

Select

```
List<Employee> employees = new List<Employee>()
{
    new Employee(){ Id=1, Name = "Tom", Email="tom@gmail.com"},
    new Employee(){ Id=2, Name = "John", Email="john@gmail.com"},
    new Employee(){ Id=3, Name = "Mark", Email="mark@gmail.com"},
    new Employee(){ Id=4, Name = "Kim", Email="kim@gmail.com"} ,
    new Employee(){ Id=5, Name = "Adam", Email="adam@gmail.com"}
};
```

```
List<Employee> employees = new List<Employee>()
{
    new Employee(){ Id=1, Name = "Tom", Email="tom@gmail.com"},
    new Employee(){ Id=2, Name = "John", Email="john@gmail.com"},
    new Employee(){ Id=3, Name = "Mark", Email="mark@gmail.com"},
    new Employee(){ Id=4, Name = "Kim", Email="kim@gmail.com"} ,
    new Employee(){ Id=5, Name = "Adam", Email="adam@gmail.com"}
};
```

```
var basicQuery = from emp in employees
                 select emp;
```

It will not executed over there only executed if perform some method like foreach, Select, ToList etc

```
var basicQuery = (from emp in employees
                  select emp).ToList();

var basicMethod = employees.ToList();

foreach (var item in basicMethod)
{
    Console.WriteLine($"Id = {item.Id}, Name = {item.Name}");
}

Console.ReadLine();
```

//getting only a property

```
var basicPropQuery = (from emp in employees
                      select emp.Id + 1).ToList();

var basicPropMethod = employees.Select(emp => emp.Id).ToList();
```

```
var basicPropQuery = (from emp in employees
                      select emp.Id.ToString()).ToList();
```

Selecting specific properties

```
var selectQuery = (from emp in employees
                   select new Employee()
                   {
                       Id = emp.Id,
                       Email = emp.Email
                   }).ToList();

foreach (var item in selectQuery)
{
    Console.WriteLine($"Id = {item.Id}, Name = {item.Name}, Email = {item.Email}");
}
```

The name will be null here

Converting one type to another type

```

namespace LinqSample1
{
    0 references
    class Student
    {
        0 references
        public int StudentId { get; set; }

        0 references
        public string StEmail { get; set; }

        0 references
        public string FullName { get; set; }
    }
}

```

Converting one object to another

```

var selectQuery = (from emp in employees
    select new Student()
    {
        StudentId = emp.Id,
        FullName = emp.Name,
        StEmail = emp.Email
    }).ToList();

var selectMethod = employees.Select(emp => new Student()
{
    StudentId = emp.Id,
    FullName = emp.Name,
    StEmail = emp.Email
}).ToList();

foreach (var item in selectMethod)
{
    Console.WriteLine($"Id = {item.StudentId}, Name = {item.FullName}, Email = {item.StEmail}");
}

Console.ReadLine();

```

Anonyms class creation no error

```

var selectQuery = (from emp in employees
    select new
    {
        CustomId = emp.Id,
        CustomName = emp.Name,
        CustomEmail = emp.Email
    }).ToList();

var selectMethod = employees.Select(emp => new
{
    CustomId = emp.Id,
    CustomName = emp.Name,
    CustomEmail = emp.Email
}).ToList();

foreach (var item in selectMethod)
{
    Console.WriteLine($"Id = {item.CustomId}, Name = {item.CustomName}, Email = {item.CustomEmail}");
}

Console.ReadLine();

```

Selecting data using index

```

var query = employees.Select((emp, index) => new { Index = index, FullName = emp.Name }).ToList();

```

query Count = 5

[0]	{ Index = 0, FullName = "Tom" }
[1]	{ Index = 1, FullName = "John" }
[2]	{ Index = 2, FullName = "Mark" }
[3]	{ Index = 3, FullName = "Kim" }
[4]	{ Index = 4, FullName = "Adam" }

Console.WriteLine(\$"Id = {item.CustomId}, Name = {item.CustomName}, Email = {item.CustomEmail}");

Object inside a object

```

new Techs() { Technology = "SQL" },
new Techs() { Technology = "SQL" }
},
new Employee() { Id=3, Name = "Mark", Email="mark@gmail.com", Programming =
new List<Techs>{
    new Techs() { Technology = "LINQ"},
    new Techs() { Technology = "MVC"},
    new Techs() { Technology = "C#" }
} },
new Employee() { Id=4, Name = "Kim", Email="kim@gmail.com", Programming = new List<Techs>() },
new Employee() { Id=5, Name = "Adam", Email="adam@gmail.com", Programming = new List<Techs>() }
};

var methodQuery = dataSource.SelectMany(emp => emp.Programming).ToList();

var querySyntax = (from emp in dataSource
    from pro in emp.Programming
    select pro).ToList();

```

Output

