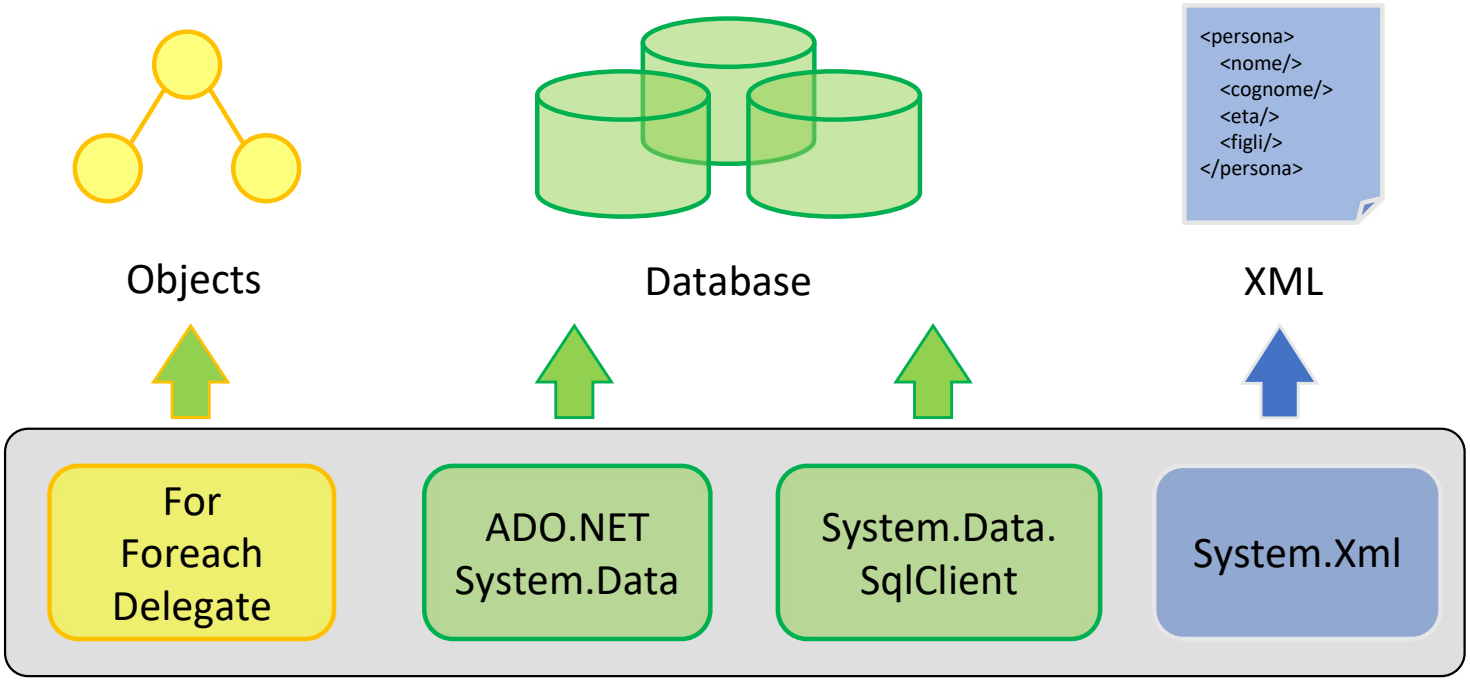
Cosa è LINQ

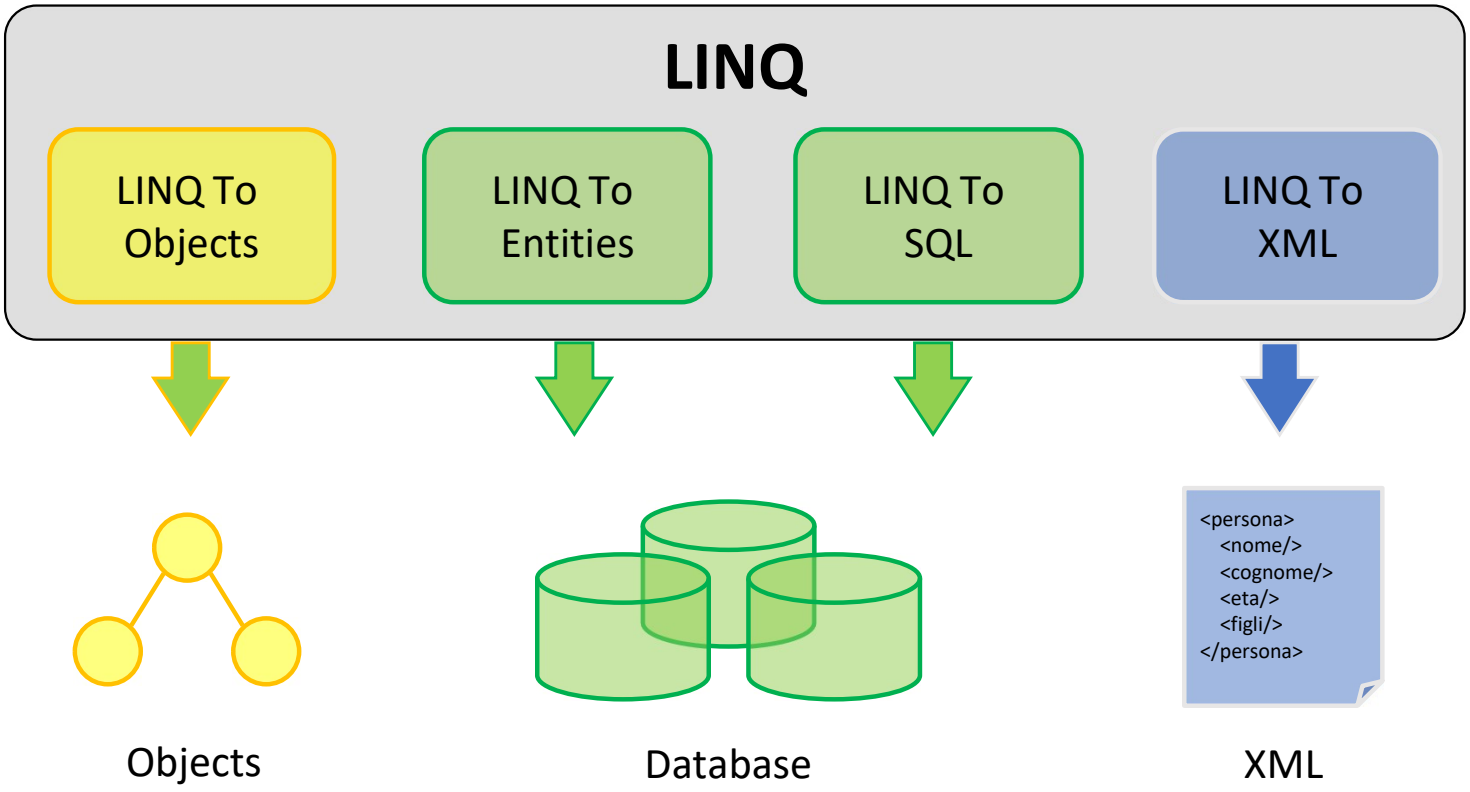
LINQ sta per **L**anguage **IN**tegrated **Q**uery

LINQ è un framework per eseguire interrogazioni su sorgenti dati all'interno del linguaggio.

Accesso ai dati senza LINQ



Accesso ai dati con LINQ



LINQ – Query Expression

Query standard per accedere a:

- Oggetti
- Dati relazionali
- Dati XML

Più di **50 operatori predefiniti**

- Aggregazione, Proiezione, Join, Partizionamento, Ordinamento

Sintassi e operatori **simile a SQL**

LINQ – Anatomia di una Query

- Due modelli di sintassi
  - Query
  - Lambda Expression
- Possibilità di utilizzare combinate
- Non modificano la sequenza originale

Query Lambda

- Più controllo e flessibilità
- Gli operatori sono applicati in sequenza
- ***Select*** può essere opzionale

LINQ – Operatori

- Utilizzo di **operatori Standard**
- Libreria di riferimento **System.Linq**
- Utilizzo con tipi **IEnumerable<T>**
- Pieno supporto ed integrazione con Intellisense

