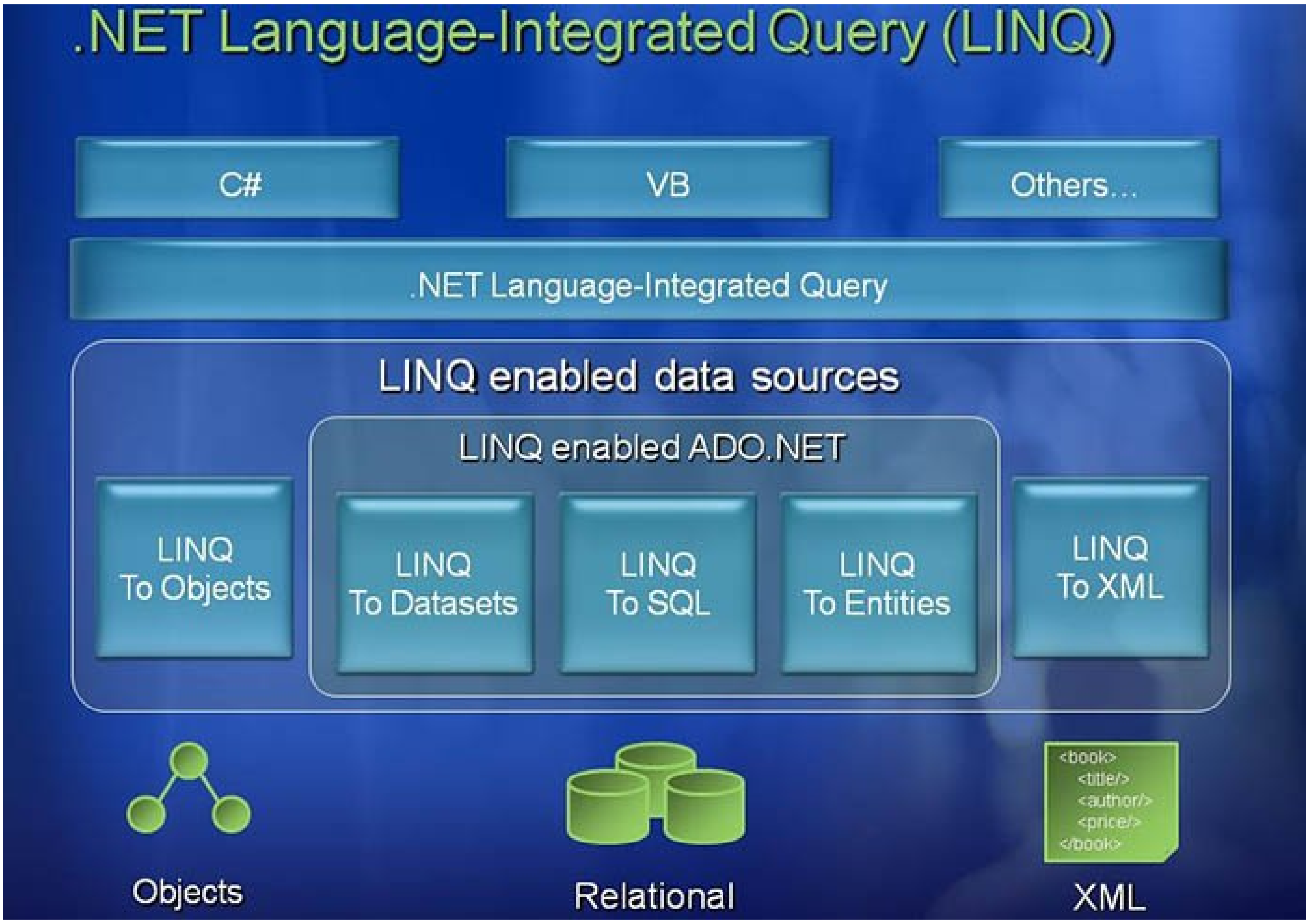
[illegible] [illegible] [illegible] [illegible] LINQ

[illegible] [illegible] [illegible] [illegible] [illegible] [illegible] ([illegible])



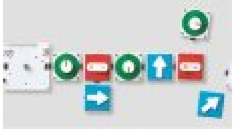
[illegible] LINQ [illegible] Language-Integrated Query [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]. [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]. [illegible].

[illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] IntelliSense [illegible] [illegible] [illegible]. [illegible]. [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]. [illegible] xml [illegible] … [illegible] [illegible] [illegible] [illegible] [illegible] [illegible]. [illegible]. [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible] [illegible].

[illegible]

[illegible] (Query) [illegible]. [illegible]. [illegible] (SQO) [illegible]. [illegible] IEnumerable<T> [illegible] IQueryable<T> [illegible]. [illegible] IQueryable [illegible] LINQ TO SQL [illegible] IEnumerable [illegible]. [illegible].

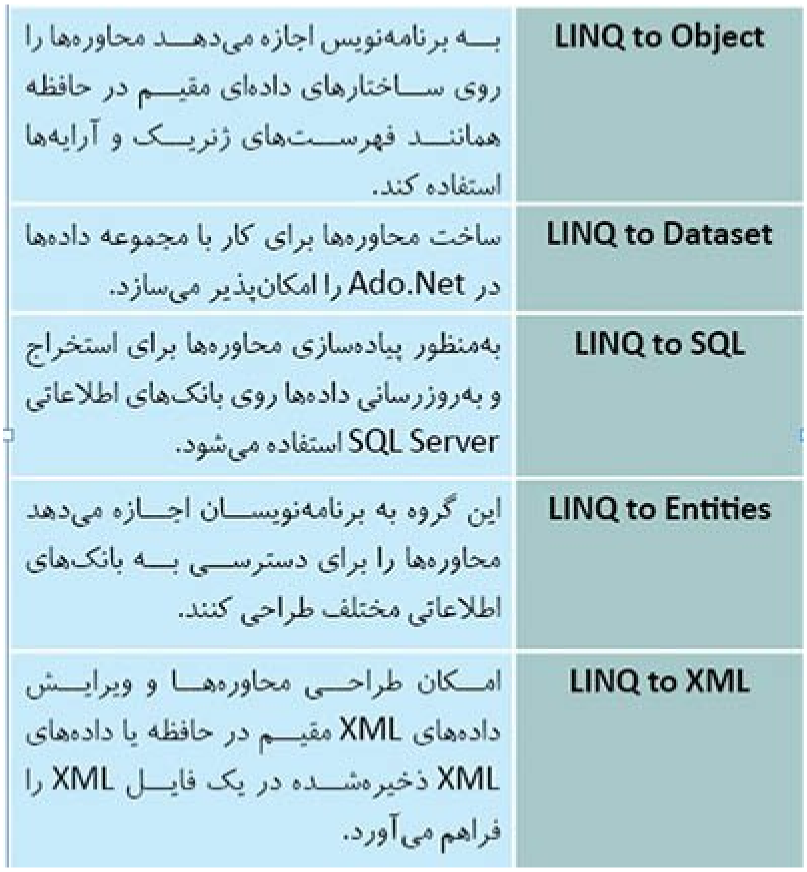
[illegible]



Bloks [illegible]

[illegible]

