# 1. Create a console application and add class named Employee with following field.

**Employee Class** EmployeelD (Integer) FirstName (String) LastName (String) Title (String)

# DOB (Date) DOJ (Date) City (String)

Solution:

using System;

using System.Collections.Generic; using System.Linq;

namespace LinQ\_Assignment

{

public class Program

{

static public void Main(String[] args)

{

Employee e = new Employee();

List<Employee> empList = new List<Employee>{

new Employee{EmployeeId = 1001, FirstName="Malcolm", LastName="Daruwalla", Title="Manager",DOB= new DateTime(1984,11,16), DOJ= new DateTime(2011,06,08), City="Mumbai" },

new Employee{EmployeeId = 1002, FirstName="Asdin", LastName="Dhalla", Title="AsstManager",DOB= new DateTime(1984,08,20), DOJ= new DateTime(2012,07,07), City="Mumbai" },

new Employee{EmployeeId = 1003, FirstName="Madhavi", LastName="Oza", Title="Consultant",DOB= new DateTime(1987,11,14), DOJ= new DateTime(2015,04,12), City="Pune" },

new Employee{EmployeeId = 1004, FirstName="Saba", LastName="Shaikh", Title="SE",DOB= new DateTime(1990,06,13), DOJ= new DateTime(2016,02,02), City="Pune" },

new Employee{EmployeeId = 1005, FirstName="Nazia", LastName="Shaikh", Title="SE",DOB= new DateTime(1991,03,08), DOJ= new DateTime(2016,02,02), City="Mumbai" },

new Employee{EmployeeId = 1006, FirstName="Amit", LastName="Pathak", Title="Consultant",DOB= new DateTime(1989,11,07), DOJ= new DateTime(2014,08,08), City="Chennai" },

new Employee{EmployeeId = 1007, FirstName="Vijay", LastName="Natrajan", Title="Consultant",DOB= new DateTime(1989,12,02), DOJ= new DateTime(2015,02,02), City="Pune" },

new Employee{EmployeeId = 1008, FirstName="Rahul", LastName="Dubey", Title="Associate",DOB= new DateTime(1993,11,11), DOJ= new DateTime(2014,11,06), City="Chennai" },

new Employee{EmployeeId = 1009, FirstName="Suresh", LastName="Mistery", Title="Associate",DOB= new DateTime(1992,12,02), DOJ= new DateTime(2014,03,12), City="Chennai" },

new Employee{EmployeeId=1010, FirstName="Sumit", LastName="Shah", Title="Manager", DOB=new DateTime(1991,04,12), DOJ = new DateTime(2016, 01,02), City="Pune"}

};

Console.WriteLine(" "); Console.WriteLine("Display All Employees");

Console.WriteLine("");

var query = from i in empList select i;

foreach (var h in query)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees Except from Mumbai city"); Console.WriteLine("");

var query1 = from i in empList where i.City != "Mumbai" select i;

foreach (var h in query1)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees Except AsstManger"); Console.WriteLine("");

var query2 = from i in empList

where i.Title != "AsstManager" select i;

foreach (var h in query2)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees Whose Lastname starts with S"); Console.WriteLine("");

var query3 = from i in empList

where i.LastName.StartsWith("S") select i;

foreach (var h in query3)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees Who were joined before 01-01-2015"); Console.WriteLine("");

var query4 = from i in empList

where i.DOJ < new DateTime(2015, 01, 01) select i;

foreach (var h in query4)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees Who were born after 01-01-1990"); Console.WriteLine("");

var query5 = from i in empList

where i.DOB > new DateTime(1990, 01, 01) select i;

foreach (var h in query5)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees Whose title is Consultant and Associate"); Console.WriteLine("");

var query6 = from i in empList

where i.Title == "Associate" || i.Title == "Consultant" select i;

foreach (var h in query6)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Total count of Employees "); Console.WriteLine("");

var query7 = (from i in empList

select i).Count(); Console.WriteLine(query7);

Console.WriteLine(" "); Console.WriteLine("Display Chennai Employees Count "); Console.WriteLine("");

var query8 = from i in empList where i.City == "Chennai" select i;

foreach (var h in query8)

{

Console.WriteLine(h.EmployeeId + " " + h.FirstName + " " + h.LastName + " " + h.Title

+ " " + h.DOB + " " + h.DOJ + " " + h.City);

}

Console.WriteLine(" "); Console.WriteLine("Display Highest employeeId");

Console.WriteLine("");

var query9 = from i in empList select i.EmployeeId;

int max = query9.Max(); Console.WriteLine(max);

Console.WriteLine(" "); Console.WriteLine("Display Employees Count Who were joined after 01-01-2015"); Console.WriteLine("");

var query10 = from i in empList

where i.DOJ > new DateTime(2015, 01, 01) select i.EmployeeId;

int c = query10.Count();

Console.WriteLine(c);

Console.WriteLine(" "); Console.WriteLine("Display Employees Count whose title is other that Associate"); Console.WriteLine("");

var query11 = from i in empList

where i.Title != "Associate" select i.EmployeeId;

int j = query11.Count(); Console.WriteLine(j);

Console.WriteLine(" "); Console.WriteLine("Display Employees Count according to their city"); Console.WriteLine("");

var query12 = empList.GroupBy(i => i.City).Select(e => new {

disnt = e.Select(l => l.EmployeeId).Distinct().Count()

});

foreach (var i in query12)

{

Console.WriteLine(i.disnt);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees Count according to their city and title"); Console.WriteLine("");

var query13 = empList.GroupBy(i => new { i.City, i.Title }).Select(e => new { disnt = e.Select(l => l.EmployeeId).Distinct().Count()

});

foreach (var i in query13)

{

Console.WriteLine(i.disnt);

}

Console.WriteLine(" "); Console.WriteLine("Display Employees who is younger in the list"); Console.WriteLine("");

var query14 = (from i in empList orderby i.DOB descending select i).Take(1);

foreach (var kk in query14)

{

Console.WriteLine(kk.EmployeeId + " " + kk.FirstName + " " + kk.Title + " " +

kk.DOB);

}

Console.WriteLine(" ");

}

}

}

using System;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LinQ\_Assignment

{

public class Employee

{

public int EmployeeId { get; set; } public string FirstName { get; set; } public string LastName { get; set; } public string Title { get; set; } public DateTime DOB { get; set; } public DateTime DOJ { get; set; } public string City { get; set; }

}

}