

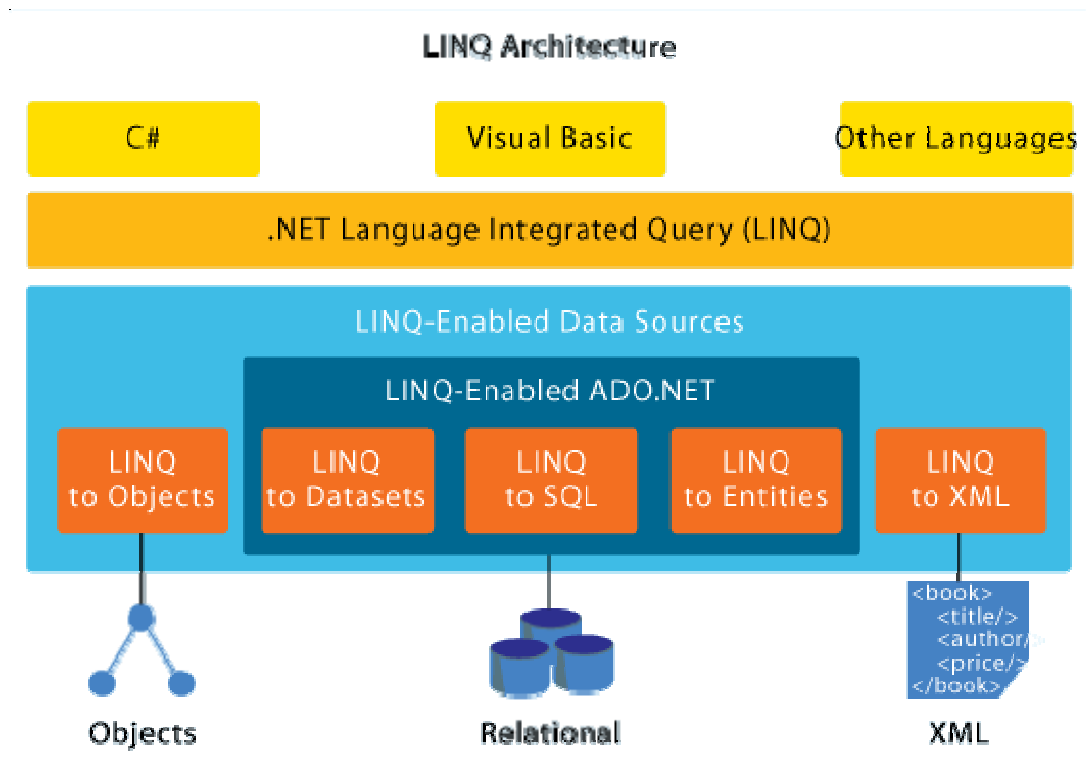
LINQ از کلمات Language-Integrated Query گرفته شده است.

✓ LINQ نوشتن Query ها رو ساده تر می کنه. درضمن اگه شما با زبان های برنامه نویسی C# یا VB کار کرده باشید لازم نیست چیزهای زیادی یاد بگیرید تا بتونید پرس و جویهای LINQ بنویسید.

✓ این ابزار Query گرفتن برای Data Source های مختلف را یکسان کرده. شما می تونید به همون روشی که روی یک بانک SQL Server پرس و جو کنید، روی Dataset یا XML Document و حتی Collection ها پرس و جو کنید.

✓ LINQ ارتباطی بین داده های Relational و دنیای Object Oriented ایجاد می کنه. چون Query روی داده ها همیشه اولین مرحله انجام پروسه کار بر روی داده هاست و با به کار گرفتن LINQ به سادگی میشه ارتباط بین Object ها و داده ها رو تعریف کرد.

✓ LINQ سرعت تولید نرم افزار رو با گرفتن خطاهای برنامه در زمان Compile افزایش می دهد و خطاهای Query ها رو می شه با ابزارهای Debug که Visual studio در اختیار می زاره تشخیص داد و تصحیح کرد.



(LINQ to SQL) SELECT STATEMENT

```
LINQ db =  
    new LINQ("Data Source=(local);Initial Catalog=AR_LINQ");  
  
var q = from c in db.Contact  
        where c.DateOfBirth.AddYears(35) > DateTime.Now  
        orderby c.DateOfBirth descending  
        select c;  
  
foreach(var c in q)  
    Console.WriteLine("{0} {1} b.{2}",  
        c.FirstName.Trim(),  
        c.LastName.Trim(),c.DateOfBirth.ToString("dd-MMM-yyyy"));
```

Output:

```
Aryadad Sepand b.17-Sep-1977  
John smith b.09-Dec-1973
```

(LINQ to SQL) INSERT STATEMENT

```
LINQ db =
```

```
    new LINQ("Data Source=(local);Initial Catalog=AR_LINQ");
```

```
Contacts newContact    = new Contacts();
```

```
newContact.FirstName    = "Aryadad";
```

```
newContact.LastName     = "Sepand";
```

```
newContact.Phone        = "021 88413800";
```

```
newContact.Email        = "info@aryadadcom";
```

```
newContact.DateOfBirth = new DateTime(2007, 10, 29);
```

```
db.Contacts.Add(newContact);
```

```
db.SubmitChanges();
```

(LINQ to SQL) UPDATE STATEMENT

```
LINQ db =  
    new LINQ("Data Source=(local);Initial Catalog=AR_LINQ");  
  
Contacts q = (from in in db.Contacts  
    where c.FirstName == "Aryadad" && c.LastName == "Sepand"  
    select c).FirstOrDefault();  
  
if (q != null) {  
    q.Email = "support@aryadad.com";  
}  
  
db.SubmitChanges();
```

(LINQ to SQL) DELETE STATEMENT

```
LINQ db =  
    new LINQ("Data Source=(local);Initial Catalog=AR_LINQ");  
  
Contacts q = from in in db.Contacts  
              where c.FirstName == "Aryadad" && c.LastName == "Sepand"  
              select c;  
  
db.Contacts.RemoveAll(q);  
db.SubmitChanges();
```

(LINQ to XML)

```
XElement xml = new XElement("contacts",  
    new XElement("contact",  
        new XAttribute("contactId", "2"),  
        new XElement("firstName", "Aryadad"),  
        new XElement("lastName", "Sepand")  
    ),  
    new XElement("contact",  
        new XAttribute("contactId", "3"),  
        new XElement("firstName", "John"),  
        new XElement("lastName", "Smith")  
    )  
);  
  
Console.WriteLine(xml);
```

Output:

```
<contacts>  
  <contact contactId="2">  
    <firstName>Aryadad</firstName>  
    <lastName>Sepand</lastName>  
  </contact>  
  <contact contactId="3">  
    <firstName>John</firstName>  
    <lastName>Smith</lastName>  
  </contact>  
</contacts>
```