

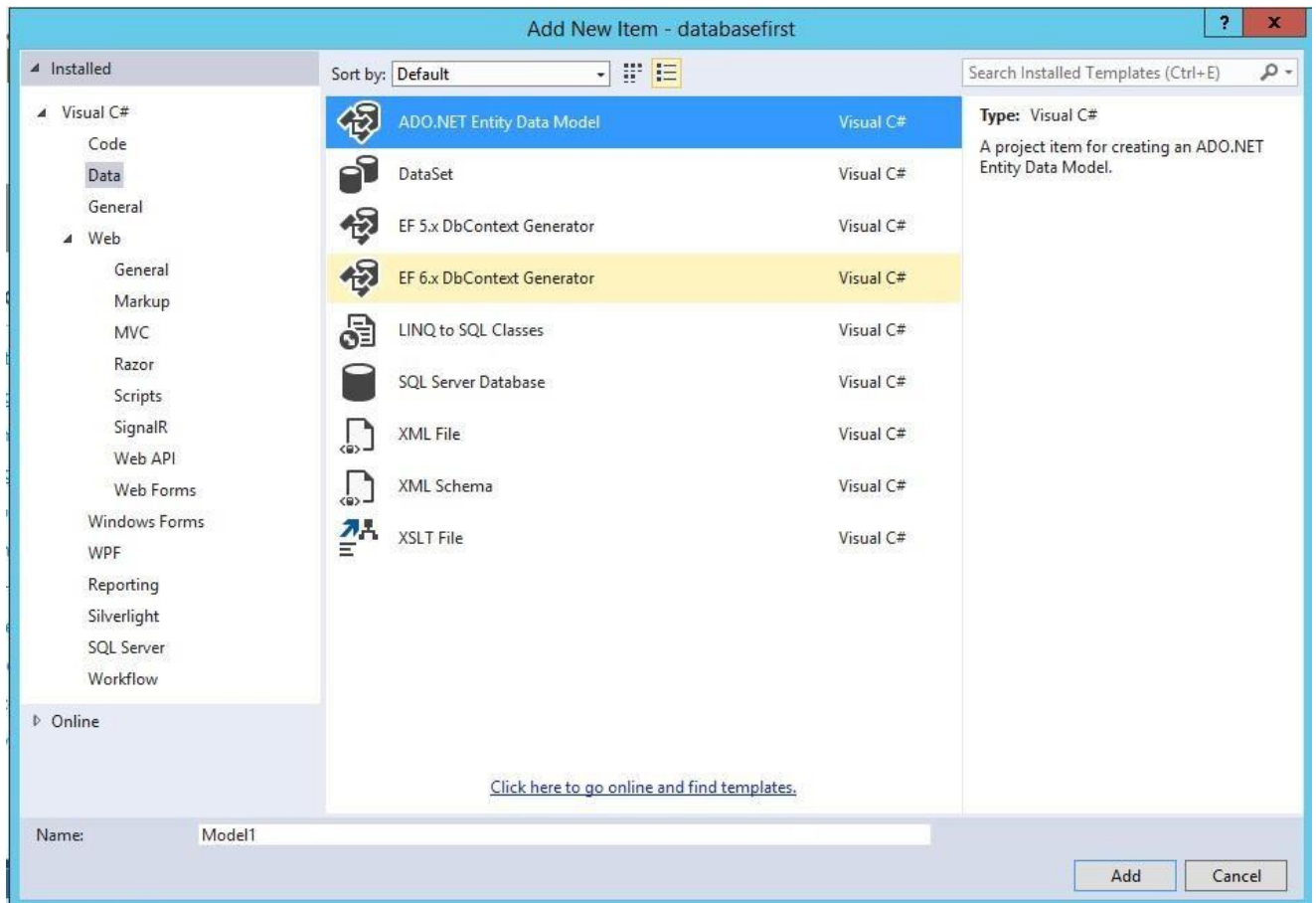


Av Erik
"Zkall"
Svensson
@esdev.se

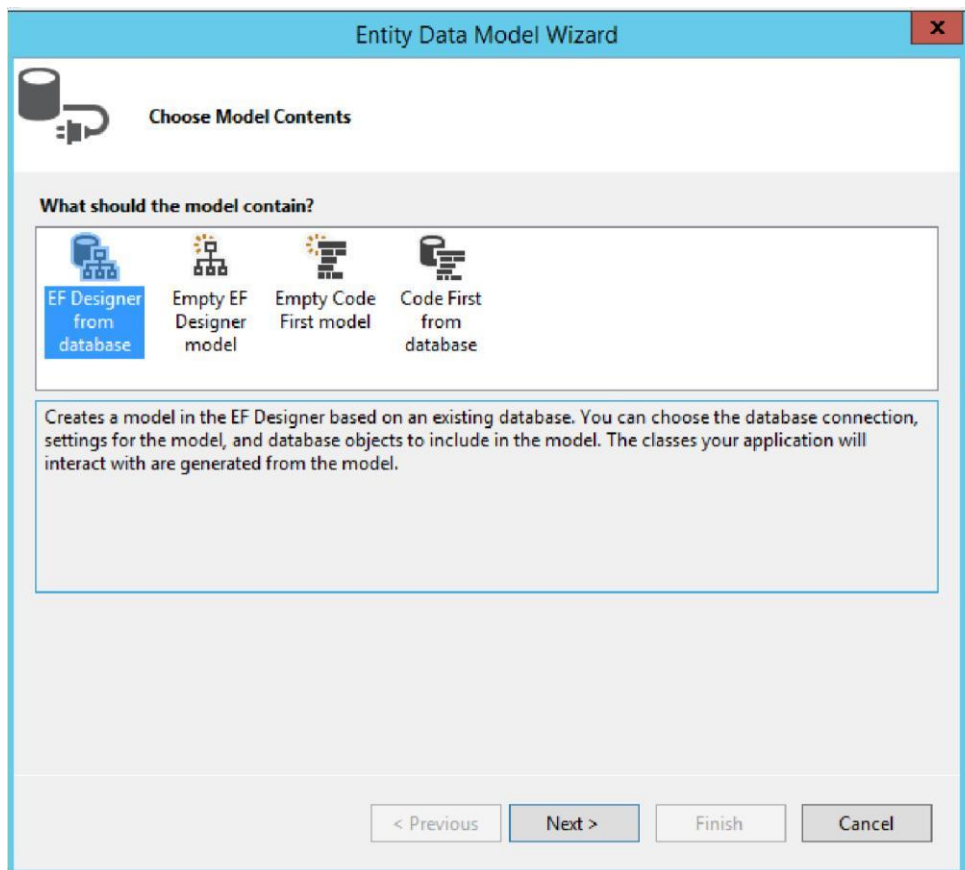


Entity Framework Database first

