* IEnumerable **is a collection of objects in memory that you can enumerate** - an in-memory sequence that makes it possible to iterate through (makes it way easy for within foreach loop, though you can go with IEnumerator only). They reside in the memory as is.
* IQueryable **is an expression tree** that will get translated into something else at some point **with ability to enumerate over the final outcome**. I guess this is what confuses most people.

IEnumerable interface is in the System.Collections namespace.

The IEnumerable object represents a set of data in memory and can move on this data only forward. The query represented by the IEnumerable object is executed immediately and completely, so the application receives data quickly.

When the query is executed, IEnumerable loads all the data, and if we need to filter it, the filtering itself is done on the client side.

IQueryable interface is located in the System.Linq namespace.

If you need the entire set of returned data, then it's better to use IEnumerable, which provides the maximum speed.

If you DO NOT need the entire set of returned data, but only some filtered data, then it's better to use IQueryable.

The IQueryable object provides remote access to the database and allows you to navigate through the data either in a direct order from beginning to end, or in the reverse order. In the process of creating a query, the returned object is IQueryable, the query is optimized. As a result, less memory is consumed during its execution, less network bandwidth, but at the same time it can be processed slightly more slowly than a query that returns an IEnumerable object.