ADO.NET
Entity Framework

Senior Lecturer, MSc, DI, MCP

Esa Salmikangas

Jyväskylä University of Applied Sciences

Finland

JAMK University of Applied Sciences

The ADO.NET Entity Framework

ADO.NET Entity Framework is an object-relational mapping (ORM) framework for the .NET Framework.

The standard definition of Microsoft for Entity Framework

- The Microsoft ADO.NET Entity Framework is an Object/Relational Mapping (ORM) framework that enables developers to work with relational data as domain-specific objects, eliminating the need for most of the data access plumbing code that developers usually need to write.
- Using the Entity Framework, developers issue queries using LINQ, then retrieve and manipulate data as strongly typed objects.
- The Entity Framework's ORM implementation provides services like change tracking, identity resolution, lazy loading, and query translation so that developers can focus on their application-specific business logic rather than the data access fundamentals.

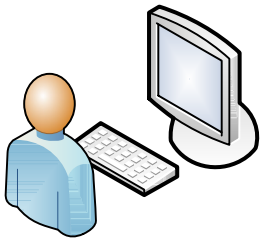
To simply say it

- Entity framework is an Object/Relational Mapping (O/RM) framework.
- It is an enhancement to ADO.NET that gives developers an automated mechanism for accessing & storing the data in database and working with the results in addition to DataReader and DataSet.
- It is a main stream nowadays!

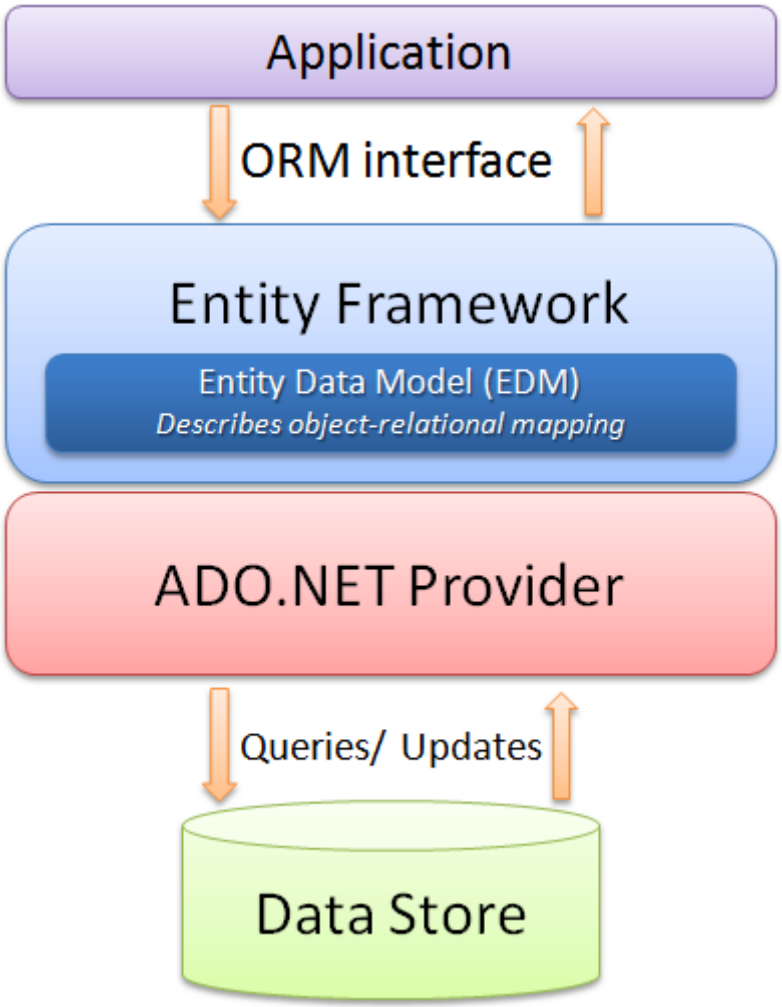
What ORM?

- "**Object-relational mapping**
  (ORM, O/RM, and O/R mapping)
  in computer software is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language. There are both free and commercial packages available that perform object-relational mapping, although some programmers opt to create their own ORM tools."
  - Source: en.wikipedia,org 21.3.2011

The Big Picture



We are programming
against EF,
not against a database



http://msdn.microsoft.com/en-us/data/aa937709

What is ORM framework and why do we need it?

- ORM is a tool for storing data from domain objects to relational database like MS SQL Server in automated way without much programming.
- ORM helps us to keep our database design separate from our domain class design.
  → This makes application maintainable and extendable.
- It also automates standard CRUD operation (Create, Read, Update & Delete)

  →a developer doesn't need to write it manually. ☺

Three main parts of ORM

- ORM includes three main parts:
  - Domain class objects
  - Relational database objects
  - Mapping information on how domain objects maps to relational database objects (tables, views & storedprocedures).

List of object-relational mapping software for .NET

- Comparison of ORM
- ORM for .NET

Why Entity Framework?

- ADO.NET Entity Framework abstracts the relational (logical) schema of the data that is stored in a database and presents its conceptual schema to the application.
  - This abstraction eliminates the object-relational impedance mismatch that is otherwise common in conventional database-oriented programs.
- ADO.NET Entity Framework is used to isolate the logical model of data from the application's model, and, in doing so, **raise the level of abstraction**

Benefits

- Reduced development time / read time is money ☺
- more application-centric terms and model
- no hard-coded dependencies on a particular data engine
- Mappings between the object model and the storage-specific schema can change without changing the application code.
- Intellisense for LINQ, no *TYOPS* in SQL

source: http://msdn.microsoft.com/en-us/data/aa937709

Entity Data Model design approaches

Data first

Model first

Code first

Entity Data Model design approaches

Data first

the new database is created first or existing database is used

→ then Entity Data Model is generated from this database

with Entity Data Model Wizard

Entity Data Model design approaches

Model first

- the development starts from scratch, At first, the conceptual model is created with Entity Data Model Designer, entities and relations are added to the model, but mapping is not created.
- After this Generate Database Wizard is used to generate storage (SSDL) and mapping (MSL) parts from the conceptual part of the model and save them to the edmx file.
  → then the wizard generates DDL script for creating database (tables and foreign keys)