Högerklicka på models foldern och välj add, new item.

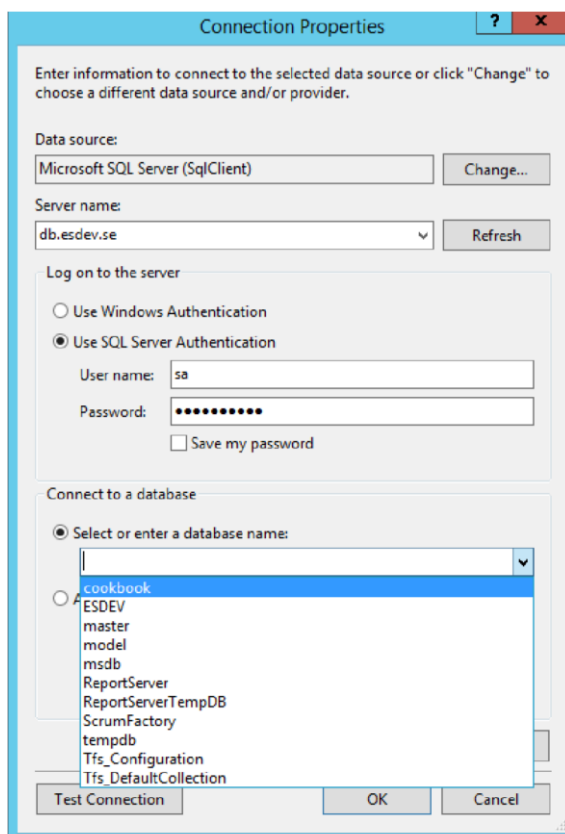


Välj ADO.NET Entity Data Model och döp den till något, t.ex DB. Klicka på Add.




