

**LINQ operators : Get, transform,
order, paginate, quantify
and aggregate elements**





Agenda

Projecting and filtering

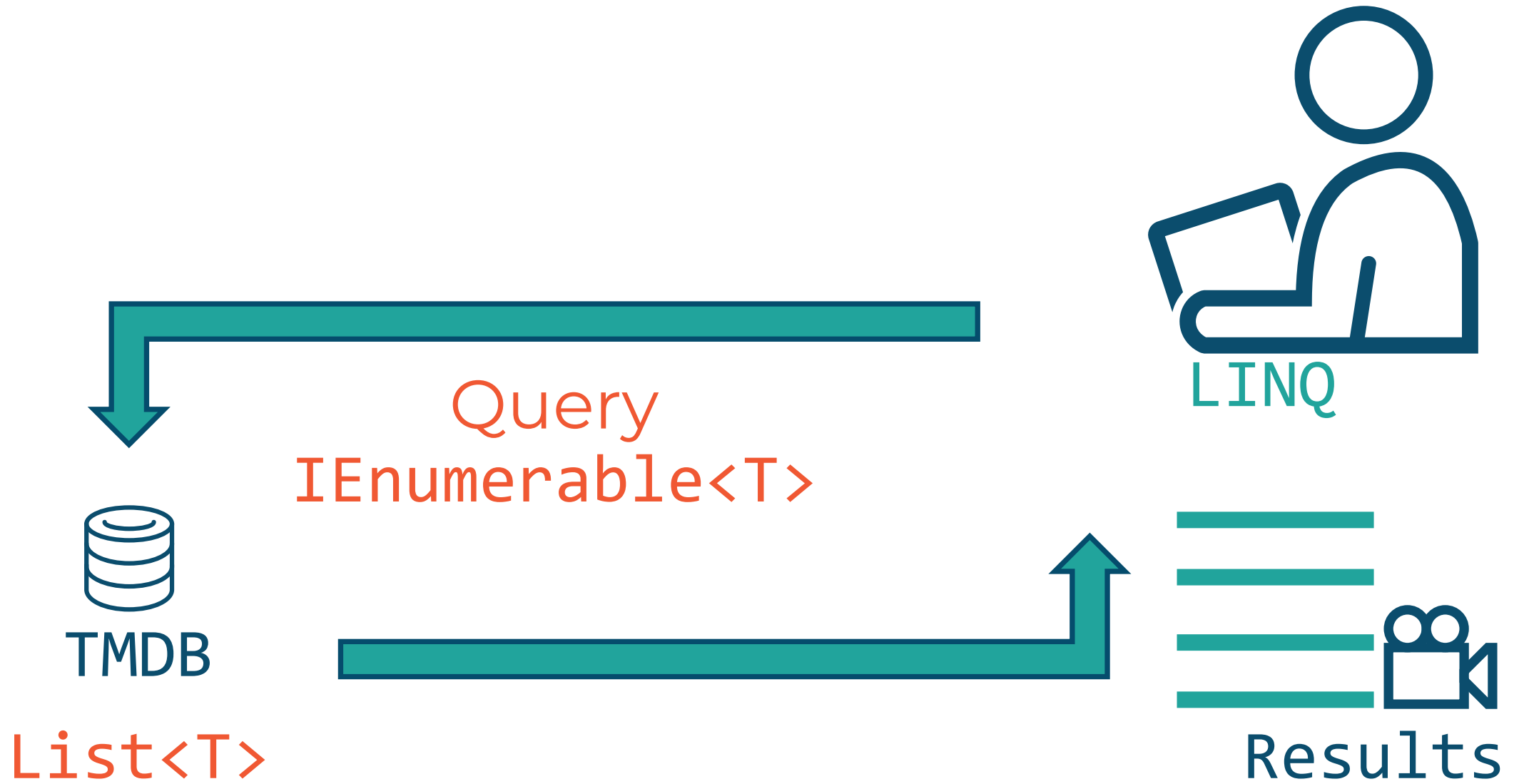
Ordering

Get distinct elements/values

Paging

Quantifying

Aggregating



Projecting and filtering elements



LINQ Operators

Projection

Projection transforms an object into a new form (type or anonymous type)

