# Визуал програмчлал

F.IT203

Лекц-11

# Гарчиг

- LINQ
- LINQ – төрөл
- LINQ жишээ

# LINQ(Language INtegrated Query)

- Өгөгдлийн эхтэй (data source) харилцан ажилладаг.
- .NET хэлнүүдийн боломжийг дээшлүүлдэг Framework-ийн хүрээнд багтсан компонент.
- Өгөгдлийн эхээс уншисан утгуудыг тодорхой шинжүүрээр шүүж харуулах бололцоотой.
- Ингэж шүүн харахдаа асуулга(Query)-г хэрэглэдэг.

# Өгөгдлийн эх

- Өгөгдлийн эх үүсвэр нь өгөгдлийн сан, XML файл болон текст файлууд байж болно.
- Өгөгдлийн эх үүсвэр рүү илгээх хүсэлт нь тодорхой шинжүүрээр шүүж харахыг хүссэн Query буюу асуулга байна.

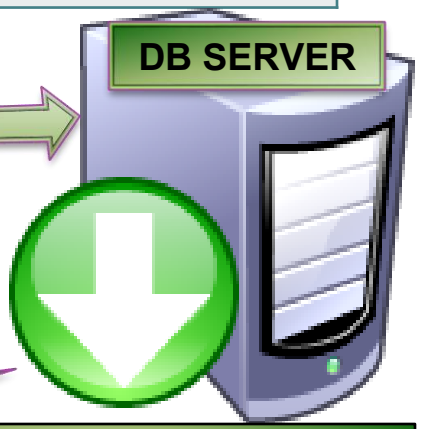
# Асуулга гэж юу вэ?

- Өгөгдлийн сангийн сервэр програмыг ямар үйлдэл хийж өгөхийг нь зааж өгдөг коммандуудын багцыг Query буюу Асуулга гэж нэрлэнэ.
- Өөрөөр хэлбэл Өгөгдлийн сангийн сервэртэй харьцаж байгаа хэл юм.

# ӨСУС-т илгээх Асуулга(Query)

| Employee | |
|---|---|
| ID | Num |
| 1 | 15 |
| 2 | 10 |

```
Select * from Employee
where Num>=13
```