Entity Data Model design approaches

Code first

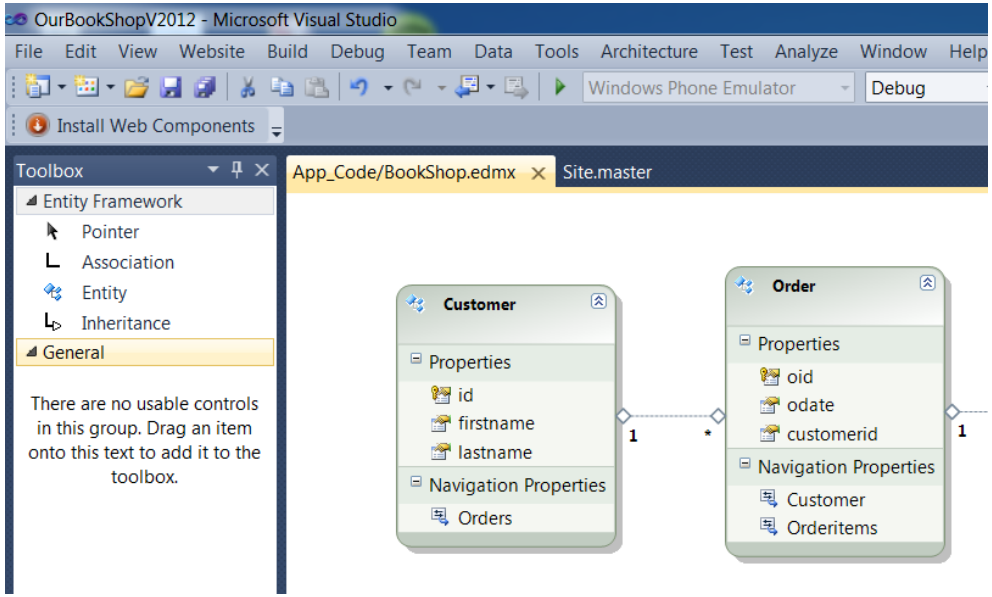
- Code First is the most recent approach included in CTP5. The model is defined via classes and configuration written by the developer and via conventions included in the framework itself

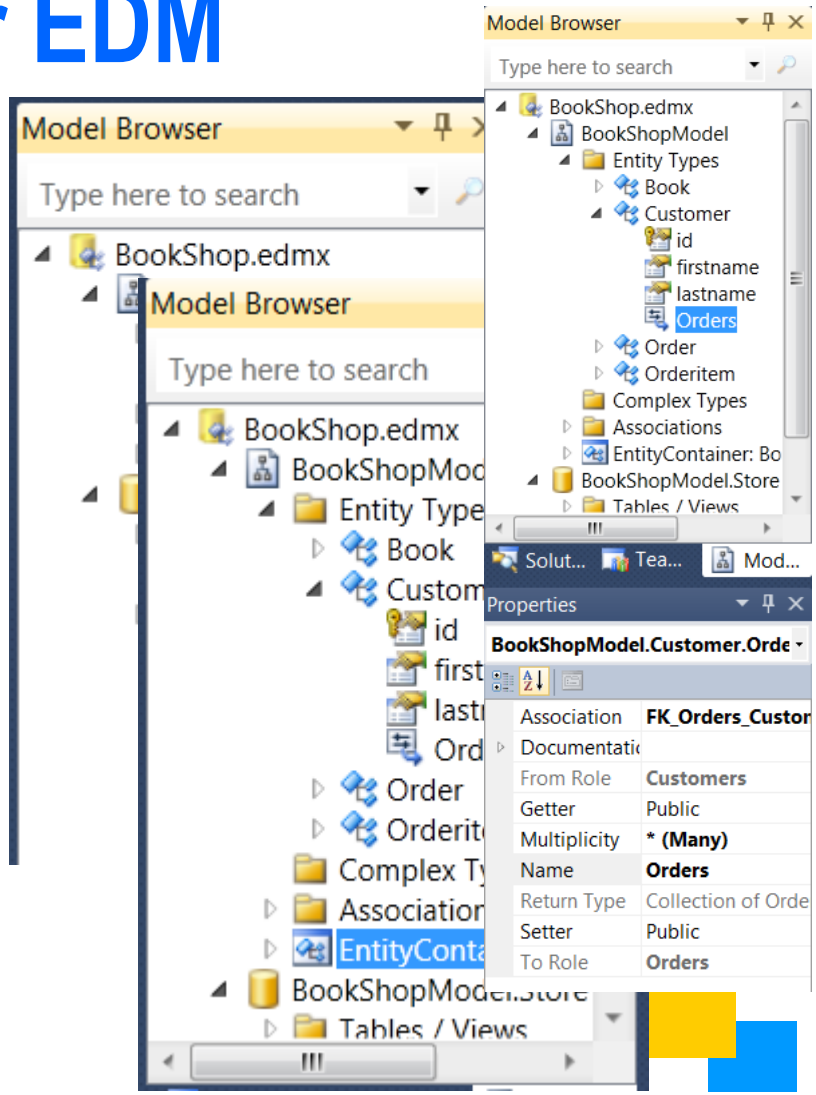
An example of a code to use EF

```
//The ViiniEntities EDMX created before Add | New Item |
ADO.NET Entity DataModel
..
//member of class
myNameSpace.ViiniEntities ctx;
..
ctx = new myNameSpace.ViiniEntities();

//Later in Eventhandlers
//using set of objects as a datasource
DataGridView1.DataSource = ctx.customers;
//creating a new object
var kunde = ctx.customers.CreateObject();
…
ctx.customers.AddObject(kunde);
ctx.SaveChanges();
```

VS2010 as a Tool For EDM

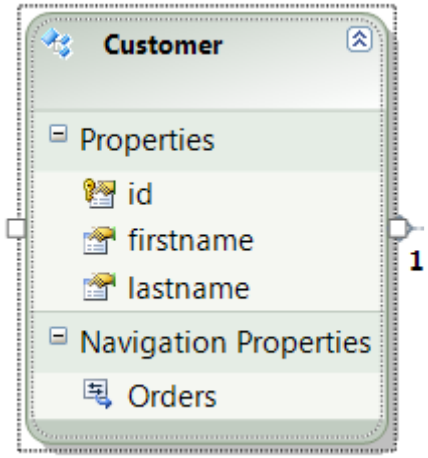




New in ADO.NET Entity Framework 4.0

- Model-first development
- Automatic pluralization
- Foreign keys in models
- POCO class support
- Lazy loading
- T4 Code Generation
- Template customization
- IObjectSet
- Virtual SaveChanges
- ObjectStateManager control
- Self-tracking entities
- SQL generation improvements
- More LINQ operator support
- LINQ extensibility
- ExecuteStoreQuery
- ExecuteStoreCommand
- SPROC import improvements
- Model defined functions
- WPF designer integration
- Code-First development (Feature CTP)

