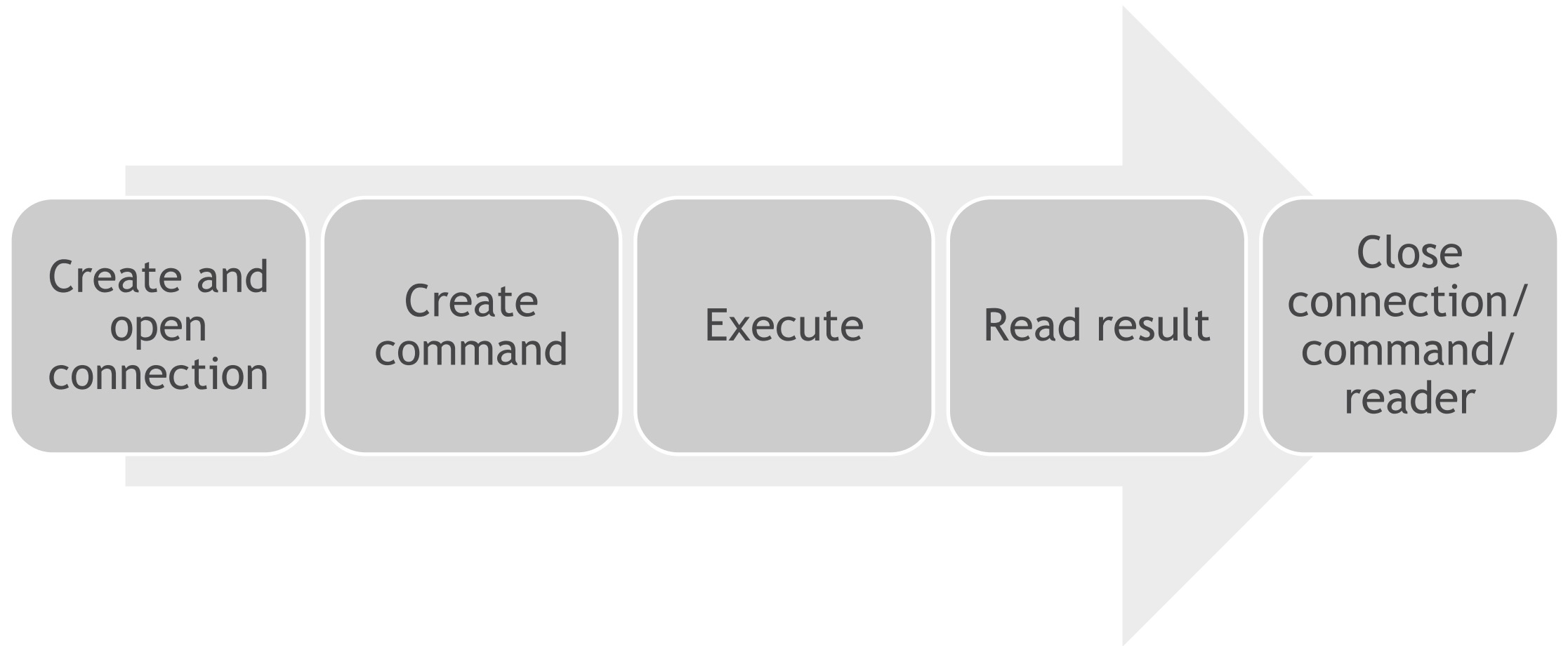
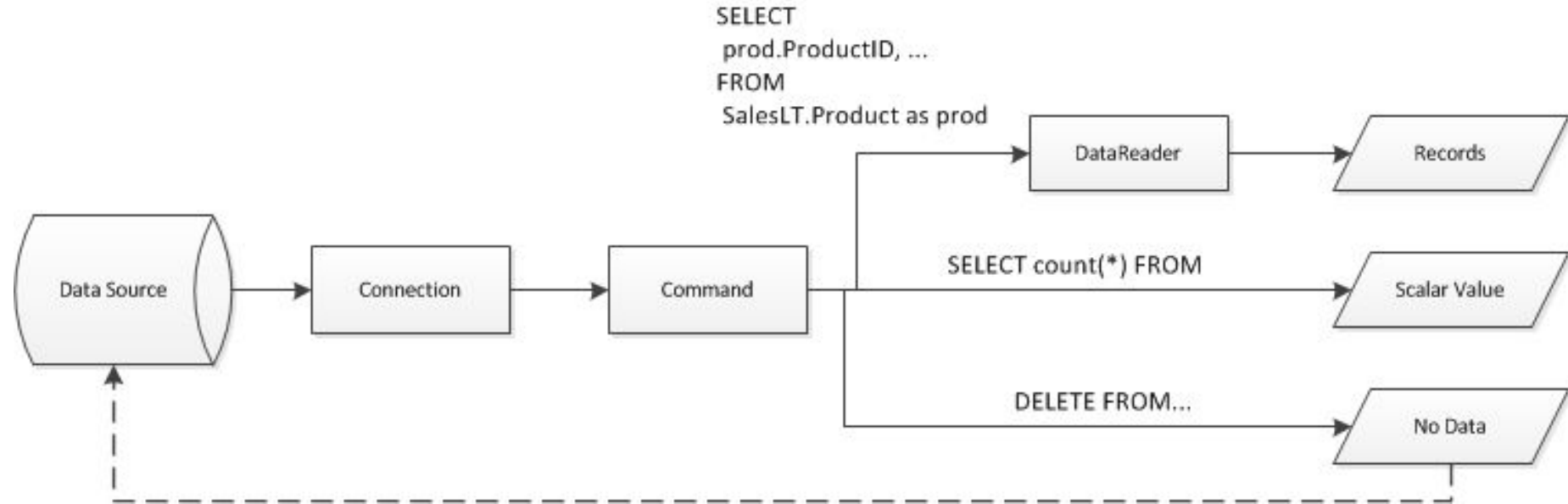