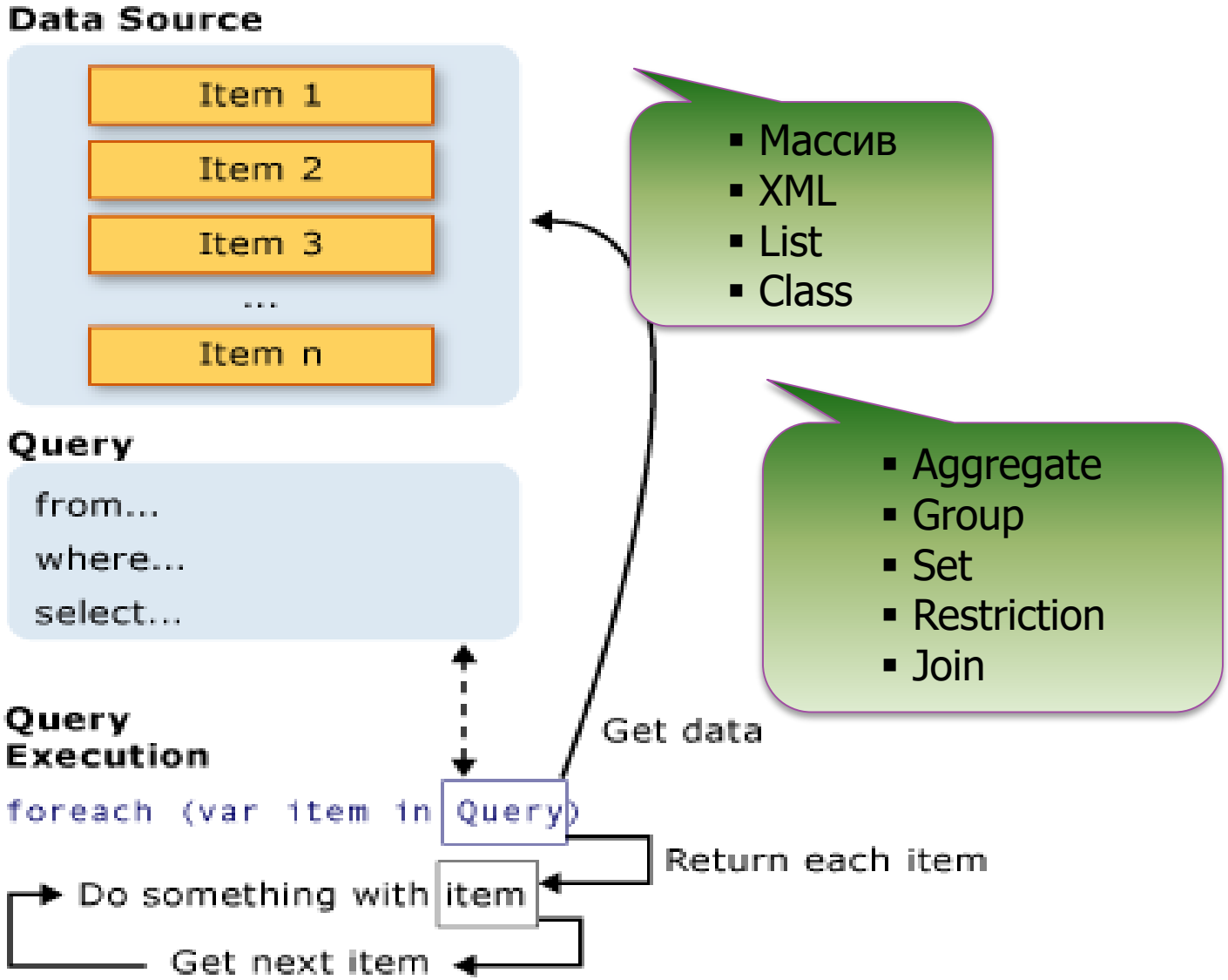



```
<?xml version="1.0”?>
<Employee>
  <id>1</to>
  <num>15</num>
</Employee>
```

# LINQ(Language INtegrated Query)

- Өөрөөр хэлбэл өгөгдлийн санд ашиглагддаг энэ асуулгыг(query) програмын хэл дээр ашиглана.
- LINQ-асуулгын биелэлт нь дараах 3 хэсгээс бүрдэнэ.
  1. Өгөгдлийн эхтэй холбох
  2. Асуулга(Query) үүсгэх
  3. Асуулгыг биелүүлэх

# Програмын хэлд ашиглагдах Асуулга буюу LINQ-н бүтэц



# LINQ Query

***Асуулга нь дараах төрлийн операторуудтай.***

- Restriction буюу хязгаарлалтын оператор
- Групплэх операторууд
- Агергат функцууд ашиглах
- Эрэмбэлэх  оператор
- Холболтын операторууд

# LINQ Хязгаарлалтын оператор

- Заасан нөхцөлийн дагуу өгөгдлийн эхээс бичлэгүүдийг сонгоно.
- Сонгох нөхцөл нь логик илэрхийлэл байна.

<атрибутын нэр><Харьцуулах үйлдэл><Тогтмол>
<атрибутын нэр><Харьцуулах үйлдэл><атрибутын нэр>

# Жишээ 1

```
class IntroToLINQ
{
    static void Main()
    {
        // The Three Parts of a LINQ Query:
        //  1. Data source.
```

```
int[] numbers = new int[7] { 0, 1, 2, 3,
4, 5, 6 };
```

# Жишээ 1 үргэлжлэл

```
// 2. Query creation.
// numQuery is an IEnumerable<int>
var numQuery =
            from num in numbers
            where (num % 2) == 0
            select num;
// 3. Query execution.
foreach (int num in numQuery)
        {
            Console.Write("{0,1} ", num);
        }
  }
}
```

# Жишээ 2

```
public void Linq1()
{
    int[] numbers = { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0 };
    var lowNums = from n in numbers
        where n < 5 select n;
    Console.WriteLine("Numbers < 5:");

    foreach (var x in lowNums){
        Console.WriteLine(x);
    }
}
```

# LINQ Бүлэглэлтийн оператор

- Өгөгдлийн эх үүсвэр дахь утгуудыг ижил утга агуулсан нэг буюу хэд хэдэн атрибутын дагуу нэгтгэж багцлахыг бүлэглэлт гэнэ.

