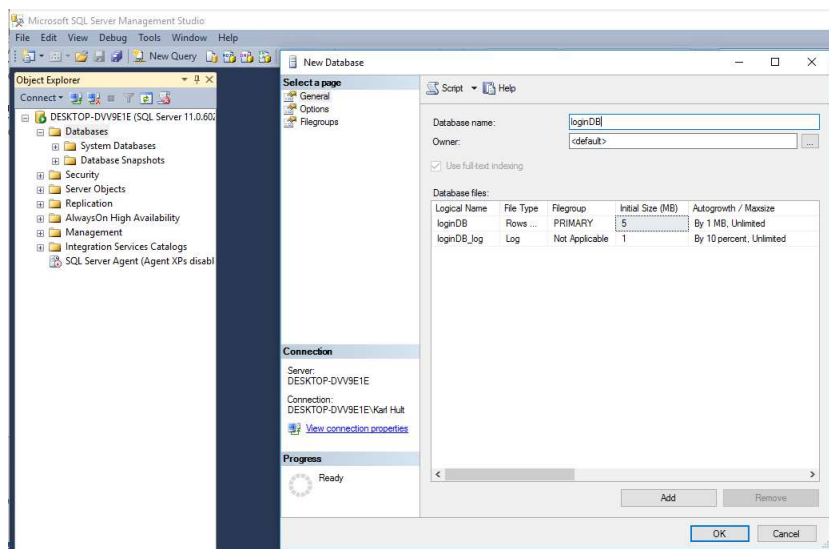


Övning MS SQL och MVC del 1

Denna övning, som är uppdelad i två delar, kommer att visa hur du kan arbeta med MS SQL och MVC. Applikationen är ett inloggningsystem som innehåller de flesta saker som vi har gått igenom under MVC-avsnittet.

Arbetsuppgift 1: Skapa databas i MS SQL

Steg 1: Öppna SQL Server Management Studio, skapa databas genom att högerklicka på katalog med namnet Databases, välj New Database.



Steg 2: Namnge databasen till loginDB, klicka på OK.

Arbetsuppgift 2: Skapa tabeller i databasen

Vi skall skapa fyra stycken tabeller i databasen, LOOKUPRole, SYSUser, SYSUsersProfile och SYSUsersProfile.

Steg 1: Klicka på New Query, alternativt kan du klicka på CTRL+N, skriv in följande query, för att skapa LOOKUPRole:

```
USE [loginDB]

GO

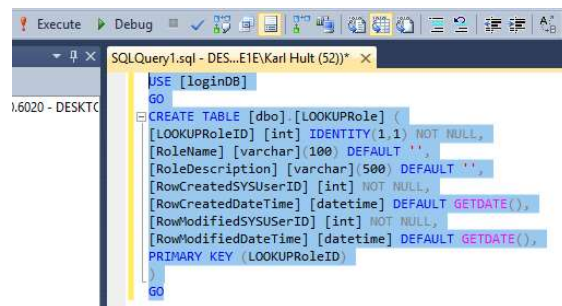
CREATE TABLE [dbo].[LOOKUPRole] (
  [LOOKUPRoleID] [int] IDENTITY(1,1) NOT NULL,
```

```

[RoleName] [varchar](100) DEFAULT '',
[RoleDescription] [varchar](500) DEFAULT '',
[RowCreatedSYSUserID] [int] NOT NULL,
[RowCreatedDateTime] [datetime] DEFAULT GETDATE(),
[RowModifiedSYSUserID] [int] NOT NULL,
[RowModifiedDateTime] [datetime] DEFAULT GETDATE(),
PRIMARY KEY (LOOKUPRoleID)
)
GO

```

Markera query du har skrivit in, klicka på Execute.



Steg 2: För att lägga till lite grunddata i tabell LOOKUPRole, skriv in följande query:

```

INSERT INTO LOOKUPRole
(RoleName, RoleDescription, RowCreatedSYSUserID, RowModifiedSYSUserID)
VALUES ('Admin', 'Can Edit, Update, Delete', 1, 1)

INSERT INTO LOOKUPRole
(RoleName, RoleDescription, RowCreatedSYSUserID, RowModifiedSYSUserID)
VALUES ('Member', 'Read only', 1, 1)

```

Markera query du har skrivit in, klicka på Execute.

LOOKUPRoleID	RoleName	RoleDescription	RowCreatedSY...	RowCreatedDa...	RowModifiedS...	RowModified...
1	Admin	Can Edit, Updat...	1	2017-06-08 20:3...	1	2017-06-08 20:3...
2	Member	Read only	1	2017-06-08 20:3...	1	2017-06-08 20:3...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Steg 3: Kör följande query för att skapa tabellen SYSUser:

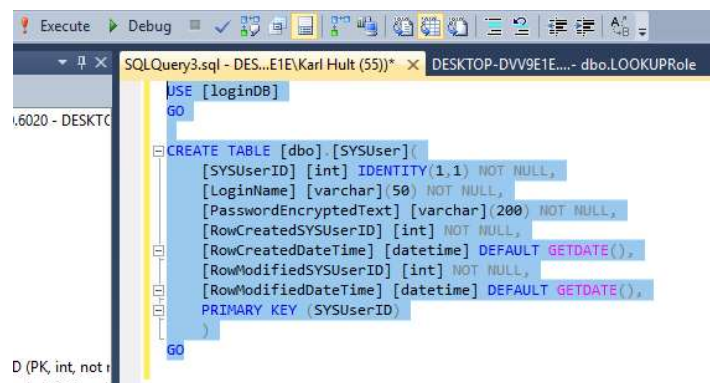
```
USE [loginDB]

GO

CREATE TABLE [dbo].[SYSUser] (
  [SYSUserID] [int] IDENTITY(1,1) NOT NULL,
  [LoginName] [varchar](50) NOT NULL,
  [PasswordEncryptedText] [varchar](200) NOT NULL,
  [RowCreatedSYSUserID] [int] NOT NULL,
  [RowCreatedDateTime] [datetime] DEFAULT GETDATE(),
  [RowModifiedSYSUserID] [int] NOT NULL,
  [RowModifiedDateTime] [datetime] DEFAULT GETDATE(),
  PRIMARY KEY (SYSUserID)
)

GO
```

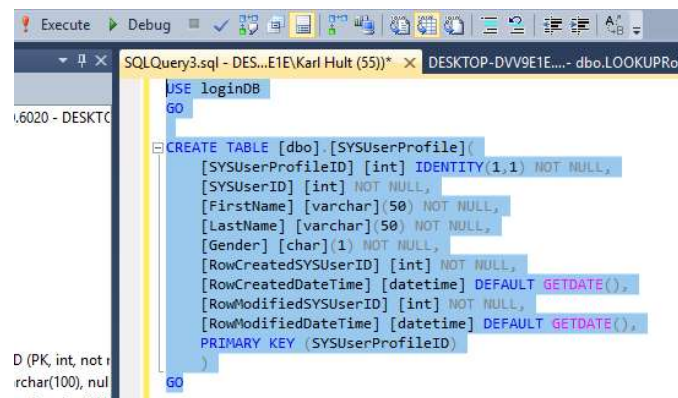
Markera query du har skrivit in, klicka på Execute.



Steg 4: Kör följande query för att skapa tabellen SYSUsersProfile:

```
SYSUserProfile table  
  
USE [loginDB]  
  
GO  
  
CREATE TABLE [dbo].[SYSUserProfile] (  
  
[SYSUserProfileID] [int] IDENTITY(1,1) NOT NULL,  
  
[SYSUserID] [int] NOT NULL,  
  
[FirstName] [varchar](50) NOT NULL,  
  
[LastName] [varchar](50) NOT NULL,  
  
[Gender] [char](1) NOT NULL,  
  
[RowCreatedSYSUserID] [int] NOT NULL,  
  
[RowCreatedDateTime] [datetime] DEFAULT GETDATE(),  
  
[RowModifiedSYSUserID] [int] NOT NULL,  
  
[RowModifiedDateTime] [datetime] DEFAULT GETDATE(),  
  
PRIMARY KEY (SYSUserProfileID)  
  
)  
  
GO
```

Markera query du har skrivit in, klicka på Execute.

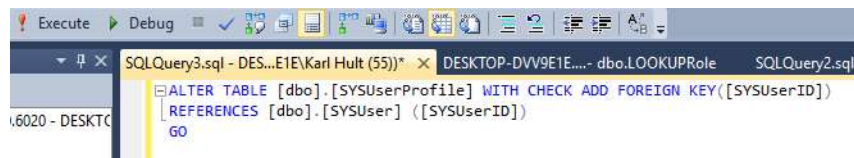


The screenshot shows a SQL Server Enterprise Manager interface. The main window displays a query window titled 'SQLQuery3.sql - DES...E1E\Karl Hult (55)*'. The query text is the same as provided in the previous block, with the following lines highlighted in blue: 'USE loginDB', 'GO', 'CREATE TABLE [dbo].[SYSUserProfile]', '[SYSUserProfileID] [int] IDENTITY(1,1) NOT NULL,', '[SYSUserID] [int] NOT NULL,', '[FirstName] [varchar](50) NOT NULL,', '[LastName] [varchar](50) NOT NULL,', '[Gender] [char](1) NOT NULL,', '[RowCreatedSYSUserID] [int] NOT NULL,', '[RowCreatedDateTime] [datetime] DEFAULT GETDATE(),', '[RowModifiedSYSUserID] [int] NOT NULL,', '[RowModifiedDateTime] [datetime] DEFAULT GETDATE(),', 'PRIMARY KEY (SYSUserProfileID)', and ')'. The 'Execute' button is visible at the top left of the query window.

Steg 5: För att länka ihop två tabeller SYSUserProfile och SYSUserID använder vi Foreign Key, kör följande query:

```
ALTER TABLE [dbo].[SYSUserProfile] WITH CHECK ADD FOREIGN KEY([SYSUserID])
REFERENCES [dbo].[SYSUser] ([SYSUserID])
GO
```

Markera query du har skrivit in, klicka på Execute.