An Example of LINQ




```
ViiniDataContext ctx= new ViiniDataContext();
      try   {
        using (ctx)        {
          var aCustomers = from cust in ctx.customers
                           where cust.Lastname.Contains("A")
                           select cust;
          foreach (var c in aCustomers)
          {
            lstLoydetyt.Items.Add(c.Firstname + " " + c.Lastname);
          }
        }
      }
```

Pit of Success

Database First

- “database is the truth”
- why? it already exists, or you want low level control over the database
- what? import model into edmx and tweak

Model First

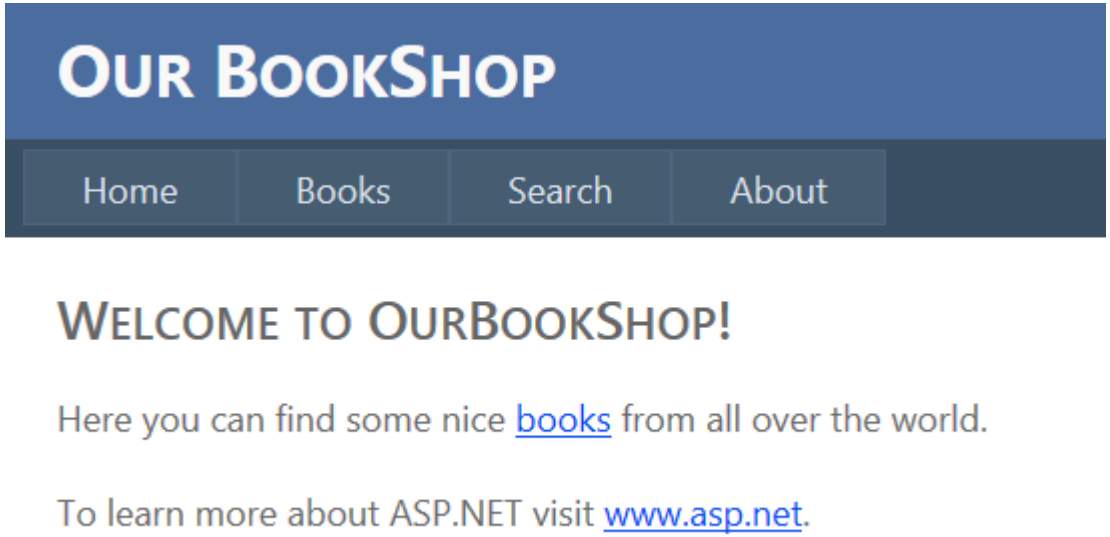
- “edmx is the truth”
- why? you want separation from code and database in a declarative format
- what? create a model and tweak

Code First

- “code is the truth”
- why? primarily focused on code shape, database is an implementation detail
- what? define classes in code, adjust shape using contextbuilder

Lazy Loading in Entity Framework

- When lazy loading is enabled in Entity Framework 4.0~4.3, the related object are loaded automatically only when the navigation properties are accessed.
  →This avoids loading unnecessary things, and then potentially saves a lot of resources.



DEMO: BOOKSHOP WITH EF

username: **test**

password: **123456**

The Goal of the Demo

Step 1: Get all customers

Step 2: Add a new customer

Step 3: Delete a customer

Step 4: Modify a customer

Step 5: Search customers

Step 6: Show all orders

Step 7: Show orders of the selected customer

First steps

- Correct an ugly Login-page

  → give proper rights for anonymous users

- Add a new Web-Page
  - add a Tab also for it in Site.master

```
<asp:Menu ID="NavigationMenu"
<items>
  <asp:MenuItem NavigateUrl="~/TestEF.aspx" Text="Test EntityFW"/>
```

- Add a Entity Data Model
- Make a context object from it and **use it**

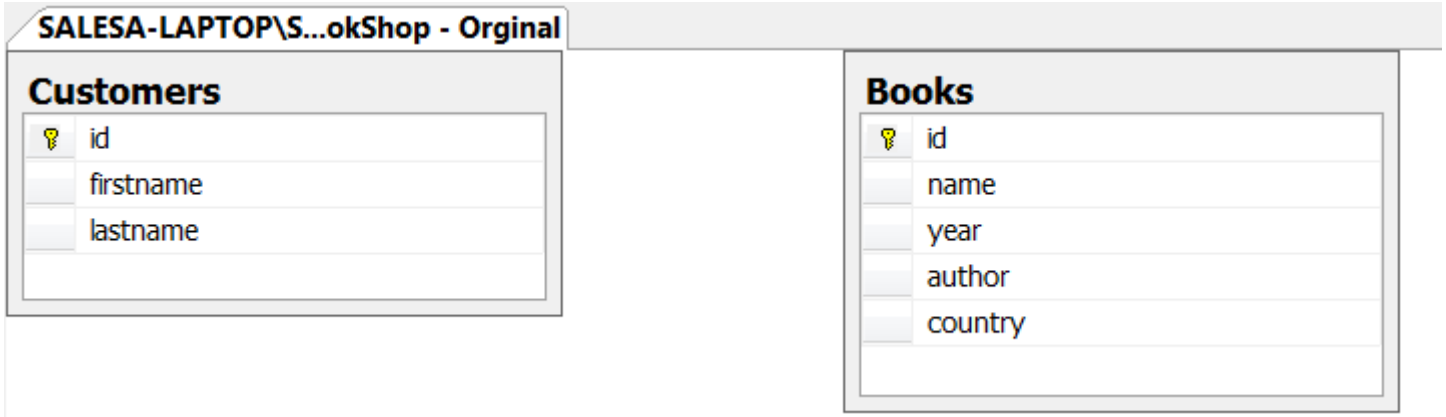
WEB.CONFIG DEMO
If the Login page is “ugly”

- the reason is that non-authenticated end-users has no privileges to read Styles-folder → We can fix it in web.config:

