

# **LINQ - 1**

## **Language Integrated Query**

Yrd. Doç. Dr. Özgü Can

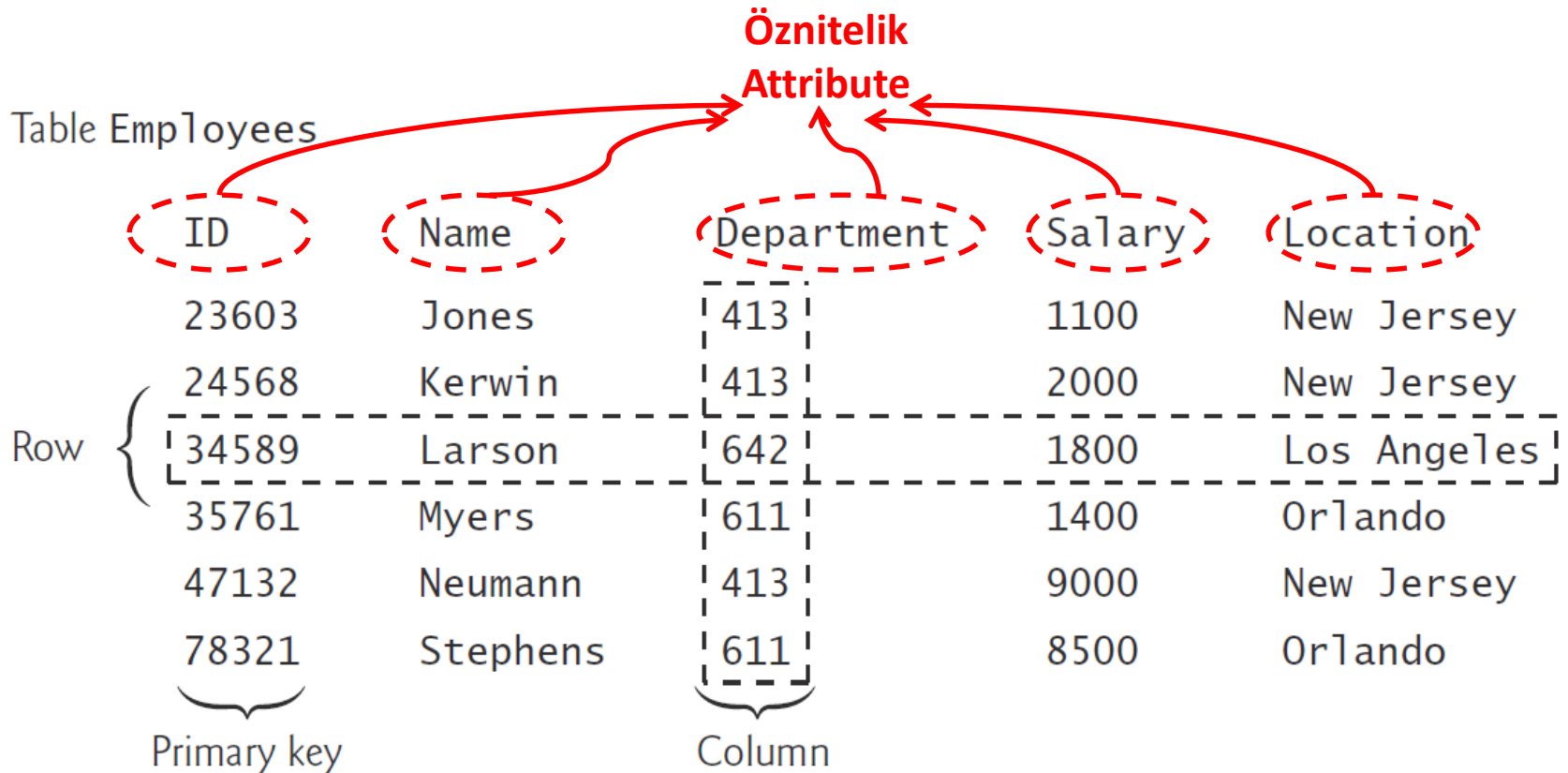
# LINQ

- Büyük miktardaki veri koleksiyonları üzerinde çalışmayı sağlar.
- Farklı veri kaynakları sorgulanabilir:
  - Nesneler
  - SQL veritabanı
  - XML dokümanları
- Sorgu sonuçları üzerinde *sıralama*, *gruplama* ve *istatistik hesaplama* işlemlerini gerçekleştirmede kolaylık sağlar.

# LINQ

- .NET Framework, **farklı veri türleri için** sorgulama çözümleri sunan çeşitli LINQ yetenekleri sunmaktadır:
  - LINQ to Objects
  - LINQ to XML
  - LINQ to ADO.NET
  - LINQ to Entities
  - LINQ to Dataset
  - LINQ to SQL
  - PLINQ

# LINQ



**Tek bir satırı belirten *unique* değer**

**Her bir tablo bir *primary key*'e ihtiyaç duyar.**

**Composite Key → İki ya da daha fazla  
sütundan oluşan  
bir *primary key*'dir.**

# Uygulama - LINQ

## Books veritabanı

- Authors Tablosu

Column	Description
AuthorID	Author's ID number in the database. In the Books database, this integer column is defined as an <b>identity</b> column, also known as an <b>autoincremented</b> column—for each row inserted in the table, the AuthorID value is increased by 1 automatically to ensure that each row has a unique AuthorID. This is the <i>primary key</i> .
FirstName	Author's first name (a string).
LastName	Author's last name (a string).

# Uygulama - LINQ

## Books veritabanı

- **Titles** Tablosu → Veritabanındaki her bir kitap ile ilgili 4 sütun

Column	Description
ISBN	ISBN of the book (a string). The table's primary key. ISBN is an abbreviation for "International Standard Book Number"—a numbering scheme that publishers worldwide use to give every book a <i>unique</i> identification number.
Title	Title of the book (a string).
EditionNumber	Edition number of the book (an integer).
Copyright	Copyright year of the book (a string).

# Uygulama - LINQ

## Books veritabanı

- **AuthorISBN** Tablosu → Her bir kitap için ISBN numarasını ve ilgili yazar ID'sini içeren 2 sütun
  - Yazar ile kitap arasında ki ilişkiyi oluşturur.

Column	Description
AuthorID	The author's ID number, a foreign key to the Authors table.
ISBN	The ISBN for a book, a foreign key to the Titles table.

- **Foreign Key** → Tablodaki bir sütun başka bir tablodaki *primary key* sütunu ile eşleşmektedir.
- **AuthorsID & ISBN** → **Composite Primary Key**

# Foreign Key

- Veritabanı bir çok tablodan oluşabilir.
- Amaç:
  - Veritabanı tabloları arasında **kopyalanmış veri miktarının *min.* olması.**
- Foreign Key → Veriyi bir çok tabloya bağlar.
- Her bir *foreign key* değeri başka bir tabloda *primary key* değeri olmalıdır.

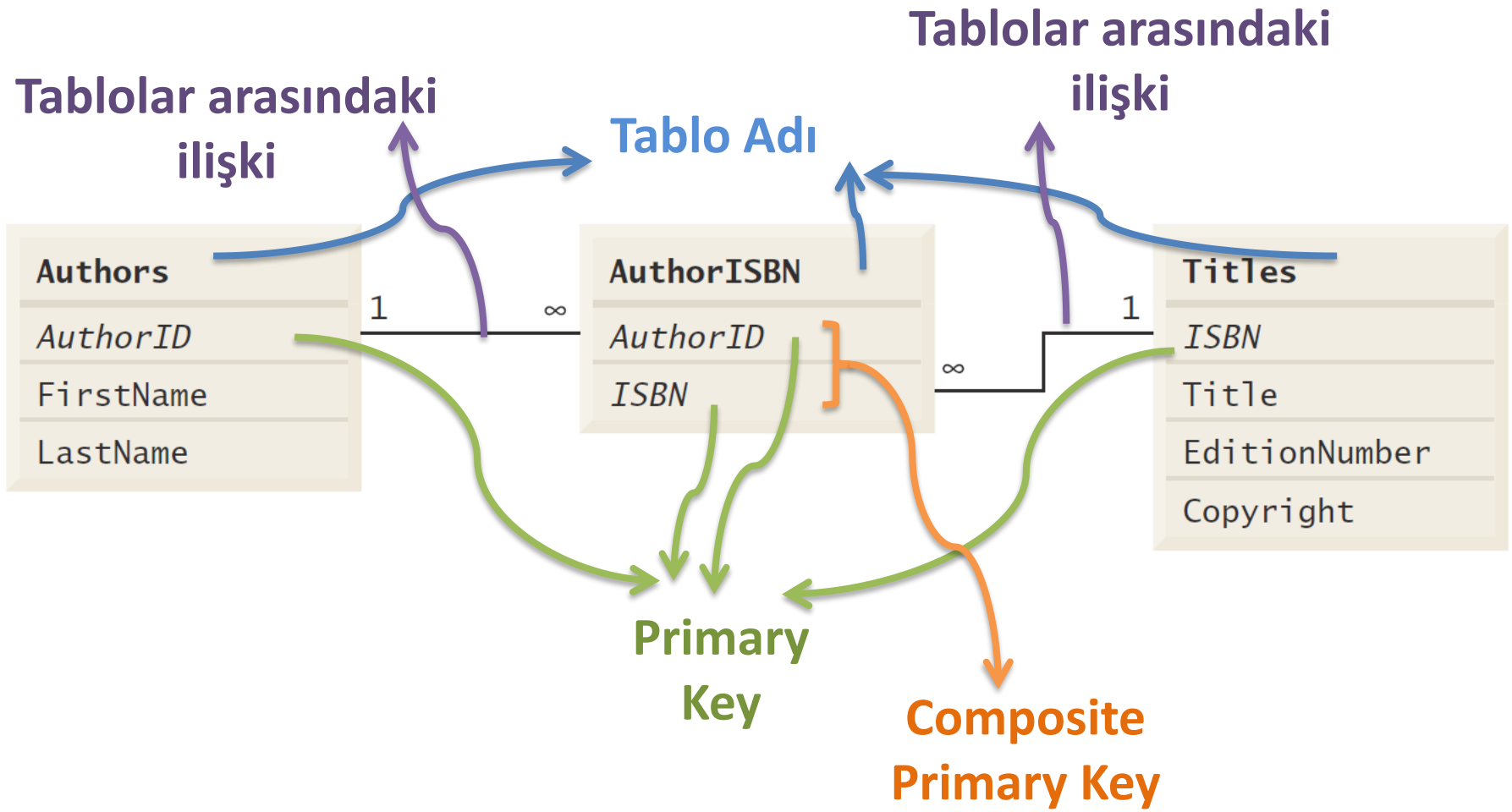


# Foreign Key

- *Primary key* ile *foreign key* arasında **one-to-many relationship** vardır.
- Bir *foreign key* kendi tablosunda **bir çok kez** yer alabilir, **ancak** diğer tabloda (*primary key* olarak) sadece **bir kez** yer alabilir.

# Uygulama - LINQ

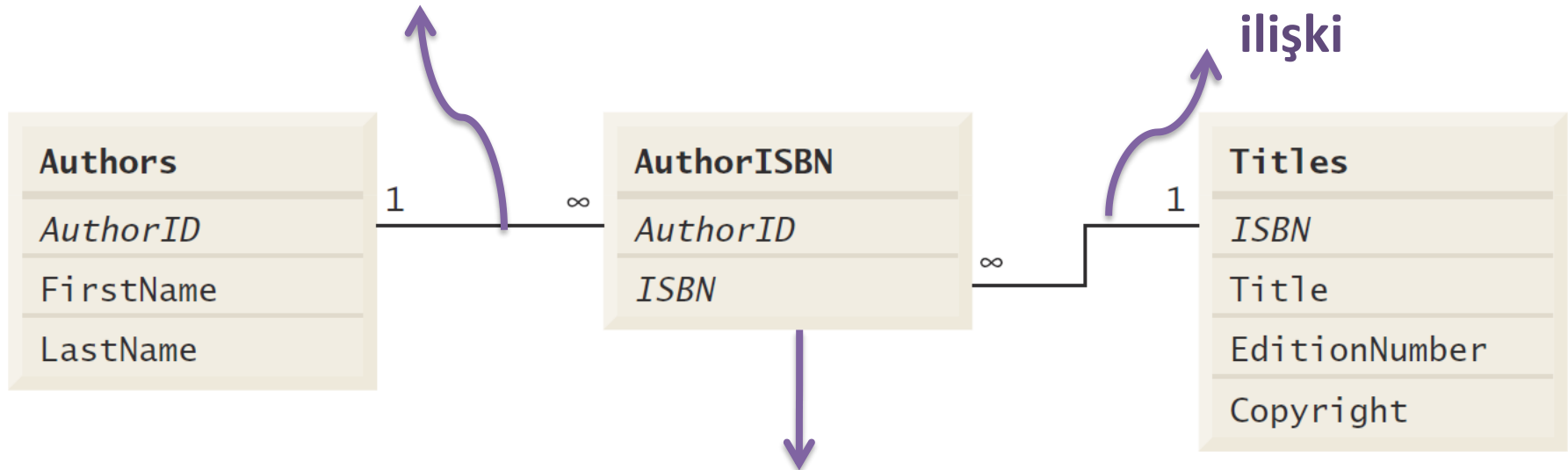
## Entity-Relationship Diagram



# Uygulama - LINQ

## Entity-Relationship Diagram

one-to-many relationship → *Bir yazar tarafından yazılmış bir çok kitap olabilir.*



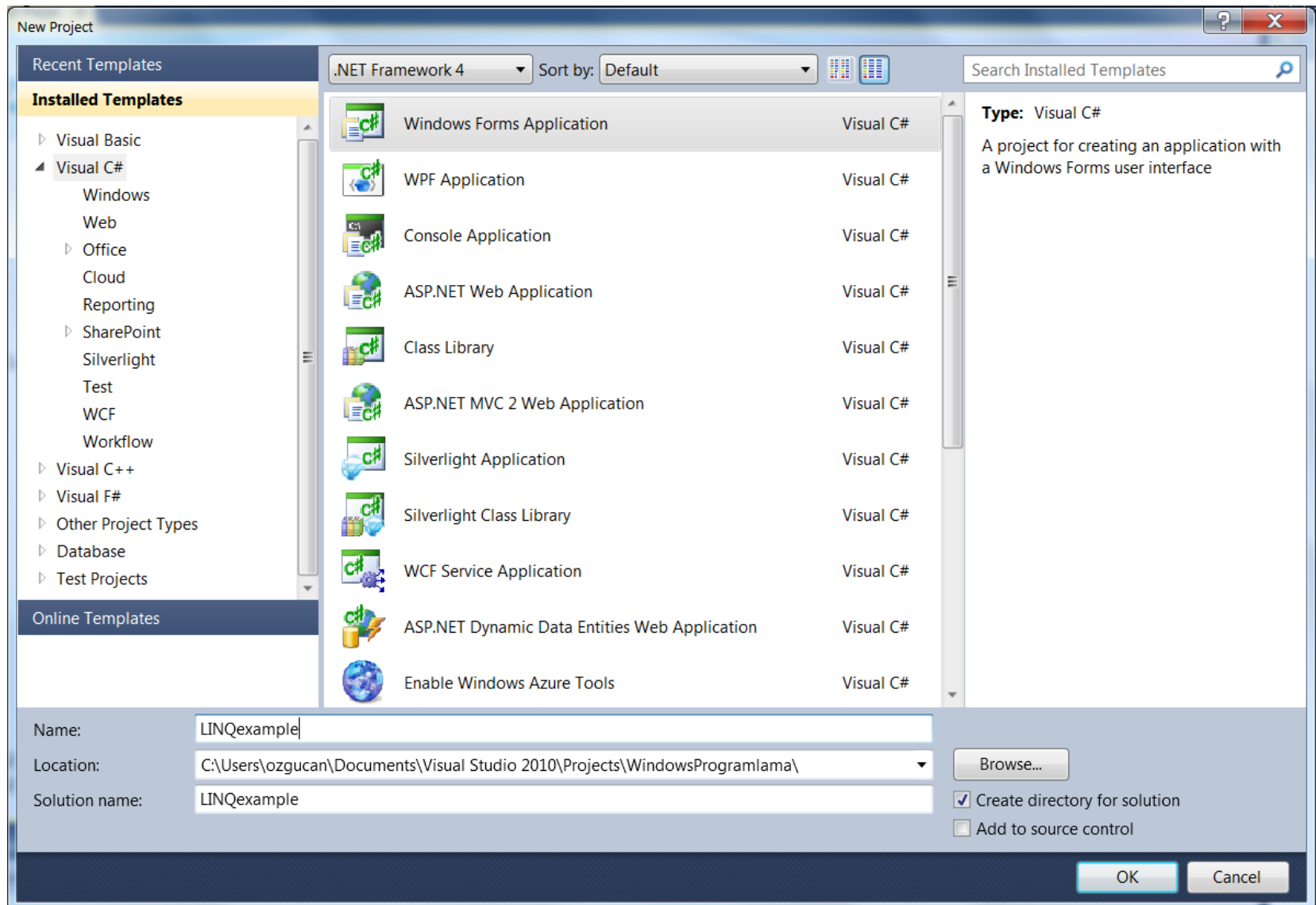
Tablolar arasındaki ilişki

**AuthorISBN** tablosu, **Authors** ve **Titles** tabloları arasında **many-to-many relationship** yaratır. *Bir yazarın birçok kitabı olabilir ve bir kitabın bir çok yazarı olabilir.*

# LINQ to SQL

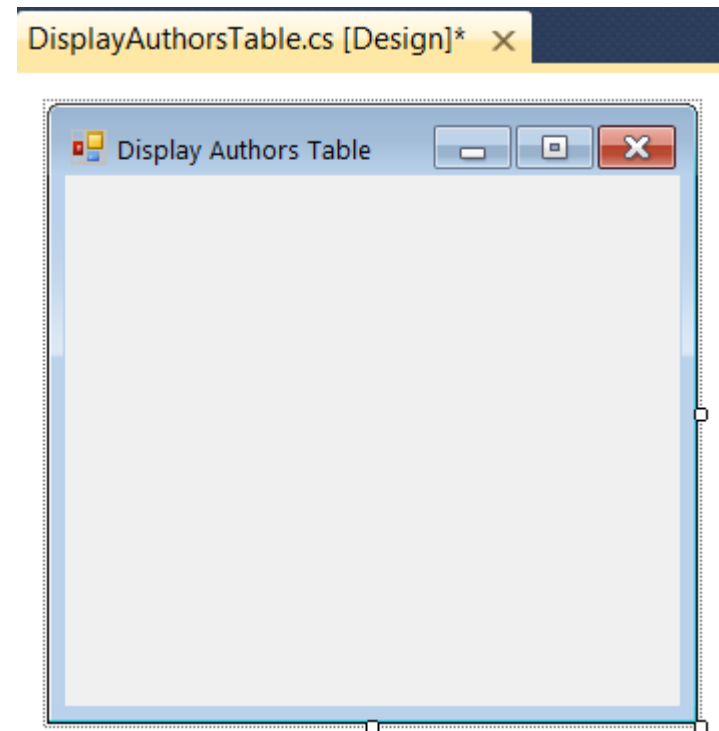
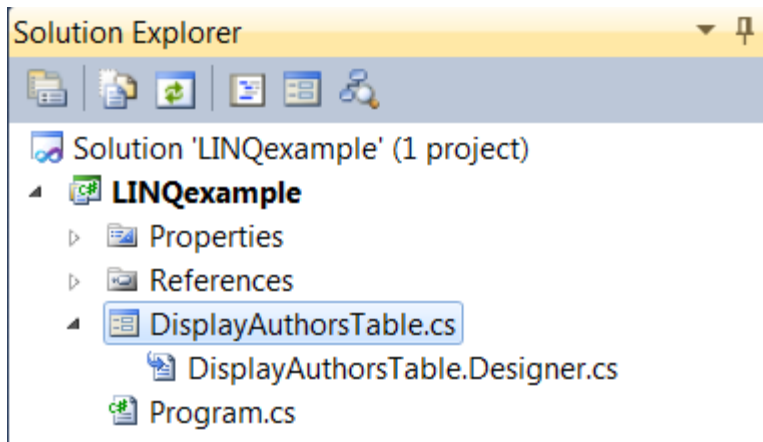
- SQL veritabanı sunucusuna LINQ sözdizimi kullanılarak erişimi sağlar.
- Veritabanına, **LINQ to SQL Designer** kullanılarak veritabanı şemasından otomatik olarak yaratılan sınıflar aracılığı ile ulaşılır.

# Uygulama - LINQ



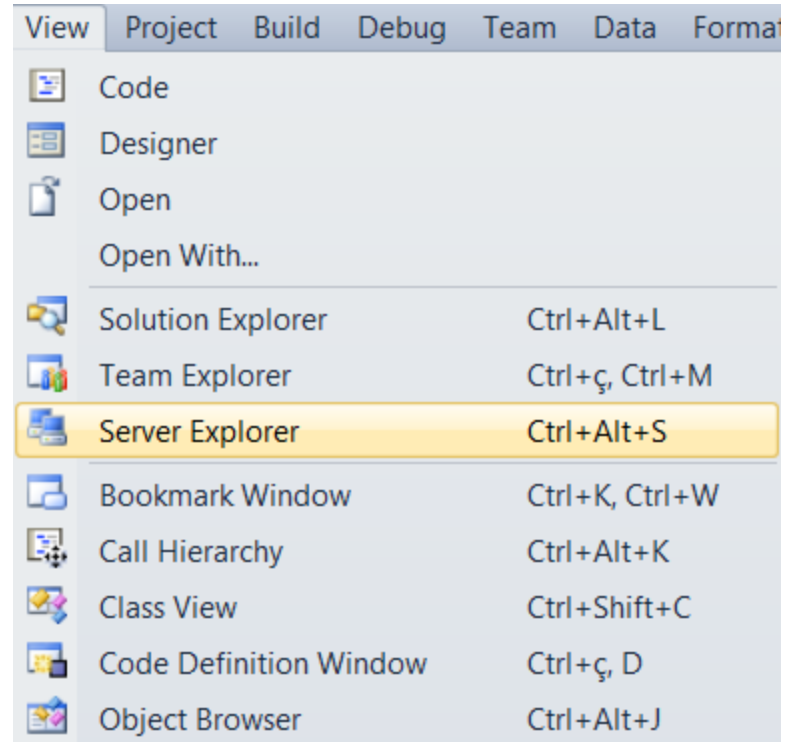
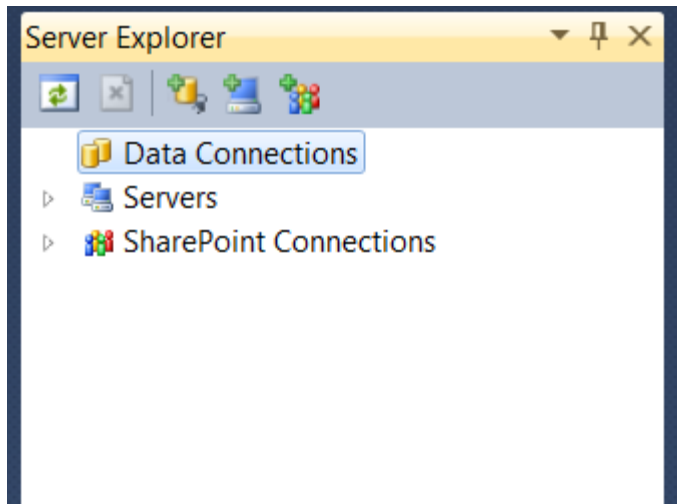
# Uygulama - LINQ