Välj EF Designer from database och tryck Next.

Välj New connection och lägg till anslutningen till din databas.



Välj sedan databasen du ska skapa model för och tryck ok.

Entity Data Model Wizard

 Choose Your Data Connection

Which data connection should your application use to connect to the database?

db.cookbook.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.Model1.csdl|res://*/Models.Model1.ssdl|
res://*/Models.Model1.msl;provider=System.Data.SqlClient;provider connection string="data
source=db.esdev.se;initial catalog=cookbook;user
id=sa;password=*****;MultipleActiveResultSets=True;App=EntityFramework"
```


☒ Save connection settings in Web.Config as:

cookbookEntities

< Previous Next > Finish Cancel

Tryck yes, include och sedan Next

Entity Data Model Wizard

 Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

☒ Tables

☒ dbo

☐ Views

☐ Stored Procedures and Functions

☐ Pluralize or singularize generated object names

☒ Include foreign key columns in the model

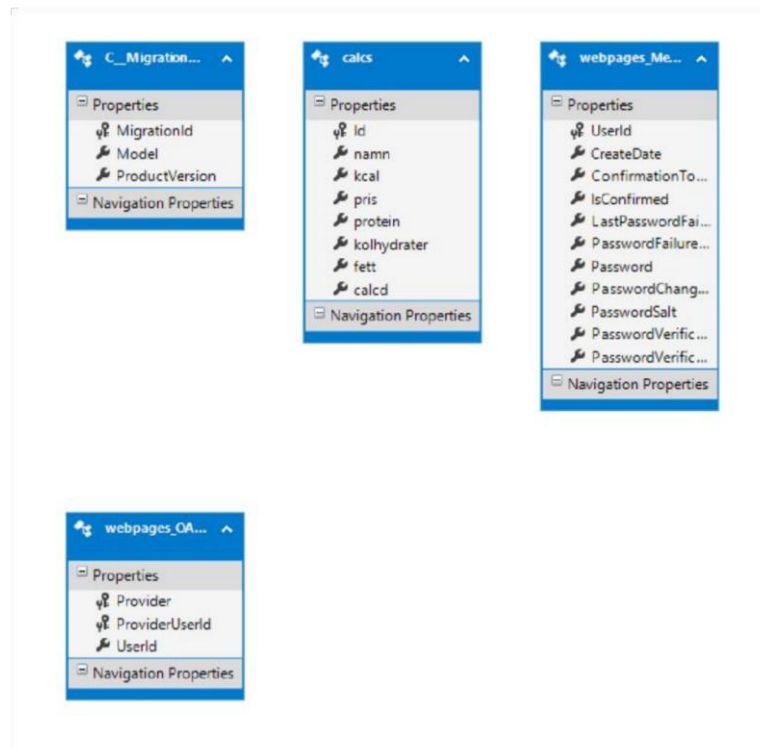
☐ Import selected stored procedures and functions into the entity model

Model Namespace:

cookbookModel

< Previous Next > Finish Cancel

Välj tables du vill göra modeller av och tryck sedan på Finish



Du möts av denna grafiska uppritning av dina modeller.