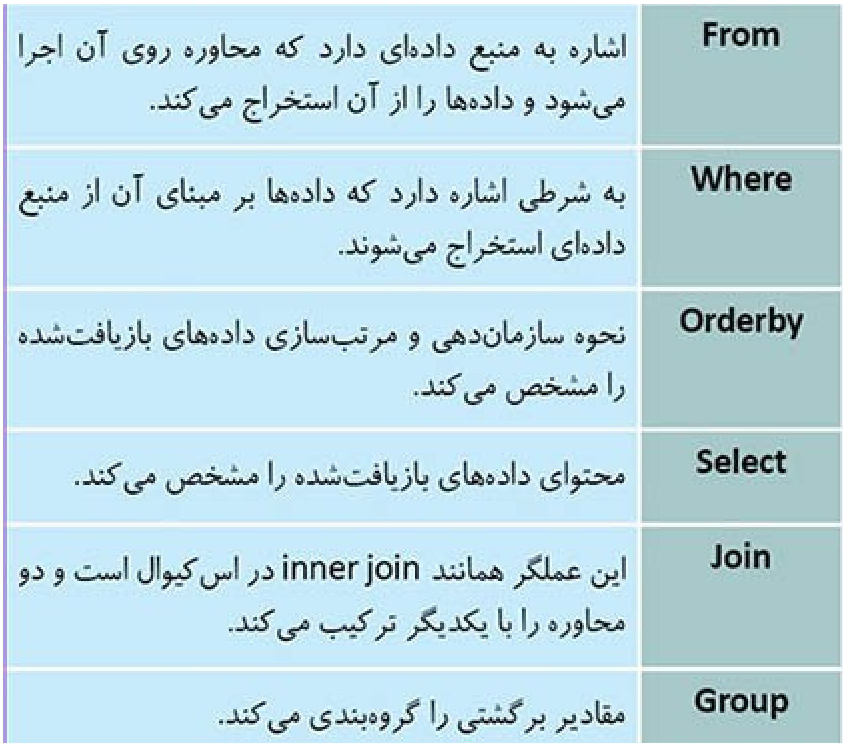
[illegible]. [illegible]. [illegible] …[illegible]. [illegible]. [illegible] SQO [illegible] Enumerable [illegible] Queryable [illegible]. [illegible]. [illegible] System.Linq [illegible]. [illegible].



LINQ to Object	به برنامه‌نویس اجازه می‌دهد محاوره‌ها را روی ساختارهای داده‌ای مقیم در حافظه همانند فهرست‌های ژنریک و آرایه‌ها استفاده کند.
LINQ to Dataset	ساخت محاوره‌ها برای کار با مجموعه داده‌ها در Ado.Net را امکان‌پذیر می‌سازد.
LINQ to SQL	به‌منظور پیاده‌سازی محاوره‌ها برای استخراج و به‌روزرسانی داده‌ها روی بانک‌های اطلاعاتی SQL Server استفاده می‌شود.
LINQ to Entities	این گروه به برنامه‌نویسان اجازه می‌دهد محاوره‌ها را برای دسترسی به بانک‌های اطلاعاتی مختلف طراحی کنند.
LINQ to XML	امکان طراحی محاوره‌ها و ویرایش داده‌های XML مقیم در حافظه یا داده‌های XML ذخیره‌شده در یک فایل XML را فراهم می‌آورد.

[illegible] 1: [illegible].

[illegible].
LINQ to Active Directory [illegible]. [illegible] LINQ to Objects [illegible] LINQ to SQL [illegible]. [illegible] SQL، [illegible] LINQ to SQL [illegible]. [illegible]. [illegible]. [illegible].



From	اشاره به منبع داده‌ای دارد که محاوره روی آن اجرا می‌شود و داده‌ها را از آن استخراج می‌کند.
Where	به شرطی اشاره دارد که داده‌ها بر مبنای آن از منبع داده‌ای استخراج می‌شوند.
Orderby	نحوه سازمان‌دهی و مرتب‌سازی داده‌های بازیافت‌شده را مشخص می‌کند.
Select	محتوای داده‌های بازیافت‌شده را مشخص می‌کند.
Join	این عملگر همانند inner join در اس‌کیوال است و دو محاوره را با یکدیگر ترکیب می‌کند.
Group	مقادیر برگشتی را گروه‌بندی می‌کند.

[illegible] :2 [illegible].