- Form1.cs → DisplayAuthorsTable.cs
- Form **Text = Display Authors Table**



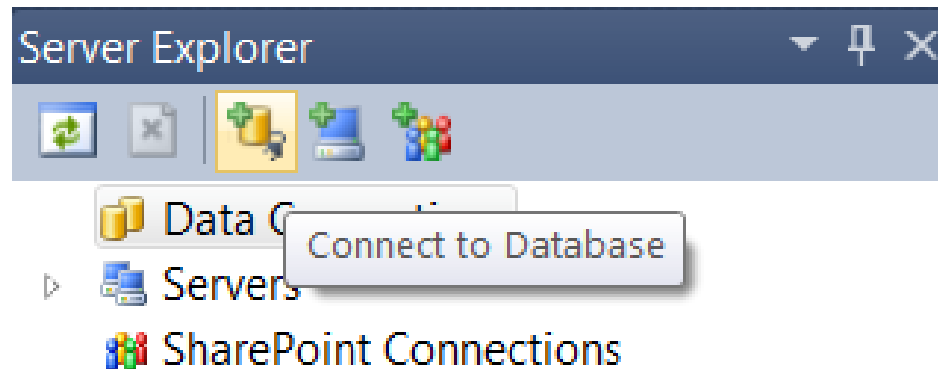
# Uygulama - LINQ

- Visual Studio C# 2010 Explorer
  - View > Other Windows > Database Explorer
- Visual Studio full version
  - View > Server Explorer



# Uygulama - LINQ

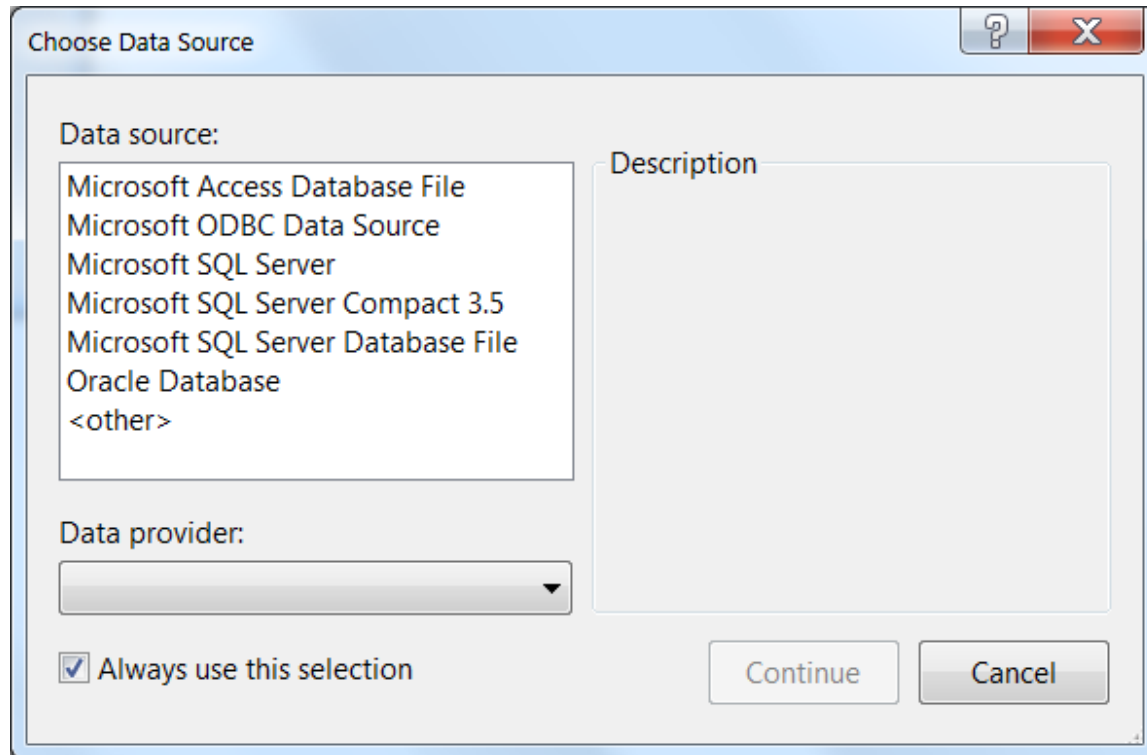
- Connect Database



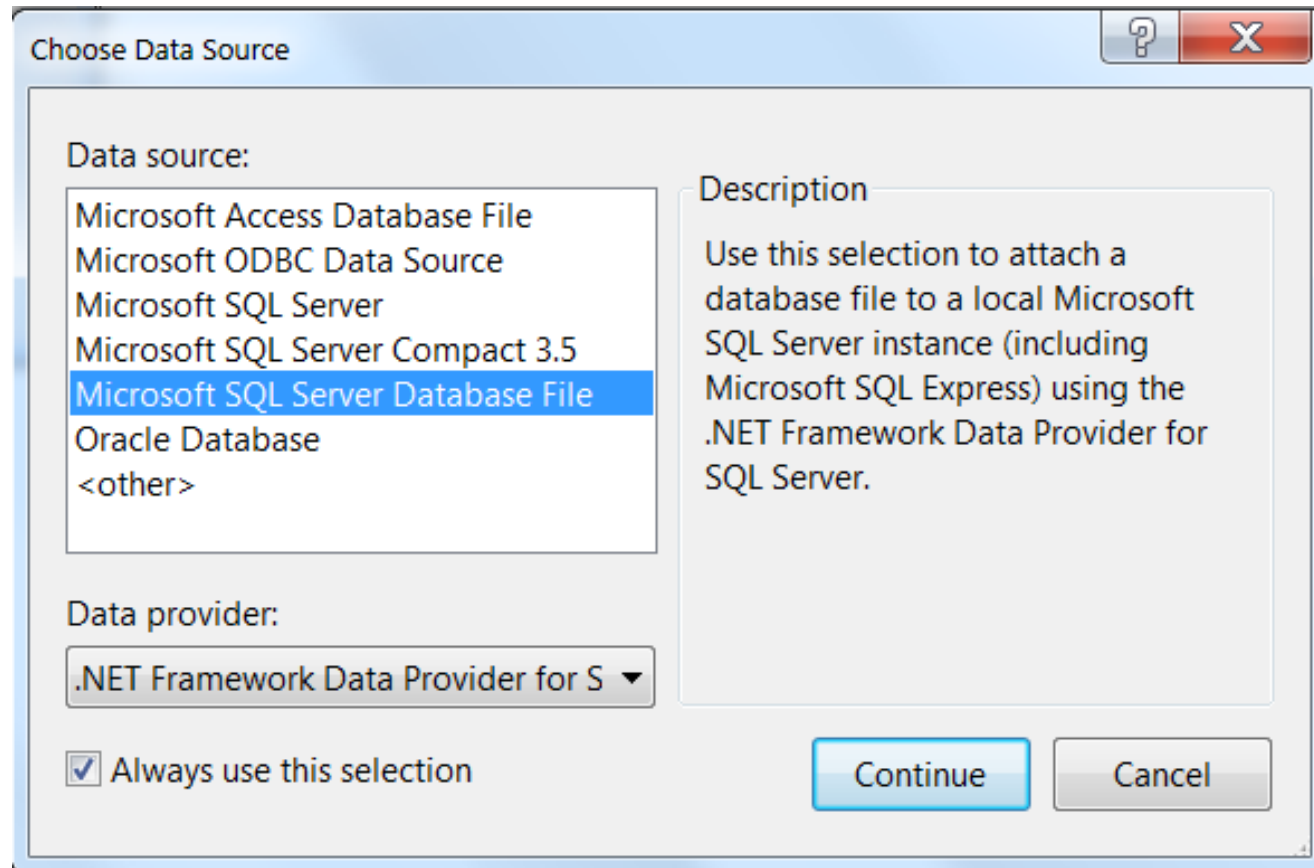


# Uygulama - LINQ

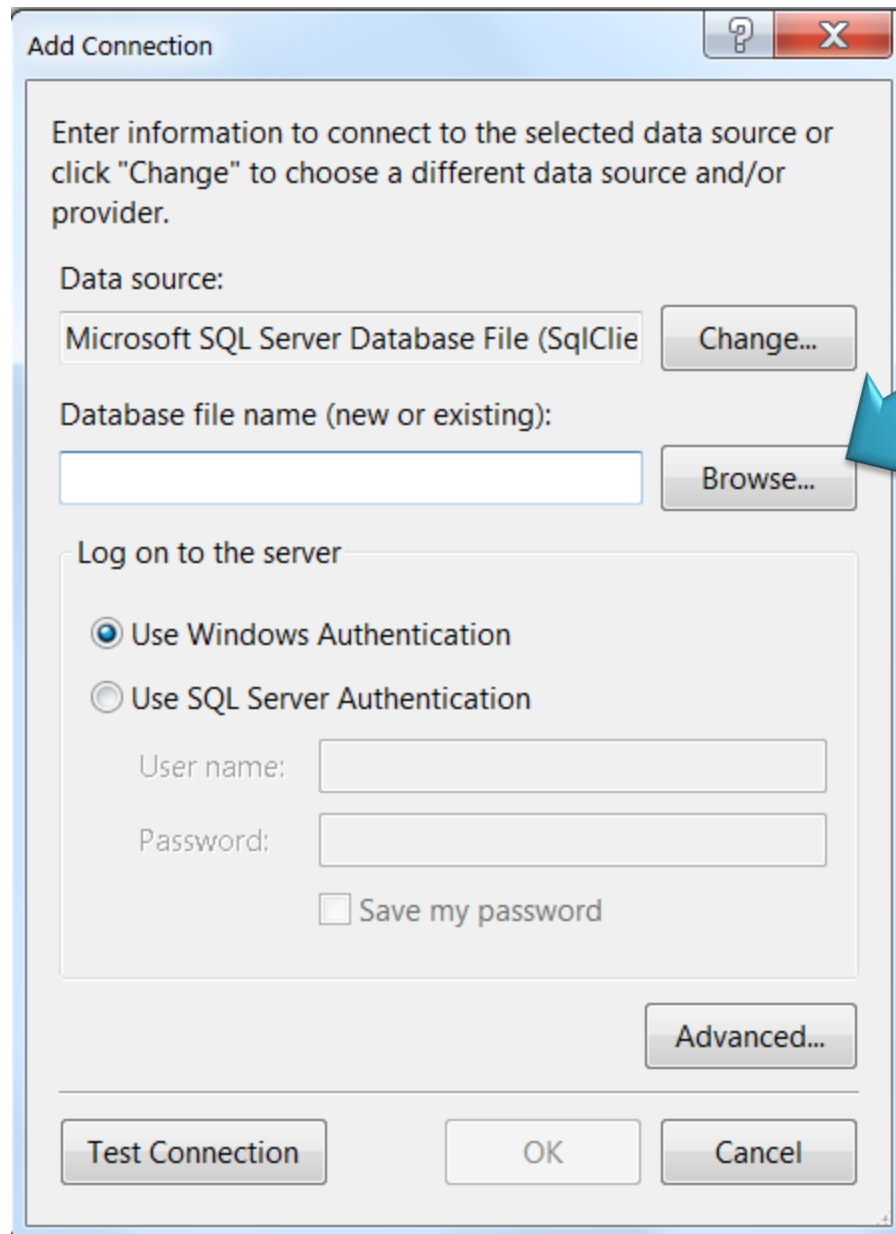
- Choose Data Source



# Uygulama - LINQ



# Uygulama - LINQ



**Add Connection**

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 Database File (SqlClient) Change...

Database file name (new or existing):  
 Browse...

Log on to the server

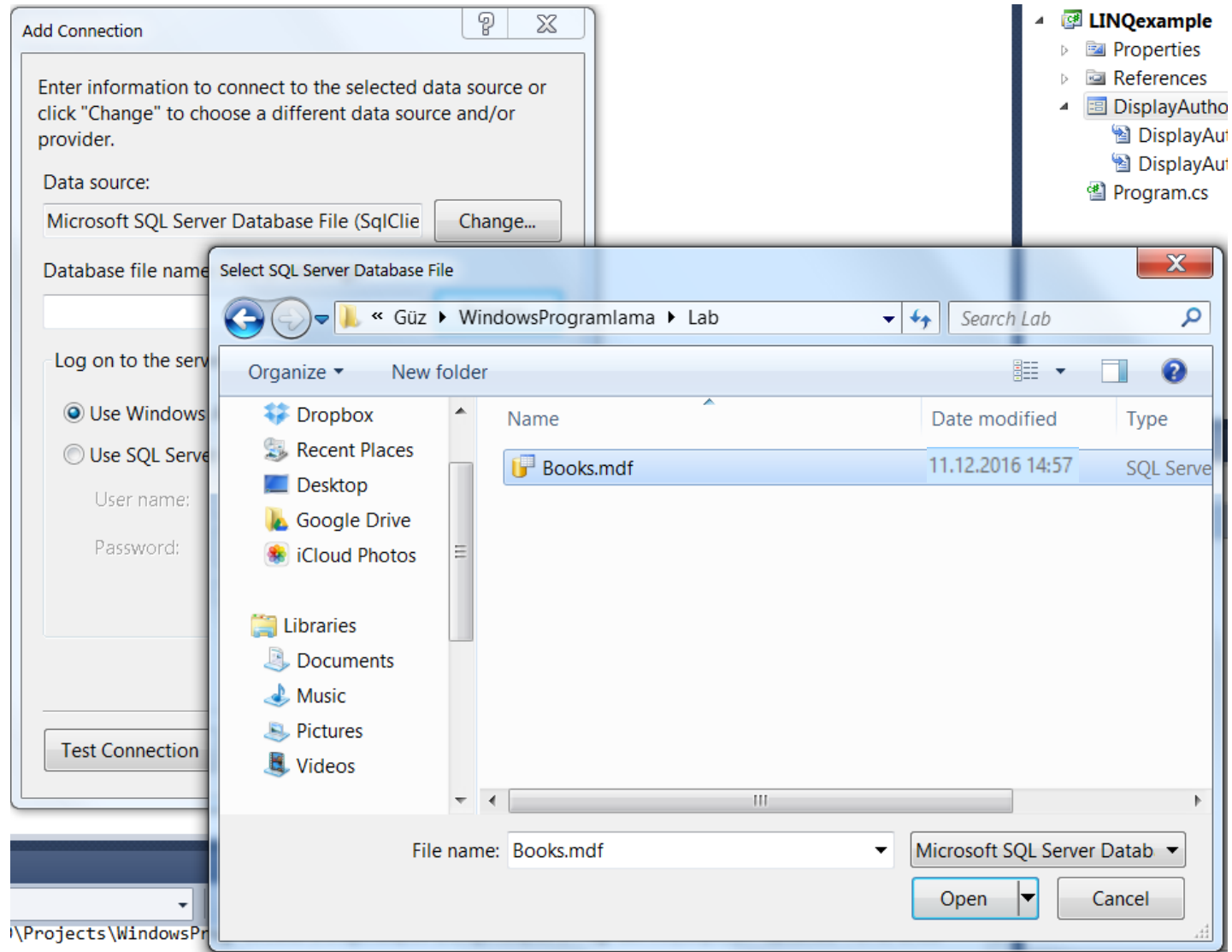
☒ Use Windows Authentication  
☐ Use SQL Server Authentication

User name:   
Password:   
☐ Save my password

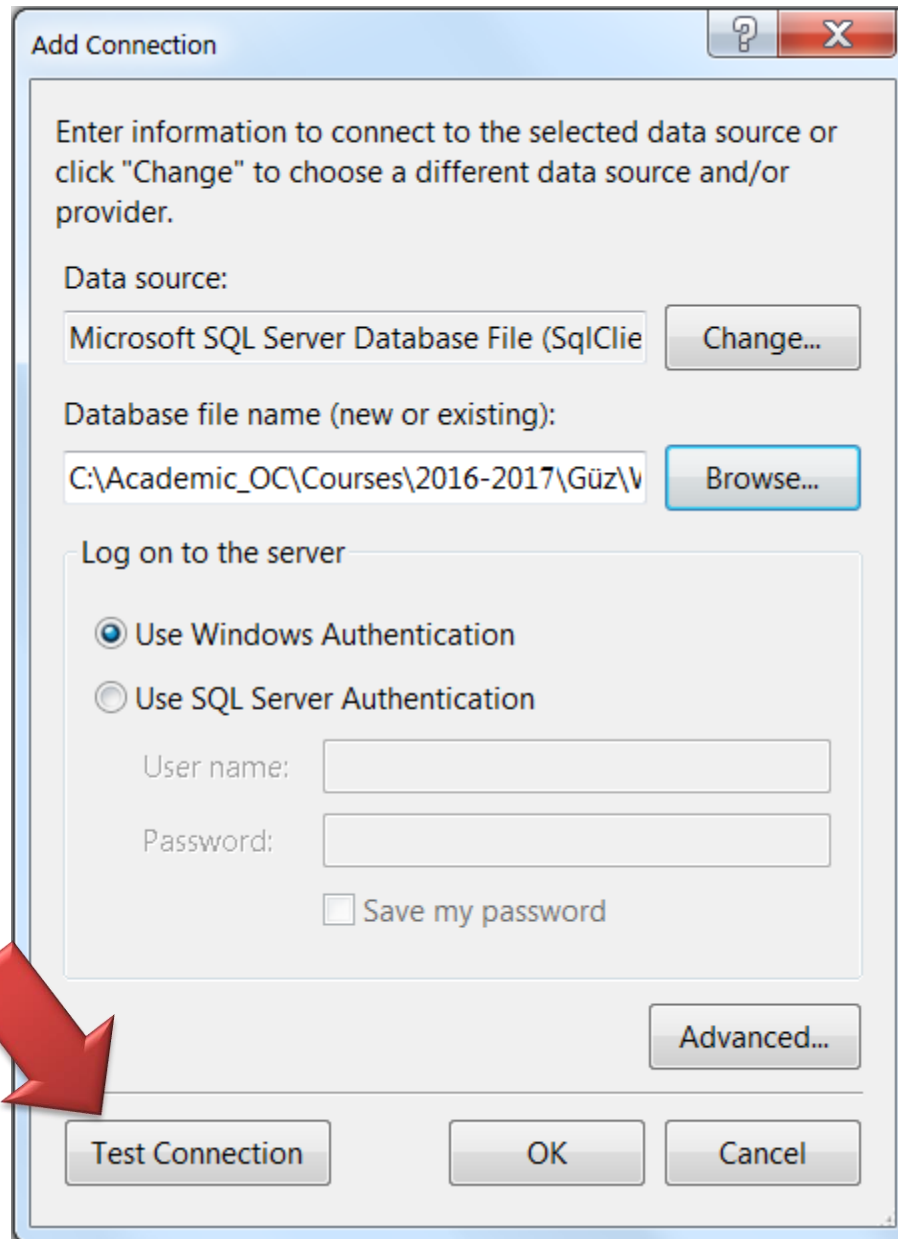
Advanced...

Test Connection OK Cancel

# Uygulama - LINQ



# Uygulama - LINQ



**Add Connection**

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 Database File (SqlClie) Change...

Database file name (new or existing):  
C:\Academic\_OC\Courses\2016-2017\Güz\W Browse...

Log on to the server

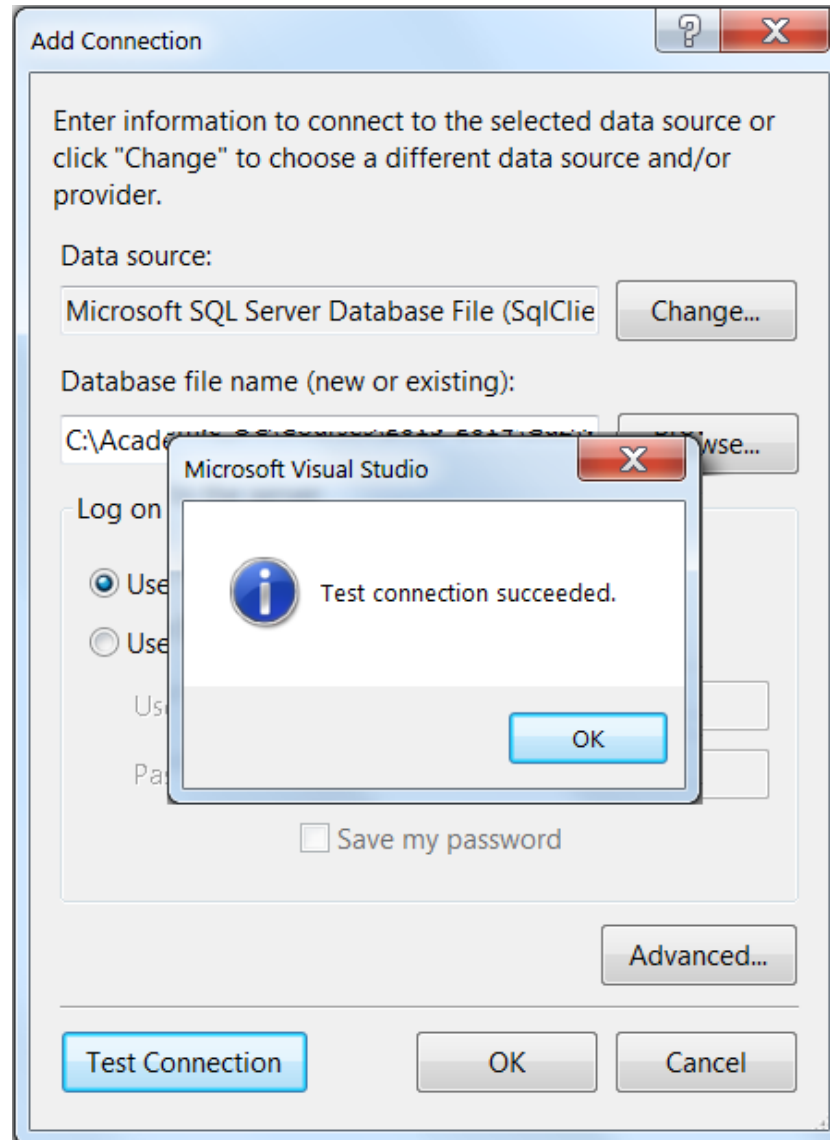
☒ Use Windows Authentication  
☐ Use SQL Server Authentication

User name:   
Password:   
☐ Save my password

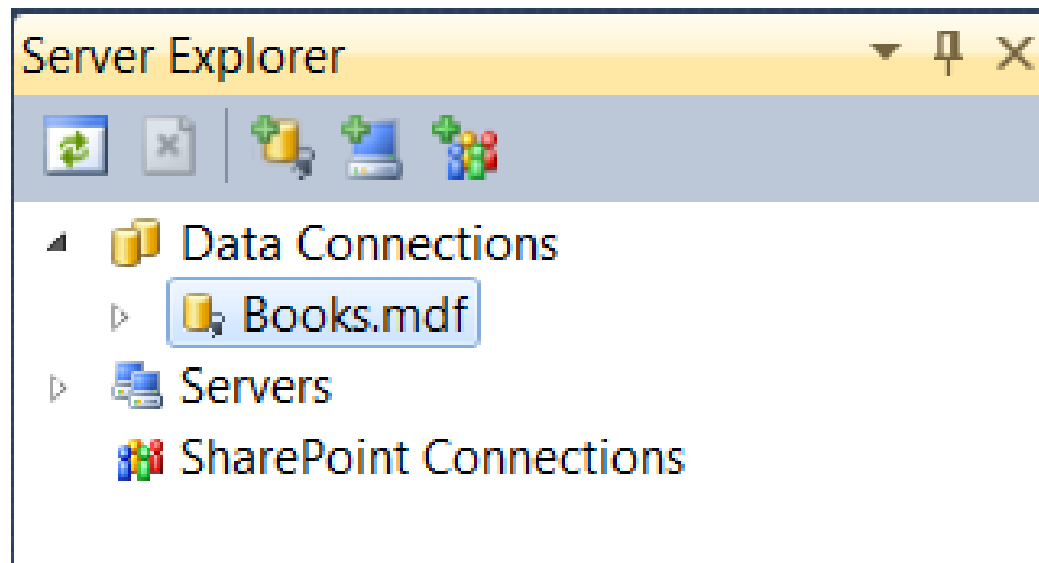
Advanced...