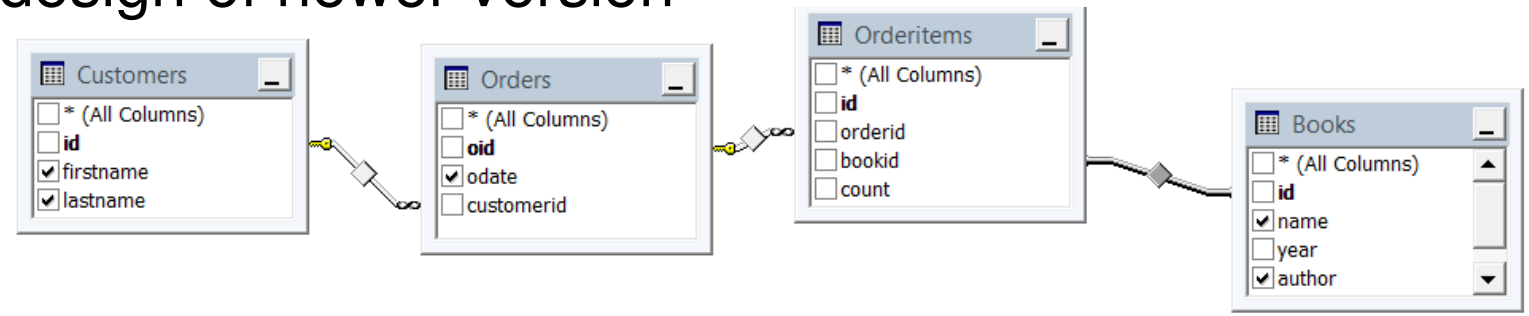
```
 <!-- This gives all users access to use
Styles-folder-->
 <location path="Styles">
   <system.web>
     <authorization>
       <allow users="*"/>
     </authorization>
   </system.web>
 </location>
```

Database

- Original database



The design of newer version

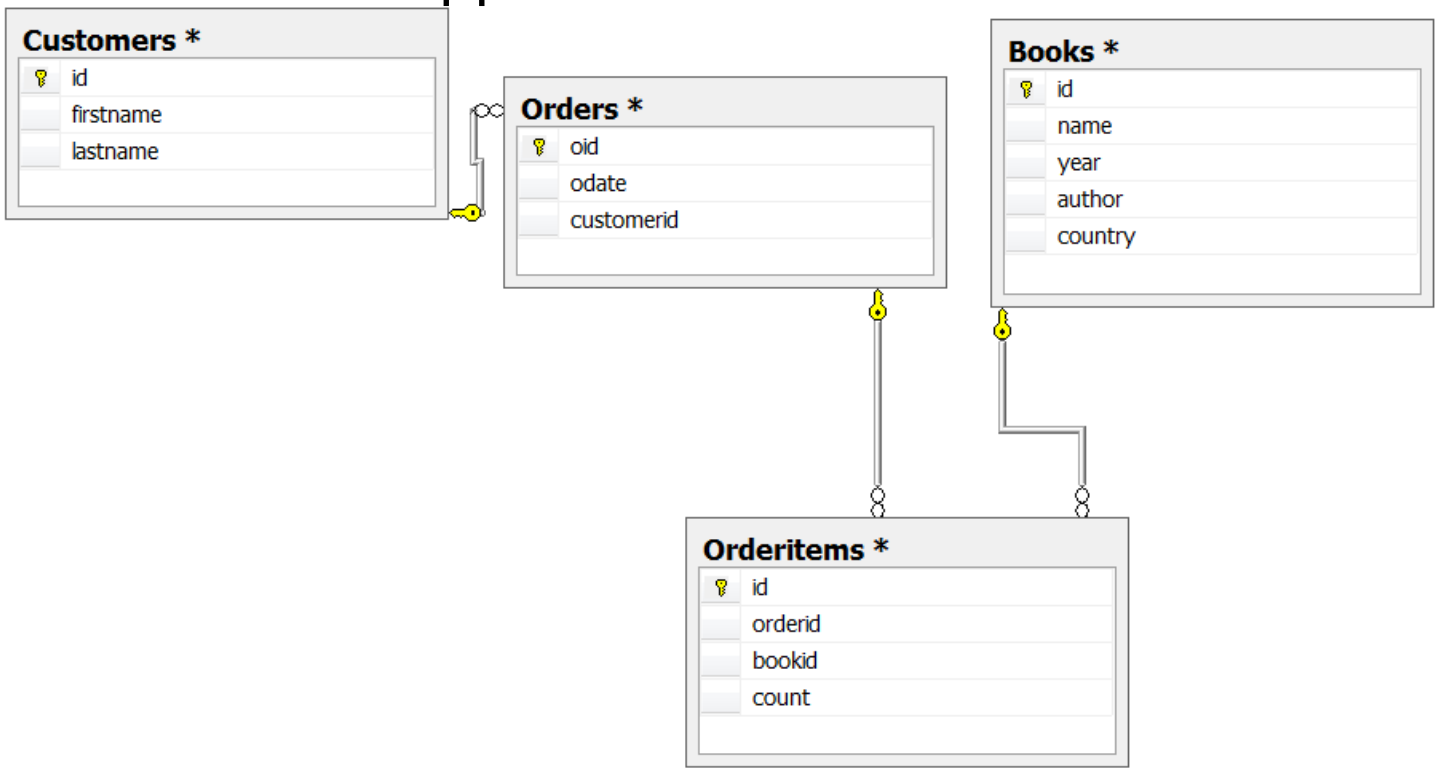


Background: Well-designed databases can pose a problem for developers.

- In the data world, a database is designed for maintainability, security, efficiency, and scalability.
- Its data is organized in a way that satisfies the demands of good database design, yet provides challenges for the developer who needs to access that data.

The Purpose of the Demo

- To show how to use Entity Framework with SQL Server in ASP.Web Application



Relations in Databases vs Relations in Entity Framework

- one-to-one
- one-to-many
- many-to-many

Also Relations in EF Entity Framework supports the relational database's one-to-many and many-to-many concepts.

