LINQ Set Methods

Method	Usage
All	Determines whether all elements of a sequence satisfy a condition.
Any	Determines whether any element of a sequence satisfies a condition.
Concat	Concatenates (links together) two sequences to produce a result set, including duplicates.
Contains	Determines whether a sequence contains a specified element.
DefaultIfEmpty	Returns the elements of the specified sequence or the type parameter's default value in a singleton collection if the sequence is empty.
Distinct	Returns distinct elements from a sequence by using the default equality comparer to compare values. If you want to return distinct elements from sequences of objects of some custom data type, you have to implement the IEquatable<T> generic interface in the class.
Except	Returns a sequence that contains the set difference of the elements of two sequences.
Intersect	Returns a sequence that contains the elements that form the set intersection (same elements) of two sequences.
Union	Produces a result set that contains all elements from two sequences, excluding duplicates.

Source: MSDN

Other LINQ Operators that you may encounter (they can also be used with LINQ to Objects and LINQ to XML):

Method	Usage
Aggregate	Applies an accumulator function over a sequence.
GroupJoin	Correlates the elements of two sequences based on equality of keys and groups the results. The default equality comparer is used to compare keys.
Last	Returns the last element of a sequence.
LastOrDefault	Returns the last element of a sequence, or a default value if no element is found.
OrderByDescending	Sorts the elements of a sequence in descending order.
Range	Generates a sequence of integral numbers within a specified range.
Repeat	Generates a sequence that contains one repeated value.
Reverse	Inverts the order of the elements in a sequence.
SelectMany	Projects each element of a sequence to an IEnumerable<T> and flattens the resulting sequences into one sequence.
SequenceEqual	Determines whether two sequences are equal by comparing the elements by using the default equality comparer for their type.
ToArray	Creates an array from a IEnumerable<T>.
ToDictionary	Creates a Dictionary<TKey, TValue> from an IEnumerable<T>.
Zip	Applies a specified function to the corresponding elements of two sequences, producing a sequence of the results.

Source: MSDN