[illegible]

[illegible] foreach [illegible].

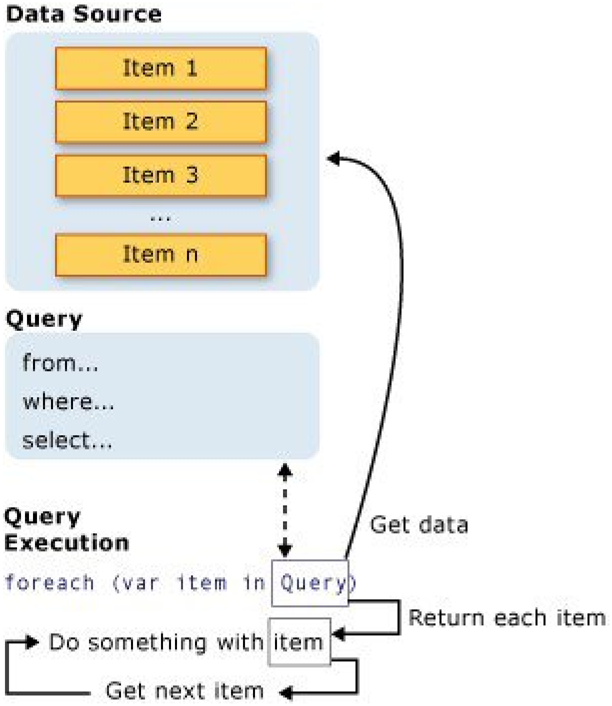
[illegible] foreach [illegible].

```
static void Main(string[] args)
        {

         int[] numbers = new int[7] { 0, 1, 2, 3, 4, 5, 6 };
        var myQuery =
            from num in numbers
            where (num % 2) == 0
            select num;
        foreach (int num in myQuery)
        {
            Console.Write("{0,1} ", num);
        }
       }
```

[illegible]

[illegible] 1 [illegible].

[illegible].



[illegible] :1 [illegible]

[illegible]

[illegible] IEnumerable<T> [illegible] IEnumerable [illegible] foreach [illegible] IEnumebrale [illegible] IEnumerable<T> [illegible] IEnumerable<T> [illegible] IQueryable<T> [illegible] SQL Server [illegible] LINQ to SQL [illegible] LINQ to Dataset [illegible] XML [illegible] LINQ to XML [illegible].

[illegible].

[illegible]

[illegible] foreach [illegible] for [illegible].

```
static void Main(string[] args) { String[] myArray = { “One”, “Two”, “Three”, “Four”, “Five” };
        var MyQuery =
            from mystring
            in myArray
          select mystring;
        foreach(var str in MyQuery)
        Console.WriteLine(str); }
```

[illegible]

[illegible]

[illegible] where [illegible] from [illegible] select [illegible] where [illegible]

```
      static void Main(string[] args)
      {

        String[] myArray = { “One”, “Two”, “Three”, “Four”, “Five” };
          var MyQuery =
from mystring
in myArray
where mystring.Length > 3
select mystring;

          foreach(var str in MyQuery)
          Console.WriteLine(str);


      }
```

[illegible]

[illegible]

[illegible] orderby [illegible] orderby [illegible] orderby [illegible]

```
    class Program
    {
      public class myclass
      {
      public string Name { get; set; }
      public int Age { get; set; }
    }
      static void Main(string[] args)
      {

        myclass[] mystrings = { new myclass { Name=”Barley”, Age=8 },
              new myclass { Name=”Boots”, Age=4 },
            new myclass { Name=”Whiskers”, Age=1 } };

        IEnumerable<myclass> query = mystrings.OrderBy(Program => Program.Age);

        foreach (myclass mystr in query)
        {
            Console.WriteLine(“{0} - {1}”, mystr.Name, mystr.Age);
        }

    }
}
```

[illegible]

[illegible]