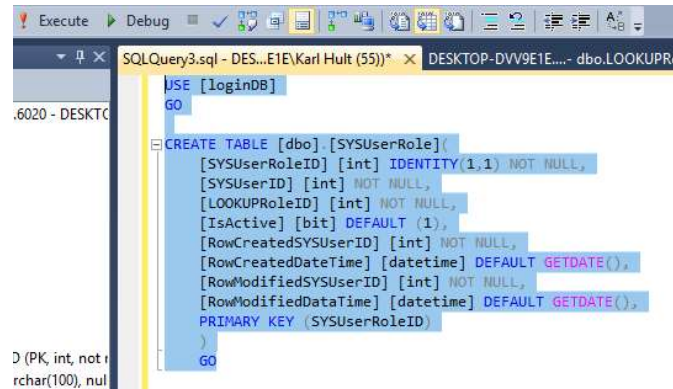
Steg 6: Kör följande query för att skapa tabellen SYSUsersRole:

```
USE [loginDB]
GO
CREATE TABLE [dbo].[SYSUserRole] (
  [SYSUserRoleID] [int] IDENTITY(1,1) NOT NULL,
  [SYSUserID] [int] NOT NULL,
  [LOOKUPRoleID] [int] NOT NULL,
  [IsActive] [bit] DEFAULT (1),
  [RowCreatedSYSUserID] [int] NOT NULL,
  [RowCreatedDateTime] [datetime] DEFAULT GETDATE(),
  [RowModifiedSYSUserID] [int] NOT NULL,
  [RowModifiedDateTime] [datetime] DEFAULT GETDATE(),
  PRIMARY KEY (SYSUserRoleID)
```

)

GO

Markera query du har skrivit in, klicka på Execute.



```
USE [loginDB]
GO
CREATE TABLE [dbo].[SYSUserRole](
  [SYSUserRoleID] [int] IDENTITY(1,1) NOT NULL,
  [SYSUserID] [int] NOT NULL,
  [LOOKUPRoleID] [int] NOT NULL,
  [IsActive] [bit] DEFAULT (1),
  [RowCreatedSYSUserID] [int] NOT NULL,
  [RowCreatedDateTime] [datetime] DEFAULT GETDATE(),
  [RowModifiedSYSUserID] [int] NOT NULL,
  [RowModifiedDateTime] [datetime] DEFAULT GETDATE(),
  PRIMARY KEY (SYSUserRoleID)
)
GO
```

Steg 7: För att länka ihop tabellerna använder vi Foreign Key, kör följande query:

```
ALTER TABLE [dbo].[SYSUserRole] WITH CHECK ADD FOREIGN KEY([LOOKUPRoleID])
```

```
REFERENCES [dbo].[LOOKUPRole] ([LOOKUPRoleID])
```

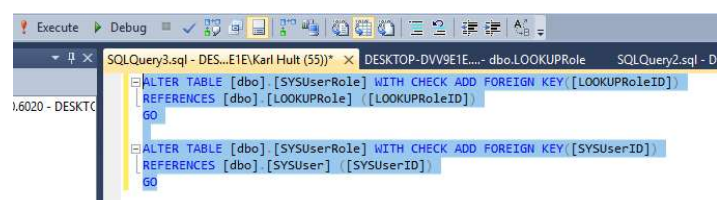
```
GO
```

```
ALTER TABLE [dbo].[SYSUserRole] WITH CHECK ADD FOREIGN KEY([SYSUserID])
```

```
REFERENCES [dbo].[SYSUser] ([SYSUserID])
```

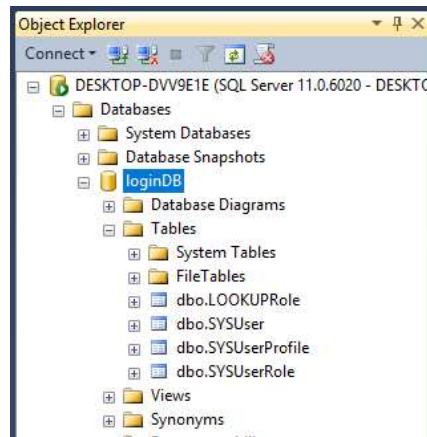
```
GO
```

Markera query du har skrivit in, klicka på Execute.



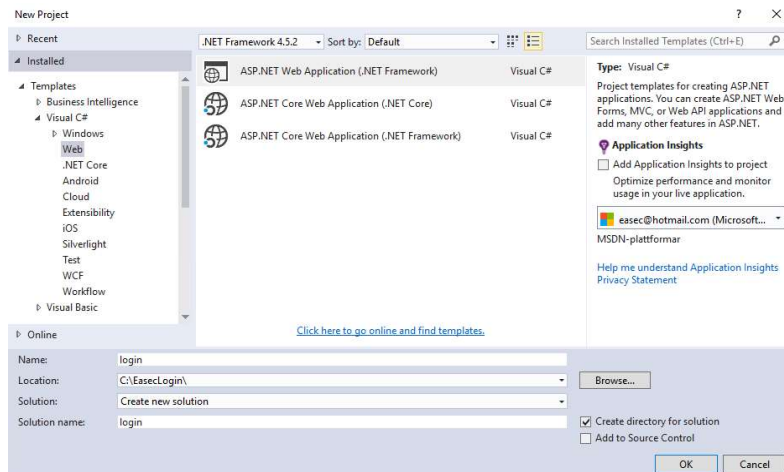
```
ALTER TABLE [dbo].[SYSUserRole] WITH CHECK ADD FOREIGN KEY([LOOKUPRoleID])
REFERENCES [dbo].[LOOKUPRole] ([LOOKUPRoleID])
GO
ALTER TABLE [dbo].[SYSUserRole] WITH CHECK ADD FOREIGN KEY([SYSUserID])
REFERENCES [dbo].[SYSUser] ([SYSUserID])
GO
```

Nu har vi skapat de fyra tabellerna med relationer.



