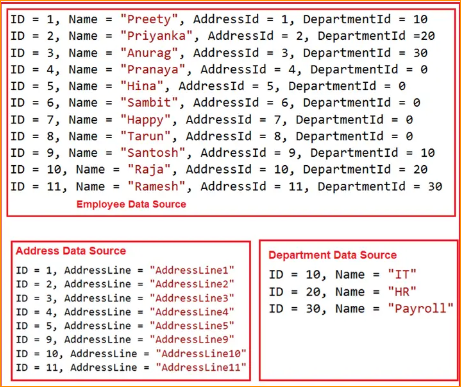
1. Examples to Understand LINQ Join with Multiple Data Sources in C#:

We are going to use the following three data sources i.e**. Employee, Address, and Department** data sources.

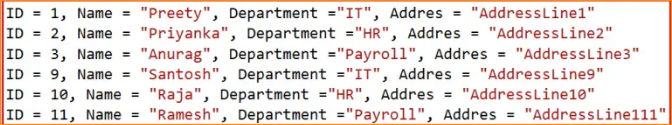
Here the **AddressId** property of the **Employee Data Source** points to the **ID property of the Address Data Source**. Similarly, the **DepartmentId** property of the Employee Data Source refers to the **ID** property of the **Department Data Source**. That means there are some common properties between these three data sources and using those common properties, we are going to perform the Join. The point that you need to remember is, in order to perform the Join Operation, we need to have common property.



* Now, our business requirement is to fetch the following data from the above three data sources.

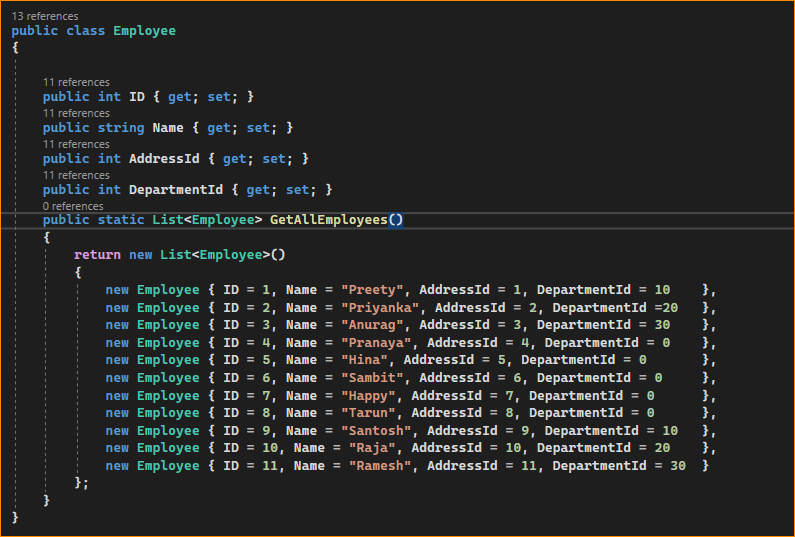
Notice, we are only required to fetch the data which exists in all three data sources. We need to ID and Name property from Employee Data Source, we need Department property from the Department Data Source and we need AddressLine Property value as Address from the Address Data Source.

Required Output-

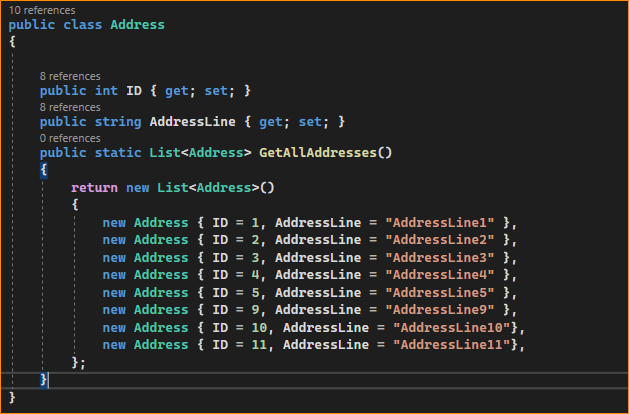


Creating the Model Classes and Data Sources:

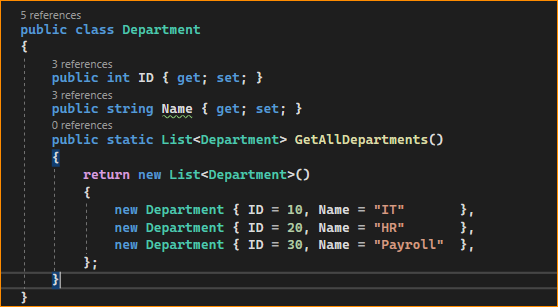
So, create a class file with the name Employee.cs and then copy and paste the following code into it. This is a very simple class having 4 properties i.e. Id, Name, and AddressId and DepartmentId. We have also created one method which is going to return a collection of Employees which is going to one of our data sources.