Select operator projects the elements of a sequence

Filtering

Where `operator` defines a filter condition

Demo

Projecting and filtering sequences

Projection

Get all the movies from the movie data list

Get all the movies and project on a new type

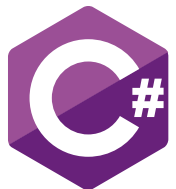
Get all the movies and project on an anonymous type

Filtering

Get all the movies with more than 25.000 voters

Get all the movies with more than 1.000.000.000 \$
revenue and more than 100 popularity

Ordering elements



LINQ Operators

Ordering

`OrderBy` orders the elements of a sequence
`OrderByDescending` orders the elements in a descending manner

`ThenBy` orders the elements of a sequence subsequently

`ThenByDescending` orders the elements of a sequence subsequently in a descending manner

Demo

Ordering sequences

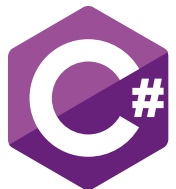
Ordering

Order movies by revenue in ascending manner

Order movies by revenue in descending manner

Order movies by number of stars descending and revenue

Get distinct elements



LINQ Operators

Distinct

`Distinct` returns the distinct elements from a sequence

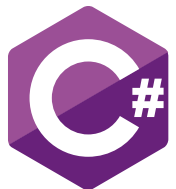
Demo

Get distinct elements using
IEqualityComparer implementation
Get distinct values

Get distinct movies (with different ids)

Get distinct popularity ordered descending

Get single element



LINQ Operators

Single

`Single` returns a specific element from a sequence

If the source sequence is not `null`

`SingleOrDefault` throws an exception if more than one element than matches the predicate parameter

First

`First` returns the first element of a sequence

If the source sequence is not `null`

`FirstOrDefault` does not throw an exception

Last

Last returns the last element of a sequence
If the source sequence is not null
LastOrDefault does not throw an exception

Demo

Get single elements using Single and First

Get single element

Get one and only one movie by id

Get first movie with more than 100 popularity

Paginate elements



LINQ Operators

Take

Take returns a specified number of elements from the start of a sequence

Skip

Skip bypasses a specified number of elements and returns the following elements

Demo

Paginate the result list with Skip and Take

Get all the movies and paginate the results

Quantifier operators



LINQ Operators

Quantifier operators

Return a Boolean indicating if one or all the elements of a sequence satisfy a condition

Operators

All determines if all the elements of a sequence satisfy a condition

Any determines if any element of a sequence satisfy a condition

Contains determines if a sequence contains a specified element

Demo

Quantifying elements of a sequence

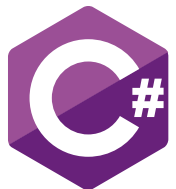
Quantifying

Checks if all the movies have a non null or empty overview

Checks if any movie has 9+ popularity

Checks if movies contain a movie

Aggregate elements



LINQ Operators

Aggregation operators

Applies mathematical operations on values
in your data list

Operators

Average calculates the average value on a sequence of values

Count determines the number of elements of a sequence satisfy a condition

Max determines the maximum value in a sequence of values

Sum determines the sum of values in a sequence of values

Demo

Determine the sum of the elements of a sequence

Quantifying

Get the revenue sum for all the movies



Summary

The Select operator projects the elements of a sequence

The Where operator filters the elements of a sequence

The OrderBy (ThenBy...) operator orders the elements of a sequence

The Single, First and Last operators return a single element of a sequence

The Take and Skip operators allow to implement paging on a sequence

The All, Any and Contains operator return a boolean indicating if one or all the elements of a sequence satisfy a condition

The aggregation apply mathematical operations on values of a sequence