**GroupBy** operator belong to **Grouping Operators**category. This operator takes a flat sequence of items, organize that sequence into groups (**IGrouping<K,V>**) based on a specific key and return groups of sequences.    
  
   
  
In short, **GroupBy**creates and returns a sequence of **IGrouping<K,V>**  
  
Let us understand GroupBy with examples.  
  
We will use the following **Employee**class in this demo

public class Employee

{

    public int ID { get; set; }

    public string Name { get; set; }

    public string Gender { get; set; }

    public string Department { get; set; }

    public int Salary { get; set; }

    public static List<Employee> GetAllEmployees()

    {

        return new List<Employee>()

        {

            new Employee { ID = 1, Name = "Mark", Gender = "Male",

                                         Department = "IT", Salary = 45000 },

            new Employee { ID = 2, Name = "Steve", Gender = "Male",

                                         Department = "HR", Salary = 55000 },

            new Employee { ID = 3, Name = "Ben", Gender = "Male",

                                         Department = "IT", Salary = 65000 },

            new Employee { ID = 4, Name = "Philip", Gender = "Male",

                                         Department = "IT", Salary = 55000 },

            new Employee { ID = 5, Name = "Mary", Gender = "Female",

                                         Department = "HR", Salary = 48000 },

            new Employee { ID = 6, Name = "Valarie", Gender = "Female",

                                         Department = "HR", Salary = 70000 },

            new Employee { ID = 7, Name = "John", Gender = "Male",

                                         Department = "IT", Salary = 64000 },

            new Employee { ID = 8, Name = "Pam", Gender = "Female",

                                         Department = "IT", Salary = 54000 },

            new Employee { ID = 9, Name = "Stacey", Gender = "Female",

                                         Department = "HR", Salary = 84000 },

            new Employee { ID = 10, Name = "Andy", Gender = "Male",

                                         Department = "IT", Salary = 36000 }

        };

    }

}

**Example 1:** Get Employee Count By Department

var employeeGroup = from employee in Employee.GetAllEmployees()

                    group employee by employee.Department;

foreach (var group in employeeGroup)

{

    Console.WriteLine("{0} - {1}", group.Key, group.Count());

}

**Output:**   
linq group by   
  
**Example 2:** Get Employee Count By Department and also each employee and department name

var employeeGroup = from employee in Employee.GetAllEmployees()

                                      group employee by employee.Department;

foreach (var group in employeeGroup)

{

    Console.WriteLine("{0} - {1}", group.Key, group.Count());

    Console.WriteLine("----------");

    foreach (var employee in group)

    {

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

    }

    Console.WriteLine(); Console.WriteLine();

}

**Output:**   
   
  
**Example 3:** Get Employee Count By Department and also each employee and department name. Data should be sorted first by Department in ascending order and then by Employee Name in ascending order.

var employeeGroup = from employee in Employee.GetAllEmployees()

                                      group employee by employee.Department into eGroup

                                      orderby eGroup.Key

                                      select new

                                      {

                                           Key = eGroup.Key,

                                           Employees = eGroup.OrderBy(x => x.Name)

                                      };

foreach (var group in employeeGroup)

{

    Console.WriteLine("{0} - {1}", group.Key, group.Employees.Count());

    Console.WriteLine("----------");

    foreach (var employee in group.Employees)

    {

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

    }

    Console.WriteLine(); Console.WriteLine();

}

**Output:**   
