

ADO.NET Database Kavramları

Windows uygulamasında tek bir kullanıcı olduğu için database işlemlerinde hata olması çok zordur. Lakin bir web uygulamasında onlarca kullanıcı olacak. Mesela bir kullanıcı bir bilgiyi **update** ederken bir başka kullanıcı bir bilgiyi **silecek**. Bu durum database'de **tutarsızlığa yol açabilir**. Böyle durumları engellemek için çok iyi database bilgisine sahip olmak gerekir. Bir kullanıcı bilgilerini değiştirirken web server database'e bağlanır ve bilgiler update edilir. Eğer **ASP.NET** ile **ADO.NET** arasında özel yapılar olmasaydı bu işlem **yavaş gerçekleşirdi**.

Visual Studio'da Database Bağlantısı Kurmak

View Tab->Server Explorer->DataConnections->Add Connection.

Açılan pencereden Microsoft Sql Server seçilir ve **continue** denilir. Yeni açılan pencerede ise, **local** computer'da kurulu olan bir sql database ismi **localhost** olduğu için, **Server Name** kısmına **localhost** yazılır. Aynı pencerenin alt kısmında "select or enter a database name" bulunur. Buradan **database** seçilir/yazılır. "Test Connection" butonuna tıklanır, bağlantı başarılı olmuşa OK butonuna tıklanıp bu pencere kapatılır. Database ekleme işlemi bitmiş olur.

SQL Basics

1. Select
2. Update
3. Insert
4. Delete

Running Queries in Visual Studio

Visual Studio'da Query oluşturmak ve test etmek için, eklediğimiz database'e sağ tıklayıp **new query** deriz. Yen bir pencere karşımıza çıkar. Bu pencerede hangi tablolarla ilgili query oluşturacaksa o tabloları seçeriz. Görsel olarak yaptığımız işlemleri hemen görebiliriz.

Query Designer->Change Type->Insert Values diyerek insert işlemi de yapabiliriz. Dört işlemi de burada yapabiliriz.

ADO.NET Basics

ADO.NET "core class"ların küçük bir kümesindeki fonksiyonelliğe dayanır. Bu "core class"ları iki kısma ayırabiliriz :

- 1) Data'ların eklenmesini ve düzenlenmesini sağlayan class'lar. **DataSet, DataTable, DataRow, DataRelation** gibi.
- 2) Belirli bir data kaynağına bağlanmayı sağlayan class'lar. **Connection,Command,DataReader** gibi

Biz **ADO.NET** konusu boyunca bu class'ları(**DataSet, DataTable, DataRow, DataRelation, Connection, Command, DataReader**) inceleyeceğiz.

DataSet, DataTable, DataRow, DataRelation class'ları generic olduğu için hangi data kaynağına bağlanırsak bağlanalım kullanımları değişmez.

Connection, Command, DataReader class'ları **Data Provider** class'ları olarak bilinirler. Bu class'ların kullanımı data kaynağına göre farklılık gösterir.

Data Provider class'ları en iyi performans için database'ten database'e göre farklı "customize" edilirler.

Data Provider class'ları hemen hemen aynı kodlarla aynı mantıkla çalışırlar. Bunun nedeni aynı base class'tan inherit edilmişlerdir, aynı interface'i implement ederler, aynı metod ve property'leri kullanırlar.

.NET dört tane provider sağlar :

1. **SQL Server Provider** : Sql server database'ine bağlanmayı sağlar.
2. **OLEDB Provider** : OLE DB driver olan herhangi bir data kaynağına bağlanmayı sağlar. Mesela Access
3. **Oracle Provider** : Oracle database'ine bağlanmayı sağlar.
4. **ODBC Provider** : ODBC driver olan herhangi bir data kaynağına bağlanmayı sağlar.

MySQL,Oracle,SQLServer,Access ve daha bir çok database OLE DB driver'ına sahiptir. Eğer bir database için data source'ına uygun bir data provider bulamazsak database bağlanmak için OLEDB provider kullanmak gerekir. Mesela SQLServer'a hem OLEDB ile hem de SQL Server Provider ile bağlanabiliriz. Böyle bir durumda SQL Server provider SQLServer'a bağlanmak için daha uygun olduğundan OLEDB ile bağlanmak mantıksız olacaktır.

