Step 1 :

**Install packages**

Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.SqlServer

Microsoft.EntityFrameworkCore.Tools

**Step 2:**

internal class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Exp { get; set; }

public string Dept { get; set; }

}

**Step 3:**

namespace EFCoreDemo.DataContext

{

internal class EmployeeDbContext : DbContext

{

public EmployeeDbContext() { }

public DbSet<Employee> Employees { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder builder)

{

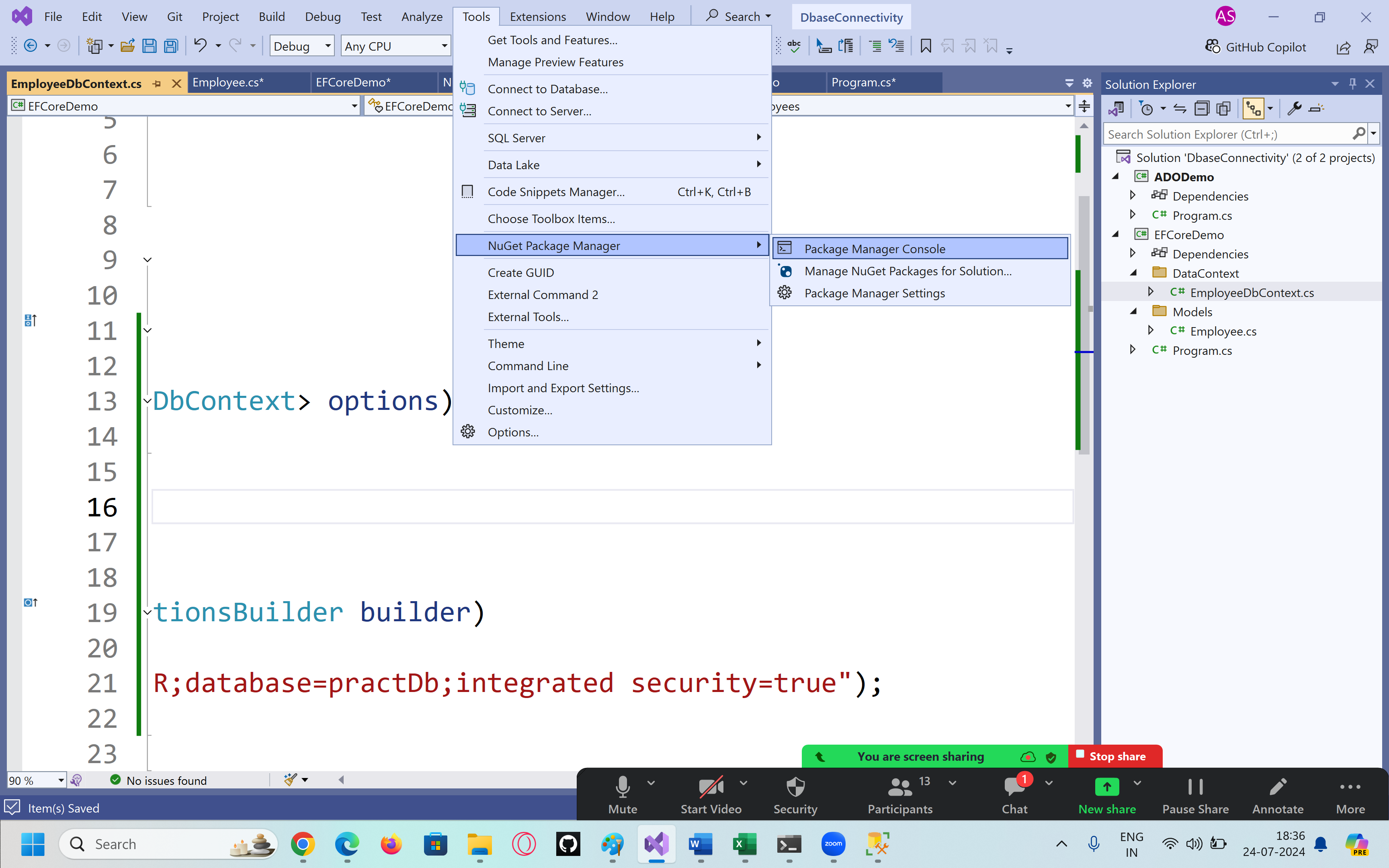
builder.UseSqlServer("server=ANAMIKA\\SQLSERVER;database=pract1Db;integrated security=true;TrustServerCertificate=True");

}

}

}

**Step 4 :**



add-migration "initial structure"

this will add Migrations folder with file named as timestamp initial\_structure

**this class basically tells what need to be done at backend**

**update-database**

**this command will run the statements written in** timestamp initial\_structure at backend

Program.cs file

using EFCoreDemo.DataContext;

using EFCoreDemo.Models;

namespace EFCoreDemo

{

internal class Program

{

static void Main(string[] args)

{

// perform CRUD operations

// C > Create / Insert

// R > Read / Get Data

// U > Update

// D > Delete

// We need instance of DbContext

EmployeeDbContext employeeDbContext = new EmployeeDbContext();

// it become mediator between collection & the table in the backend

// we will use LINQ

// Get records

//var employees = (from x in employeeDbContext.Employees

// select x).ToList();

var employees = employeeDbContext.Employees.ToList();

Console.WriteLine("Employees List");

if (employees.Count() > 0)

{

foreach (var employee in employees)

{

Console.WriteLine(employee.Name + " " + employee.Id + " " + employee.Dept + " " + employee.Exp);

}

}

else

{

Console.WriteLine("there are no records");

}

// add record

// object initializer by using properties

Employee employee1 = new Employee()

{

Name = "Vijay",

Dept = "HR",

Exp = 9

};

employeeDbContext.Employees.Add(employee1);

employeeDbContext.SaveChanges();

// update record

//Employee empToChange = employeeDbContext.Employees

// .FirstOrDefault(x => x.Id == 1);

foreach (var employee in employees)

{

if (employee.Id == 2)

{

employee.Name = "new name";

}

}

employeeDbContext.SaveChanges();

// delete record

//Employee empToDelete = employeeDbContext.Employees.FirstOrDefault(x => x.Id == 1);

//foreach (var employee in employees)

//{

// if (employee.Id == 1)

// {

// employeeDbContext.Employees.Remove(employee);

// }

// employeeDbContext.SaveChanges();

//}

}

}

}