One-to-many relationship

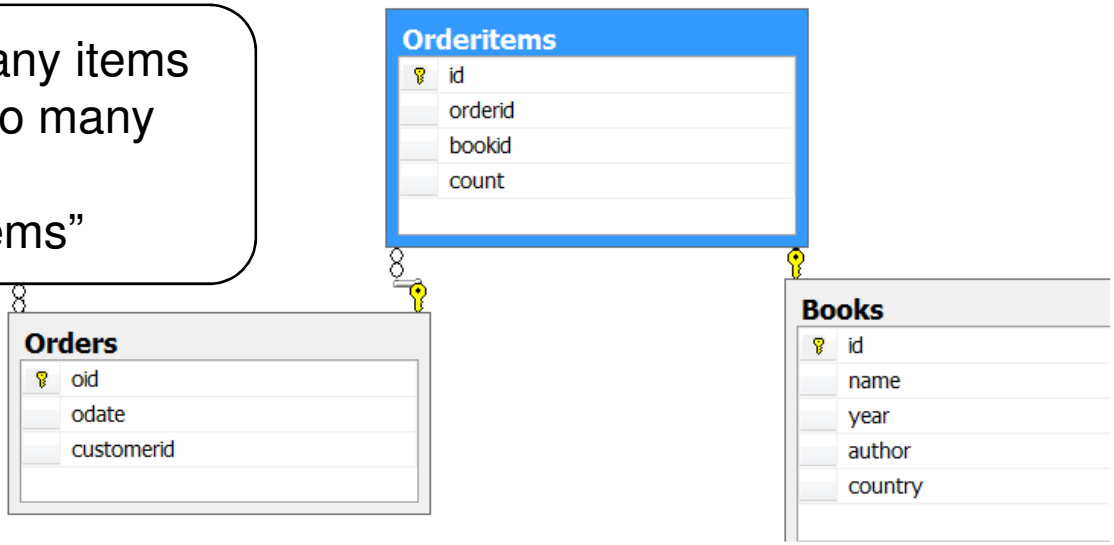
- Entity A can have many Entity B, but Entity B can have only one Entity A, then A and B is one-to-many relationship.
- In our demo: A customer can have many orders, but an order can have only one customer

Many-to-many

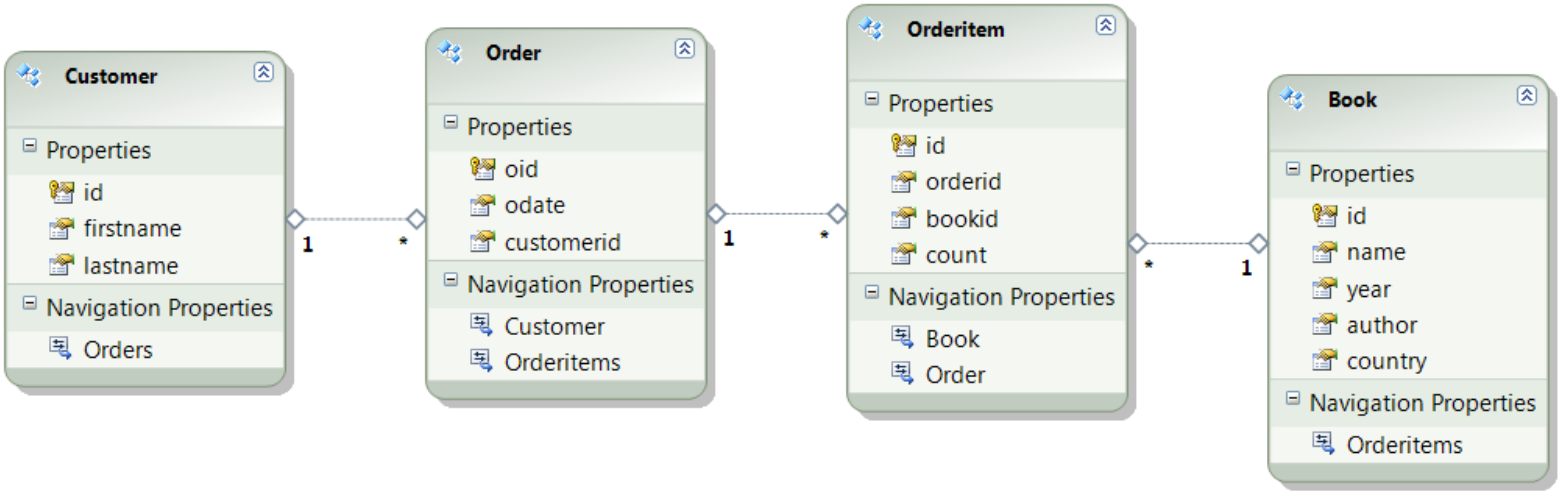
- Entity A can have many Entity B;
  Entity B can have many Entity A.

- This relationship usually needs a matching table in a relational database for modeling. Technically, after a matching table is added, a many-to-many relationship has been broken into two one-to-many relationships.

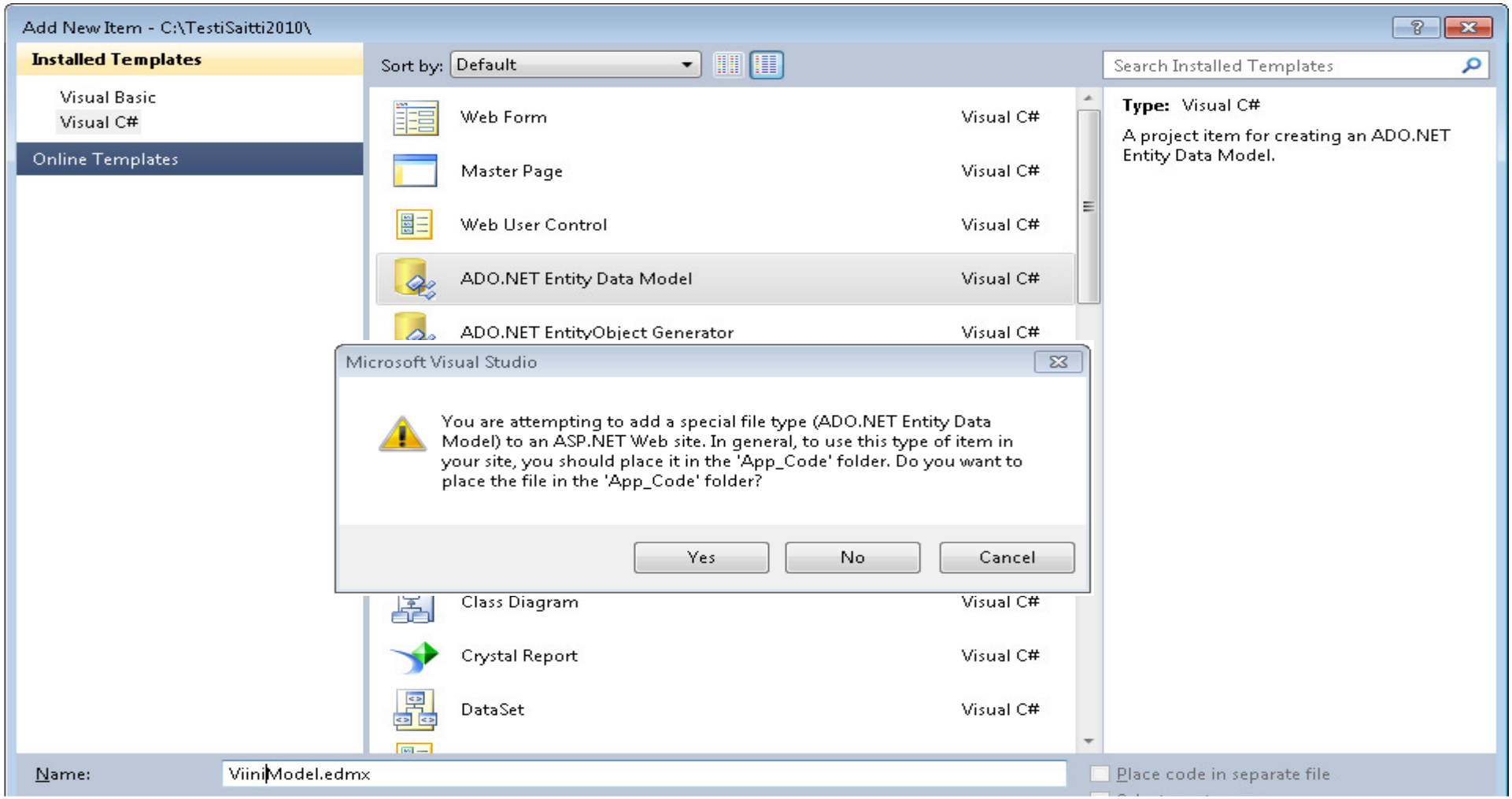
In our demo: Orders can have many items (books), and items can belong to many orders
→ we need a table “OrderItems”