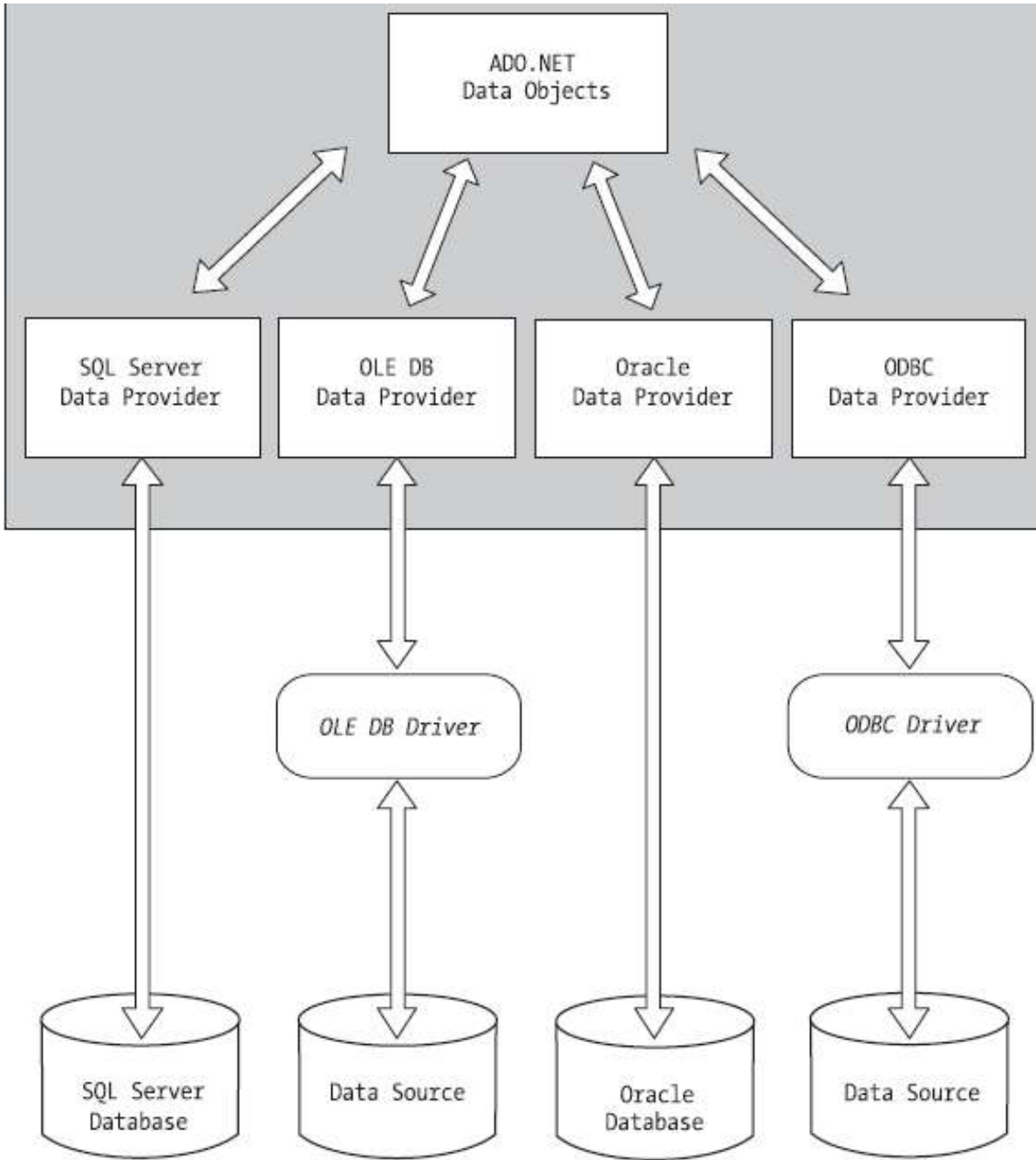


Figure 15-8. *The layers between your code and the data source*

Data Namespaces

ADO.NET data işlemlerini sağlayan bir çok namespace'e sahiptir.

Bunlar :

- 1) **System.Data** : **DataSet**, **DataTable**, **DataRow**, **DataRelation** class'larını barındırır.
- 2) **System.Data.OleDb** : **OleDb** driver'a sahip olan database'lere bağlanmayı sağlayan class'ları içerir.
- 3) **System.Data.SqlClient** : SQL Server'a bağlanmayı sağlayan class'ları içerir.
- 4) **System.Data.SqlTypes** : Sql Server'da kullanılan özel data tiplerini içerir, **SqlMoney**, **SqlDateTime** gibi. Bu class'ları SQL Server ile çalışırken doğabilecek hataları önlemek için **.NET** kütüphanesinde tanımlanmış standart tip class'larına dönüştürmeksizin, **SqDateTime** türünden olan bir veriyi **DateTime** türüne dönüştürmek gibi, kullanabiliriz.
- 5) **System.Data.OracleClient** : **OracleConnection**, **OracleCommand** gibi Oracle database'i ile bağlantı kurmayı sağlayan class'ları içerir.
- 6) **System.Data.Odbc** : Odbc driver'a sahip olan database'lerle bağlantı kurmayı sağlayan **OdbcConnection**, **OdbcCommand** gibi class'ları içerir.

The Data Provider Classes

	SQLServer	OleDb	Oracle	ODBC
Connection	SqlConnection	OleDbConnection	OracleConnection	OdbcConnection
Command	SqlCommand	OleDbCommand	OracleCommand	OdbcCommand
DataReader	SqlDataReader	OleDbDataReader	OracleDataReader	OdbcDataReader
DataAdapter	SqlDataAdapter	OleDbDataAdapter	OracleDataAdapter	OdbcDataAdapter

Bu class'lar neredeyse aynı işlevleri yaparlar.

Farklılıkları şunlardır :

1. İsimleri
2. ConnectionString'leri
3. Bazı küçük ek özellikleri