LANGUAGE-INTEGRATED QUERY (**LINQ**)

Seminari - Desenvolupament de Programari en Visual Studio 2019

Objectius

- Aprendre les característiques fonamentals del llenguatge LINQ per a accés a dades de diferents fonts en la plataforma .NET (C#)

Language-Integrated Query (LINQ)

- **LINQ** és un conjunt de funcionalitats del framework .NET de Microsoft que estén la sintaxi dels llenguatges de programació acceptats (i.g. C# o Visual Basic) amb expressions o patrons que faciliten la recuperació d'informació de fonts de dades diferents (i.g. base de dades, col·leccions, o objectes ADO)

- Les expressions o patrons acceptats es pareixen a les funcions i expressions lambda de llenguatges de programació funcionals como Haskell, OCaML, o F#.
  - Hui en dia, les expressions lambda estan presents en la majoria de llenguatges de programació, inclòs Java des de la versió 8, i són molt comunes en llenguatges basats en esdeveniments com Apple Swift o Google Kotlin.

De LINQ a SQL

- Anem a utilitzar **LINQ** per a accedir de forma senzilla a bases de dades relacionals.

- Les expressions **LINQ** es tradueixen internament en expressions **SQL.**

- Permeten oblidar-nos dels detalls interns de cada arquitectura de base de dades i deixar els detalls de connexió , accés i consulta al sistema de traducció de **LINQ** a **SQL**.

Sintaxis de LINQ

- Una expressió **LINQ** consta de tres parts diferenciades:

1. Obtenir la font de dades

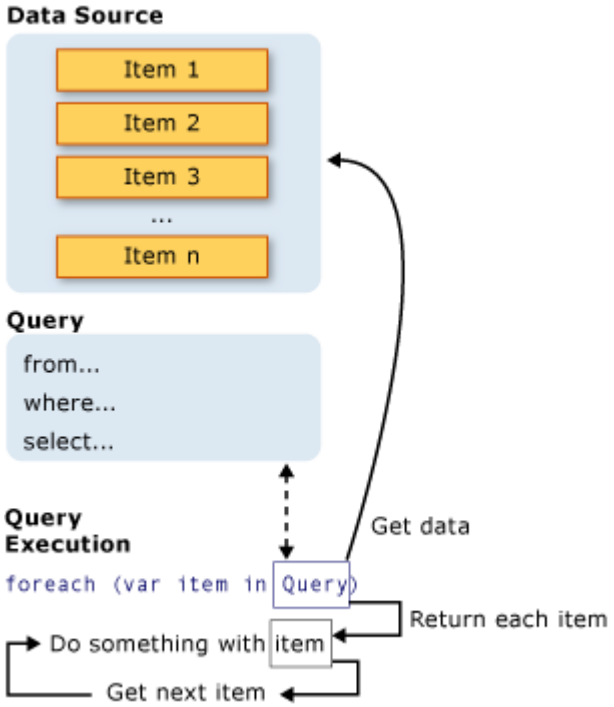
```
var bos = dbcontext.offices
```

2. Crear la consulta

```
var offices_query =
    from office in bos
    where office.Id == “Valencia”
    select office;
```

3. Executar la consulta

```
foreach (BranchOffice office in offices_query)
    Console.WriteLine(office.Id,office.address);
```



Lambda expressions en LINQ

- **LINQ** permet una forma abreviada per a introduir expressions, coneguda com *method-based*. Esta forma utilitza expressions *lambda* com els llenguatges funcionals.
- Els dos troços de codi (1 i 2) tornen lo mateix.
- Codi 1:

```
var offices_query =
    from office in dbcontext.offices
    where office.Id == “Valencia”
    select office;
foreach (BranchOffice office in offices_query)
    Console.WriteLine(office.Id,office.address);
```

- Codi 2:

```
var offices_query = dbcontext.offices.Where(office => office.Id == “Valencia”)
foreach (BranchOffice office in offices_query)
    Console.WriteLine(office.Id,office.address);
```

Where en LINQ

- El mètode ***Where*** en una expressió **LINQ** torna un objecte de tipus *IEnumerable*.
- La interfície *IEnumerable* té molts altres mètodes que poden resultar útils:
  - ...
  - *OrderBy,*
  - *Distinct,*
  - *FirstOrDefault,*
  - etc.

LINQ en C#

- LINQ està integrat en diferents funcionals de C#:
  - Consultes, com s'ha mostrat anteriorment
  - Variables sense tipus declarat.

```
var number = 5;
var name = "Virginia";
var query = from str in stringArray
            where str[0] == 'm'
            select str;
```

  - Inicializadores d'objectes i col·leccions.

```
Customer cust = new Customer { Name = "Mike", Phone = "555-1212" };
```

  - Tipus anònims

```
select new {name = cust.Name, phone = cust.Phone};
```

  - Extensió de mètodes
  - Expresions Lambda
  - Propietats auto implementades

```
public string Name {get; set;}
```

Conclusions

- Funcionalitats de llenguatges de programació funcionals integrades en llenguatges de programació, incloent Microsoft .NET i Java

- Diverses extensions i aplicacions: SQL, objectes ADO, interfícies gràfiques, etc.