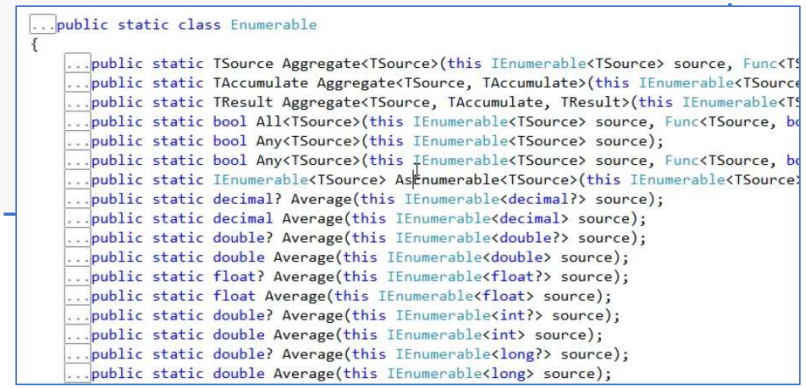
Operatori

Tipologia	Operatore
Projection	Select, SelectMany, (From)
Ricerca	Where
Ordinamento	OrderBy, OrderByDescending, Reverse, ThenBy, ThenByDescending
Raggruppamento	GroupBy
Aggregazione	Count, LongCount, Sum, Min, Max, Average, Aggregate,
Paginazione	Take, TakeWhile, Skip, SkipWhile
Insiemistica	Distinct, Union, Intersect, Except
Generazione	Range, Repeat, Empty
Condizionali	Any, All, Contains
Altri	Last, LastOrDefault, ElementAt, ElementAtOrDefault, First, FirstOrDefault, Single, SingleOrDefault, SequenceEqual, DefaultIfEmpty

LINQ - Operatori

- Reference: **System.Linq**
- Estende le funzionalità di **IEnumerable<T>** e **IQueryable<T>**

```
public static class Enumerable
{
  static public Ienumerable<Tsource> Where(this IEnumerable<TSource> source, Func<TSource, bool> predicate)
    …
    …
    …
}
```



LINQ

Sostituzione di **foreach** con query Linq

Definizione

```
IEnumerable<Employee> employee =
  from p in employees
  where p.Name == "Scott"
  select p.Name;
```

Esecuzione

```
foreach (var emp in employee)
{
  …
  …
}
```

LINQ – Lambda Expression

```
IEnumerable<string> filteredList = cities.Where(StartsWithL);


public bool StartsWithL(string name)
{
  return name.StartWith("L");
}
```

```
  IEnumerable<string> filteredList = cities.Where(name => name.StartWith("L"));
```

LINQ – Lambda Expression

- Rappresentazione sintetica
- (input-parameters) => expression

- Utilizzo dell'operatore **=>**
  - **A sinistra:** firma della funzione
  - **A destra:** statement della funzioni

LINQ – Lambda Expression

Parametri ed i tipi opzionali

- Non sono richieste parametri, quando sono impliciti

Logica negli statement

- Utilizzo di variabili locali
- Attenzione: le lambda expression dovrebbero essere tenute più semplici possibile

```
IEnumerable<string> filteredList =
cities.Where((string s) =>
  {
    string temp = s.ToLower();
    return temp.StartsWith("L");
  }
);
```

LINQ – Lambda Expression

Lambda Expression usano particolari **delegate:**

- **Action<T>**
  - Non ritornano un valore
- **Func<T>** e **Expression<T>**
  - Ritornano un valore

```
Func<int, int> square = x => x * x;
Func<int, int, int> mult = (x, y) => x * y;
Action<int> print = x => Console.WriteLine(x);
print(square(mult(3, 5)));
```

LINQ – Query Expression

- **Extension Methods**
- **Lambda expressions**
  - Delegati
  - Expression Trees
- **Query Expression**

LINQ – Query Expression

- Extension Methods
- Lambda expressions
- **Query Expression**

```
IEnumerable<string> filterCities =
  from city in cities
  where city.StartsWith("L") && city.Lenght <15
  orderby city
  select city;
```

Demo

LinQ – Lambda Expression VS Query Expression

Esercitazione 5

Riprendere l'esercizio precedente:

- Creare una List di almeno 10 Shape
- Scrivere le query LINQ per
  - Elencare tutte le Shape con un'Area superiore a 20
  - Elencare tutte le Shape con il Nome che inizi per 'A'
  - Elencate solo i Nomi delle Shape
  - Elencare tutte le Shape in ordine Alfabetico per Nome e poi per Area decrescente

Le query devono essere scritte in entrambe le sintassi (Extension Methods e Query Expression)