ADO.NET CONNECTED MODEL

-
- Connection
 - Command
 - DataReader
 - Transactions



КОМПОНЕНТЫ **CONNECTED MODEL**



- Connection
- Command
- DataReader

CONNECTION

СОЗДАНИЕ **CONNECTION**

```
var conn = new SqlConnection(  
    "Data Source=(local);Initial Catalog=AdventureWorksLT;Integrated Security=True");  
  
conn.Open();  
  
// ...  
  
conn.Close();
```

```
using (var conn = new SqlConnection(  
    "Data Source=(local);Initial Catalog=AdventureWorksLT;Integrated Security=True"))  
{  
    conn.Open();  
  
    // ...  
}
```

CONNECTION STRINGS

Data
Source=(local);

Initial
Catalog=AdventureWorksLT;

Integrated
Security=True



Server

The diagram consists of three callout lines. The first line starts from the word '(local)' in the 'Data Source' string and points to an oval containing the word 'Server'. The second line starts from the text 'AdventureWorksLT' in the 'Initial Catalog' string and points to an oval containing the words 'Data Base'. The third line starts from the text 'Integrated Security=True' and points to an oval containing the words 'Windows-authentic' and 'ation' on two lines.

Data Base

Windows-authentic
ation

- Общая структура
param1=value; param2=value; ...
- Свои элементы

SQL Client	"Persist Security Info=False;Integrated Security=true;Initial Catalog=Northwind;server=(local)"
OleDb (MS Access)	"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\bin\LocalAccess40.mdb"
ODBC (Excel)	"Driver={Microsoft Excel Driver (*.xls)};DBQ=c:\bin\book1.xls"

[Connection Strings \(ADO.NET\)](https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-strings)