```
protected override void OnModelCreating(DbModelBuilder modelBuilder)
{
    throw new UnintentionalCodeFirstException();
}

// references
public virtual DbSet<C__MigrationHistory> C__MigrationHistory { get; set; }
// references
public virtual DbSet<calcs> calcs { get; set; }
// references
public virtual DbSet<favoriter> favoriter { get; set; }
// references
public virtual DbSet<handlingslista> handlingslista { get; set; }
// references
public virtual DbSet<ingredienser> ingredienser { get; set; }
// references
public virtual DbSet<kommentarer> kommentarer { get; set; }
// references
public virtual DbSet<maträtt_maxipicture> maträtt_maxipicture { get; set; }
// references
public virtual DbSet<maträtter> maträtter { get; set; }
// references
public virtual DbSet<maträtter_likes> maträtter_likes { get; set; }
// references
public virtual DbSet<maträtter_picture> maträtter_picture { get; set; }
// references
public virtual DbSet<user_pictures> user_pictures { get; set; }
// references
public virtual DbSet<UserProfile> UserProfile { get; set; }
// references
public virtual DbSet<webpages_Membership> webpages_Membership { get; set; }
// references
public virtual DbSet<webpages_OAuthMembership> webpages_OAuthMembership { get; set; }
// references
public virtual DbSet<webpages_Roles> webpages_Roles { get; set; }
// references
public virtual DbSet<veckolistas> veckolistas { get; set; }
}
```

Om du kollar i modellen så ser du att massor av classer har skapats och att du även har ovanstående data context skapat. En connection string i web.config har även skapats på automatik.

För att se hur man kommer åt databas modellen i koden se slutet av code first dokumentet.

Hur kommer man åt databasen från koden

Du refererar till namnet av db classen (rekommenderat namn är DB då det är rätt uppenbart när du ska deklarera vad som är vad. ,så du skriver:

```
private DB db = new DB();  
0 references  
public ActionResult Index()  
{  
    var model = db.Apor.ToList();  
    return View();  
}
```

Sedan skriver du bara db.Pluralnamn av dbset du skapade och sen db.pluralnamn.add/remote/update/ Efter du har gjort en ändring måste du skriva db.SaveChanges(); annars sparas inte ändringarna.

Lägga till en model i databasen

```
private DB db = new DB();  
0 references  
public ActionResult Index()  
{  
    var temp = new ApaModel { namn="Adolf" };  
    db.Apor.Add(temp);  
    db.SaveChanges();  
    return View();  
}
```

Id behöver du inte skriva, utan det sköts per automatik.

Uppdatera en model i databasen

```
private DB db = new DB();  
0 references  
public ActionResult Index()  
{  
    var apan = db.Apor.First(x => x.namn == "Adolf");  
    apan.namn = "Nisse";  
    db.SaveChanges();  
    return View();  
}
```

Hämta först det du ska uppdatera, ändra sedan variablerna du vill ändra, och avsluta med db.SaveChanges();

Ta bort en model i databasen

```
private DB db = new DB();
0 references
public ActionResult Index()
{
    var apan = db.Apor.First(x => x.namn == "Adolf");
    db.Apor.Remove(apan);
    db.SaveChanges();
    return View();
}
```

Samma som förra exemplet, hämta objektet du ska modifieraa. Skriv sedan db.Apor.Remove(apan); Avsluta med db.SaveChanges();

Hämta flera

```
private DB db = new DB();
0 references
public ActionResult Index()
{
    var model = db.Apor.ToList();
    return View();
}
```

.ToList() för List > IEnumerable imo :P

Hämta flera med parametrar

```
private DB db = new DB();
0 references
public ActionResult Index()
{
    var model = db.Apor.Where(x => x.namn == "Nisse").ToList();
    return View();
}
```

.Where(x => x.namn == "Nisse") hämtar alla apor som heter Nisse. **X => X == x.namn** är lambda som jag går in på mer i ett annat dokument, för nu kan ni tänka på att det är så det funkar bara.

Code first versus Database first versus Designer first

1 Code first

Bra när man väll lärt sig, då kan du skapa alla classer själv och ha bättre kontroll över dina modeller och deras namngivning. Kan ha en liten inlärningskurva och du får själv göra mycket av arbetet själv.

2 Designer first

Väldigt bra om du är ny då det mesta automatisk skapar och du med ett enkelt gränssnitt där du kan göra databas kopplingar och få dom kopplade i koden.

3 Database first

Inte bra på något sätt, vilket som är anledningen jag inte gjorde nån djupdykning på det, bra att veta att det finns men inget värt att lägga för mycket tid på att lära sig.