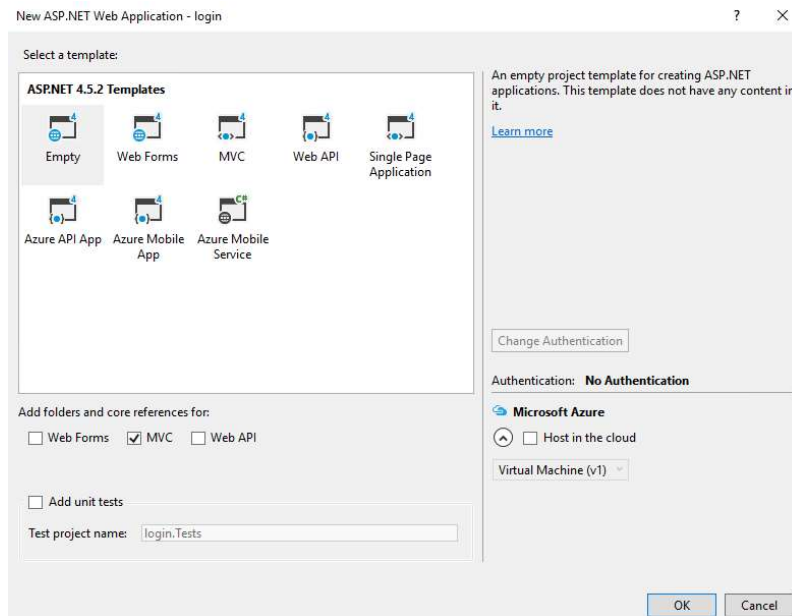
Arbetsuppgift 2: Skapa MVC-applikation

Steg 1: Starta Visual Studio.



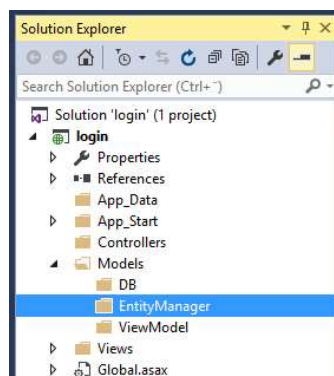
Steg 2: Välj File – New – Project, i dialogfönstret för New Project välj Templates – ASP.NET Web Application.

Namnge ditt projekt till login, klicka på OK.



Steg 3: Välj Empty under Templates, klicka i boxruta till vänster om MVC, verifiera att inte Microsoft Azure är valt. Klicka på OK.

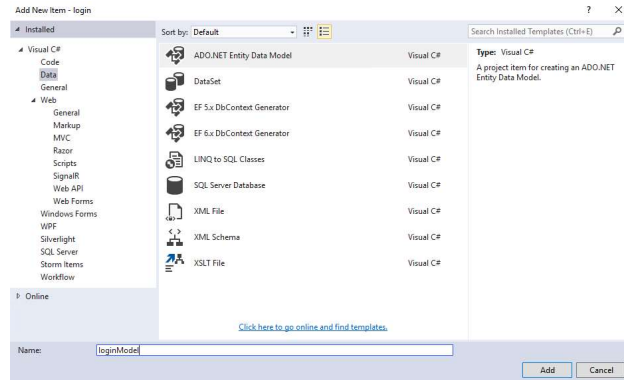
Arbetsuppgift 3: Skapa Entity Models



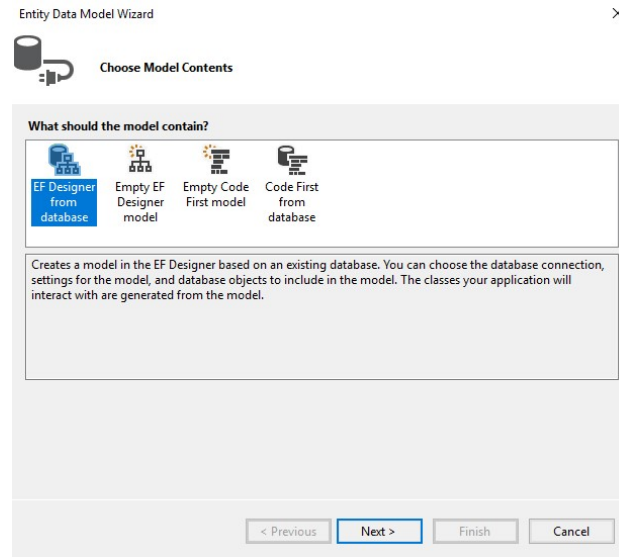
Steg 1: Högerklicka på katalogen Model, välj Add – New Folder, lägg till DB.

Skapa på samma sätt katalogerna EntityManager och ViewModel.

I katalogen DB kommer vi att lagra vår entity model (.EDMX).



Steg 2: Lägg till entity, genom att högerklicka på katalogen DB, välj sedan Add – New Item – Data – ADO.NET Entity Data Model. Namnge Model till loginModel, klicka på Add.



Steg 3: Markera EF Designer from database, klicka på Next.

Steg 4: Klicka på knappen New Connection.

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
DESKTOP-DVV9E1E Refresh

Log on to the server
Authentication: Windows Authentication

User name:
Password:
 Save my password

Connect to a database
 Select or enter a database name:
loginDB
 Attach a database file:
Browse...
Logical name:

Advanced...

Test Connection OK Cancel

Steg 5: Verifiera att Microsoft SQL Server (SqlClient) är vald, klicka på Refresh, markera din dator under Server name.

Välj loginDB under Select or enter a database name.