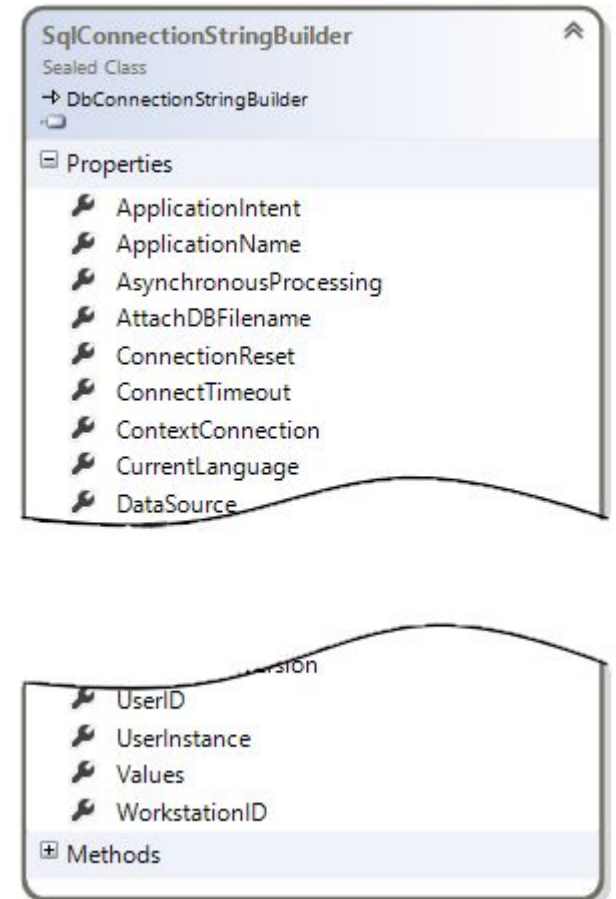
<https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/connection-strings>

<http://www.connectionstrings.com>

CONNECTION STRING BUILDER ПРИМЕР

```
var connectionStringBuilder = new SqlConnectionStringBuilder
{
    DataSource = "(local)",
    InitialCatalog = "Northwind",
    IntegratedSecurity = true
};

using (var connection =
    new SqlConnection(connectionStringBuilder.ConnectionString))
{
    connection.Open();
}
```



COMMON CONNECTION PARAMETERS (SQLCLIENT)

Parameter	Samples
Data Source / Server	(local) np:(local), tcp:(local), lpc:(local) W406811-DB11\PrimaryInstance
Initial Catalog / Database	Northwind
Integrated Security / Trusted_Connection	True
User ID / UID	Ivan_ivanov
Password	123456
AttachDBFilename / Initial File Name	DataDirectory \data\YourDB.mdf
Connect Timeout / Timeout	30

[Connection parameters](https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx)

[https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.connectionstring\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx)