Test Connection OK Cancel

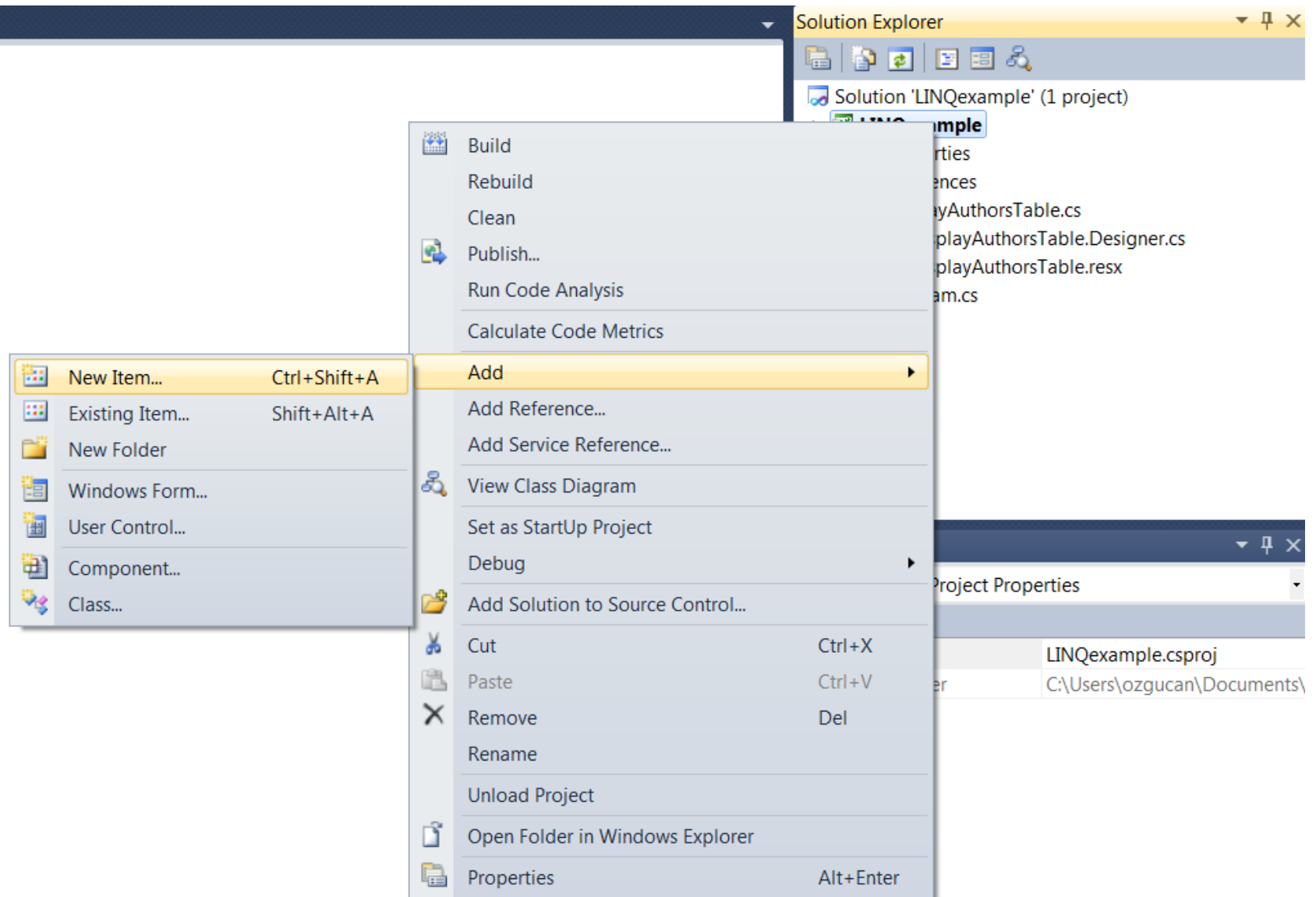
# Uygulama - LINQ



# Uygulama - LINQ

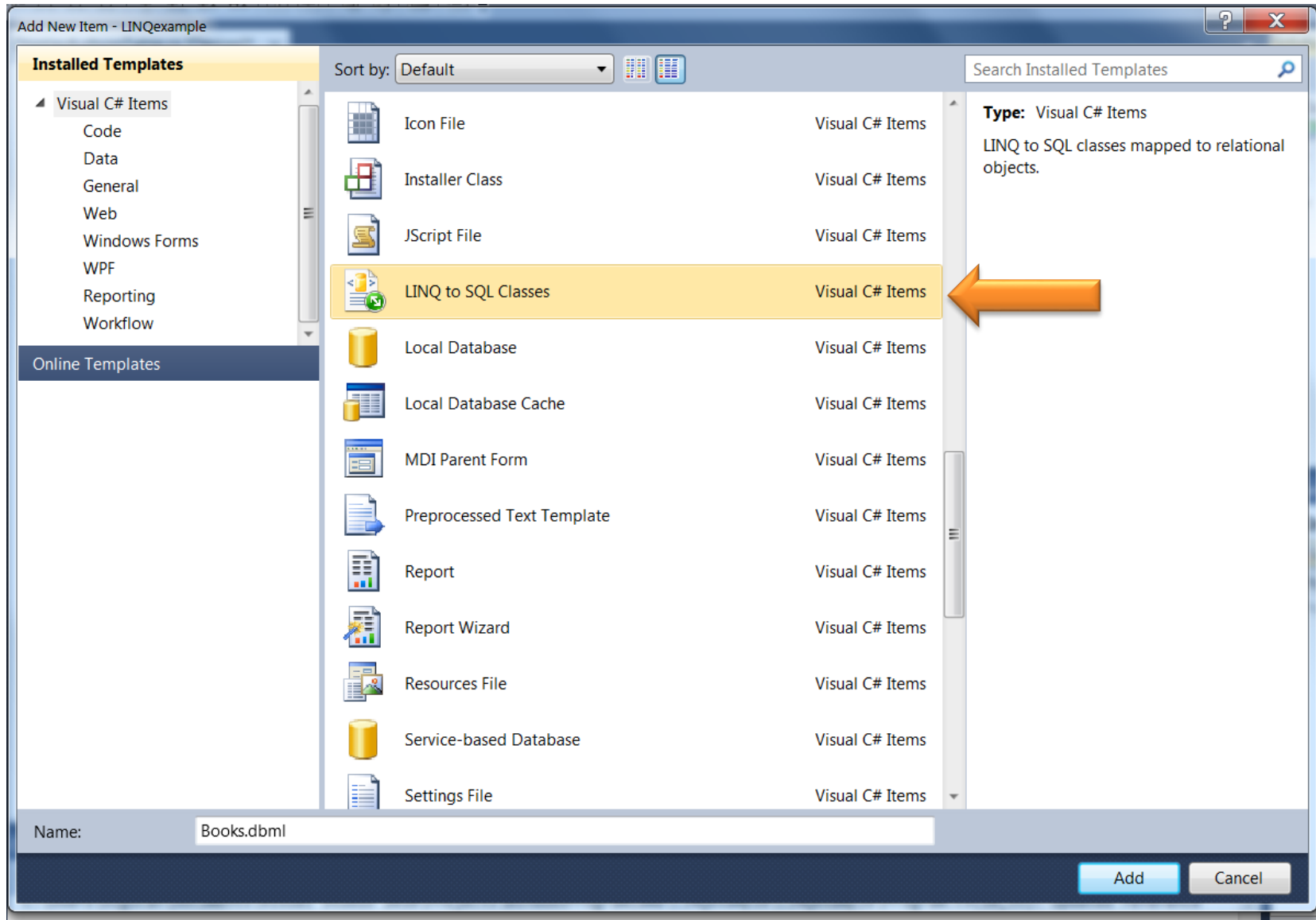


# Uygulama - LINQ



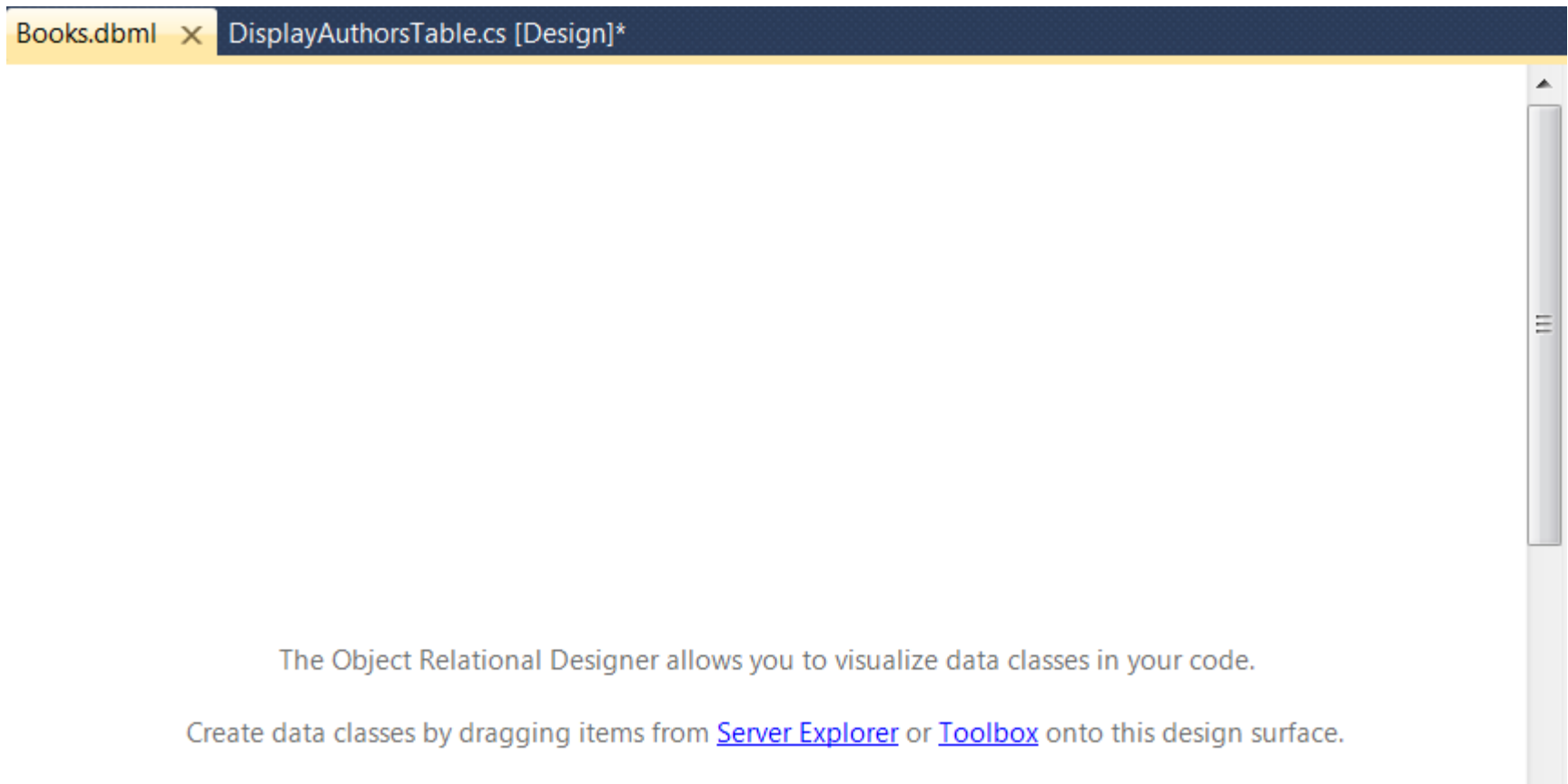


# Uygulama - LINQ

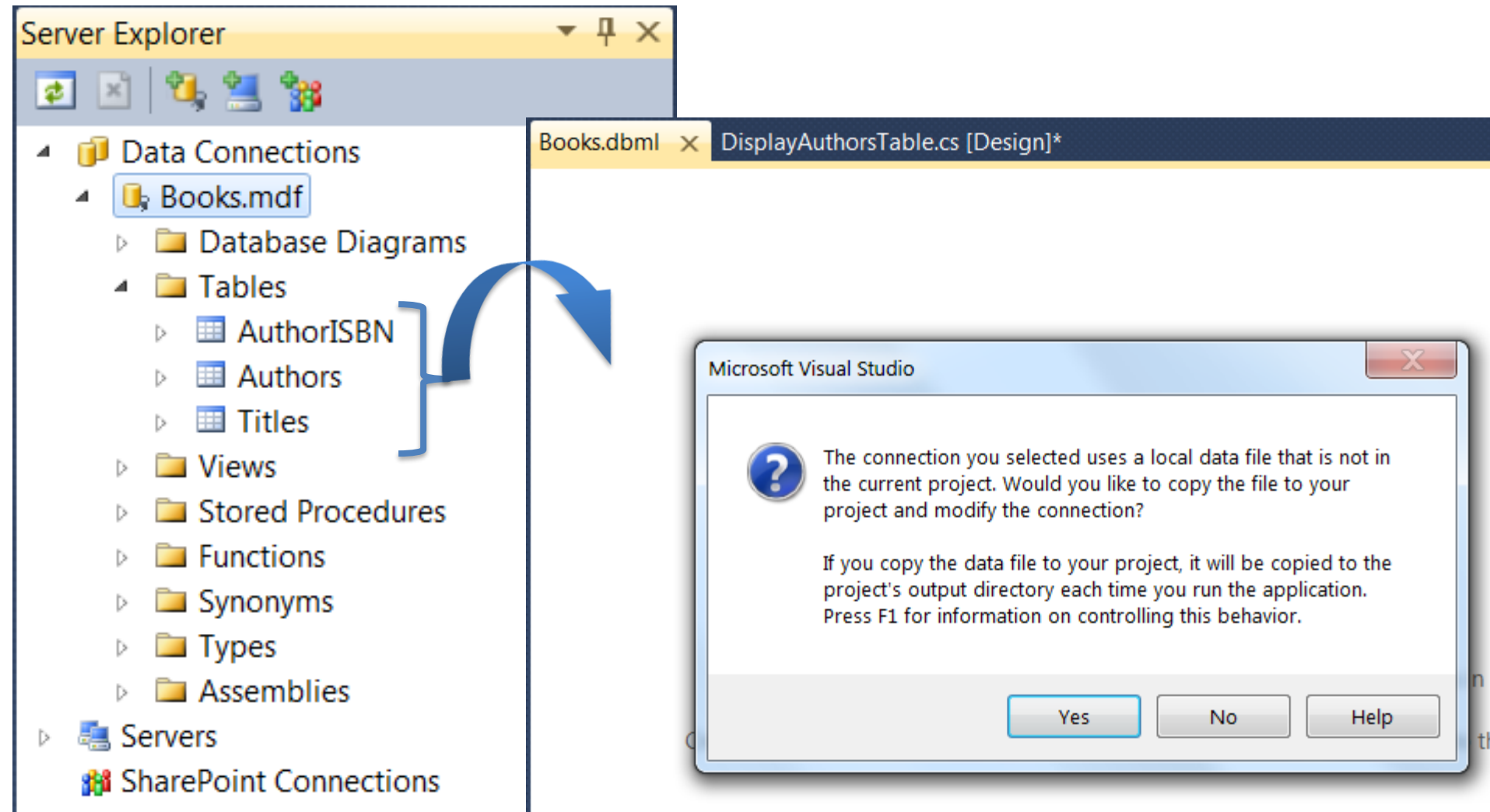


# Uygulama - LINQ

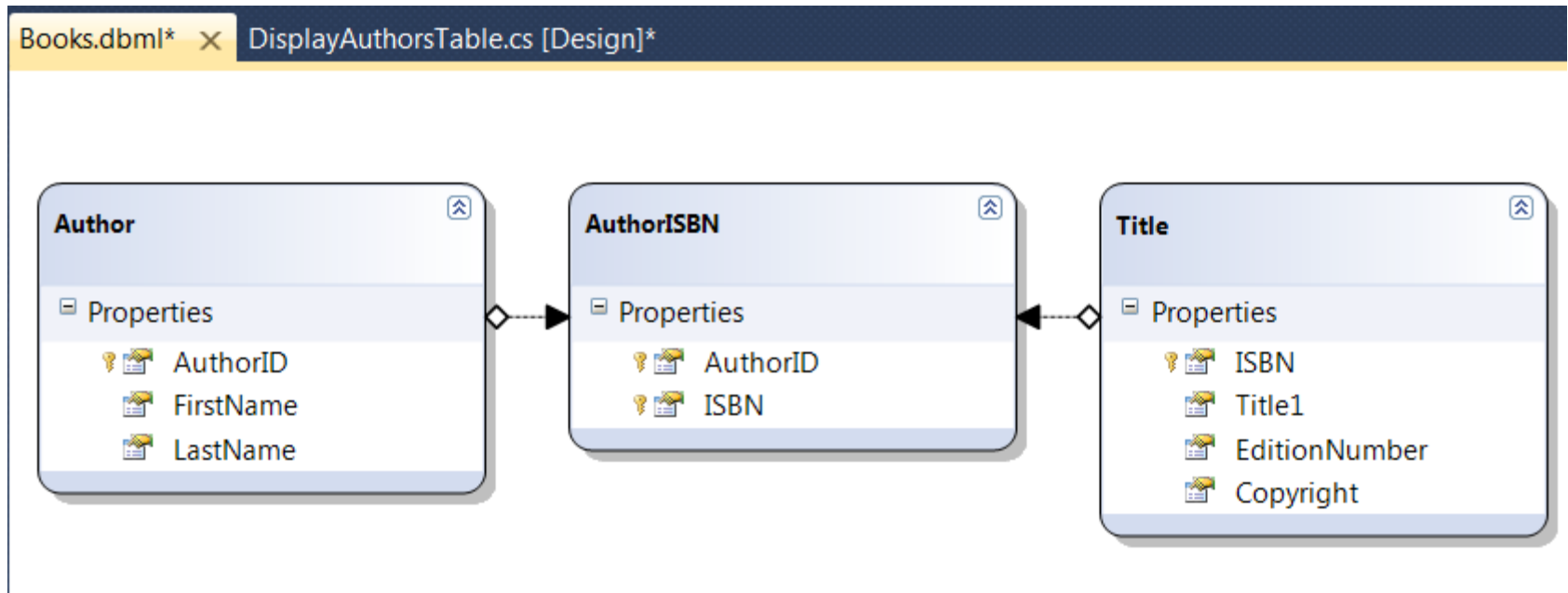
- **Object Relational Designer window**



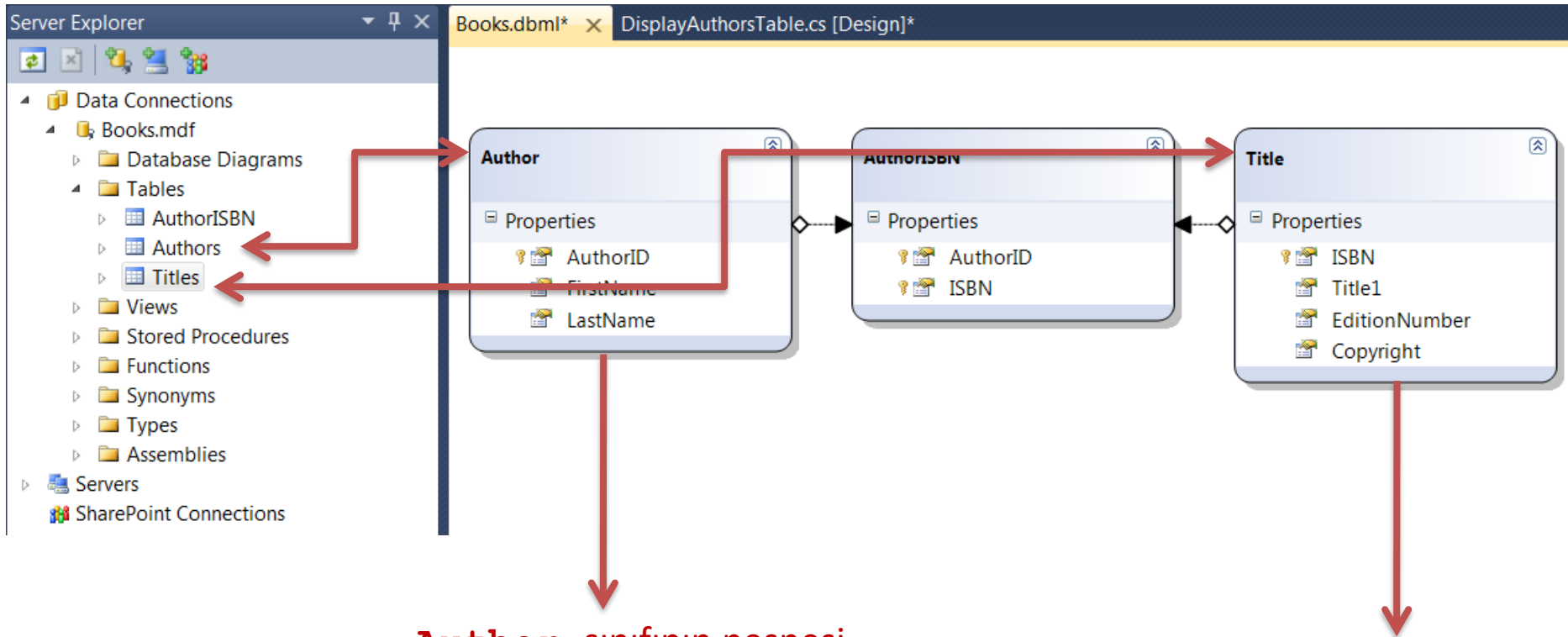
# Uygulama - LINQ



# Uygulama - LINQ

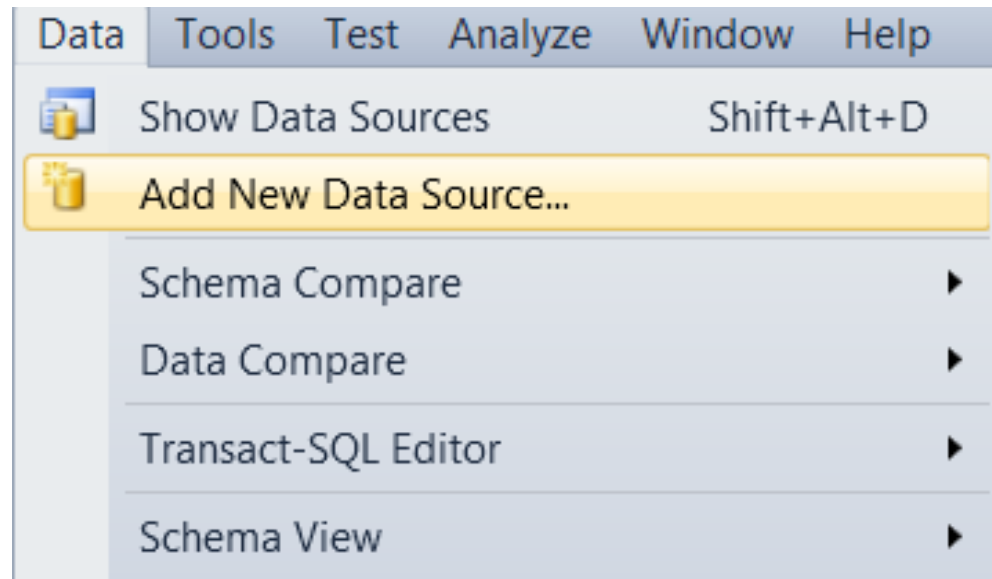


# Uygulama - LINQ



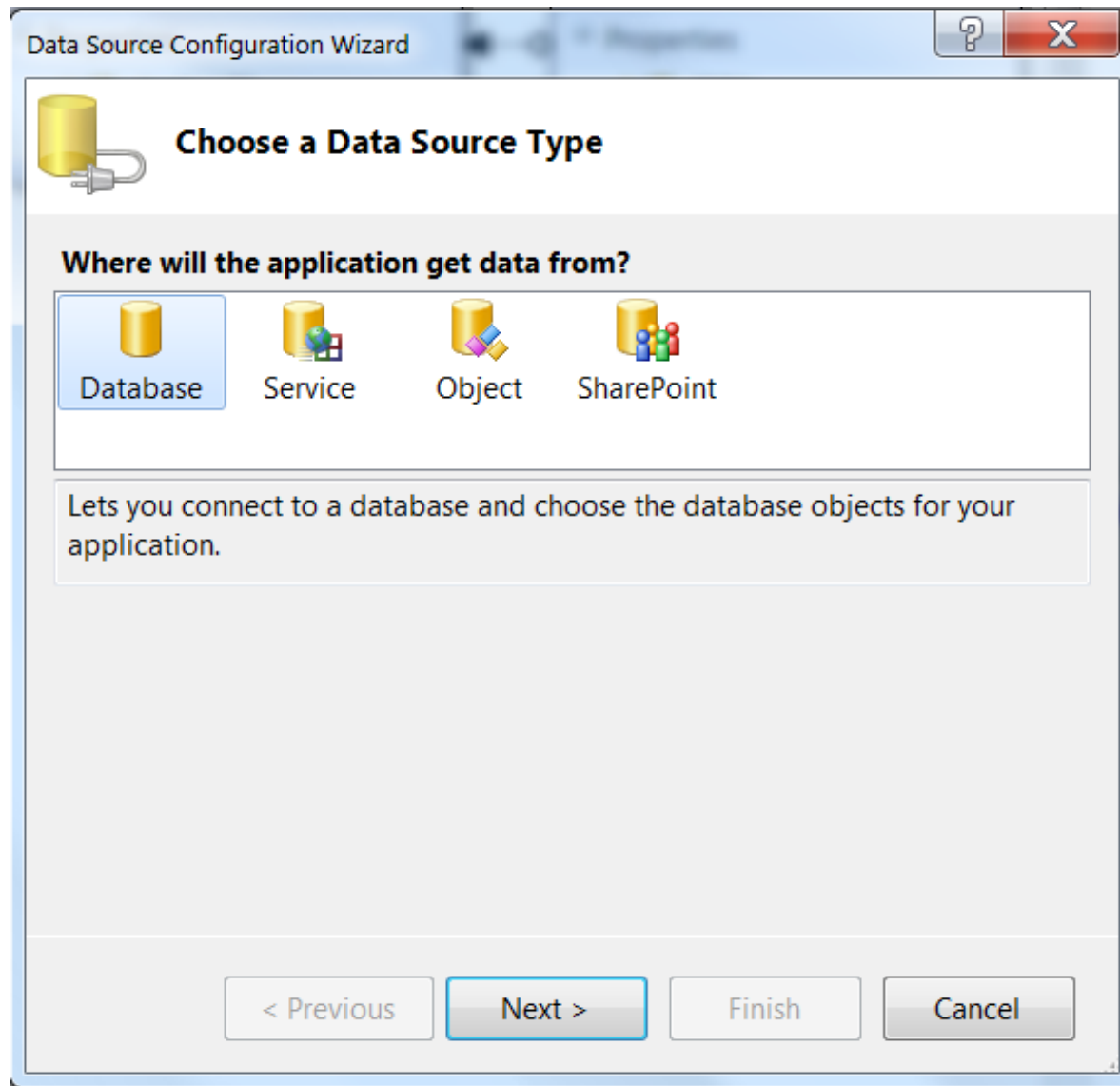
# Uygulama - LINQ

**Author *LINQ to SQL* sınıfının *Data Source* olarak eklenmesi**



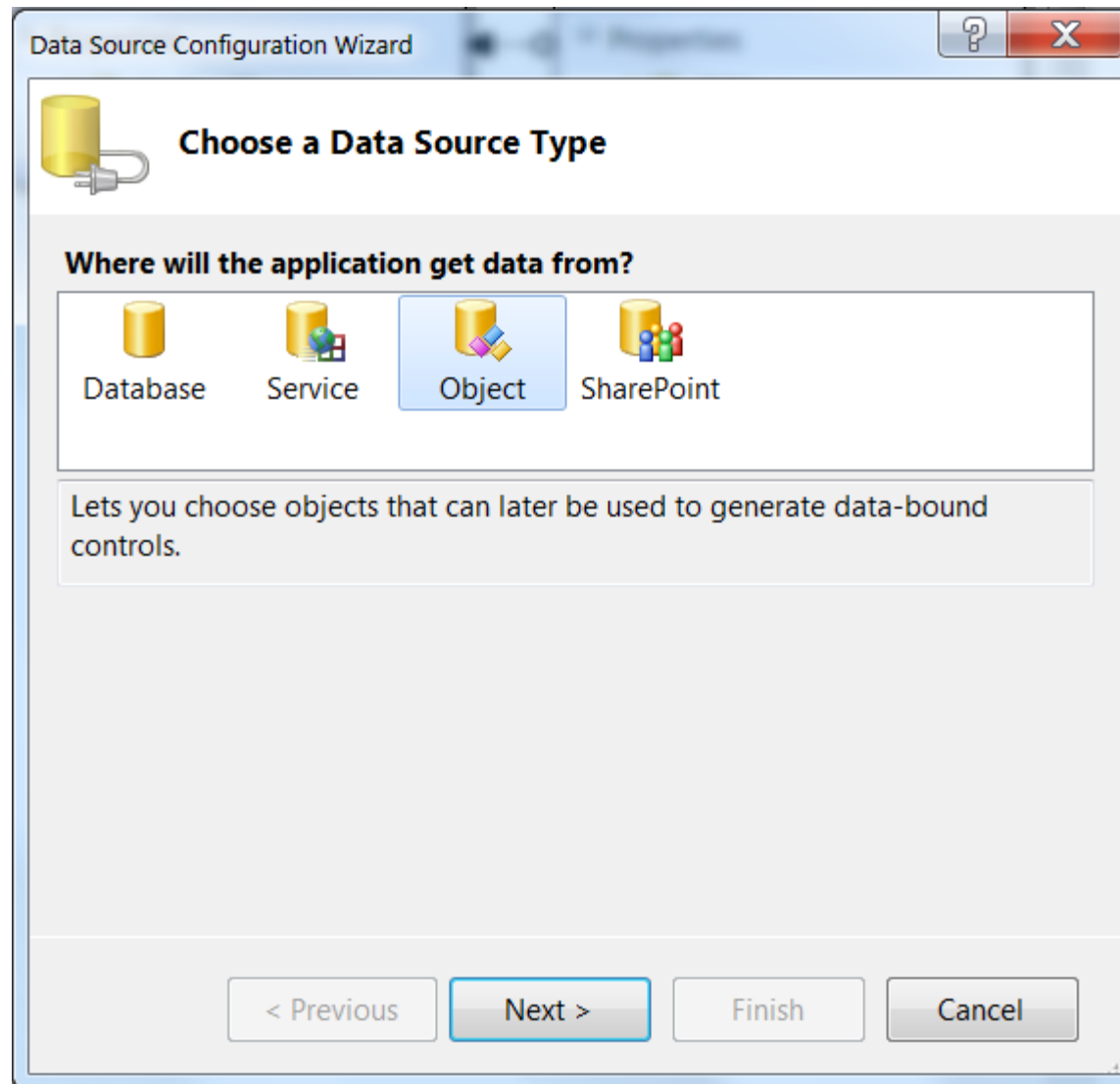
Visual Studio 2015 → **View > Other Windows > Data Source > Add New Data Source**

# Uygulama - LINQ



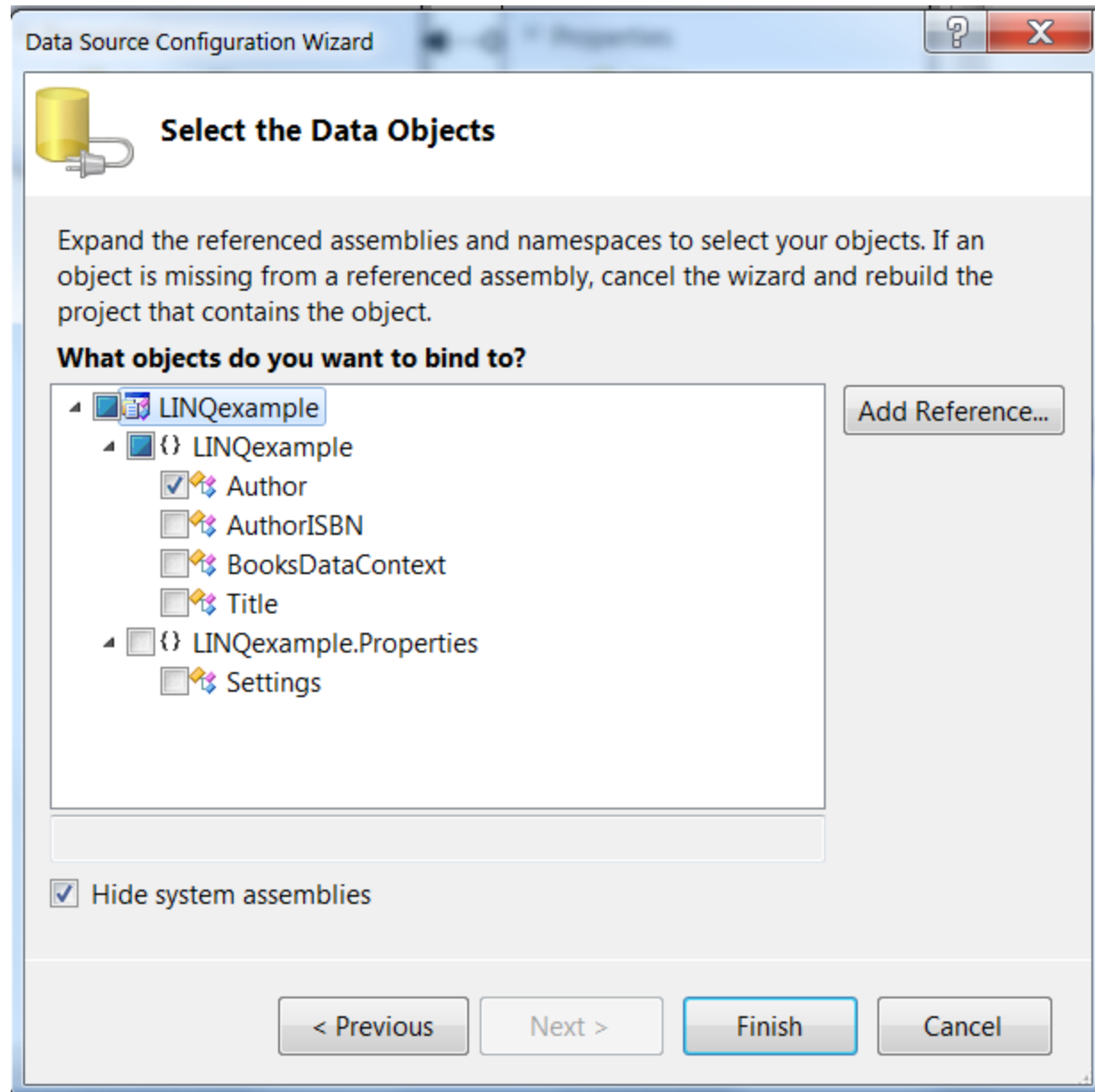
