Left Join or Left Outer join

Left Join

In left join or left outer join -

"In this join all data from first data source is returned, regardless of whether it has any related data in second data source."

If for few (or all) records data is not available in second data source then null value is returned for second data source.

Left join or left outer join both are same because outer keyword is optional.


```
0 references
static void Main(string[] args)
{
    var students = new List<Student>()
    {
        new Student(){ Id =1, Name = "Maria",AddressId = 1},
        new Student(){ Id =2, Name = "Amelia", AddressId = 2},
        new Student(){ Id =3, Name = "Rebecca"},
        new Student(){ Id =4, Name = "Una", AddressId = 4},
        new Student(){ Id =5, Name = "Victoria", AddressId = 5},
    };

    var address = new List<Address>()
    {
        new Address(){Id = 1, AddressLine = "Maria address" },
        new Address(){Id = 2, AddressLine = "Amelia address" },
        new Address(){Id = 3, AddressLine = "Una address" },
    };



    Console.ReadLine();
```

```
var qs = (from std in students
          join add in address on std.AddressId equals add.Id into stdAddress
          from studentAddess in stdAddress.DefaultIfEmpty()
          select new { StudentName = std.Name, StudentAddress = studentAddess != null ? studentAddess.AddressLine : "NA" }).

var ms = students.GroupJoin(address, std => std.AddressId, add => add.Id,
                            (std, add) => new { std, add })

                            .SelectMany(x =>
                            x.add.DefaultIfEmpty(),
                            (studentData, addressData) =>
                            new
                            {
                                studentData.std,
                                addressData
                            }).ToList();


Console.ReadLine();
```

Output

```
    new Address(){Id = 2, AddressLine = "Amelia address" },
    new Address(){Id = 3, AddressLine = "Una address" },
};


var qs = (from std in students
          qs Count = 5                ddress on std.AddressId equals add.Id into stdAddress
            [0]   { StudentName = "Maria", StudentAddress = "Maria address" }
            [1]   { StudentName = "Amelia", StudentAddress = "Amelia address" }   dentAddess != null ? studentAdde
            [2]   { StudentName = "Rebecca", StudentAddress = "NA" }
            [3]   { StudentName = "Una", StudentAddress = "Una address" }
            [4]   { StudentName = "Victoria", StudentAddress = "NA" }
            Raw View
Console.
```