CONNECTION STRING + APP.CONFIG + PROVIDER FACTORIES

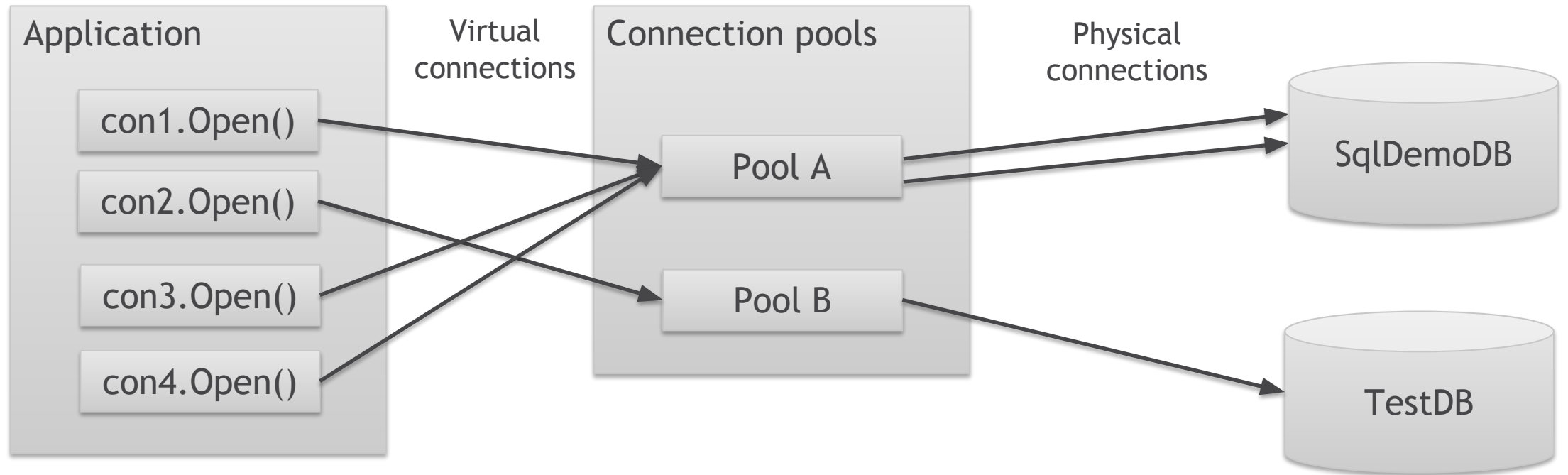
```
<configuration>
  <connectionStrings>
    <add name="NorthwindConection"
          providerName="System.Data.SqlClient"
          connectionString="Data Source=(local);Initial Catalog=Northwind;Integrated Security=True"/>
  </connectionStrings>
</configuration>
```

```
var connectionStringItem = ConfigurationManager.ConnectionStrings["NorthwindConection"];
var connectionString = connectionStringItem.ConnectionString;
var providerName = connectionStringItem.ProviderName;

var factory = DbProviderFactories.GetFactory(providerName);

using (var connection =factory.CreateConnection())
{
    connection.ConnectionString = connectionString;
    connection.Open();
}
```

CONNECTION POOLS



~~Data Source=(local);Initial Catalog=**SqlDemoDB**;Integrated Security=True~~

Data Source=(local);Initial Catalog=**TestDB**;Integrated Security=True

Data Source=(local);Initial Catalog=**SqlDemoDB**;Integrated Security=True

Data Source=(local);Initial Catalog=**SqlDemoDB**;Integrated Security=True

[Connection Pooling](#)

CONNECTION ПРОБЛЕМЫ И BEST PRACTICES

- Держите соединение с источником минимальное кол-во времени
- Всегда закрывайте все созданные вами объекты Connection или DataReader, когда вы завершаете с ними работать

Best
Practice

COMMAND

СОЗДАНИЕ **COMMAND**

Command should be associated with Connection

```
using (IDbConnection connection = new SqlConnection(connectionString))
{
    connection.Open();

    var command = connection.CreateCommand();
}
```

```
using (IDbConnection connection = new SqlConnection(connectionString))
{
    connection.Open();

    IDbCommand command = new SqlCommand();
    command.Connection = connection;
}
```

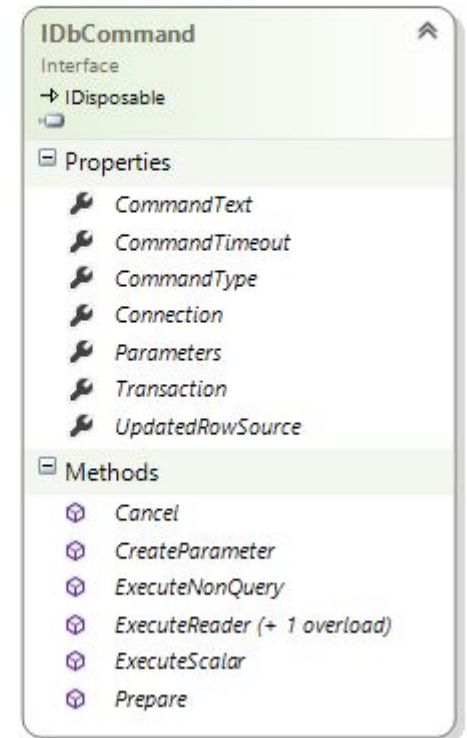
ОБЩИЕ СВОЙСТВА **COMMAND**

```
using (IDbConnection connection = new SqlConnection(connectionString))
{
    connection.Open();

    var command = connection.CreateCommand();

    command.CommandText = "select count(*) from Northwind.Customers";
    command.CommandType = CommandType.Text;

    var customersCount = command.ExecuteScalar();
    Console.WriteLine(customersCount);
}
```