*LINQ to SQL* sınıfı, veritabanında ki tabloları temsil eden **nesnelerin yaratılması için** kullanılmaktadır.

## Object Data Source

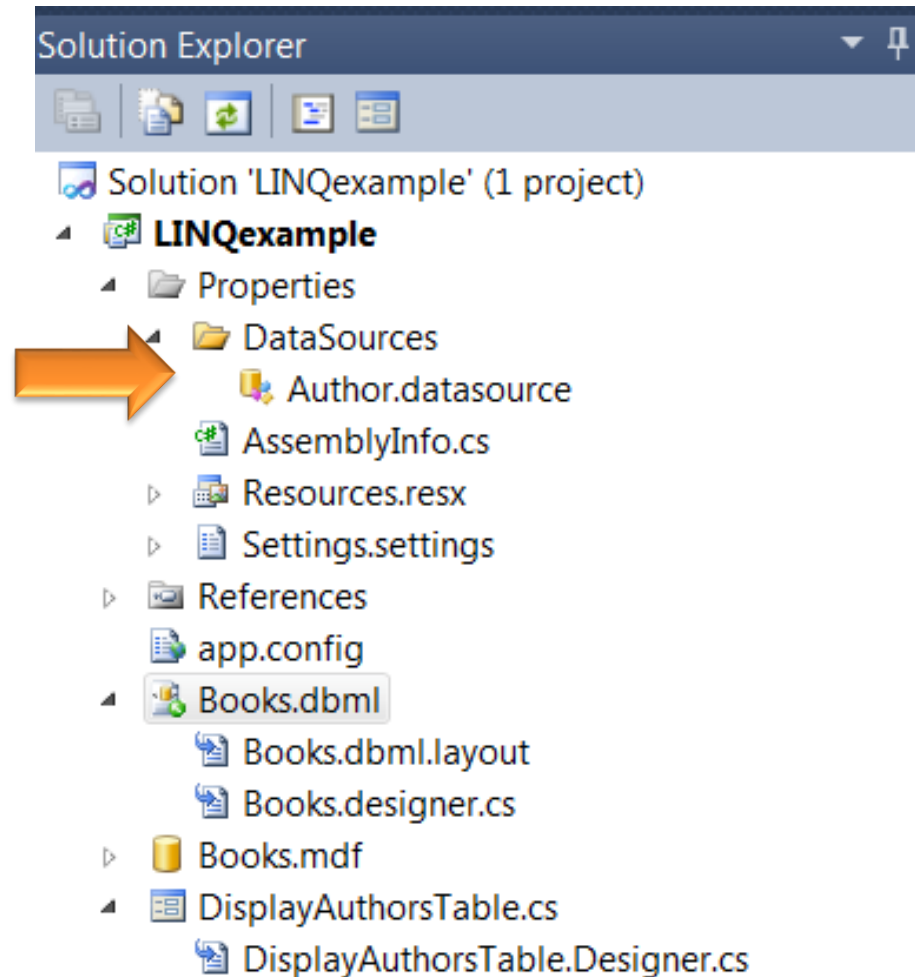


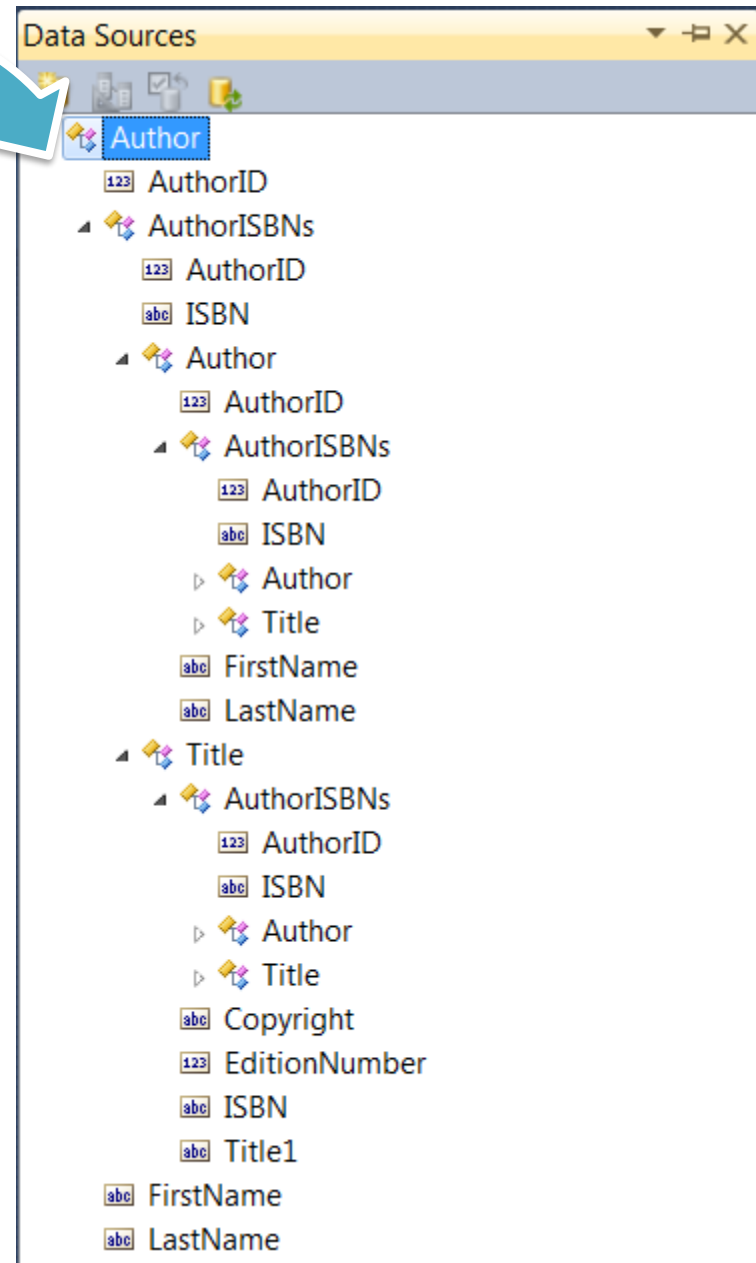
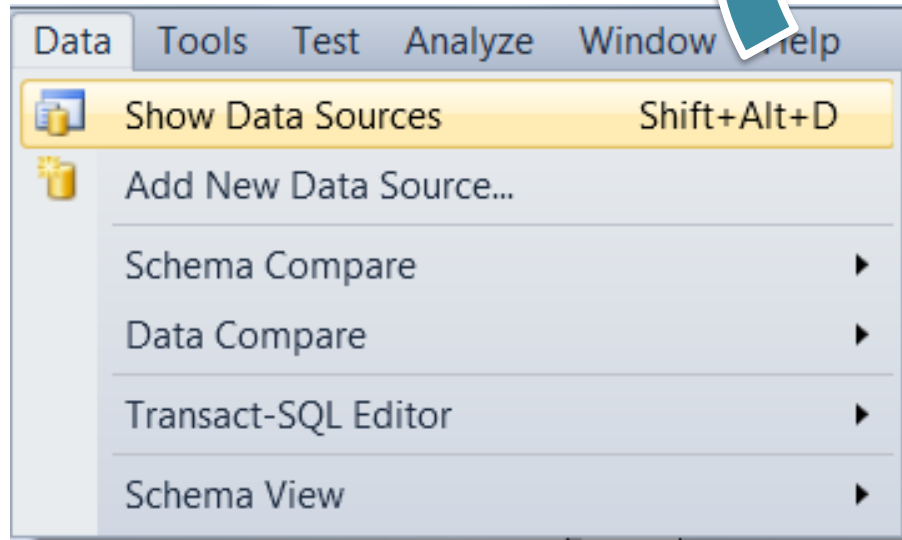


# Uygulama - LINQ



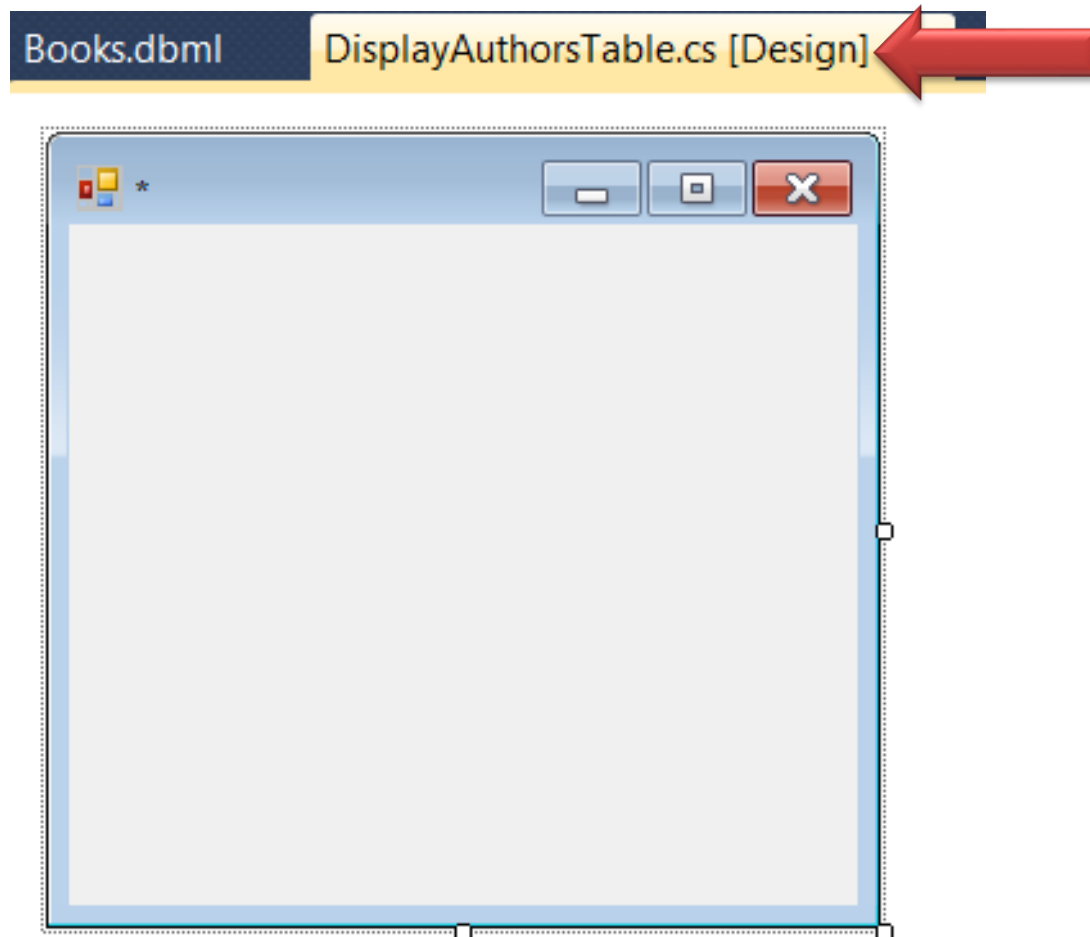
# Uygulama - LINQ



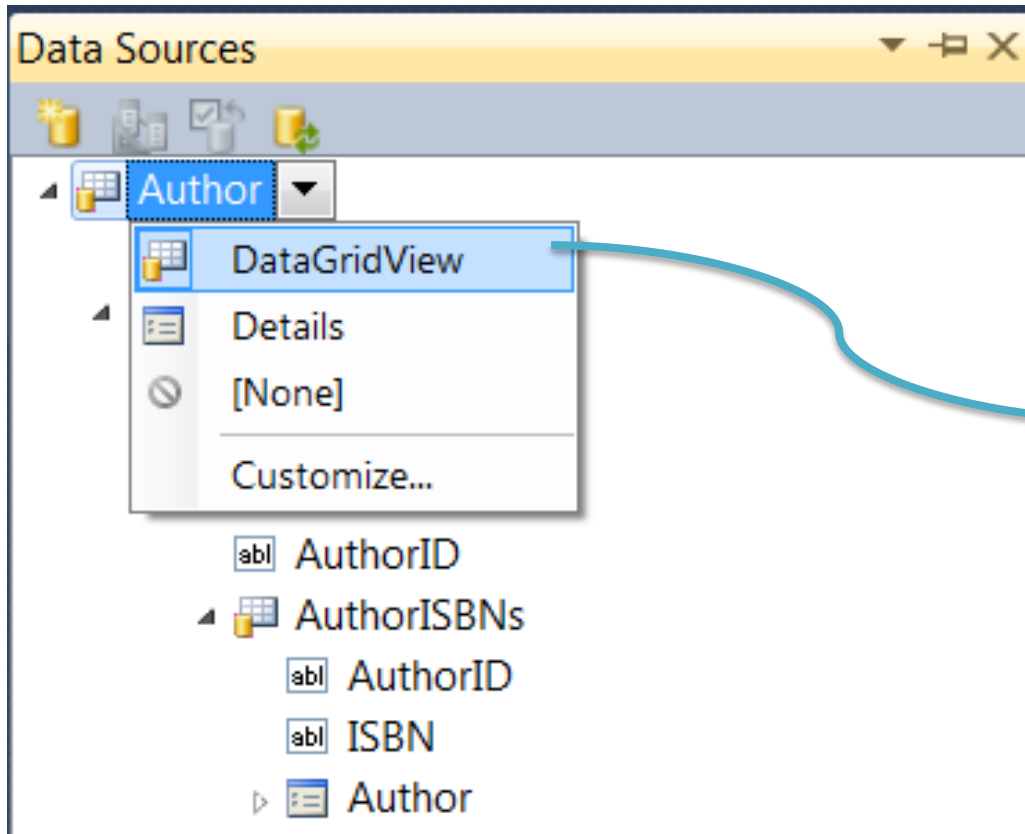


# Uygulama - LINQ

**Authors tablosunun verilerini görüntüleme**



# Uygulama - LINQ

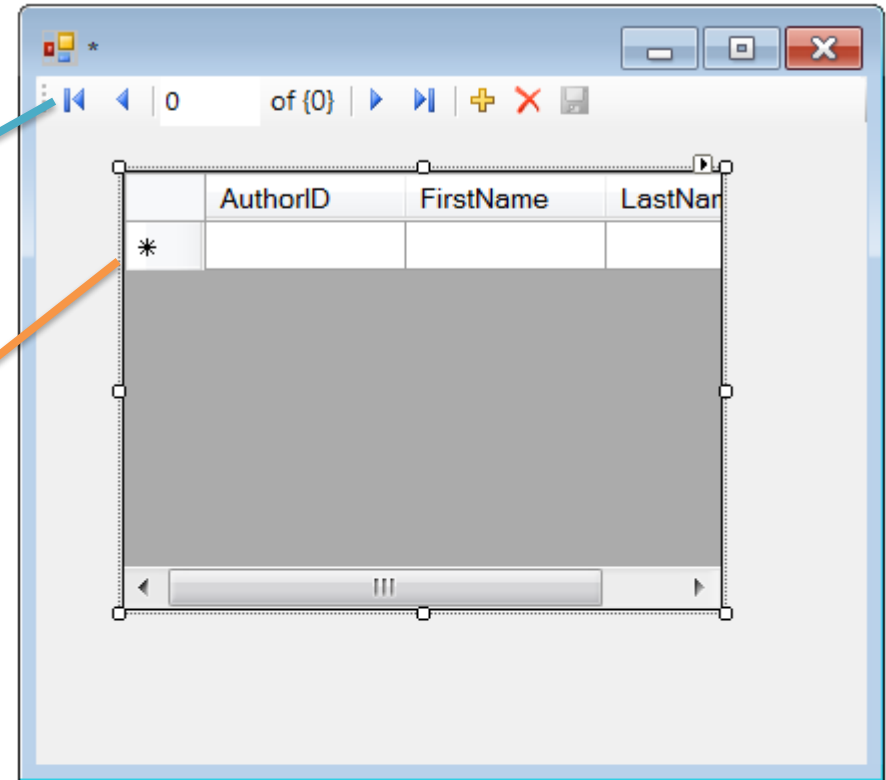


Veriyi  
görmüleyecek  
ve veri ile  
etkileşim kuracak  
olan  
GUI kontrolü

**Author** node'unu  
**Data Sources'dan Form'a**  
sürükleyip bırakın.

**BindingNavigator**  
için  
GUI kontrolleri

**Authors tablosu**  
sütun adlarının  
**DataGridView** ile  
görüntülenmesi



## DataBindingNavigator

Button'lar aracılığı ile:

- Girdiler arasında dolaşma
- Girdi ekleme-silme
- Değişiklikleri kaydetme

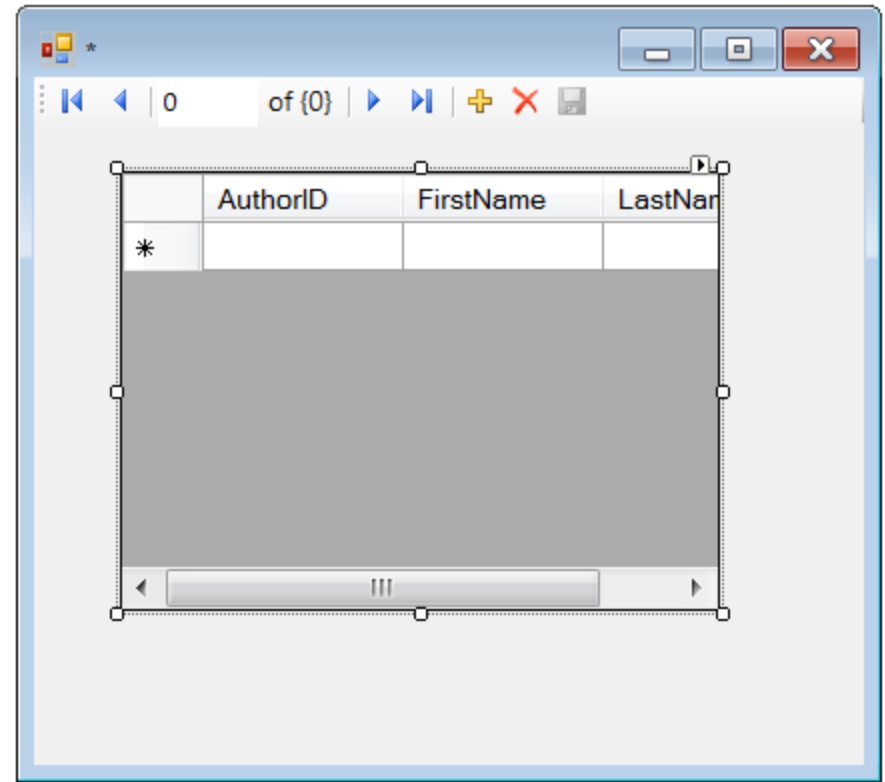
işlemlerini gerçekleştirir.



authorBindingSource



authorBindingNavigator



## BindingSource

Verinin  
data source  
ve  
data-bound kontrolü  
arasındaki transferi ile  
ilgilenir.



authorBindingSource

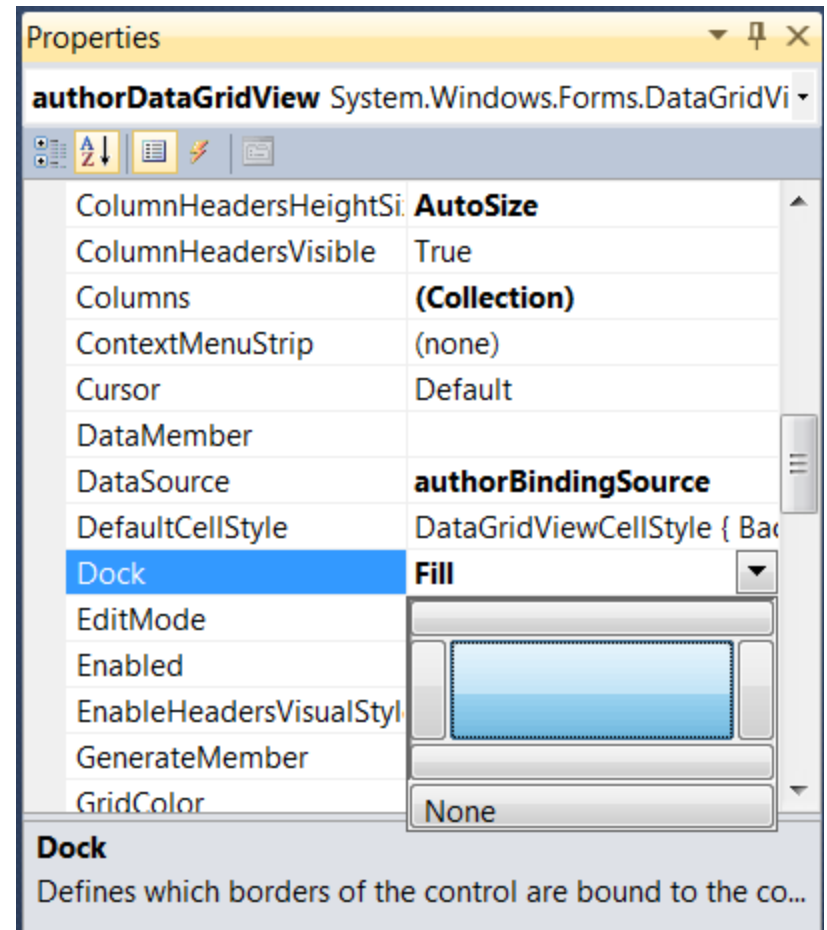


authorBindingNavigator

# Uygulama - LINQ

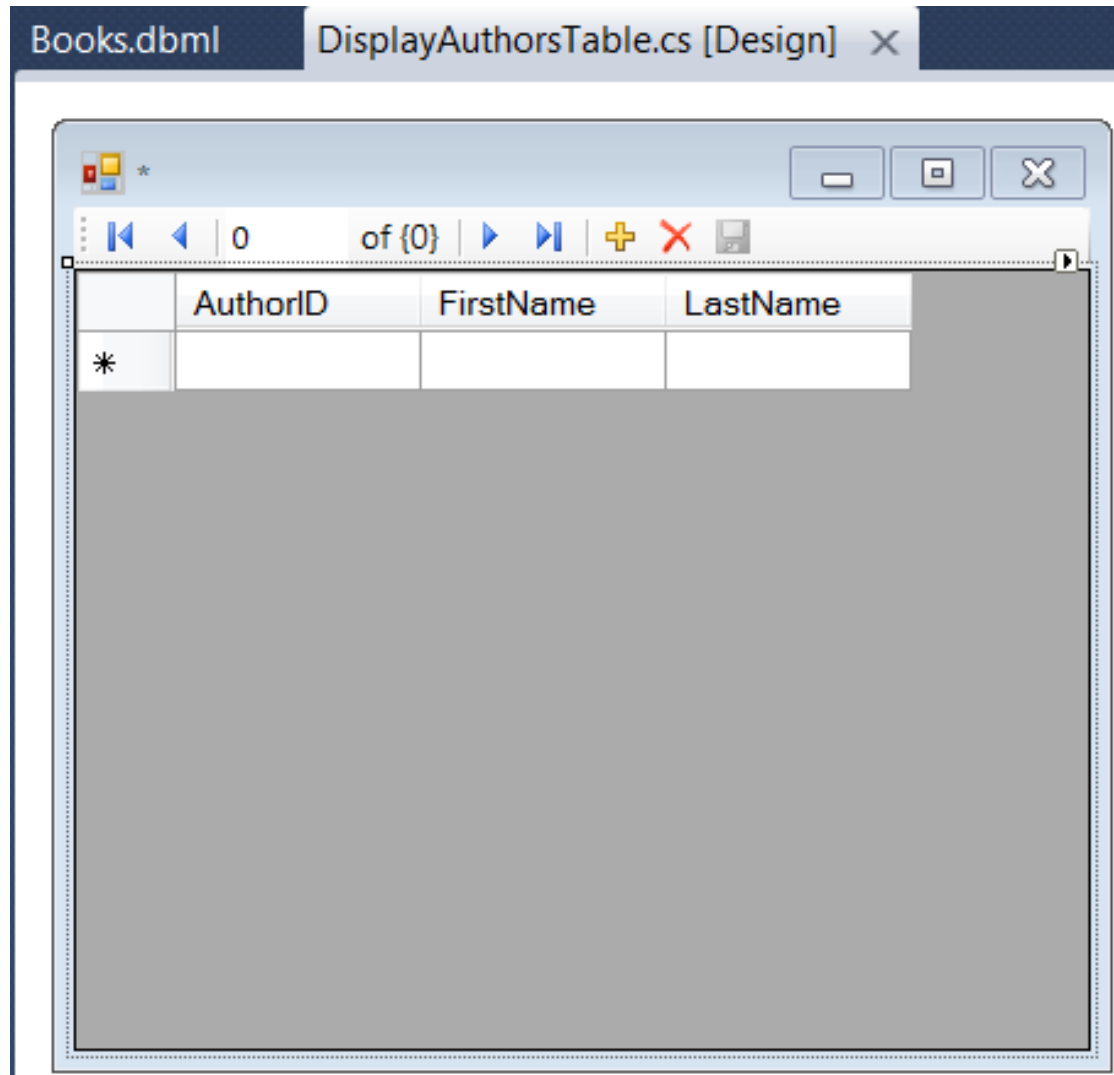
- DataGridView'un tüm pencereyi işgal etmesi için:

– Dock = Fill



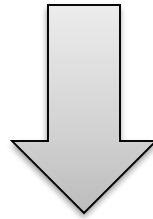


# Uygulama - LINQ



# Uygulama - LINQ

Uygulamanın veritabanı ile  
etkileşim kurması için



**BooksDataContext**'in  
**authorBindingSource**'a bağlanması

# Uygulama - LINQ

```
// LINQ to SQL data context
private BooksDataContext database = new BooksDataContext();

//load data from database into DataGridView
private void DisplayAuthorsTable_Load(object sender, EventArgs e)
{
    // use LINQ to order the data for display
    authorBindingSource.DataSource = from author in database.Authors
                                     orderby author.AuthorID
                                     select author;
}
```

# Uygulama - LINQ

- Default → **Save Data = Disabled**

**Save Data**

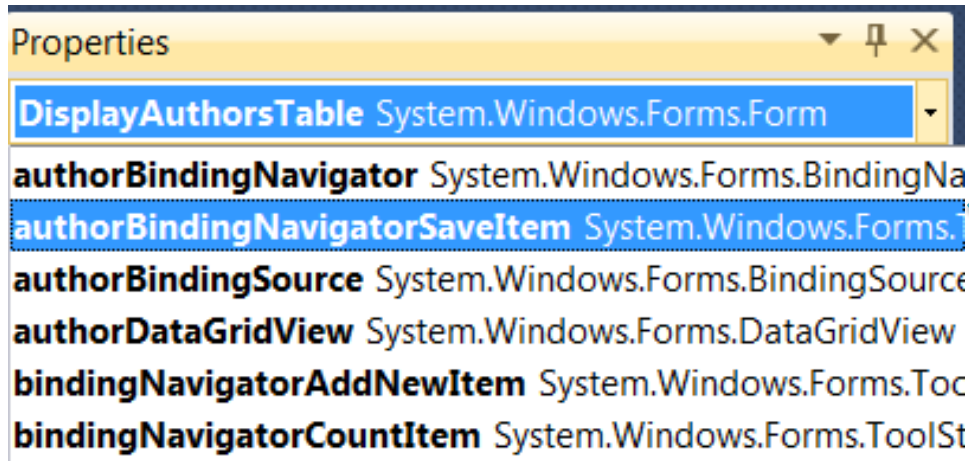


**Enabled**

The screenshot shows the Visual Studio IDE with the 'DisplayAuthorsTable.cs [Design]' window active. The window displays a table with columns 'AuthorID', 'FirstName', and 'LastName'. A context menu is open over the table, showing options like 'View Code', 'Set Image...', 'Enabled', 'Alignment', 'DisplayStyle', 'Convert To', 'Insert', 'Select', 'Cut', 'Copy', 'Paste', 'Delete', 'Document Outline', and 'Properties'. The 'Enabled' option is highlighted. A red arrow points from the 'Enabled' text to the 'Save Data' icon in the table's toolbar.

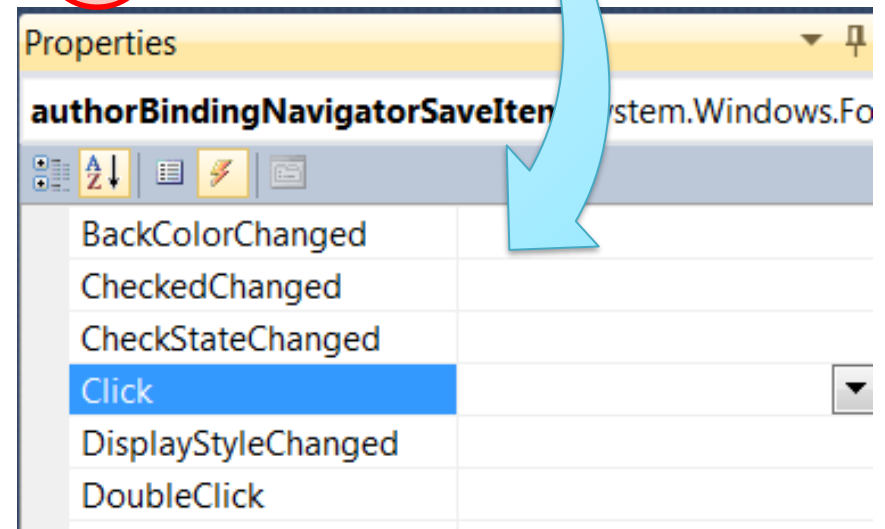
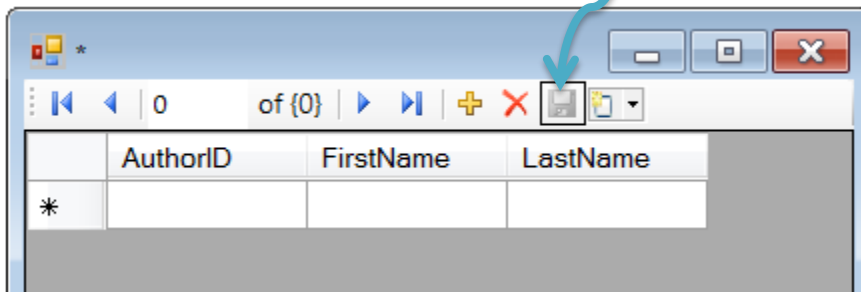
	AuthorID	FirstName	LastName
*			

# Uygulama - LINQ



Veri üzerinde  
yapılan  
değişiklerin  
kaydedilmesi için

2 ya da  
double-lik



# Uygulama - LINQ

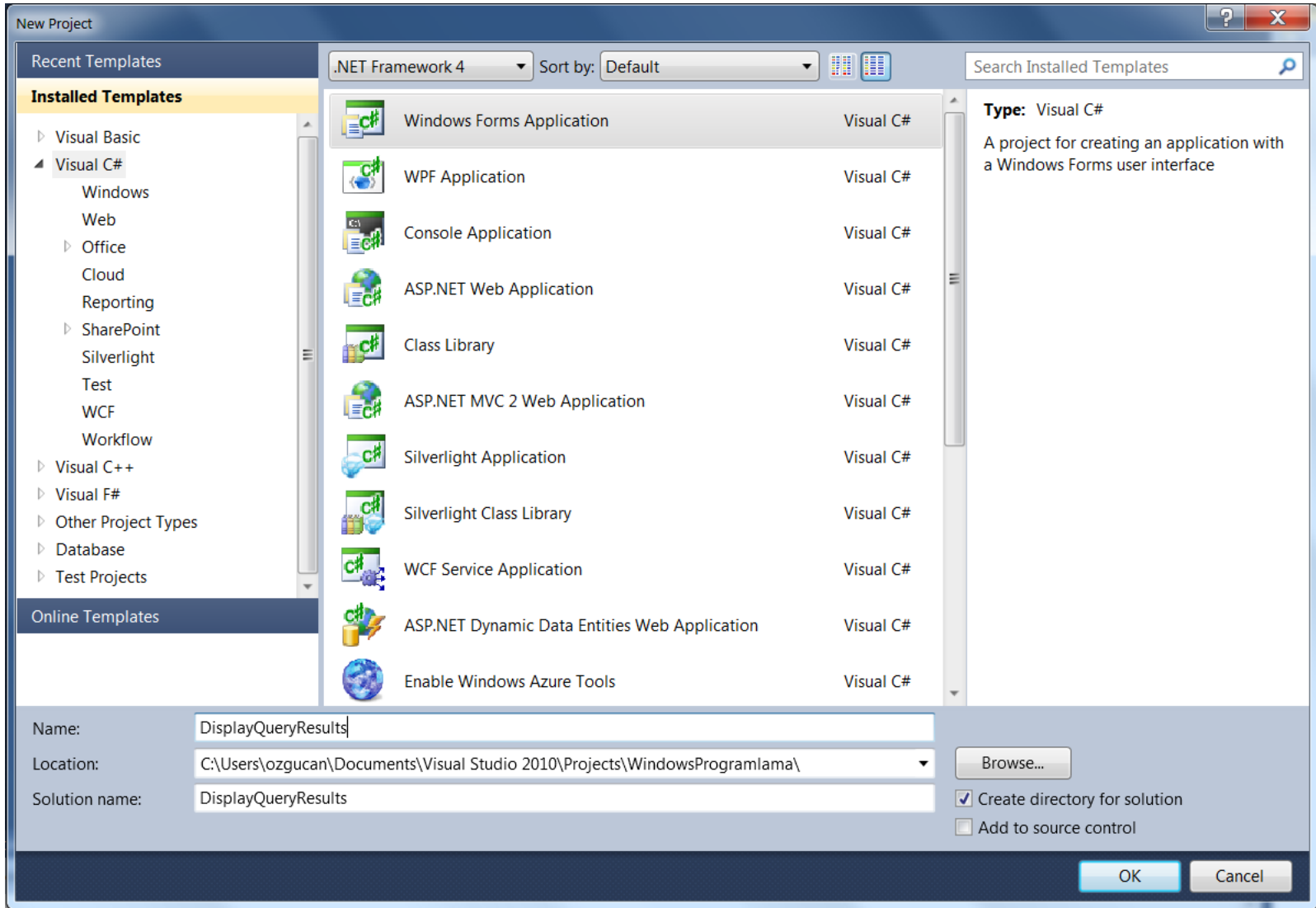
```
// click event handler for the Save Button in the BindingNavigator saves the changes made to the data
private void authorBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    Validate(); // validate input fields
    authorBindingSource.EndEdit(); // indicate edits are complete
    database.SubmitChanges(); // write changes to database file
}
```

Form'daki bütün  
kontroller "**validate**"  
edilir ve "**validation**"  
olayı ile ilgili bir olay-  
işleyici varsa çalıştırılır.

