In this video, we will discuss the **difference between Group Join and Inner Join in LINQ** with examples. We will be using the following Department and Employee classes in this video.   
  
 

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"},

            new Department { ID = 3, Name = "XX"},

        };

    }

}

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 },

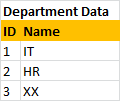
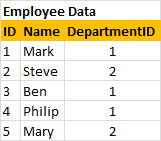
            new Employee { ID = 4, Name = "Philip", DepartmentID = 1 },

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

        };

    }

}

**Department**data returned by **GetAllDepartments()**method is shown below   
   
  
**Employee**data returned by **GetAllEmployees()**method is shown below   
   
  
**The following query performs a GroupJoin on the 2 lists**

var result = from d in Department.GetAllDepartments()

                    join e in Employee.GetAllEmployees()

                    on d.ID equals e.DepartmentID into eGroup

                    select new

                    {

                       Department = d,

                       Employees = eGroup

                    };

Notice that we are using the **join**operator and the into keyword to group the results of the join. To perform group join using extension method syntax, we use **GroupJoin()**Extension method as shown below.

var result = Department.GetAllDepartments()

                                        .GroupJoin(Employee.GetAllEmployees(),

                                         d => d.ID,

                                         e => e.DepartmentID,

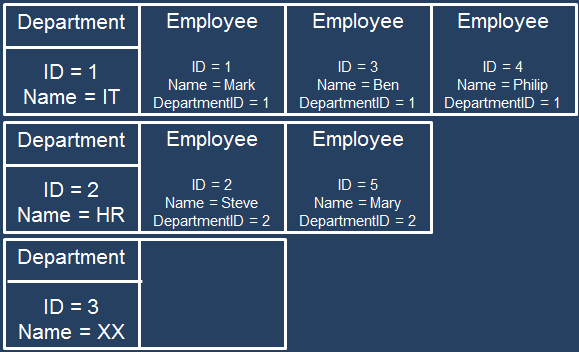
                                         (department, employees) => new

                                         {

                                              Department = department,

                                              Employees = employees

                                         });

The above 2 queries **groups employees by department**and would produce the following groups.   
  
  
To print the **Department**and **Employee**Names we use 2 foreach loops as shown below.

foreach (var department in result)

{

    Console.WriteLine(department.Department.Name);

    foreach (var employee in department.Employees)

    {

        Console.WriteLine(" " + employee.Name);

    }

    Console.WriteLine();

}

The following query performs an **Inner Join**on the 2 lists

var result = from e in Employee.GetAllEmployees()

                    join d in Department.GetAllDepartments()

                    on e.DepartmentID equals d.ID

                    select new { e, d };

To perform an **inner join**using extension method syntax, we use **Join()**Extension method as shown below.

var result = Employee.GetAllEmployees()

                                     .Join(Department.GetAllDepartments(),

                                      e => e.DepartmentID,

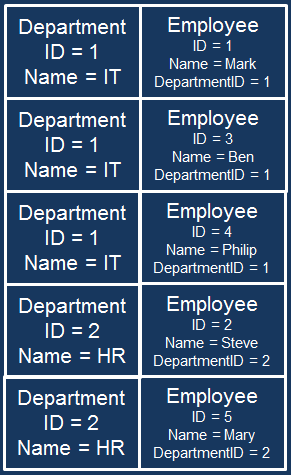
                                      d => d.ID, (employee, department) => new

                                      {

                                           e = employee,

                                           d = department

                                      });

The above 2 queries would produce a **flat result set**as shown below   
   
  
To print the **Department**and **Employee**Names we use just 1 foreach loop as shown below.

foreach (var employee in result)

{

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

}

In short, **Join**is similar to **INNER JOIN**in SQL and **GroupJoin**is similar to **OUTER JOIN**in SQL