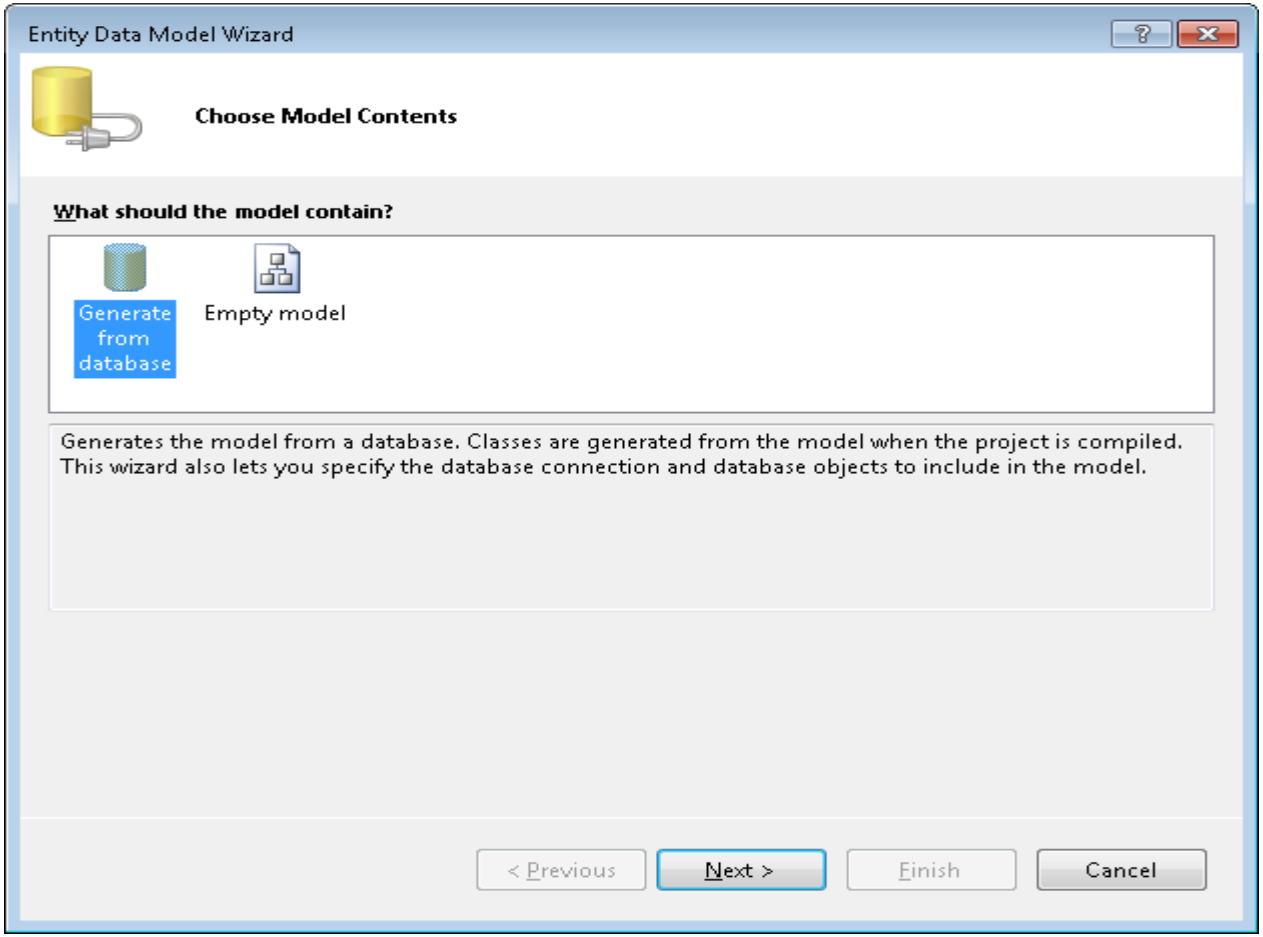
Entities in our Demo



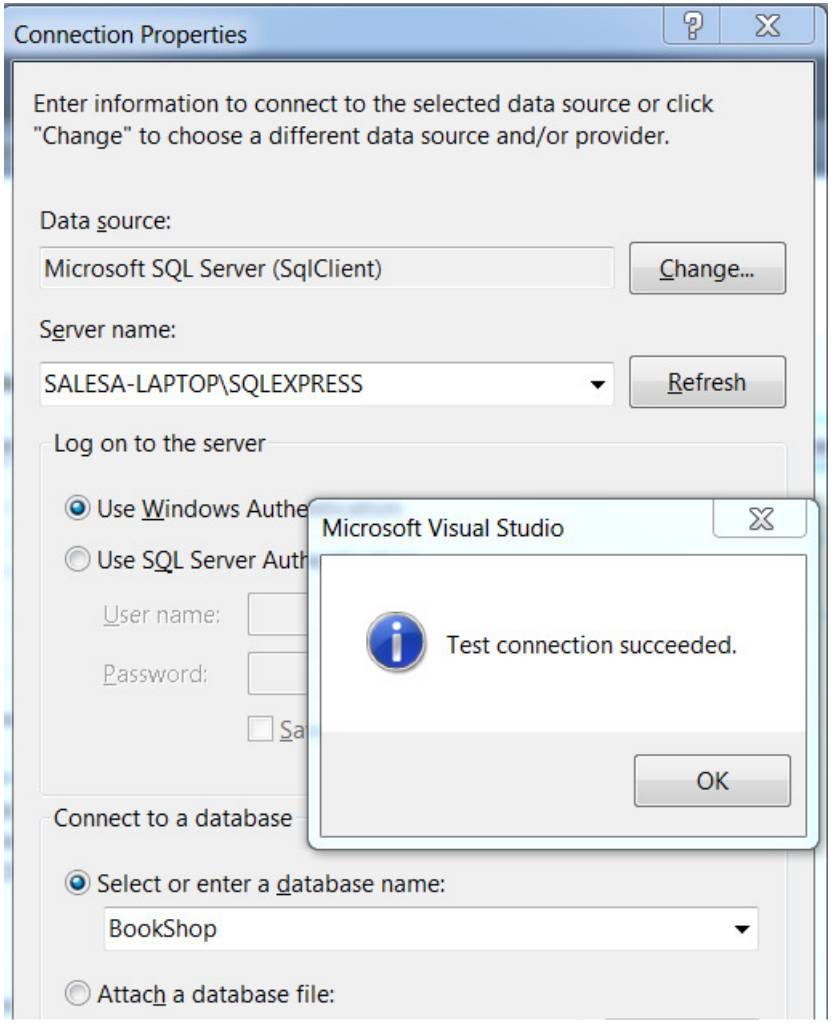
Entity Data Model Wizard

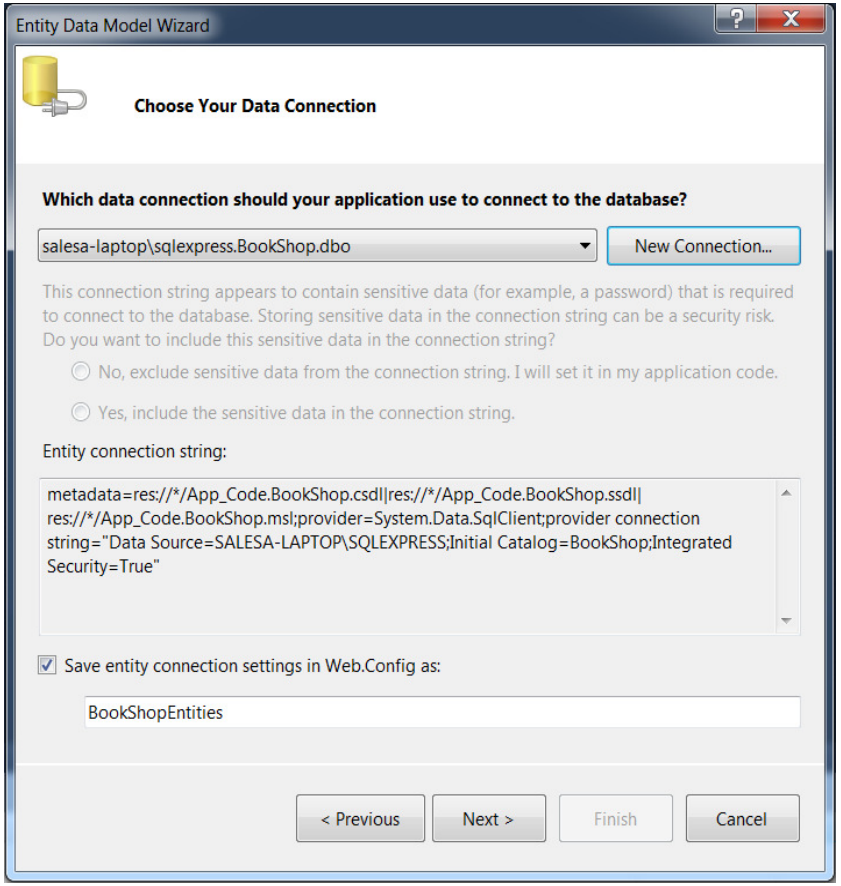


Entity Data Model Wizard

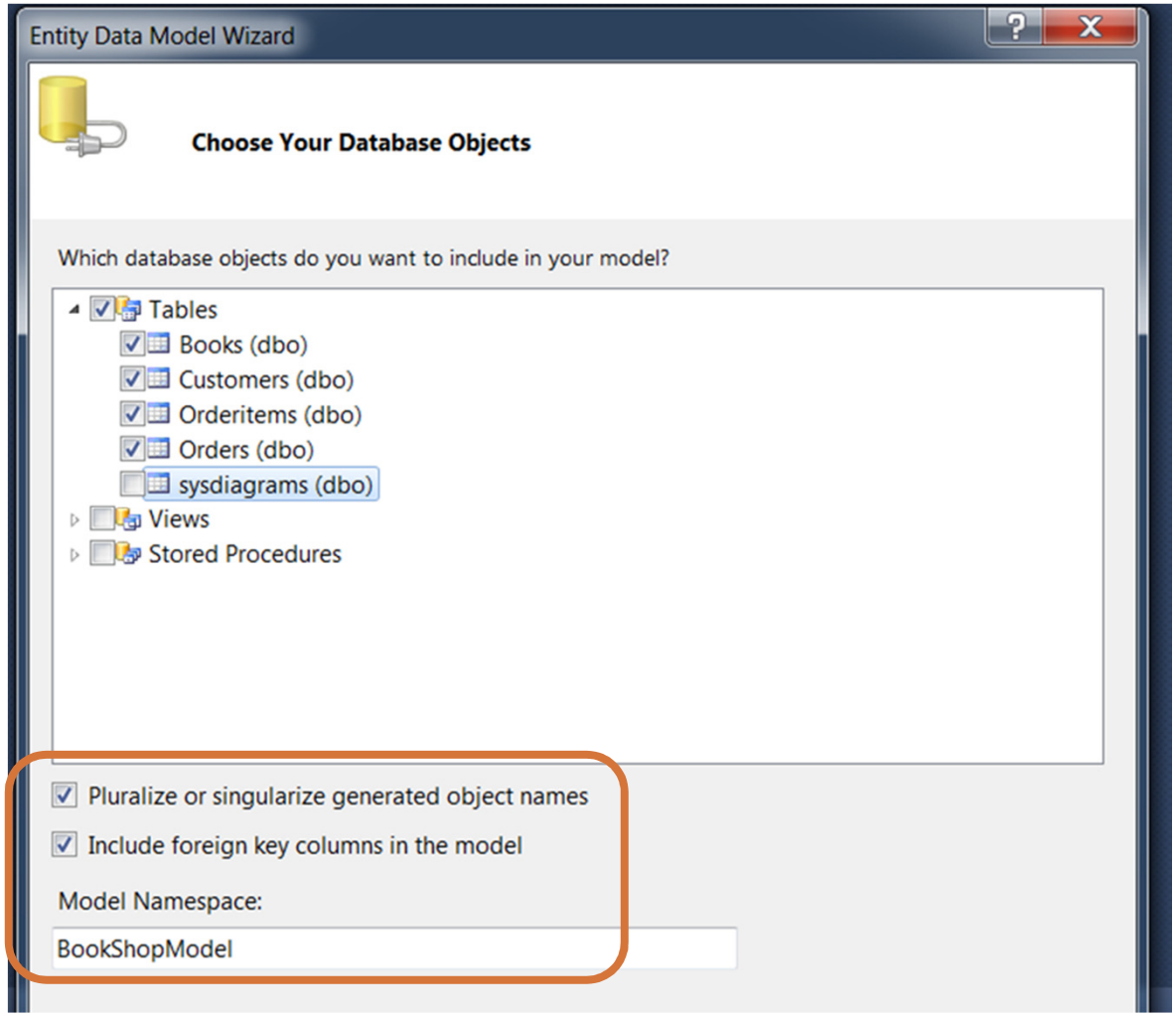


Choose Your Data Connection

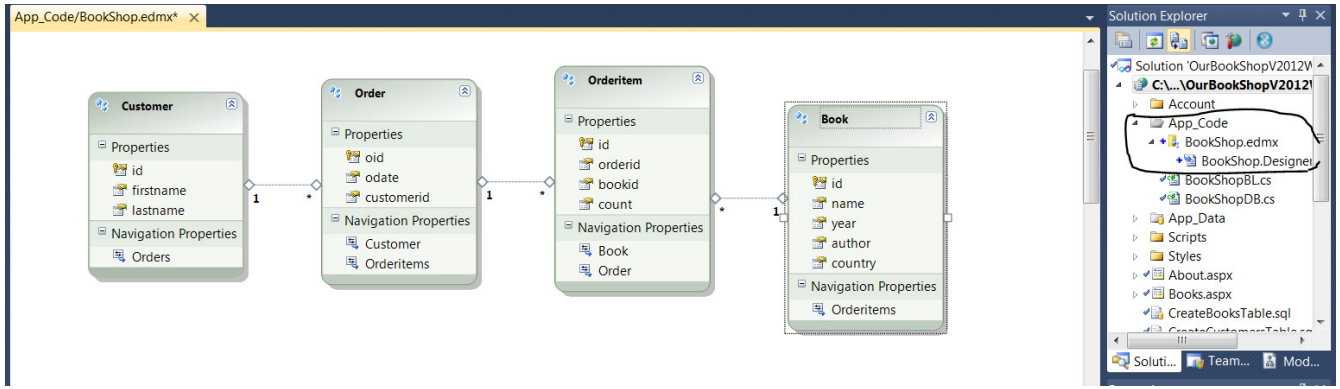




Choose Database Objects

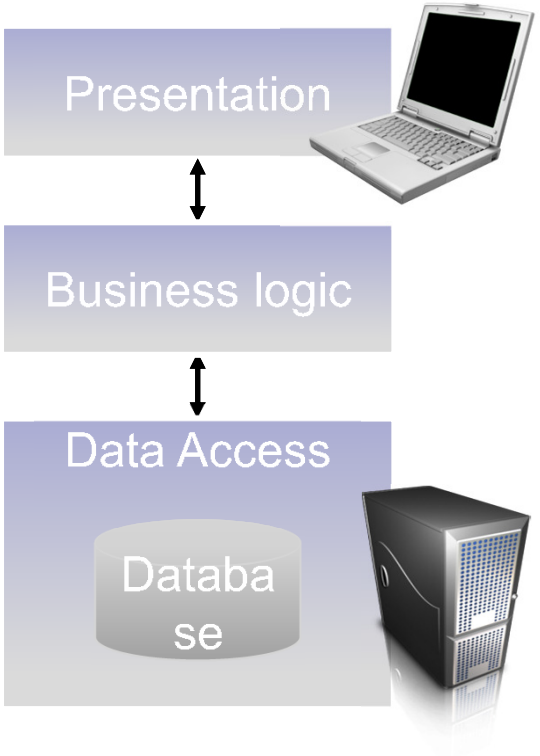


BookShop.edmx



AND ONE MORE THING

N-Tier with Entity Framework



- Why:
  - App split by machine boundary
  - Need to serialize object graphs
  - Save changes made elsewhere
- How: use WCF with
  - Self-Tracking Entities
  - WCF Data Services, OData
  - WCF RIA Services
  - Your own data transfer objects

Summary

- Entity Framework is Microsoft's implementation of ORM.
- It is widely used, it is mainstream.
- it gives lot of benefits for programmers.
  - easy to use
  - raise to abstraction level
  - save time, save money
  - etc
- if interested start here: http://msdn.microsoft.com/en-us/data/ef.aspx

Main resources

- Julia Lerman: Programming Entity Framework
- http://msdn.microsoft.com/en-us/default.aspx
  - http://msdn.microsoft.com/en-us/library/gg696172%28v=vs.103%29.aspx
  - http://msdn.microsoft.com/en-us/data/ef.aspx
  - etc
- http://entityframeworktutorial.net/
- http://en.wikipedia.org/wiki/Object-relational_mapping
- http://en.wikipedia.org/wiki/Entity_framework

Thanks for your attention question and comments, bitte