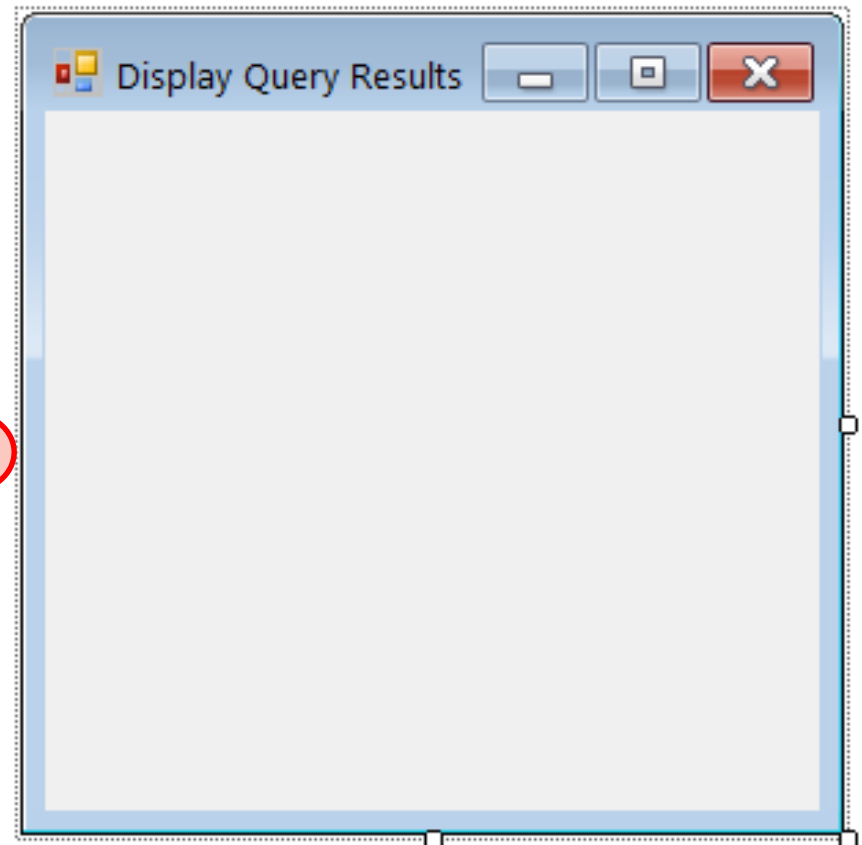
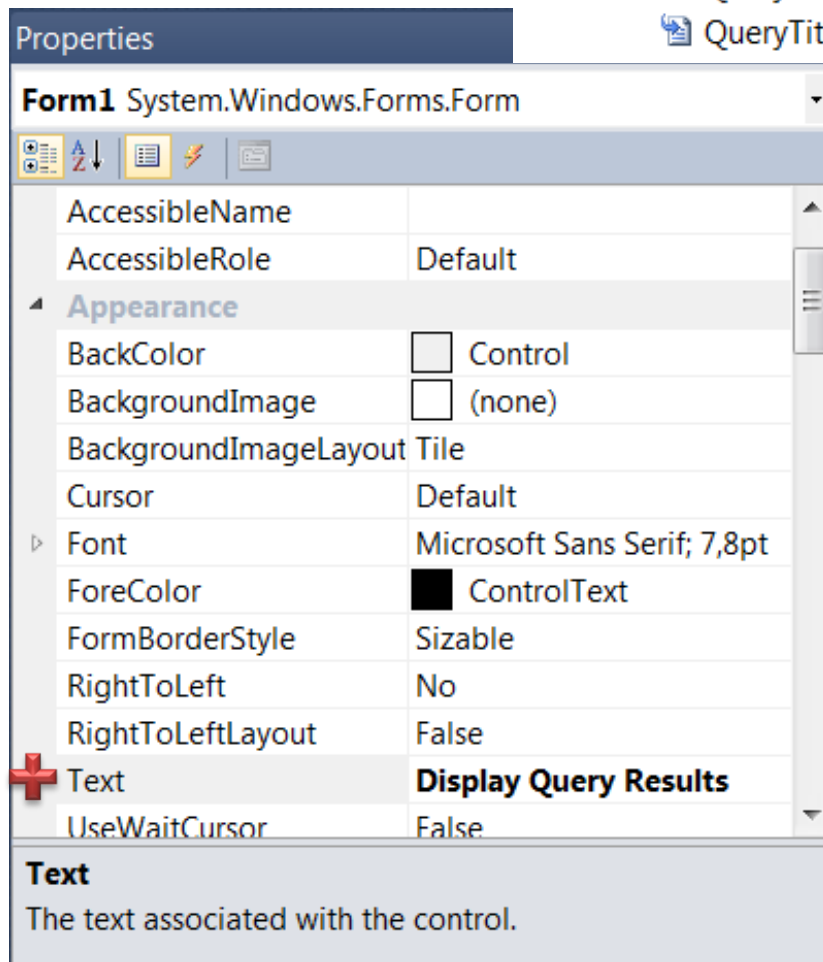
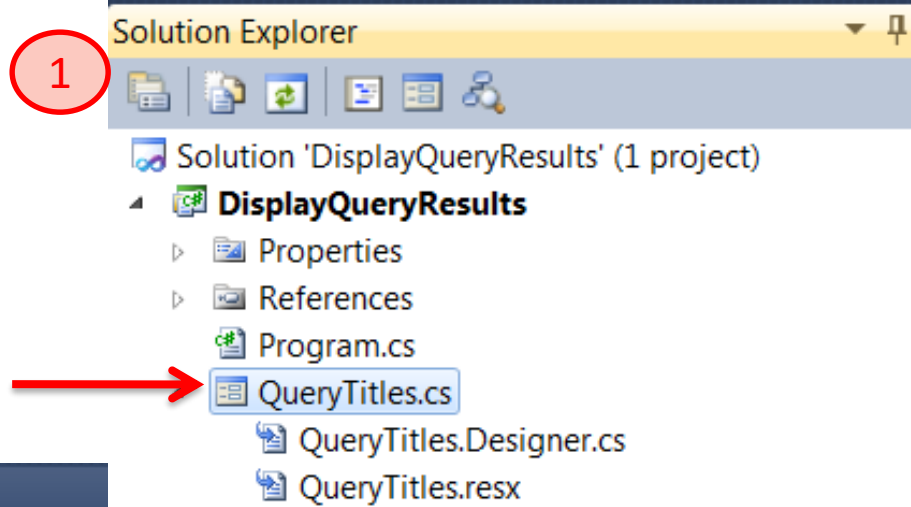
1 of 4			
	AuthorID	FirstName	LastName
▶	1	Elif	Safak
	2	Orhan	Pamuk
	3	Ayşe	Kulin
	4	Ahmet	Ümit
*			

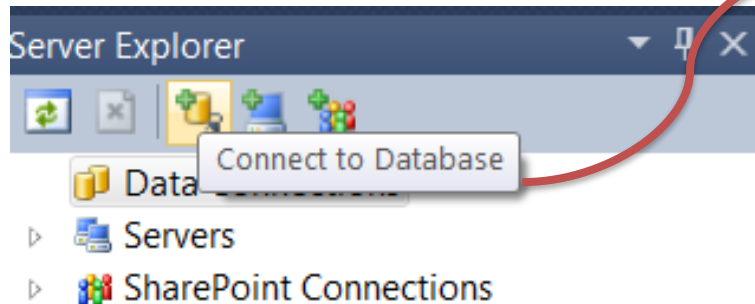
5 of 5			
	AuthorID	FirstName	LastName
	1	Elif	Safak
	2	Orhan	Pamuk
	3	Ayşe	Kulin
	4	Ahmet	Ümit
✎	5	Zülfü	Livaneli
*			

# Uygulama – LINQ Sorgu Sonuçları









**Add Connection**

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 Database File (SqlClie) Change...

Database file name (new or existing):  
 Browse...

Log on to the server

☒ Use Windows Authentication

☐ Use SQL Server Authentication

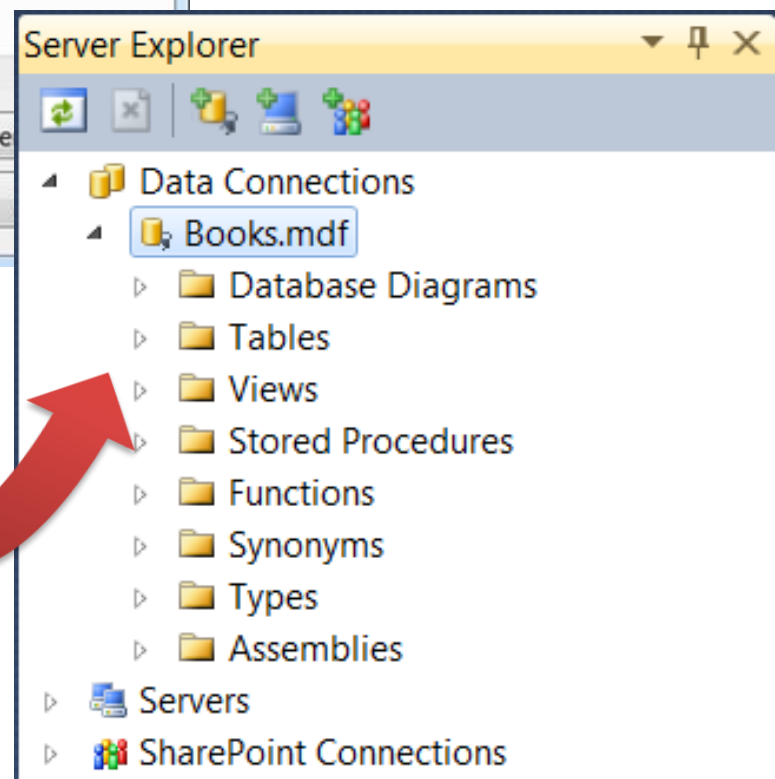
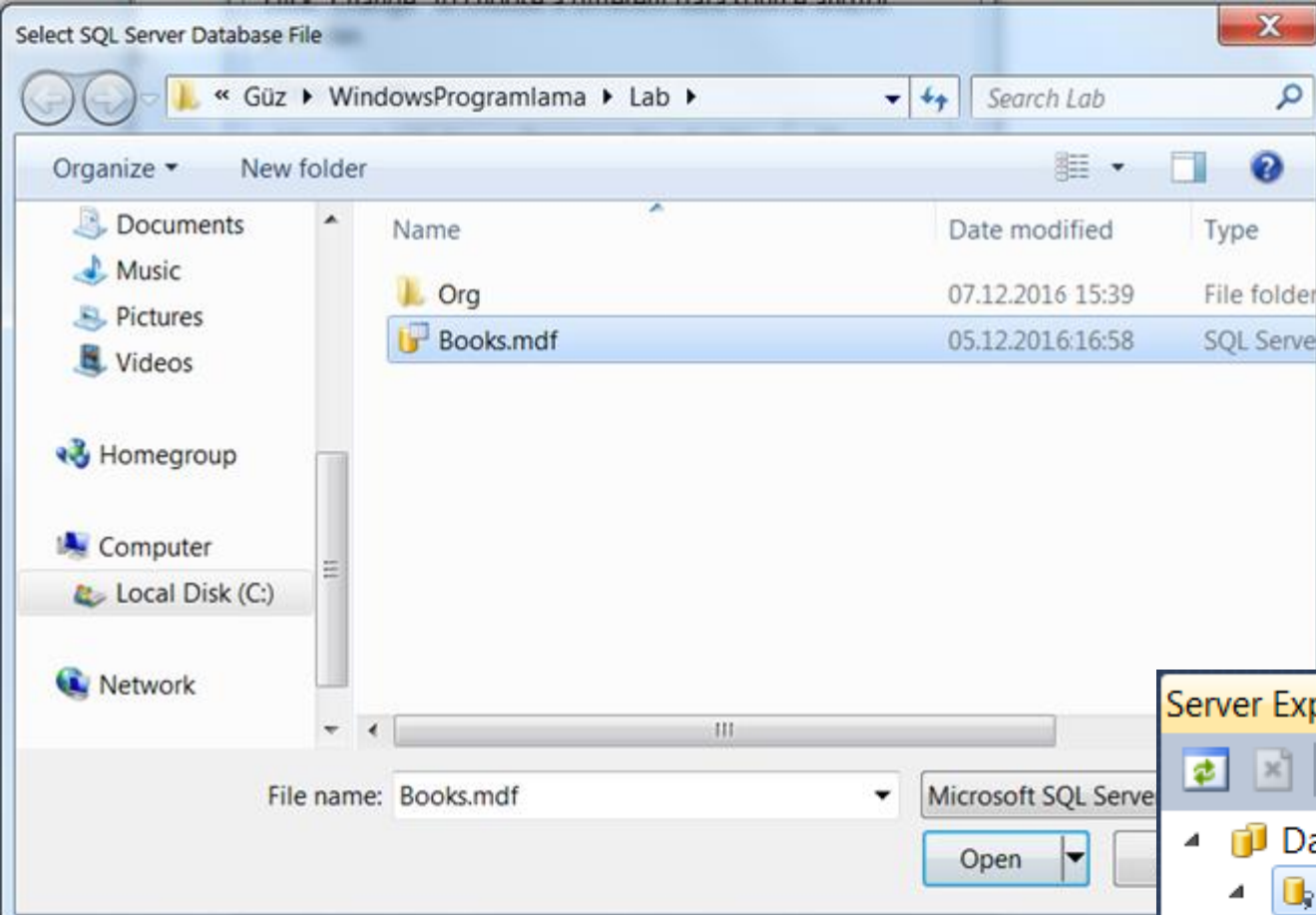
User name:

Password:

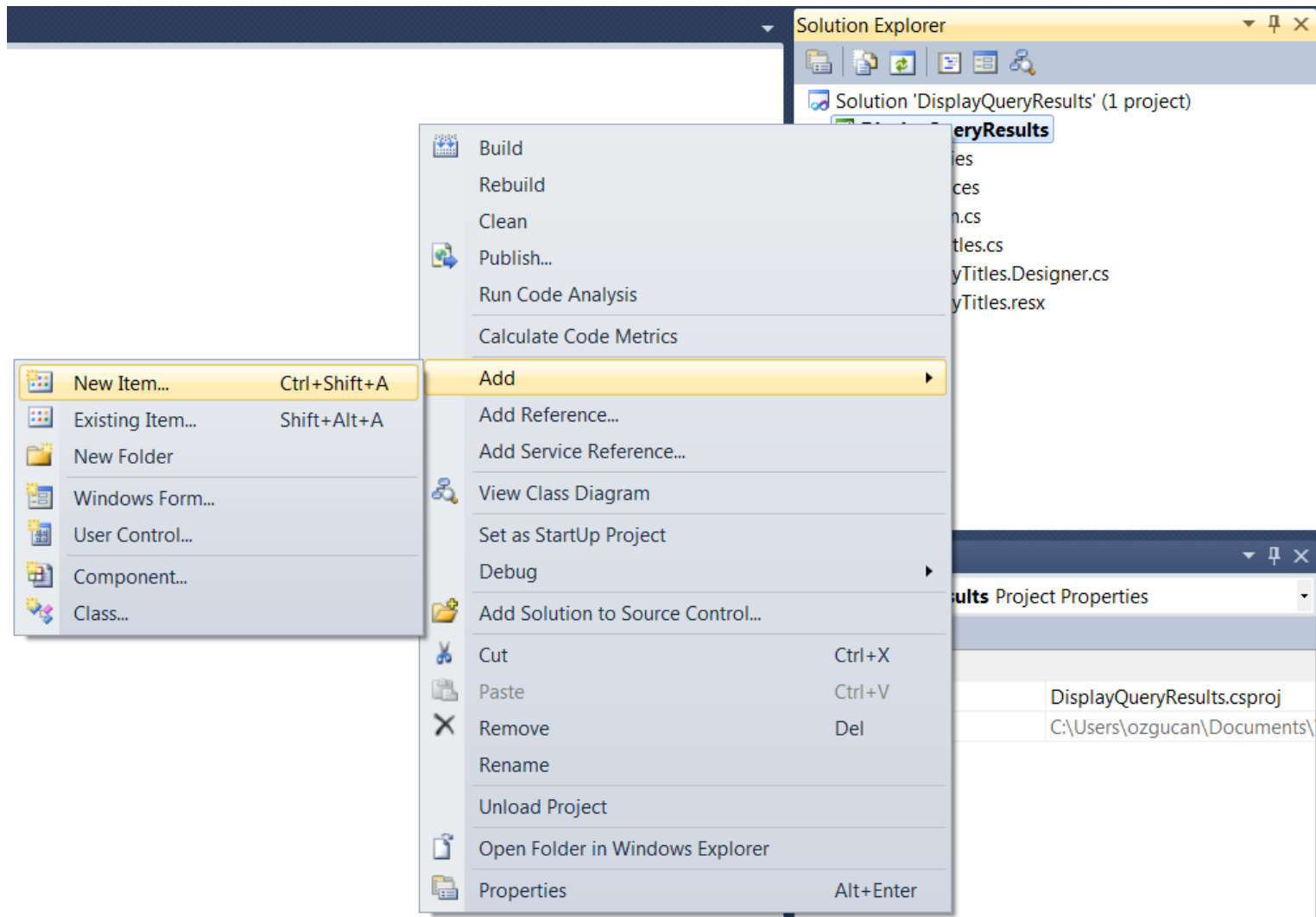
☐ Save my password

Advanced...

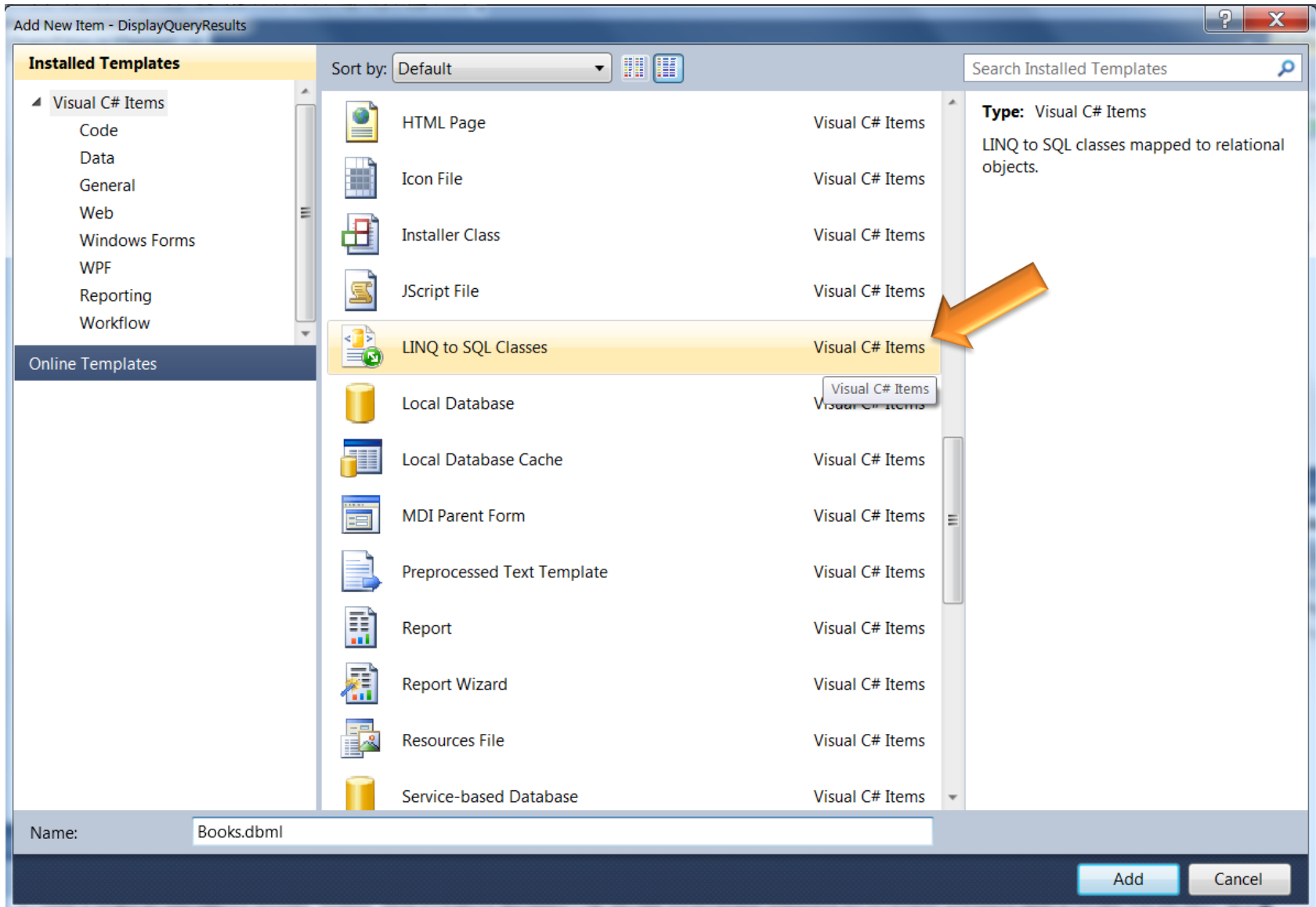
Test Connection OK Cancel

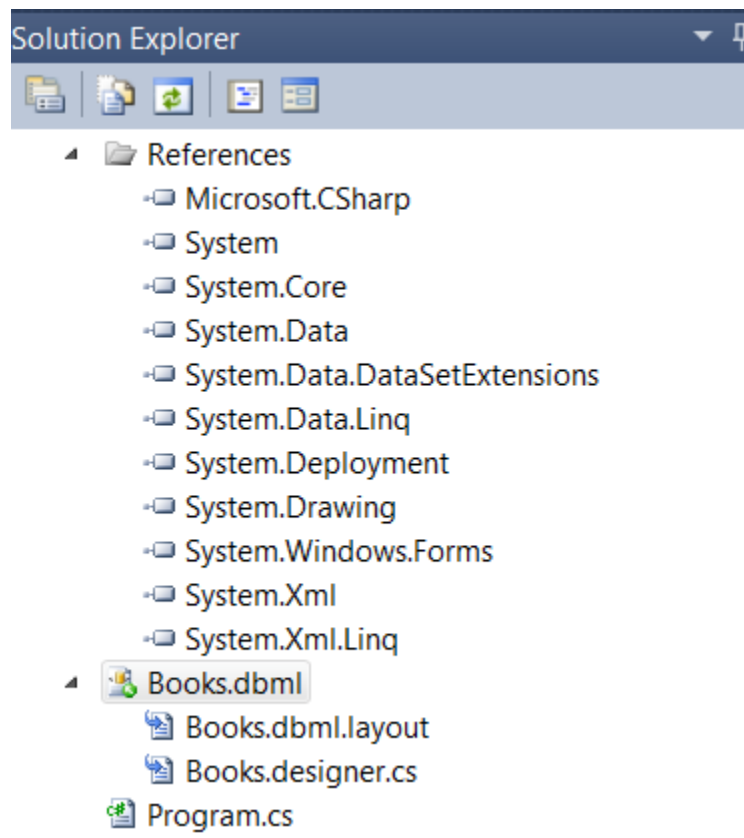
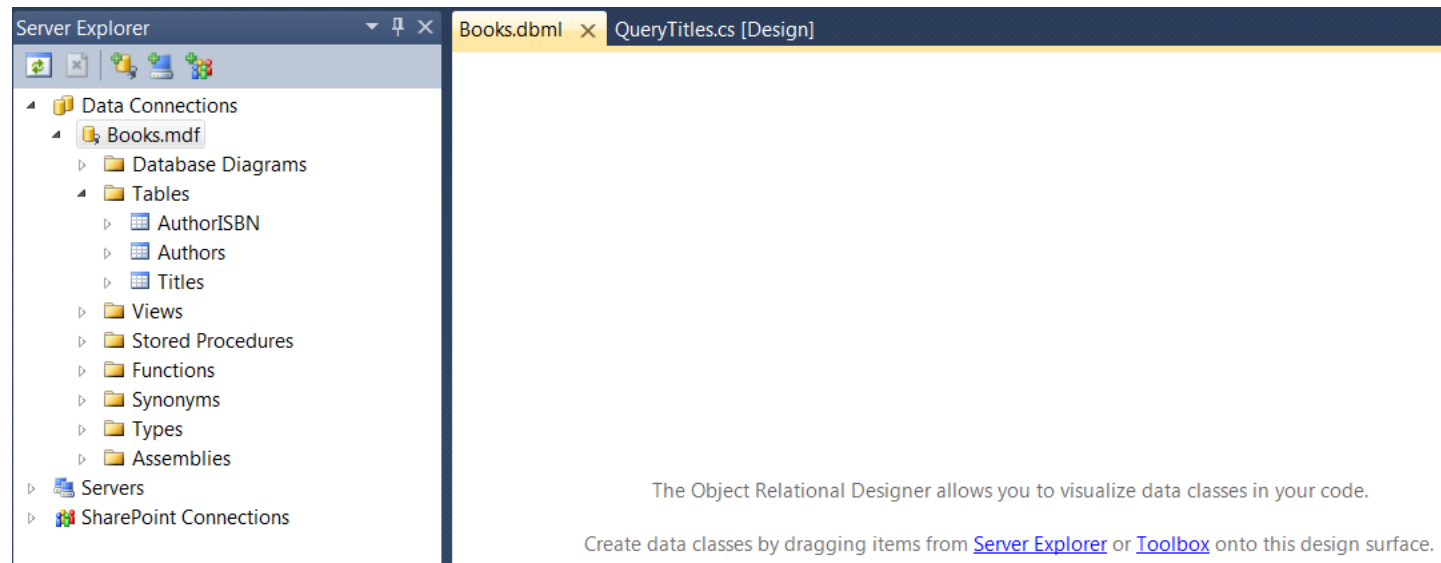


# Uygulama – LINQ Sorgu Sonuçları



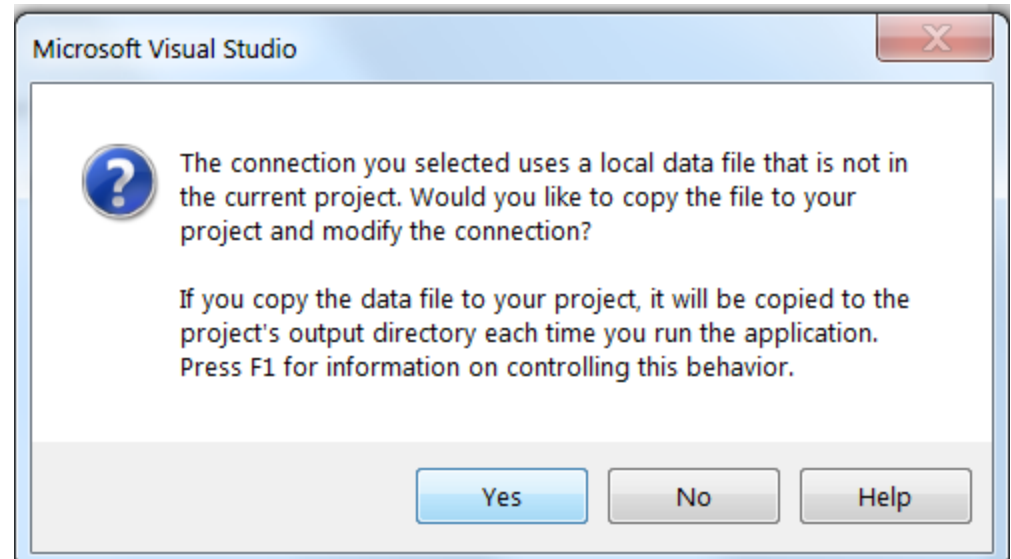
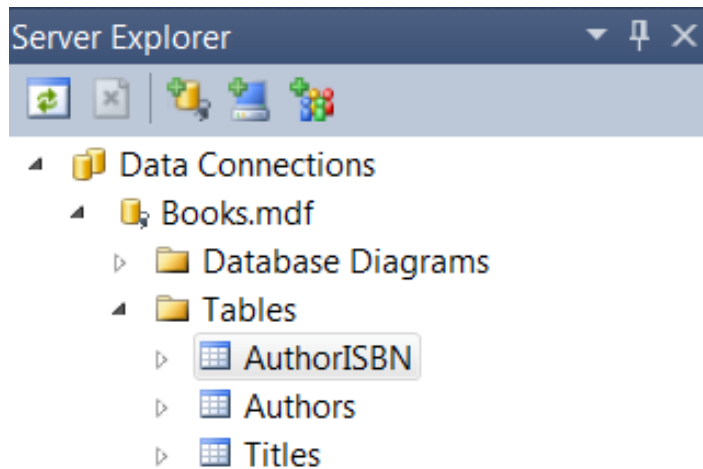
# Uygulama – LINQ Sorgu Sonuçları