Now, create another class file with the name Address.cs and then copy and paste the following code into it. This is a very simple class having 2 properties i.e. Id, and AddressLine. We have also created one method which is going to return a collection of addresses which is going to be the second data source going to be used in the Inner Join.



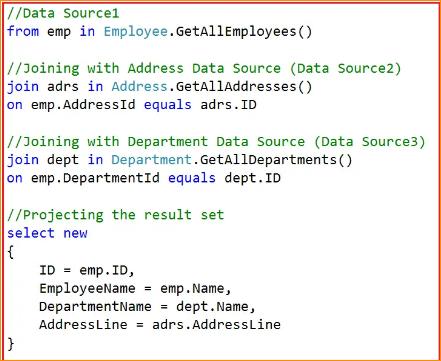
Finally, create another class file with the name Department.cs and then copy and paste the following code into it. This is a very simple class having 2 properties i.e. Id, and Name. We have also created one method which is going to return a collection of departments which is going to be the third data source.



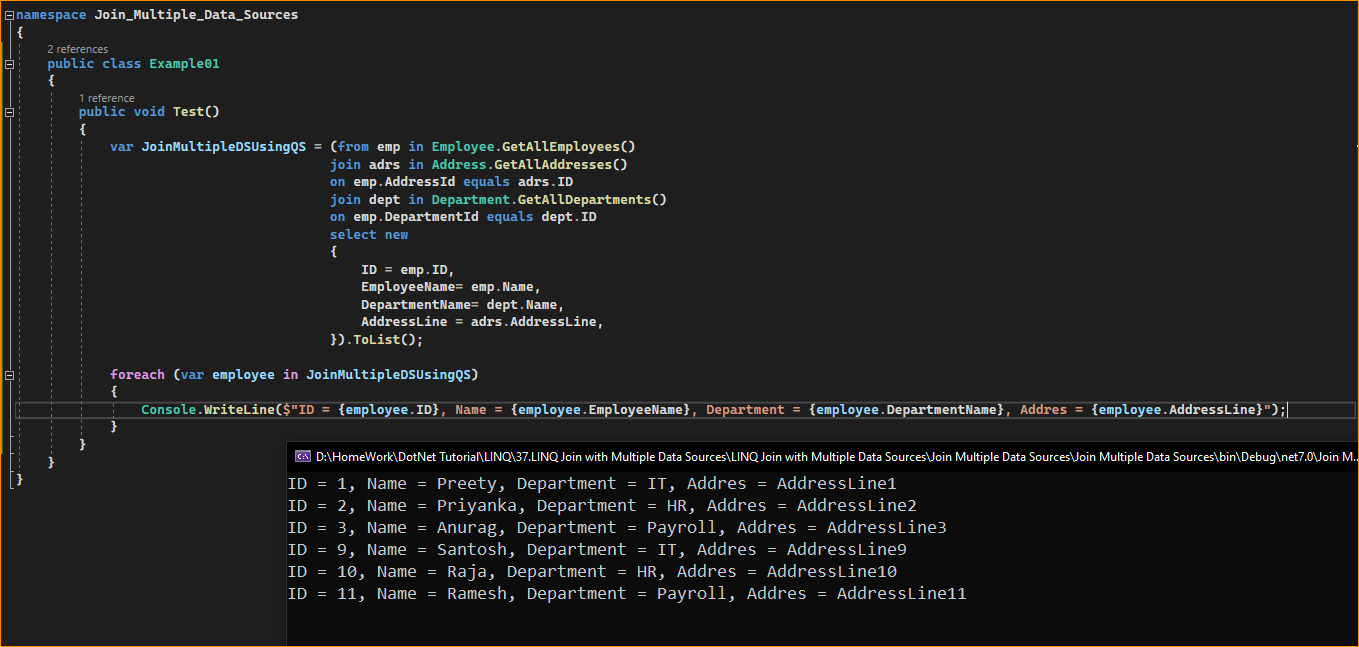
The following code snippet shows how to join three data sources using Linq Query Syntax in C#. This is very much similar to SQL Join.

Here, first, we are fetching the employee’s collection using the emp variable and then joining it with the addresses collection using the join operator (with the joining condition as on emp.AddressId equals adrs.ID).

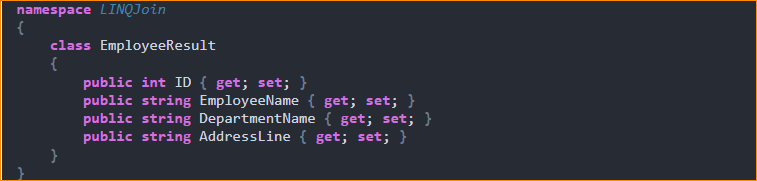
We are accessing the addresses collection using the adrs variable. Then again, we are joining the employee collection with the department collection using the join operator (with the joining condition as on emp.DepartmentId equals dept.ID). We are accessing the department collection using the dept variable. And finally projecting the result into an Annonymous Type. Here, we are fetching ID and EmployeeName properties from the Employees collection, DepartmentName from the Department collection, and AddressLine from the Address Collection.