COMMAND TYPES

Command Type	Samples / Comments
Text (default)	<pre>command1.CommandText = "SELECT * FROM Northwind.Products"; command1.CommandType = CommandType.Text;</pre> <pre>command2.CommandText = "exec sp_helpdb"; command2.CommandType = CommandType.Text;</pre>
StoredProcedure	<pre>command3.CommandText = "sp_helpdb"; command3.CommandType = CommandType.StoredProcedure;</pre>
TableDirect	<p>Поддерживается только в .NET Framework Data Provider для OLE DB</p> <pre>command.CommandText = "Northwind.Customers"; command.CommandType = CommandType.TableDirect;</pre>

COMMAND RESULTS

Result Type	Samples
Row set	<pre>command.CommandText = "SELECT CompanyName FROM Northwind.Customers"; SqlDataReader reader = command.ExecuteReader();</pre>
Single value	<pre>command.CommandText = "SELECT count(*) FROM Northwind.Customers"; int count = (int)command.ExecuteScalar();</pre>
No result	<pre>command.CommandText = "UPDATE Northwind.Products SET UnitPrice = UnitPrice - 0.0002"; int affected = command.ExecuteNonQuery();</pre>
Xml	<pre>command.CommandText = "SELECT * FROM Northwind.Customers FOR XML AUTO, ROOT('Customers')"; XmlReader xmlReader = command.ExecuteXmlReader();</pre>

ПАРАМЕТРИЗОВАННЫЕ ЗАПРОСЫ. **SQL** ИНЪЕКЦИИ

```
string.Format(
    "select top 1 * from dbo.Users where Login = '{0}' and Password = '{1}'", login, password);
```

Logi n	<input type="text" value="user"/>
Passwor d	<input type="text" value="123"/>

```
select top 1 * from dbo.Users
where Login = 'user' and Password = '123'
```

Logi n	<input type="text" value="' OR 1 = 1 /*"/>
Passwor d	<input type="text" value="*/ --"/>

```
select top 1 * from dbo.Users
where Login = '' OR 1 = 1 /*' and Password = '123'*/
--
```

COMMAND PARAMETERS

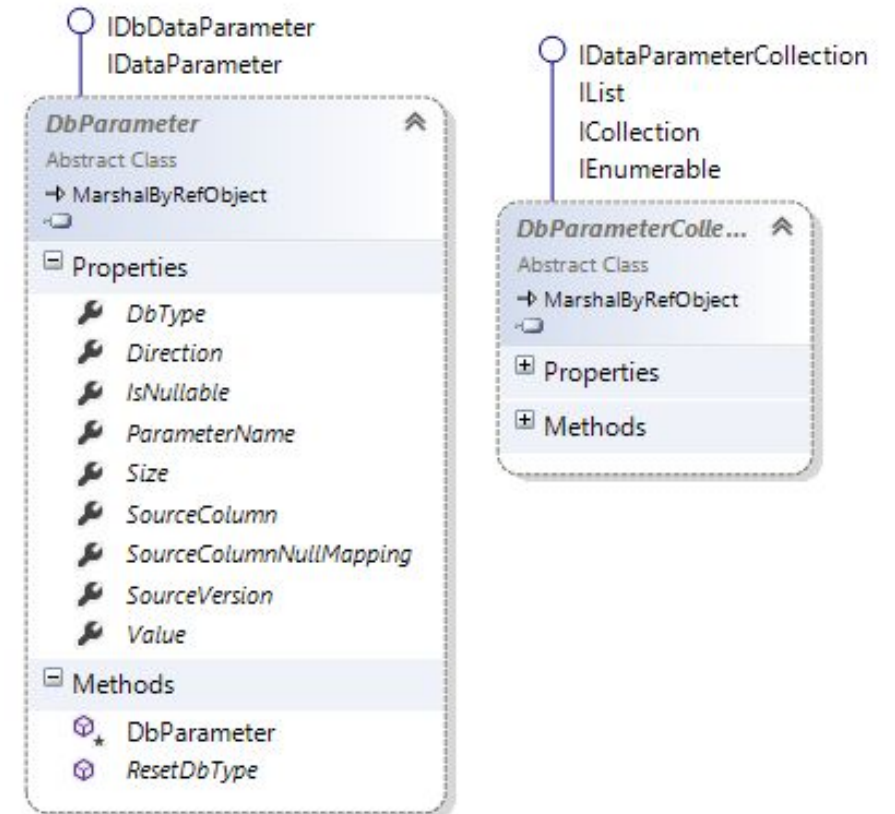
```
command.CommandText =  
    "SELECT count(*) FROM Northwind.Products  
    WHERE UnitPrice >= @minPrice";
```