Klicka på Test Connection. Klicka på OK för att bekräfta att kommunikation fungerade. Klicka på OK.

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?
desktop-dvv9e1e.loginDB.dbo New Connection...

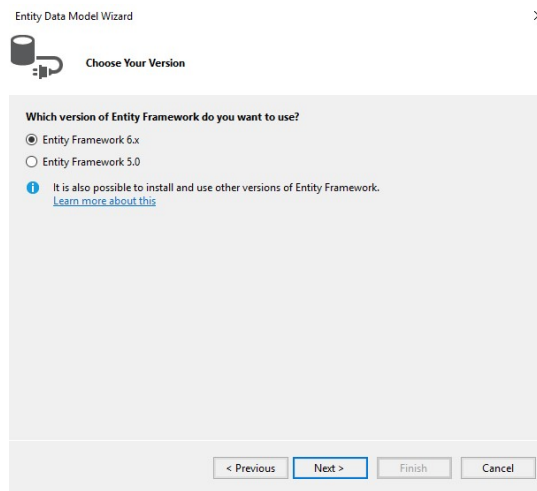
This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?
 No, exclude sensitive data from the connection string. I will set it in my application code.
 Yes, include the sensitive data in the connection string.

Connection string:
 metadata=res://*/Models.DB.loginModel.csdl|res://*/Models.DB.loginModel.csdl|res://*/Models.DB.loginModel.msl|provider=System.Data.SqlClient;provider connection string="data source=DESKTOP-DVV9E1E;initial catalog=loginDB;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

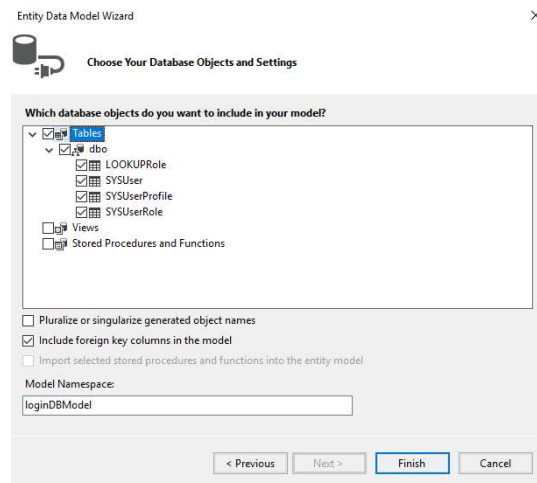
Save connection settings in Web.Config as:
loginDBEntities

< Previous Next > Finish Cancel

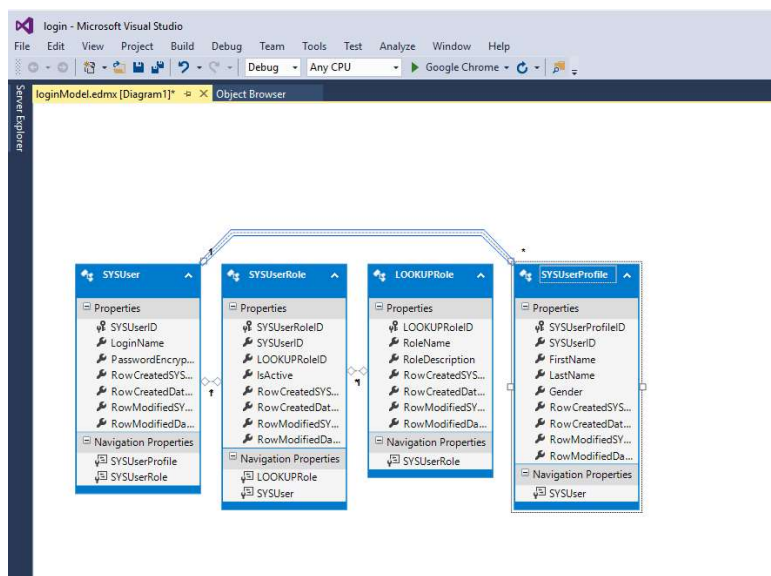
Anslutning genereras automatiskt, klicka på Next.



Steg 6: Verifiera att Entity Framework 6.x är vald, klicka på Next.

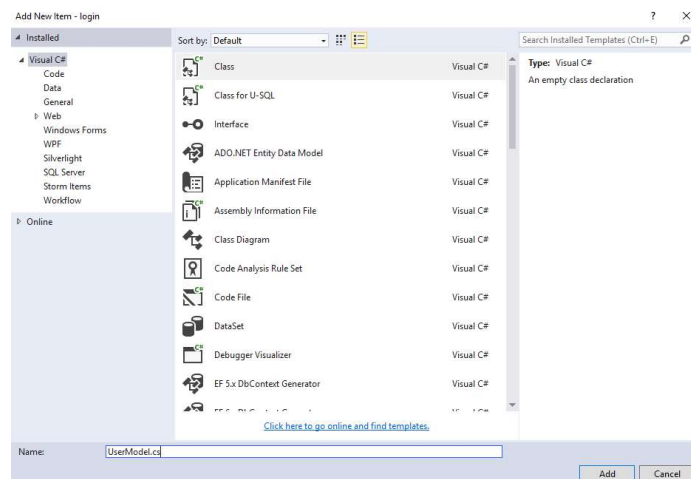


Steg 7: Klicka i dbo, expandera för att kontrollera att boxrutor för de fyra tabellerna är ikryssade. Klicka på Finish.