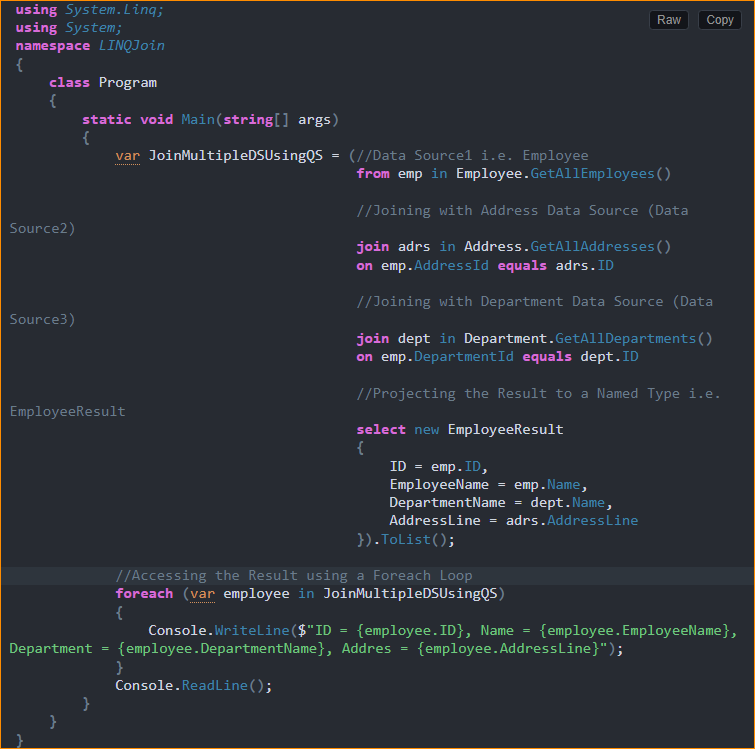
If you want to join the fourth data source then you need to write another join within the query. The complete example code is given below. You can access the query result using a for each loop which is shown in the below example. The following example code is self-explained, so please go through the comment lines.

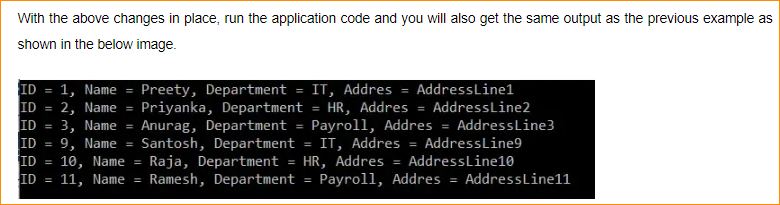


Instead of projecting the result to an anonymous type, can we project the result to a named type? Yes, it is possible. Let us see how we can do this. First, create a class file with the name EmployeeResult.cs with the required properties that you want in the result set. As per our requirement, we have created the class with the following four properties.



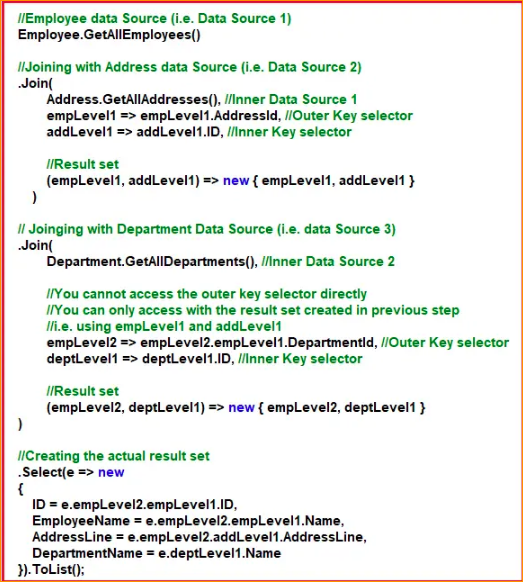
Next, modify the Main method of the Program class as follows. Here, you can see, we are projecting the result to the above-created EmployeeResult type.





1. LINQ Method Syntax to Perform Inner Join using Multiple Data Sources in C#:

Let’s see how to rewrite the previous example using LINQ Method Syntax to Join three data sources using C#. Using LINQ Method Syntax, it is a little difficult to write the join query when joining more than two data sources. The following code snippet shows how to write the query to fetch data from three data sources using the LINQ Method syntax.



Or this way

Instead of projecting the result to an anonymous type, can we project the result to a named type? Yes, it is possible. Let us see how we can do this. First, create a class file with the name EmployeeResult.cs with the required properties that you want in the result set. As per our requirement, we have created the class with the following four properties.