IDbCommand

```
var minPrice = command.CreateParameter();  
minPrice.ParameterName = "@minPrice";  
minPrice.DbType = DbType.Decimal;  
minPrice.Value = 50;  
  
command.Parameters.Add(minPrice);
```

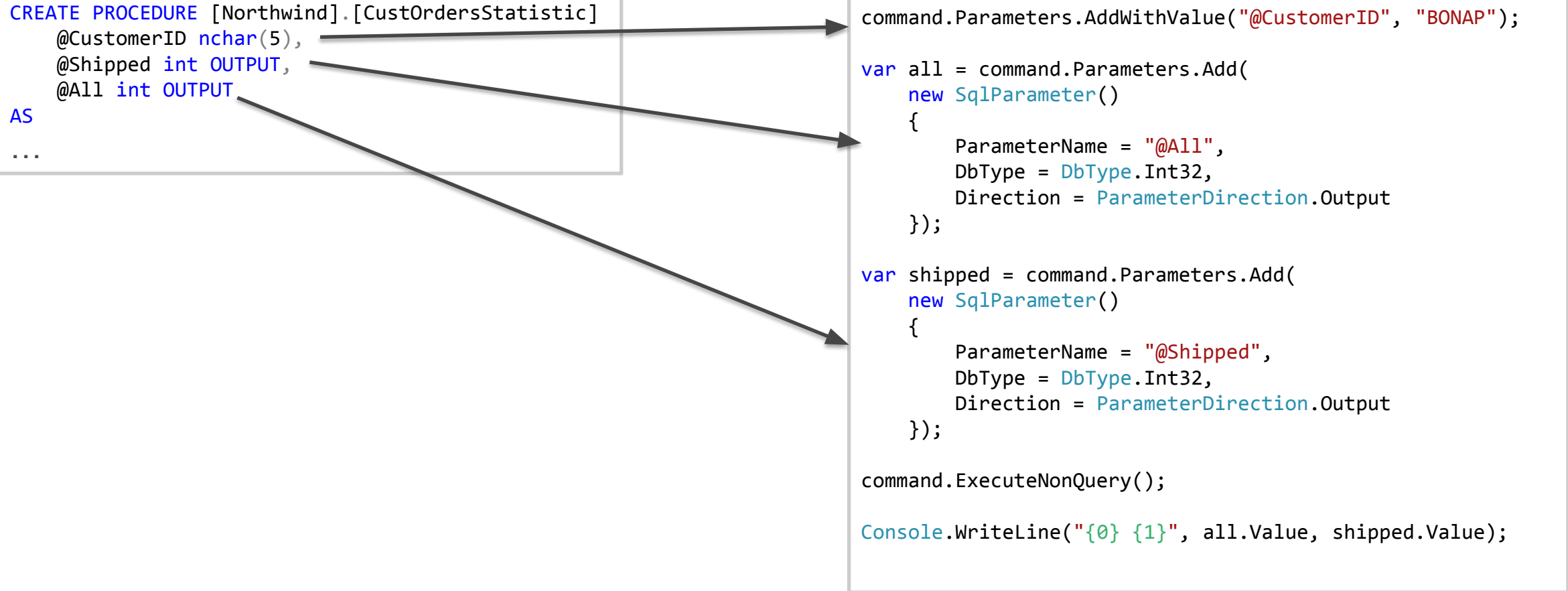
SqlCommand

```
command.Parameters.AddWithValue("@minPrice", 50m);
```