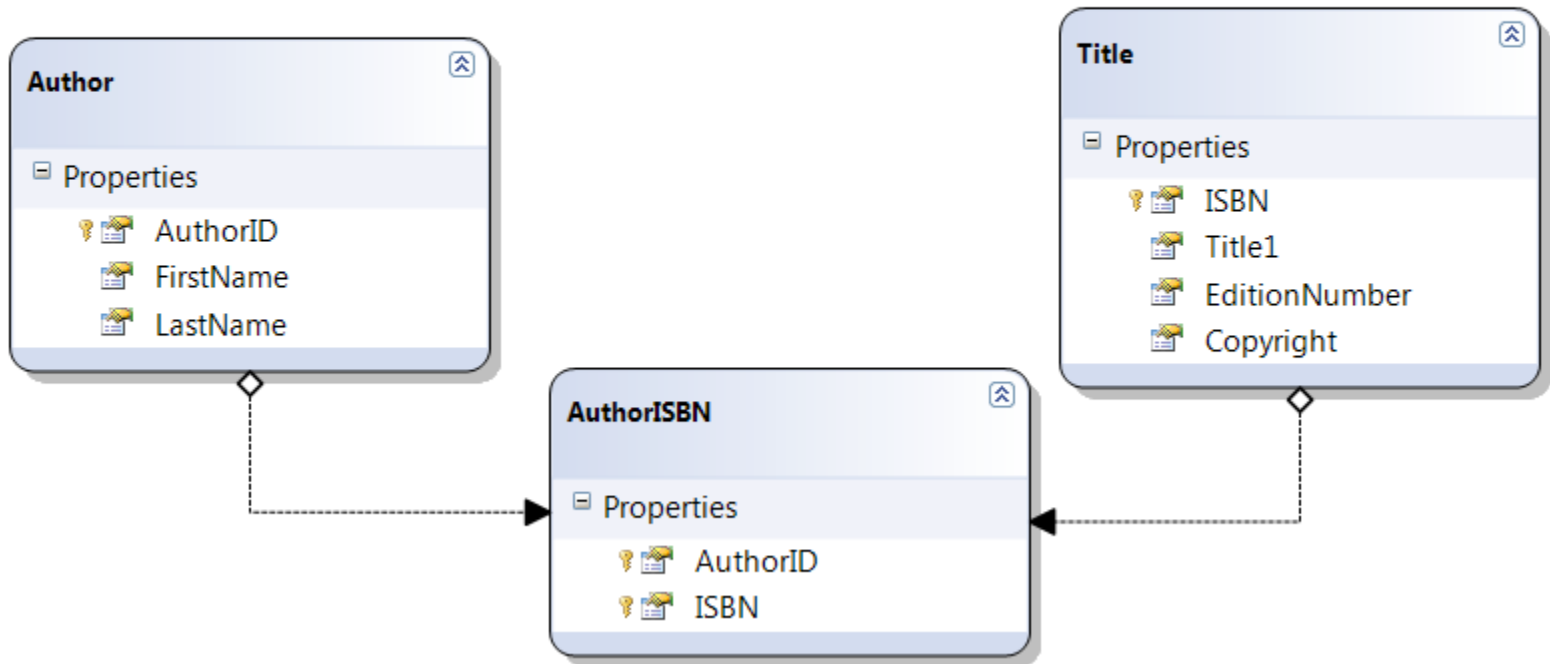
# Uygulama – LINQ Sorgu Sonuçları

- **AuthorsISBN, Authors, Titles** tablolarını **Books .dbml** içerisine sürükleyin.

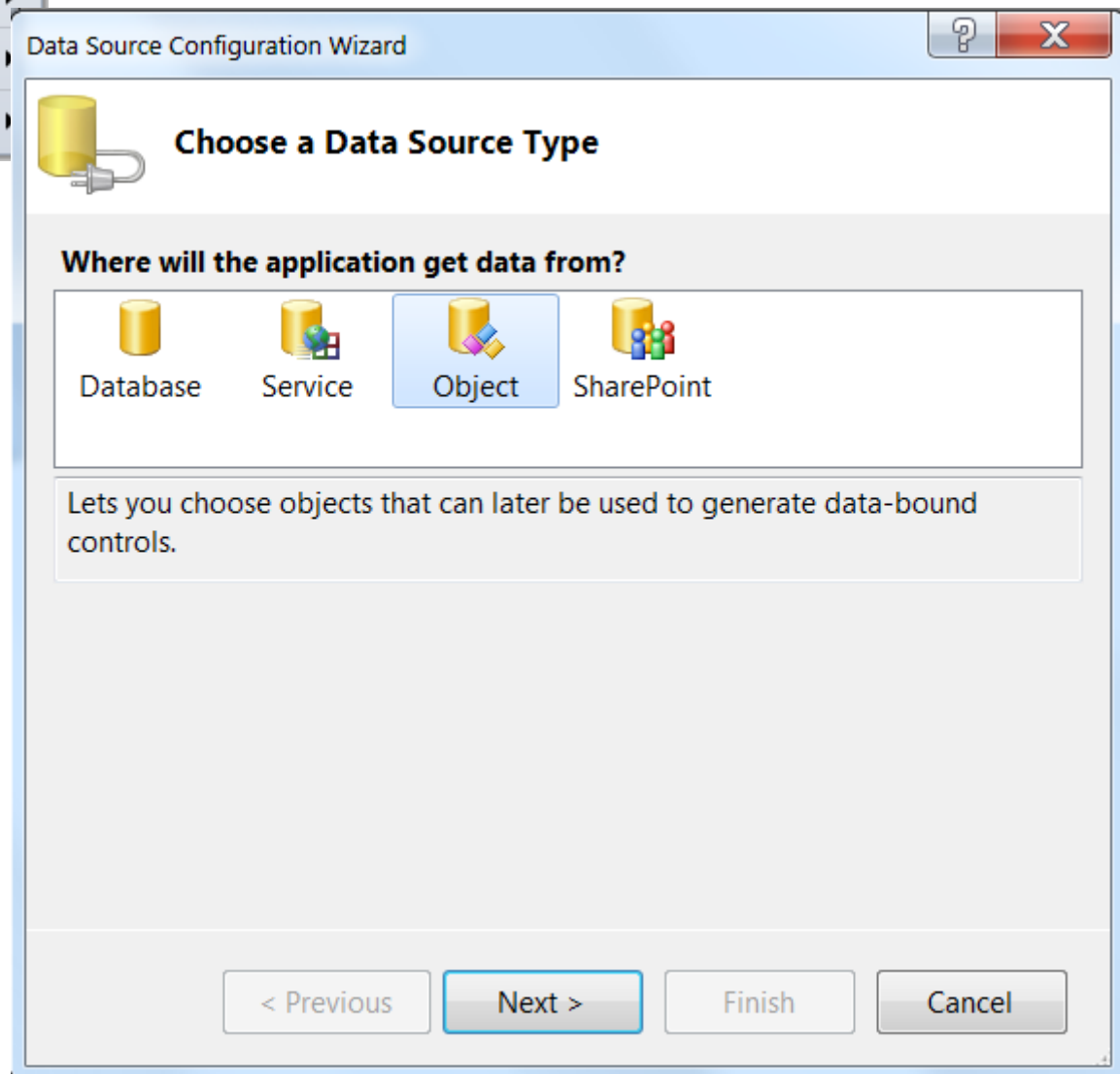
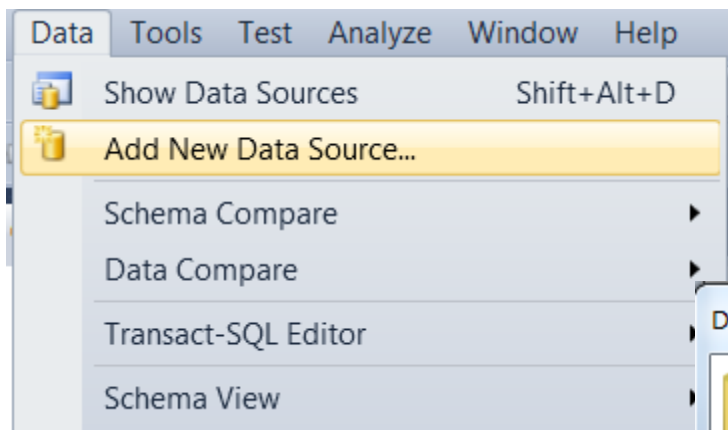


# Uygulama – LINQ Sorgu Sonuçları

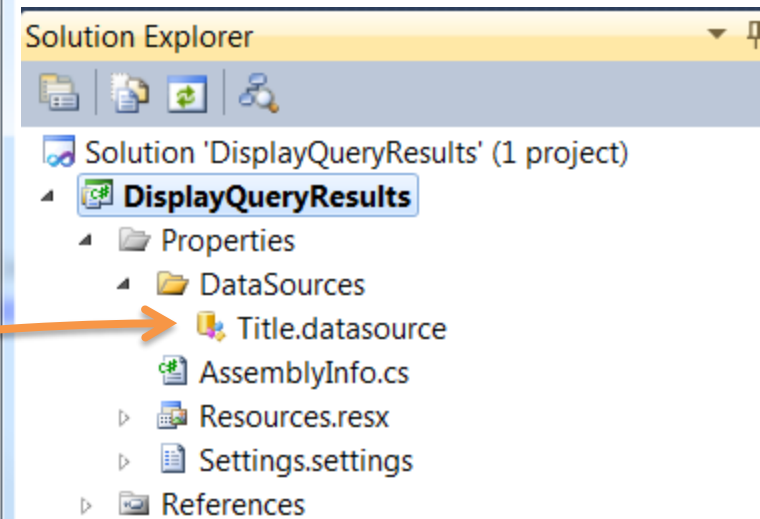
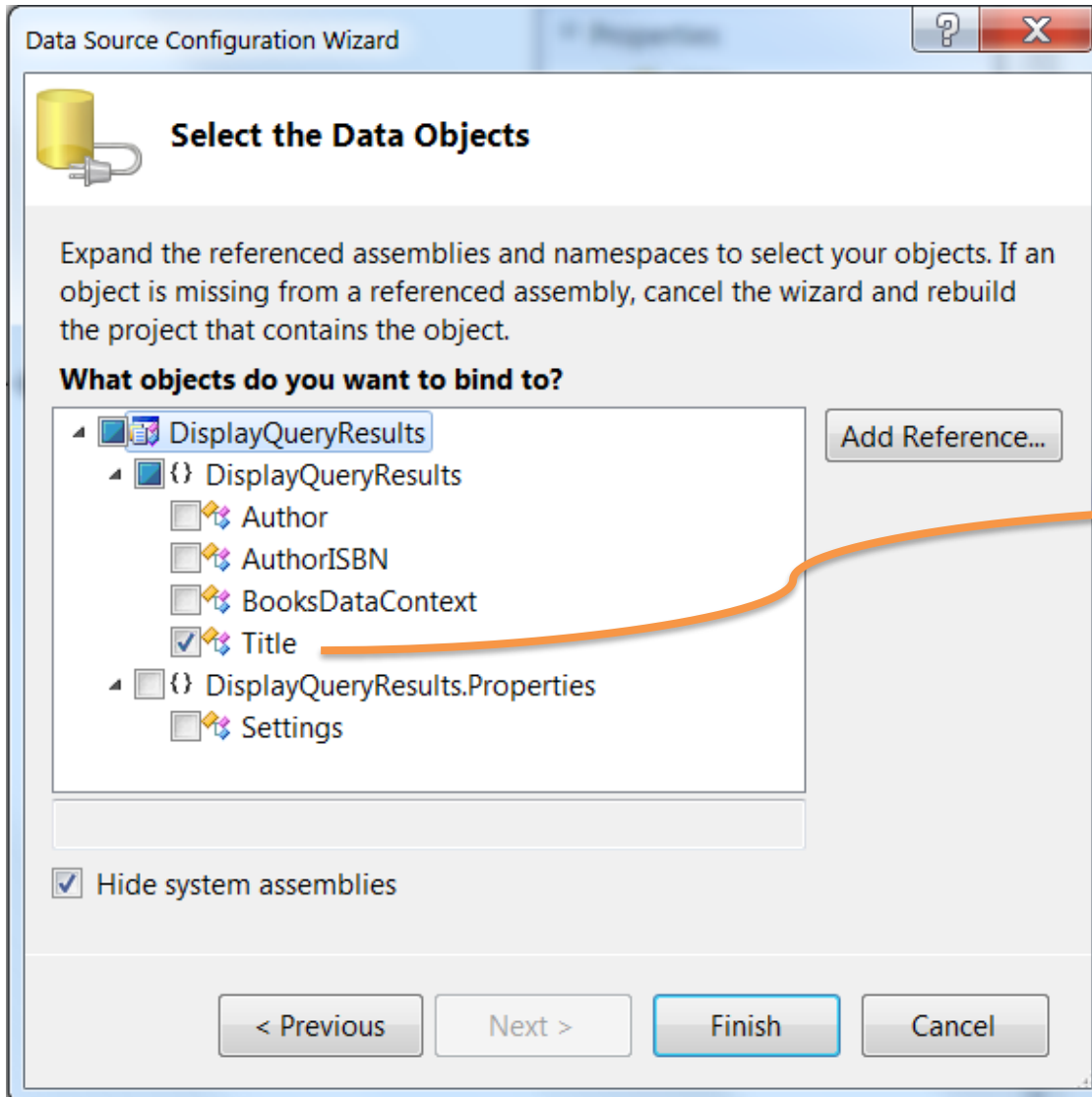
Books.dbml\* x QueryTitles.cs [Design]



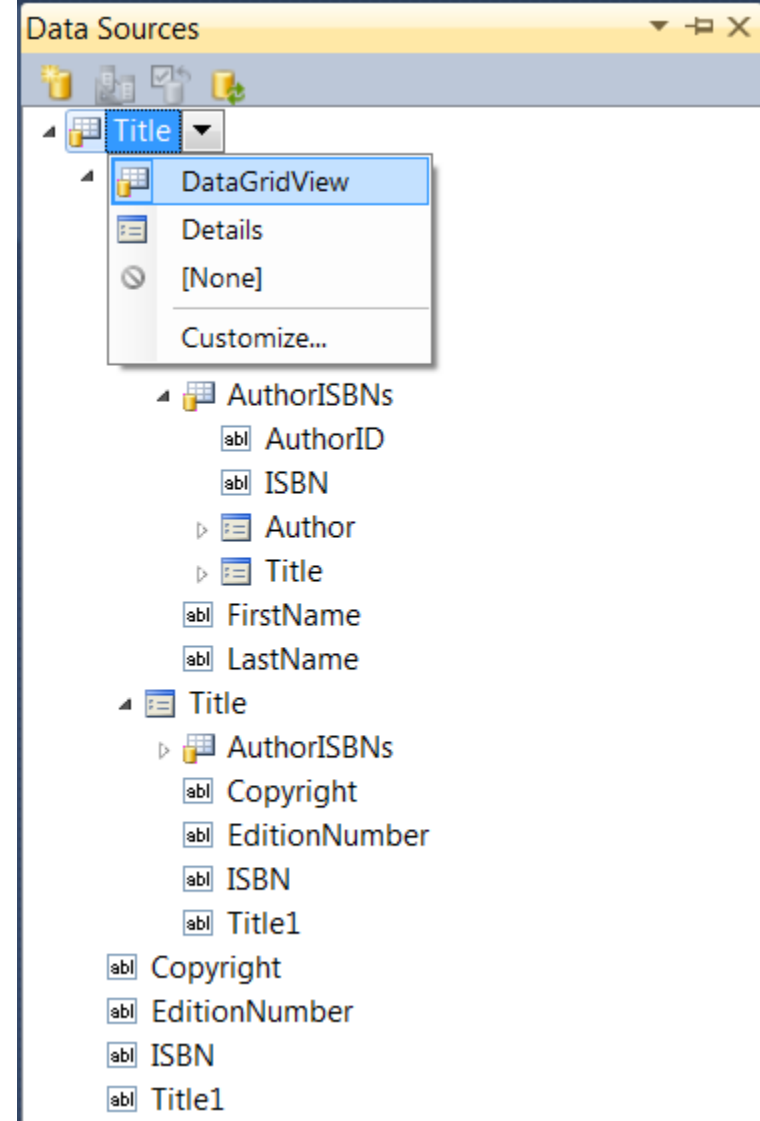
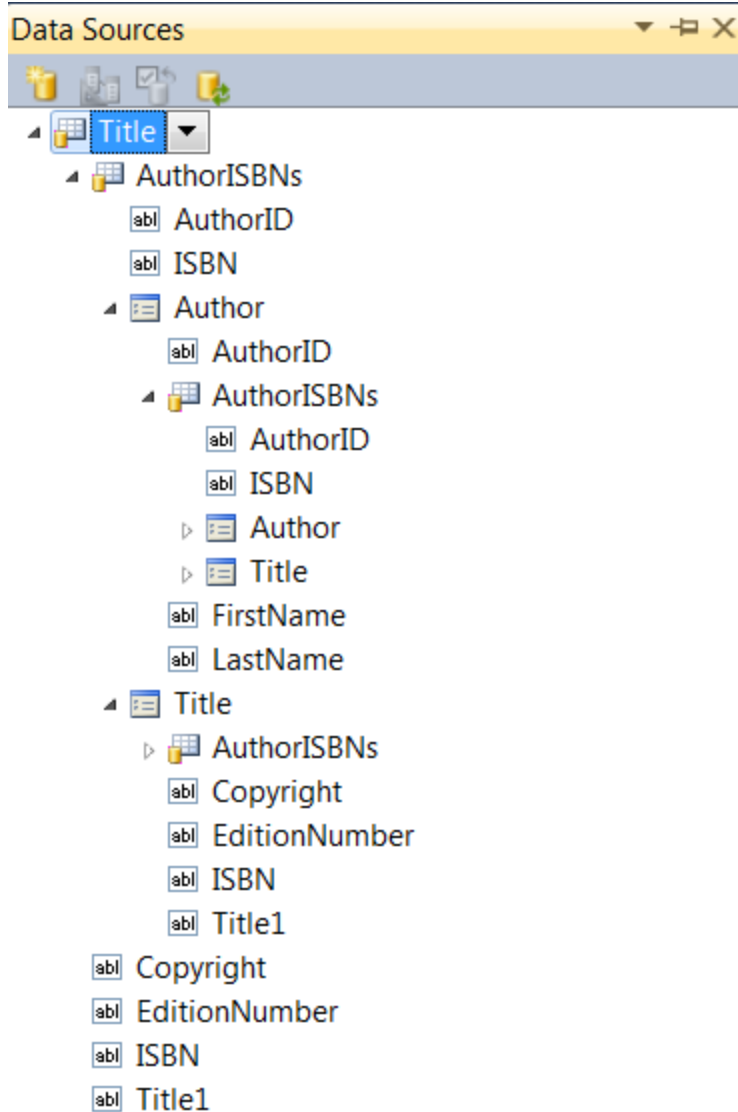