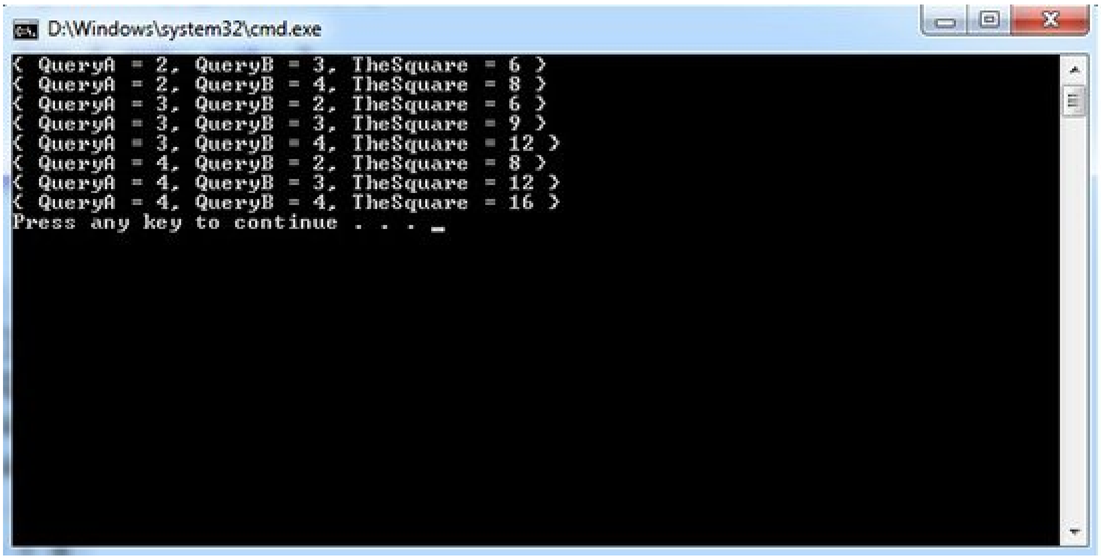
[illegible]

```
static void Main(string[] args)
    {

        Int32[] FirstArray= {1,2,3,4,5};
        Int32[] SecondArray = { 6,7,8,5,4};

        var MyQuery = from QueryA in FirstArray
                      from QueryB in SecondArray
                       where QueryA == QueryB
                       select new { QueryA, QueryB };

        foreach(var str in MyQuery)
        Console.WriteLine(str);
    }
```

[illegible]

[illegible]

[illegible] Let [illegible] Let [illegible] (2 [illegible]



```
{ QueryA = 2, QueryB = 3, TheSquare = 6 }
{ QueryA = 2, QueryB = 4, TheSquare = 8 }
{ QueryA = 3, QueryB = 2, TheSquare = 6 }
{ QueryA = 3, QueryB = 3, TheSquare = 9 }
{ QueryA = 3, QueryB = 4, TheSquare = 12 }
{ QueryA = 4, QueryB = 2, TheSquare = 8 }
{ QueryA = 4, QueryB = 3, TheSquare = 12 }
{ QueryA = 4, QueryB = 4, TheSquare = 16 }
Press any key to continue . . .
```

[illegible] :2 [illegible]

```
static void Main(string[] args)
    {

       Int32[] ArrayA = {1,2,3,4};
        Int32[] ArrayB = { 1,2,3,4};
        var MyQuery =
from QueryA in ArrayA
from QueryB in ArrayB
let TheSquare = QueryA * QueryB
where TheSquare > 4
select new { QueryA, QueryB, TheSquare };
        foreach(var str in MyQuery)
        Console.WriteLine(str); }
```

[illegible]

.[illegible]

:[illegible]
[[illegible]](#)
:[illegible]
[[illegible]](#)
:[illegible]
10:05 - 24/11/1395
:[illegible]
[[illegible]](#) - [LINQ](#)

---

https://www.shabakeh-mag.com/workshop/6506:[illegible]