BI3OB STORED PROCEDURES

```
CREATE PROCEDURE [Northwind].[CustOrdersStatistic]
    @CustomerID nchar(5),
    @Shipped int OUTPUT,
    @All int OUTPUT
AS
...
```



```
var command = connection.CreateCommand();
command.CommandText = "[Northwind].[CustOrdersStatistic]";
command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@CustomerID", "BONAP");

var all = command.Parameters.Add(
    new SqlParameter()
    {
        ParameterName = "@All",
        DbType = DbType.Int32,
        Direction = ParameterDirection.Output
    });

var shipped = command.Parameters.Add(
    new SqlParameter()
    {
        ParameterName = "@Shipped",
        DbType = DbType.Int32,
        Direction = ParameterDirection.Output
    });

command.ExecuteNonQuery();

Console.WriteLine("{0} {1}", all.Value, shipped.Value);
```

DATAREADER

READ RESULT

```
using (IDbConnection connection =
    new SqlConnection(ConnectionString))
{
    var command = connection.CreateCommand();
    command.CommandText =
        "SELECT CompanyName, City, Region FROM Northwind.Customers";

    connection.Open();

    using (IDataReader reader = command.ExecuteReader())
    {
        while (reader.Read())
        {
            Console.WriteLine("{0} - {1}, {2}",
                reader["CompanyName"],
                reader["City"],
                reader["Region"]);
        }
    }
}
```

- Side-by-side execution can only take place in **different connections**
 - Every readers should be closed before next command start

DATAREADER МЕТОДЫ

- DataReader
 - Navigation
 - Read()
 - NextResult()
 - HasRows
- Get fields value
 - By field name
 - [“field_name”]
 - By field index
 - GetString(i)
 - GetDateTime(i)
 - GetBoolean(i)
 - ...

READ MANY RESULT SETS

```
var command = connection.CreateCommand();
command.CommandText =
    "SELECT * " +
    "FROM Northwind.Orders " +
    "where OrderID = @orderId;" +

    "SELECT p.ProductName, ods.UnitPrice, ods.Quantity " +
    "FROM Northwind.[Order Details] ods " +
    "LEFT JOIN Northwind.Products p ON p.ProductID = ods.ProductID " +
    "WHERE ods.OrderID = @orderId;";

command.Parameters.AddWithValue("@orderId", 10262);

using (var reader = command.ExecuteReader())
{
    reader.Read();
    Console.WriteLine("{0} ({1})", reader["OrderID"], reader["OrderDate"]);

    reader.NextResult();

    while (reader.Read())
        Console.WriteLine("\t{0} - {1}", reader["ProductName"], reader["UnitPrice"]);
}
```

TRANSACTIONS

```
using (IDbConnection connection = new SqlConnection(ConnectionString))
{
    connection.Open();

    using (var transaction = connection.BeginTransaction())
    {
        var command = connection.CreateCommand();
        command.CommandText =
            "delete from Northwind.[Order Details] where OrderID = @orderId;";
        command.CommandText +=
            "delete from Northwind.Orders where OrderID = @orderId;";

        var orderIdParam = command.CreateParameter();
        orderIdParam.ParameterName = "@orderId";
        orderIdParam.Value = orderId;
        command.Parameters.Add(orderIdParam);

        command.Transaction = transaction;

        command.ExecuteNonQuery();

        transaction.Commit();
    }
}
```