Arbetsuppgift 4: Skapa klass för kommande view

För att skapa en sign-up view längre fram, lägger du till ViewModel. Entity Framework kommer att generera objekt och hantera tillgång till informationen inom applikationen. EF skapar klasserna LOOKUPRole, SYSUserRole, SYSUser och SYSUserProfile, dessa innehåller alla fälten från tabellerna, som egenskaper för varje klass. Om du inte vill använda dessa klasser direkt i en view, skapar du en separat klass för de egenskaper som du vill använda i view.



Steg 1: Högerklicka på katalogen ViewModel, välj sedan Add – Class. Skriv in UserModel.cs i rutan för namn, klicka på Add.

```

login
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;

namespace login.Models.ViewModel
{
    public class UserSignUpView
    {
        [Key]
        public int SYSUserID { get; set; }
        public int LOOKUPRoleID { get; set; }
        public string RoleName { get; set; }
        [Required(ErrorMessage = "e*")]
        [Display(Name = "Login ID")]
        public string LoginName { get; set; }
        [Required(ErrorMessage = "e*")]
        [Display(Name = "Password")]
        public string Password { get; set; }
        [Required(ErrorMessage = "e*")]
        [Display(Name = "Förnamn")]
        public string FirstName { get; set; }
        [Required(ErrorMessage = "e*")]
        [Display(Name = "Efternamn")]
        public string LastName { get; set; }
        public string Gender { get; set; }
    }
}

```

Steg 2: Skriv in programkod enligt bild ovan.

Arbetsuppgift 5: Lägga till klassen UserManager

Nästa steg är att skapa klassen UserManager. Denna klass skall hantera CRUD-operationer på specifik tabell.

Steg 1: Högerklicka på katalogen EntityManager, klicka på Add – Class, namnge klasser till userManager.cs. Klicka på Add.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using login.Models.DB;
using login.Models.ViewModel;

namespace login.Models.EntityManager
{
    public class UserManager
    {
        public void AddUserAccount(UserSignUpView user)
        {
            using (loginDBEntities db = new loginDBEntities())
            {
                SYSUser SU = new SYSUser();
                SU.LoginName = user.LoginName;
                SU.PasswordEncryptedText = user.Password;
                SU.RowCreatedSYSUserID = user.SYSUserID > 0 ? user.SYSUserID : 1;
                SU.RowModifiedSYSUserID = user.SYSUserID > 0 ? user.SYSUserID : 1;
                SU.RowCreatedDateTime = DateTime.Now;
                SU.RowModifiedDateTime = DateTime.Now;

                db.SYSUser.Add(SU);
                db.SaveChanges();

                SYSUserProfile SUP = new SYSUserProfile();
                SUP.SYSUserID = SU.SYSUserID;
                SUP.FirstName = user.FirstName;
                SUP.LastName = user.LastName;
                SUP.Gender = user.Gender;
                SUP.RowCreatedSYSUserID = user.SYSUserID > 0 ? user.SYSUserID : 1;
                SUP.RowModifiedSYSUserID = user.SYSUserID > 0 ? user.SYSUserID : 1;
                SUP.RowCreatedDateTime = DateTime.Now;

                db.SYSUserProfile.Add(SUP);
                db.SaveChanges();

                if (user.LOOKUPRoleID > 0)
                {
                    SYSUserRole SUR = new SYSUserRole();
                    SUR.LOOKUPRoleID = user.LOOKUPRoleID;
                    SUR.SYSUserID = user.SYSUserID;
                    SUR.IsActive = true;
                    SUR.RowCreatedSYSUserID = user.SYSUserID > 0 ? user.SYSUserID : 1;
                    SUR.RowModifiedSYSUserID = user.SYSUserID > 0 ? user.SYSUserID : 1;
                    SUR.RowCreatedDateTime = DateTime.Now;
                    SUR.RowModifiedDataTime = DateTime.Now;

                    db.SYSUserRole.Add(SUR);
                    db.SaveChanges();
                }
            }
        }
    }
}
```

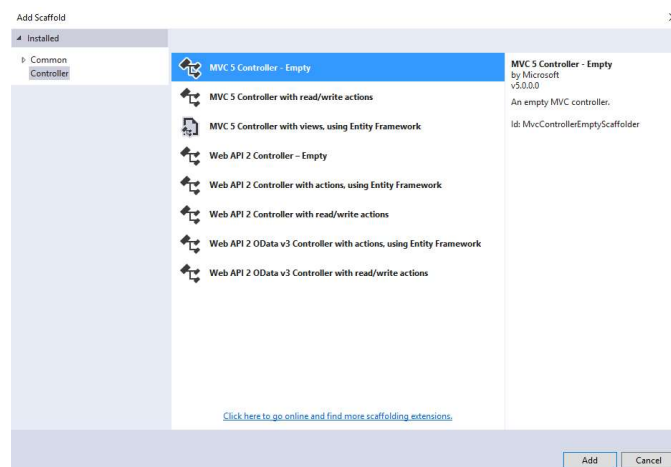
Steg 2: Skriv in programkod enligt ovan.

Steg 3: Klicka på Return två gånger.

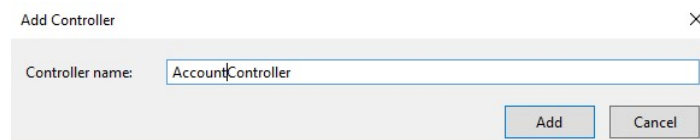
```
public bool IsLoginNameExist(string loginName)
{
    using (loginDBEntities db = new loginDBEntities())
    {
        return db.SYSUser.Where(o => o.LoginName.Equals(loginName)).Any();
    }
}
}
```

Steg 4: Skriv in programkod enligt ovan.

Arbetsuppgift 6: Skapa klassen AccountController (Controller)



Steg 1: Högerklicka på katalogen Controllers, välj Add – Controller – MVC 5 Controller – Empty, klicka på Add.



Steg 2: Namnge controller till AccountController.

Steg 3: Klicka på Add, för att skapa klassen.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using login.Models.ViewModel;
using login.Models.EntityManager;

namespace login.Controllers
{
    public class AccountController : Controller
    {
        // GET: Account
        public ActionResult SignUp()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Signup(UserSignUpView USV)
        {
            if (ModelState.IsValid)
            {
                UserManager UM = new UserManager();
                if (!UM.IsLoginNameExist(USV.LoginName))
                {
                    UM.AddUserAccount(USV);
                    FormsAuthentication.SetAuthCookie(USV.FirstName, false);
                    return RedirectToAction("Welcome", "Home");
                }
                else
                {
                    ModelState.AddModelError("", "Inloggningsnamn redan tagit.");
                }
            }
            return View();
        }
    }
}

