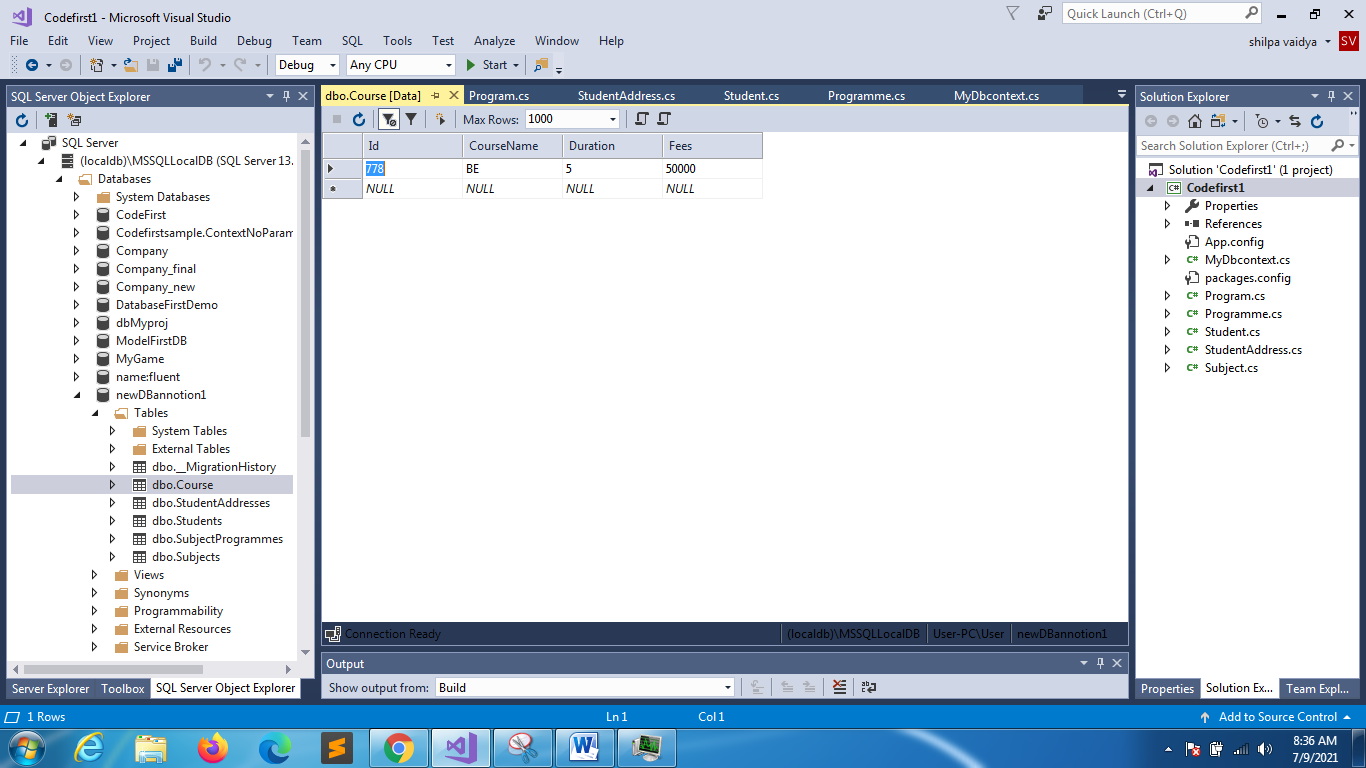
Before executing



After executing



PREVIOUS DATA IS LOST



**DropCreateDatabaseAlways**

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; }

}

}

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 = 779;

p.Title = "BA";

p.Duration = 5;

p.Fees = 20000;

db.Programmes.Add(p);

db.SaveChanges();

Console.WriteLine("db created");

}

catch(Exception ob)

{

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

}

}

}

}