**The following are the different types of joins in LINQ**  
Group Join - [Part 21](http://csharp-video-tutorials.blogspot.com/2014/07/part-21-group-join-in-linq.html)  
Inner Join - [Part 22](http://csharp-video-tutorials.blogspot.com/2014/08/part-22-inner-join-in-linq.html)  
Left Outer Join - [Part 24](http://csharp-video-tutorials.blogspot.com/2014/08/part-24-left-outer-join-in-linq.html)  
Cross Join - We will discuss in this video   
  
   
  
In this video we will discuss implementing **CROSS JOIN in LINQ**.   
  
**Cross join** produces a cartesian product i.e when we cross join two sequences, every element in the first collection is combined with every element in the second collection. The total number of elements in the resultant sequence will always be equal to the product of the elements in the two source sequences. The on keyword that specfies the JOIN KEY is not required.  
  
Let us understand implementing Cross Join with an example. Consider the following **Department**and **Employee**classes. 

public class Department

{

    public int ID { get; set; }

    public string Name { get; set; }

    public static List<Department> GetAllDepartments()

    {

        return new List<Department>()

        {

            new Department { ID = 1, Name = "IT"},

            new Department { ID = 2, Name = "HR"},

        };

    }

}

public class Employee

{

    public int ID { get; set; }

    public string Name { get; set; }

    public int DepartmentID { get; set; }

    public static List<Employee> GetAllEmployees()

    {

        return new List<Employee>()

        {

            new Employee { ID = 1, Name = "Mark", DepartmentID = 1 },

            new Employee { ID = 2, Name = "Steve", DepartmentID = 2 },

            new Employee { ID = 3, Name = "Ben", DepartmentID = 1 },

            new Employee { ID = 4, Name = "Philip", DepartmentID = 1 },

            new Employee { ID = 5, Name = "Mary", DepartmentID = 2 },

        };

    }

}

**Example 1 :**Cross Join **Employees**collection with **Departments**collections.

var result = from e in Employee.GetAllEmployees()

                    from d in Department.GetAllDepartments()

                    select new { e, d };

foreach (var v in result)

{

    Console.WriteLine(v.e.Name + "\t" + v.d.Name);

}

**Output:**We have 5 elements in **Employees**collection and 2 elements in **Departments**collection. In the result we have 10 elements, i.e the cartesian product of the elements present in Employees and Departments collection. Notice that every element from the Employees collection is combined with every element in the Departments collection.   
   
  
**Example 2 :** Cross Join **Departments**collections with **Employees**collection

var result = from d in Department.GetAllDepartments()

                    from e in Employee.GetAllEmployees()

                    select new { e, d };

foreach (var v in result)

{

    Console.WriteLine(v.e.Name + "\t" + v.d.Name);

}

**Output:** Notice that the output in this case is slightly different from **Example 1**. In this case, every element from the Departments collection is combined with every element in the Employees collection.   
   
  
**Example 3 :** Rewrite **Example 1**using extension method syntax  
  
To implement **Cross Join**using extension method syntax, we could either use SelectMany() method or Join() method  
  
**Implementing cross join using SelectMany()**

var result = Employee.GetAllEmployees()

                        .SelectMany(e => Department.GetAllDepartments(), (e, d) => new { e, d });

foreach (var v in result)

{

    Console.WriteLine(v.e.Name + "\t" + v.d.Name);

}

**Implementing cross join using Join()**

var result = Employee.GetAllEmployees()

                                     .Join(Department.GetAllDepartments(),

                                               e => true,

                                               d => true,

                                               (e, d) => new { e, d });

foreach (var v in result)

{

    Console.WriteLine(v.e.Name + "\t" + v.d.Name);

}