```

Steg 4: Skriv in programkod enligt ovan.

Arbetsuppgift 7: Skapa klassen HomeController (Controller)

Steg 1: Högerklicka på katalogen Controllers, välj Add – Controller – MVC 5 Controller – Empty, klicka på Add.

Steg 2: Namnge controller till HomeController.

Steg 3: Klicka på Add, för att skapa klassen.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

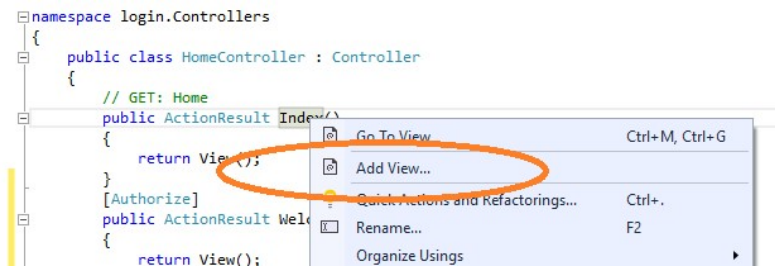
namespace login.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home
        public ActionResult Index()
        {
            return View();
        }

        [Authorize]
        public ActionResult Welcome()
        {
            return View();
        }
    }
}

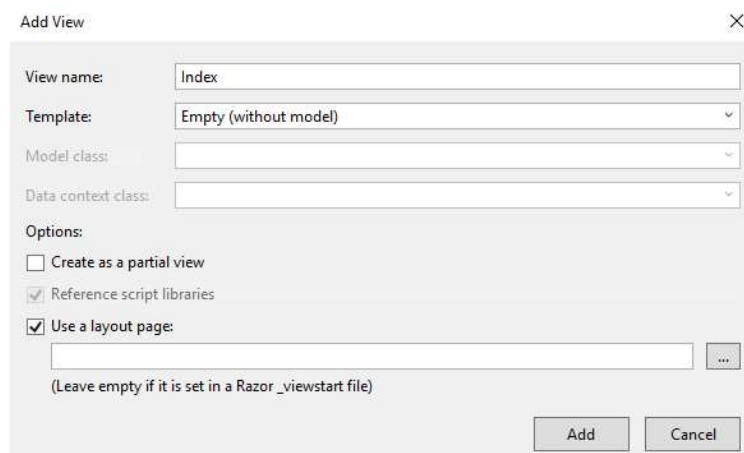
```

Steg 4: Skriv in programkod enligt ovan.

Arbetsuppgift 8: Skapa view



Steg 1: I HomeController, markera Index och högerklicka. Välj Add View i dialogruta som dyker upp.



Steg 2: Klicka på Add.



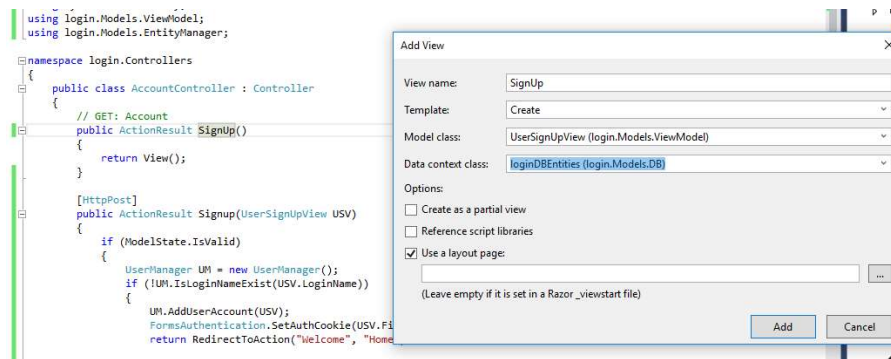
Steg 3: Skriv in programkod enligt ovan.

Steg 4: I HomeController, markera Welcome och högerklicka. Välj Add View i dialogruta som dyker upp. Klicka på Add.



Steg 4: Skriv in programkod enligt ovan.

Steg 5: Skifta till AccountController.



Steg 6: Högerklicka på SignUp. Välj Add View i dialogruta som dyker upp. Klicka på Add. Välj Create i Template, UserSignUpView (login.Models.ViewModel) för Model class och loginDBEntities (login.models.DB) för Data context class.

Klicka på Add.