***Ач холбогдол***

- Өгөгдлийн эх үүсвэр дахь бичлэгүүдийг нэгтгэж харах
- Бүлэг дотор тооцоо бодолт хийх

# Жишээ 3

```
public void Linq41()
{
    string[] words = { "blueberry", "chimpanzee",
   "abacus", "banana", "apple", "cheese" };

    var wordGroups =
        from w in words
        group w by w[0];
```

# Жишээ 3 үргэлжлэл

```
foreach (var g in wordGroups)
    {
        Console.WriteLine("Words that start with the letter
                                        '{0}':", g.FirstLetter);

        foreach (var w in g.Words)
        {
            Console.WriteLine(w);
        }
    }
}
```

# LINQ агергат функц

Бүлэгт агергат функц ашиглаж нэгтгэл хийх шаардлага их гардаг. Агергат функцуудыг ихэвчлэн Group оператортой хамт ашиглана.

- Sum(нийлбэр олох)
- Бүлэг дэх бичлэгийн тоог олох-(Count)
- Бүлэг дэх хамгийн их бага утгуудыг олох-(max, min)
- Бичигдэх хэлбэр:

<атрибутын жагсаалт> (<бүлэглэх түлхүүр>[<томъёо><агергат функц>](<өгөгдлийн эх>))

# LINQ агергат функц

***Тайлбар:***

- Атрибутын жагсаалт – бүлэглэх утгын хувьд зөвхөн нэг утгатай байх хадгалсан ба бүлгийн хувьд томъёо функцээр гаргаж авсан атрибут
- томъёо /функц нь бүлэглэх түлхүүрийн хувьд зөвхөн нэг утга буцаах ёстой.

# Жишээ - 4

```
public void Linq73()
{
    int[] a = { 2, 2, 3, 5, 5 };

    int b = a.Distinct().Count();

    Console.WriteLine("There are {0} unique factors of
   massiv", b);
}
```

# Жишээ 5

```
public void Linq74()
{
    int[] numbers = { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0 };

    int oddNumbers = numbers.Count(n => n % 2 == 1);

    Console.WriteLine("There are {0} odd numbers in
   the list.", oddNumbers);
}
```

# Жишээ 6

```
public void Linq79()
{
    string[] words = { "cherry", "apple", "blueberry" };

    double totalChars = words.Sum(w => w.Length);

    Console.WriteLine("There are a total of {0}
   characters in these words.", totalChars);
}
```

# Жишээ 7

```
public void Linq85()
{
    int[] numbers = { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0 };

    int maxNum = numbers.Max();

    Console.WriteLine("The maximum number is {0}.",
   maxNum);
}
```

# LINQ Range

```
public void Linq65()
{
    var numbers =
        from n in Enumerable.Range(100, 50)
        select new { Number = n, OddEven = n % 2 == 1 ? "odd" :
    "even" };
    foreach (var n in numbers)
    {
        Console.WriteLine("The number {0} is {1}.", n.Number,
    n.OddEven);
    }
}
```

# LINQ Ordering

- Эрэмбийг нэг буюу олон төвшнөөр зохион байгуулж болно.
- Эрэмбэлэх түлхүүрийг нэг утгаар заасан үед нэг төвшний эрэмбэлэлт
- Харин нэгээс илүү атрибутаас заасан үед олон төвшний эрэмбэлэлт хийгдэнэ.

# Жишээ 8

```
public void Linq28()
{
    string[] words = { "cherry", "apple", "blueberry" };

    var sortedWords =
        from w in words
        order by w
        select w;

    Console.WriteLine("The sorted list of words:");
    foreach (var w in sortedWords)
    {
        Console.WriteLine(w);
    }
}
```

# Жишээ-9

```
public void Linq29(){
    string[] words = { "cherry", "apple", "blueberry" };

    var sortedWords = from w in words orderby
   w.Length select w;

    Console.WriteLine("The sorted list of words (by
   length):");
    foreach (var w in sortedWords){
        Console.WriteLine(w);
    }
}
```