# Uygulama – LINQ Sorgu Sonuçları

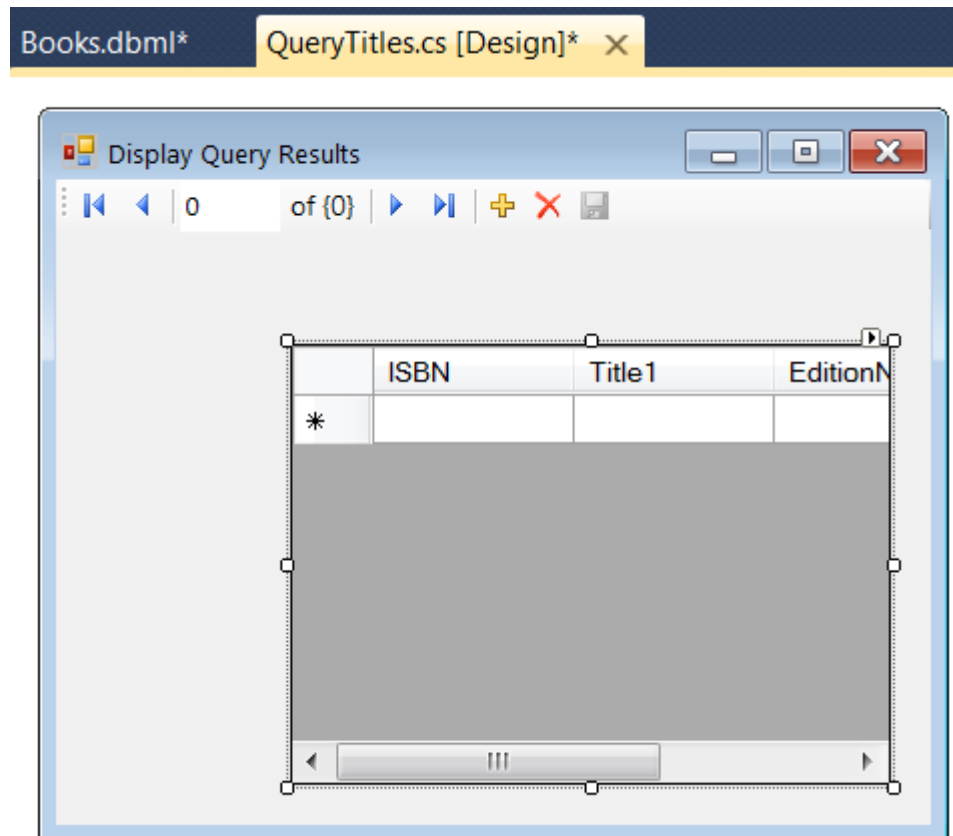


# Uygulama – LINQ Sorgu Sonuçları

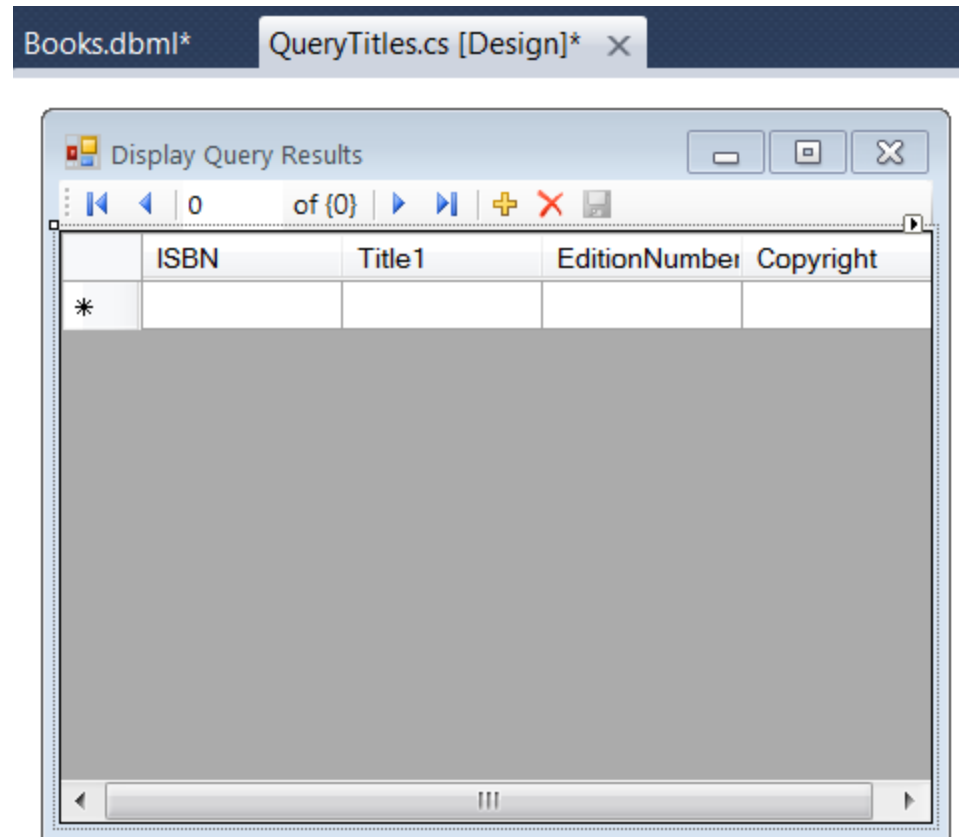
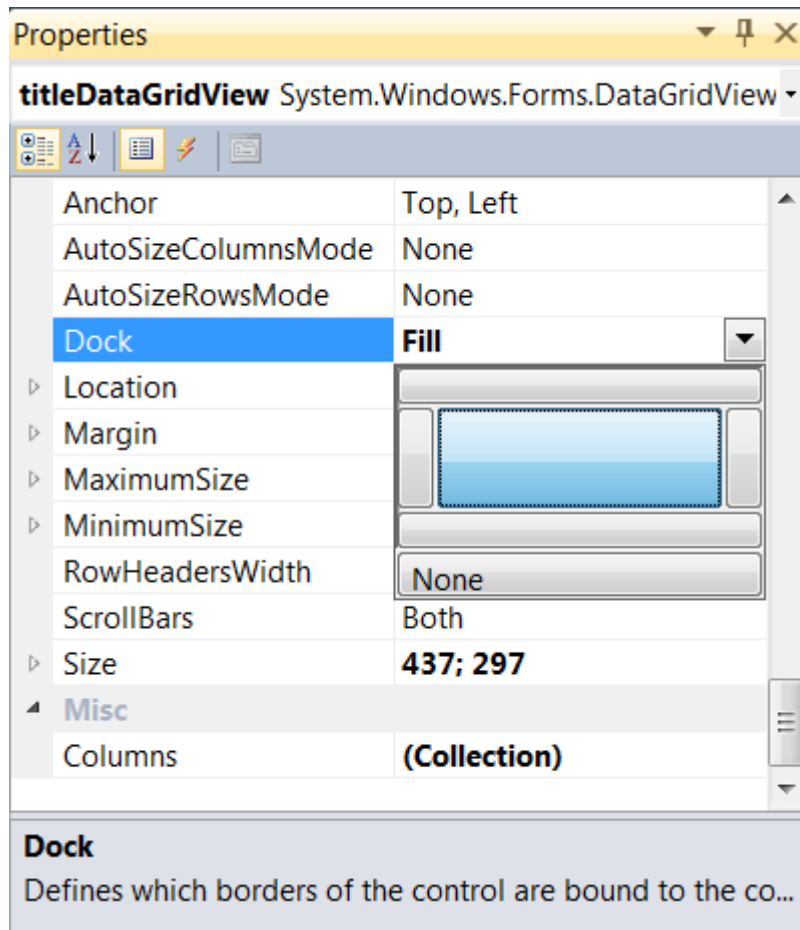


# Uygulama – LINQ Sorgu Sonuçları

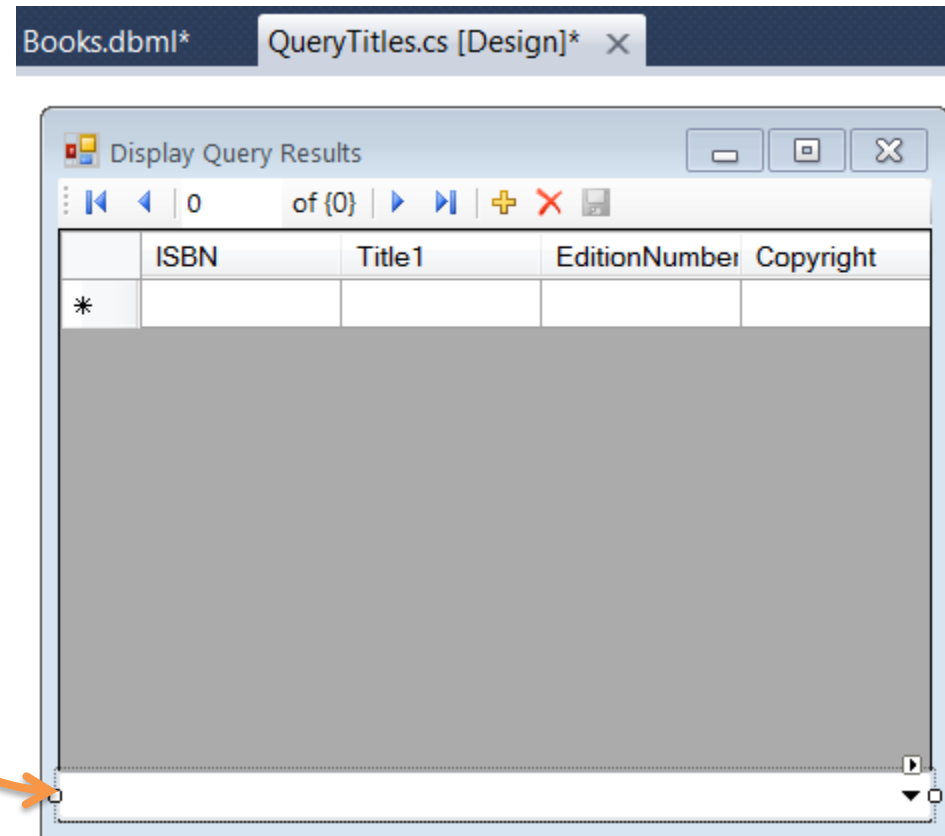
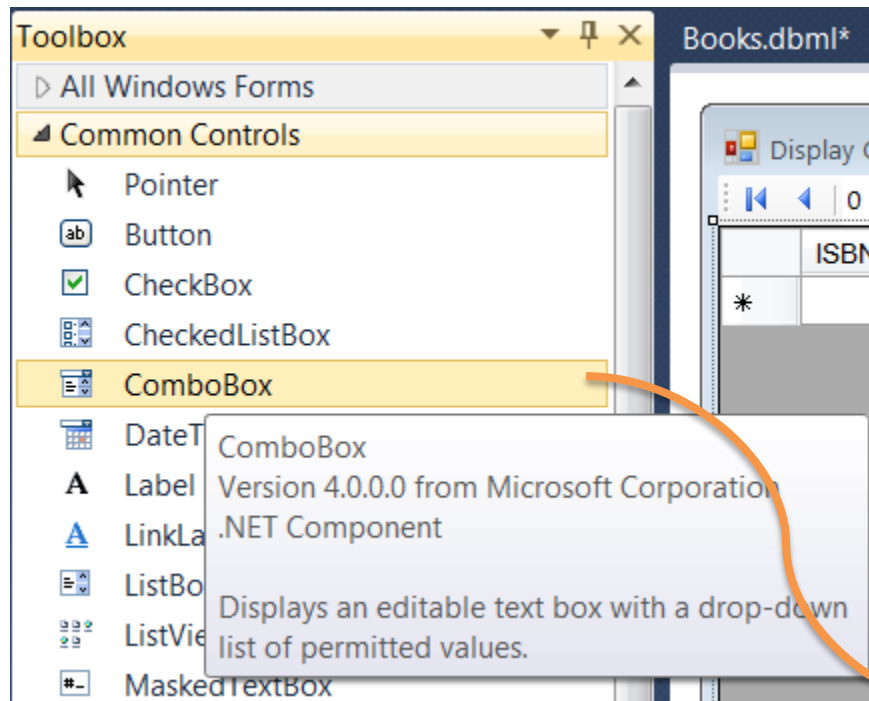
- **Title** node'unu **Display Query Results** içerisine sürükleyiniz.



# Uygulama – LINQ Sorgu Sonuçları

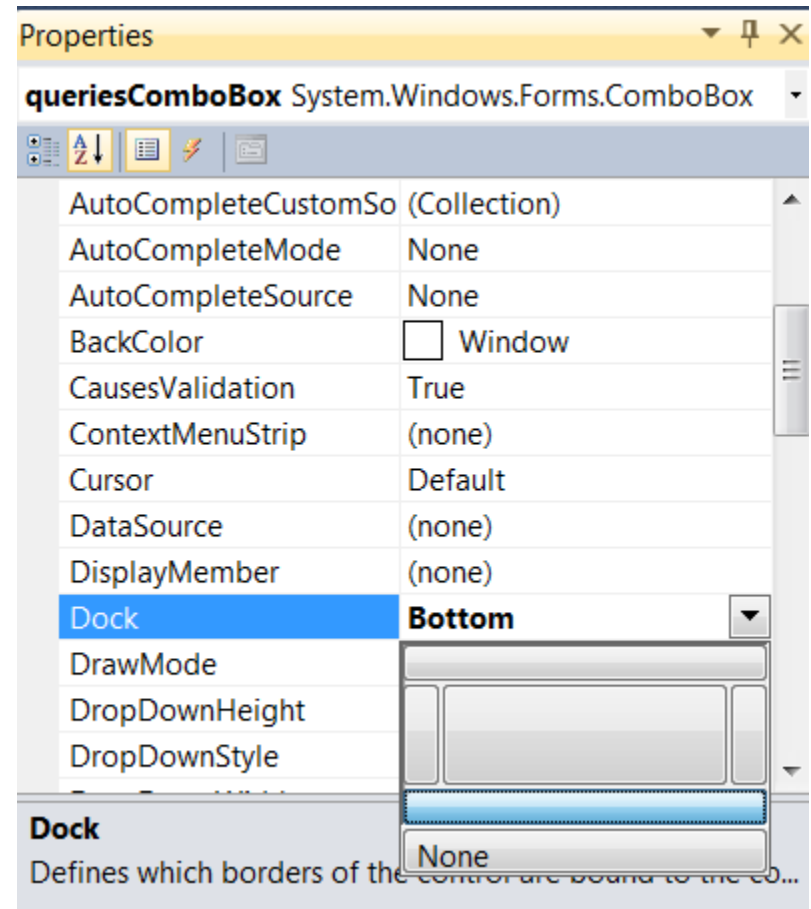
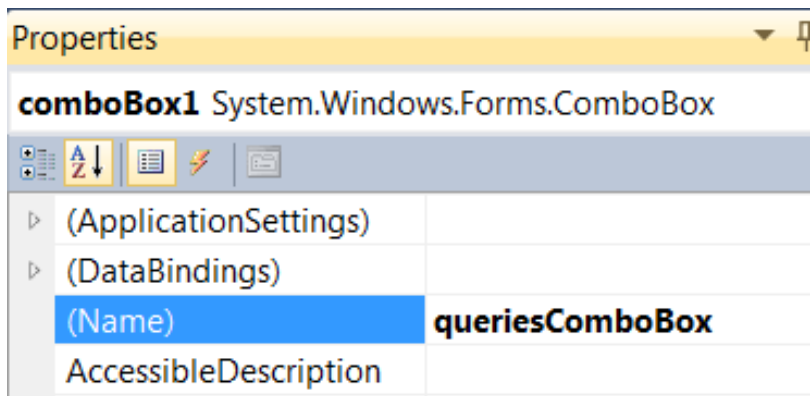


# Uygulama – LINQ Sorgu Sonuçları



# Uygulama – LINQ Sorgu Sonuçları

- **ComboBox Properties**



# Uygulama – LINQ Sorgu Sonuçları

Books.dbml\* QueryTitles.cs [Design]\*

Display Query Results

0 of {0}

	ISBN	Title1	EditionNumber	Copyright
*				

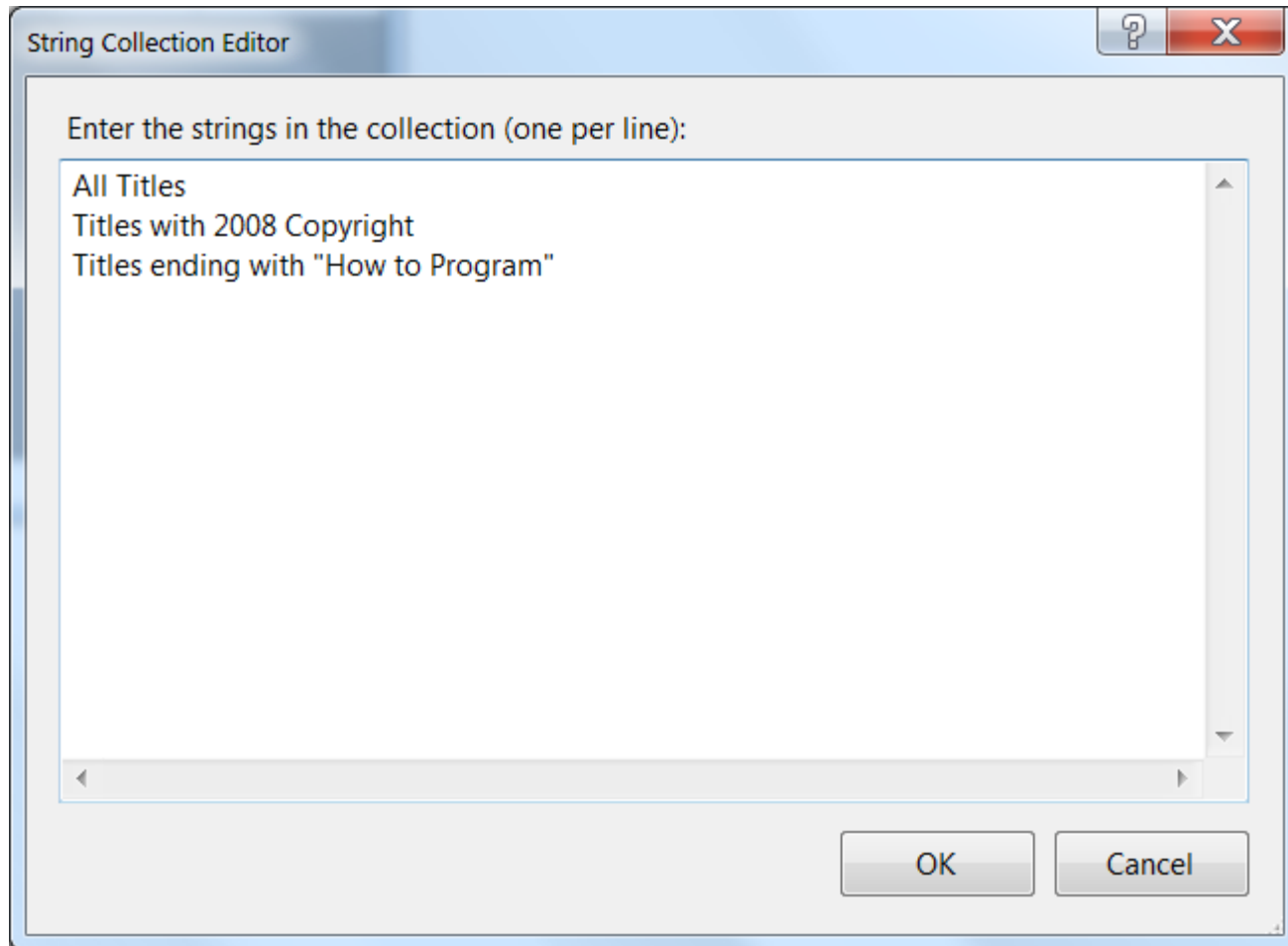
ComboBox Tasks

- ☐ Use Data Bound Items
- Unbound Mode**
- [Edit Items...](#)

Opens the Items collection editor

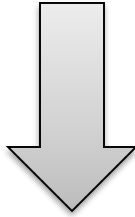


# Uygulama – LINQ Sorgu Sonuçları



# Uygulama – LINQ Sorgu Sonuçları

Double click → `queriesComboBox`



`queriesComboBox_SelectedIndexChanged`

```
// LINQ to SQL data context
private BooksDataContext database = new BooksDataContext();

private void queriesComboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    // set the data displayed according to what is selected
    switch (queriesComboBox.SelectedIndex)
    {
        case 0: // all titles // use LINQ to order the books by title
            titleBindingSource.DataSource = from book in database.Titles
                                            orderby book.Title1
                                            select book;

            break;
        case 1: // titles with 2008 copyright // use LINQ to get titles with 2008 copyright and sort them
            titleBindingSource.DataSource = from book in database.Titles
                                            where book.Copyright == "2008"
                                            orderby book.Title1
                                            select book;

            break;
        case 2: // titles ending with "How to Program"
            // use LINQ to get titles ending with "How to Program" and sort them by title
            titleBindingSource.DataSource = from book in database.Titles
                                            where book.Title1.EndsWith("How to Program")
                                            orderby book.Title1
                                            select book;

            break;
    } // end switch

    titleBindingSource.MoveFirst(); // move to first entry
}
```

---

# Uygulama – LINQ Sorgu Sonuçları

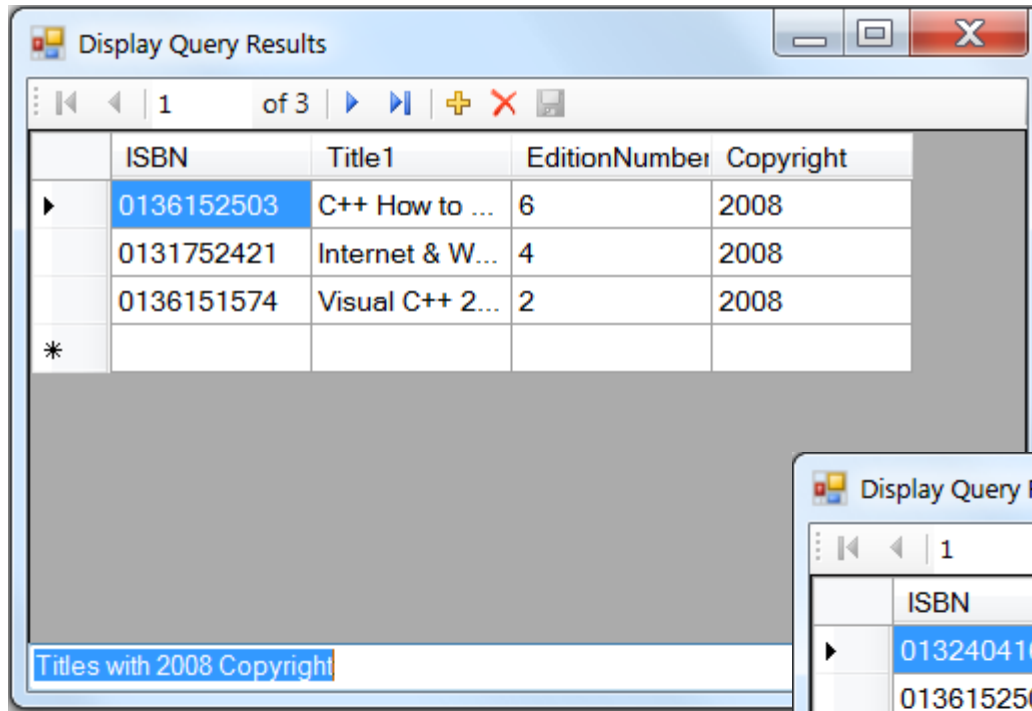
```
private void QueryTitles_Load(object sender, EventArgs e)
{
    // write SQL to standard output stream
    database.Log = Console.Out;

    // set the ComboBox to show the default query that
    // selects all books from the Titles table
    queriesComboBox.SelectedIndex = 0;
}
```

Display Query Results				
1 of 8				
	ISBN	Title1	EditionNumber	Copyright
▶	0132404168	C How to Pr...	5	2007
	0136152503	C++ How to ...	6	2008
	0131752421	Internet & W...	4	2008
	0132222205	Java How to...	7	2007
	0136053033	Simply Visu...	3	2009
	013605305X	Visual Basic...	4	2009
	013605322X	Visual C# 20...	3	2009
	0136151574	Visual C++ 2...	2	
*				
All Titles				

Display Query Results				
1 of 8				
	ISBN	Title1	EditionNumber	Copyright
▶	0132404168	C How to Pr...	5	2007
	0136152503	C++ How to ...	6	2008
	0131752421	Internet & W...	4	2008
	0132222205	Java How to...	7	2007
	0136053033	Simply Visu...	3	2009
	013605305X	Visual Basic...	4	2009
	013605322X	Visual C# 20...	3	2009
	0136151574	Visual C++ 2...	2	2008
*				
All Titles				
All Titles				
Titles with 2008 Copyright				
Titles ending with "How to Program"				

# Uygulama – LINQ Sorgu Sonuçları

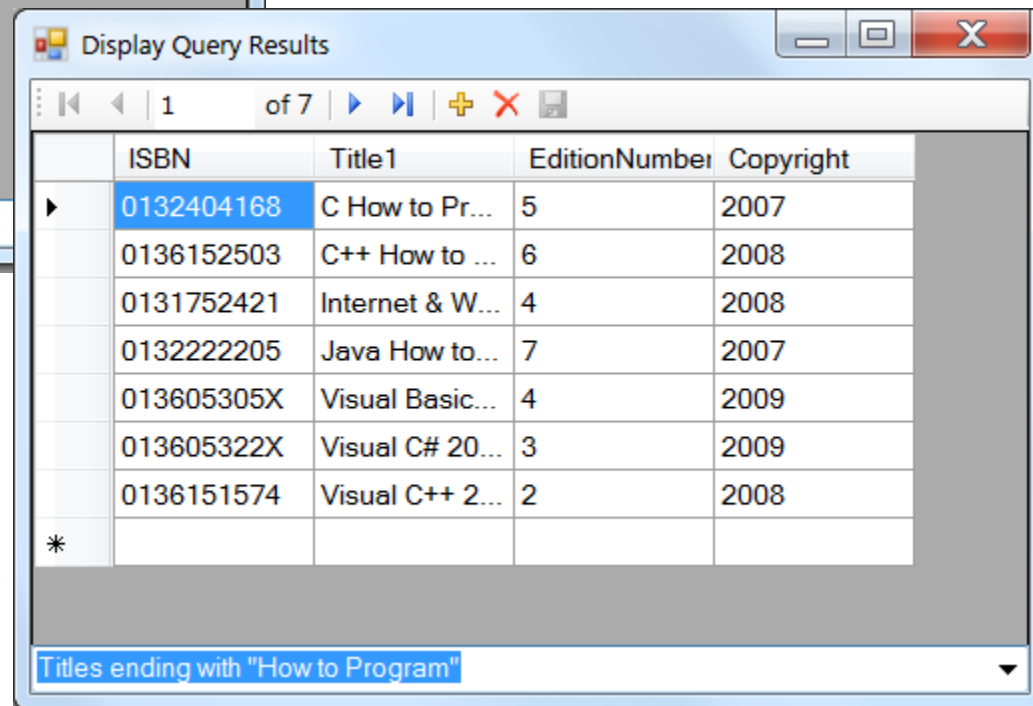


Display Query Results

1 of 3

	ISBN	Title1	EditionNumber	Copyright
▶	0136152503	C++ How to ...	6	2008
	0131752421	Internet & W...	4	2008
	0136151574	Visual C++ 2...	2	2008
*				

Titles with 2008 Copyright



Display Query Results

1 of 7

	ISBN	Title1	EditionNumber	Copyright
▶	0132404168	C How to Pr...	5	2007
	0136152503	C++ How to ...	6	2008
	0131752421	Internet & W...	4	2008
	0132222205	Java How to...	7	2007
	013605305X	Visual Basic...	4	2009
	013605322X	Visual C# 20...	3	2009
	0136151574	Visual C++ 2...	2	2008
*				

Titles ending with "How to Program"