```

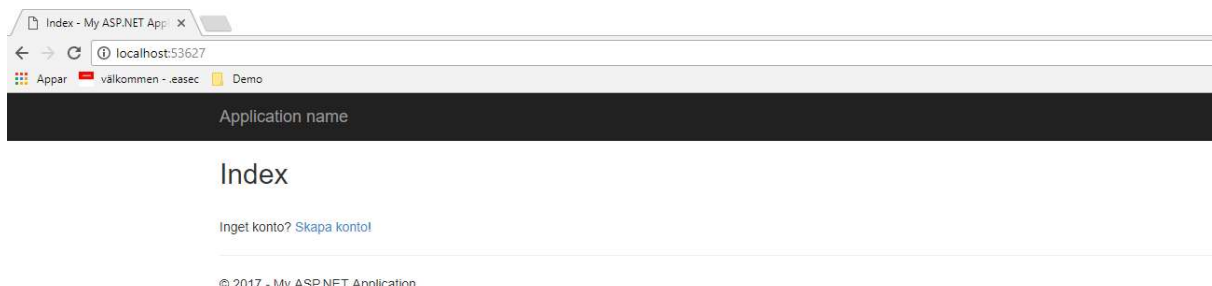
<div>
    @Html.ActionLink("Back to Main", "Index", "Home")
</div>

```

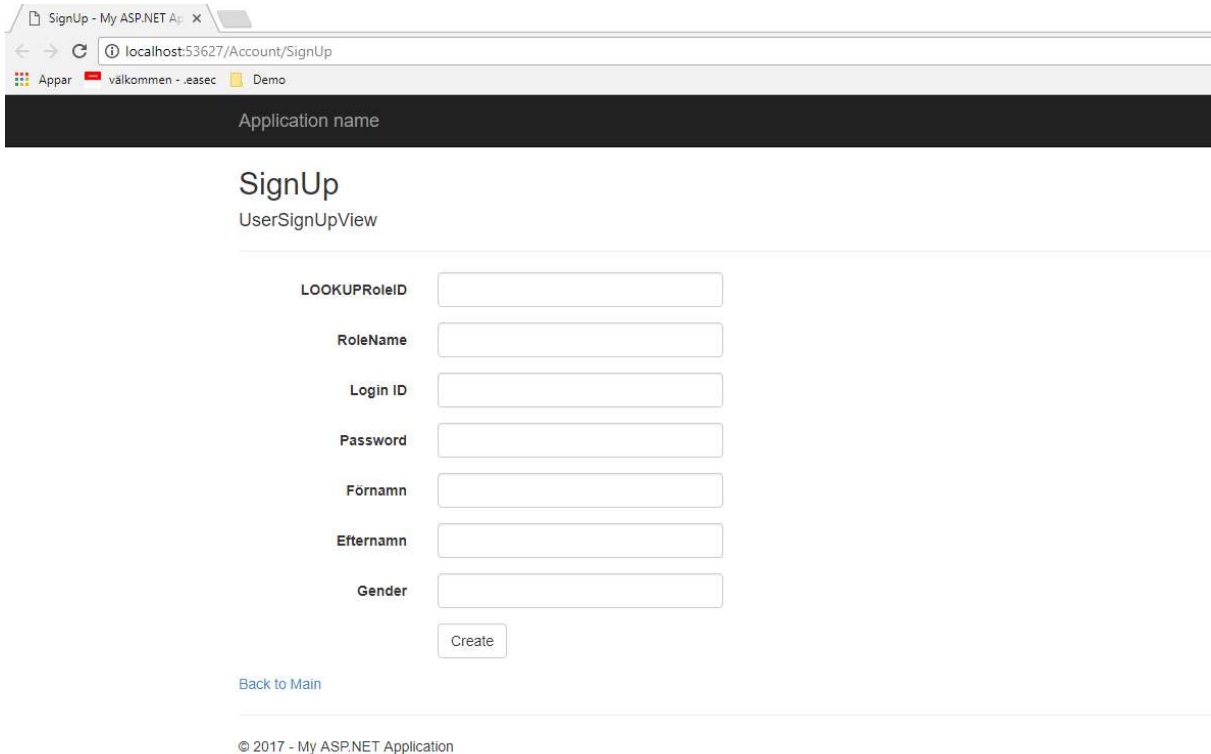
Steg 7: Nederst i denna fil, ändra länken enligt ovan.

Arbetsuppgift 9: Testkör

Steg 1: Klicka på Debug – Start Debugging



Steg 2: Klicka på Skapa konto!



SignUp - My ASP.NET A: X

localhost:53627/Account/SignUp

Appar välkommen - .eassec Demo

Application name

SignUp

UserSignUpView

LOOKUPRoleID

RoleName

Login ID

Password

Förnamn

Efternamn

Gender

Create

[Back to Main](#)

© 2017 - My ASP.NET Application

Steg 3: Klicka på länken Back to Main.

Steg 3: Spara ditt projekt.

Detta markerar slutet för del 1 av övningen. I nästa del så skall vi ändra och förfina SignUp-sidan, bli rensa bland fälten, lägga till dropdown-lista mm. Nu finns det bara möjlighet att lägga till, men i nästa del läggs de andra CRUD-bitarna till.