

Aggregation Lab

Lab 1: Count() Method

Open the **Program.cs** file and replace the contents with the following:

```
using LINQLab.EntityClasses;
using LINQLab.RepositoryClasses;

// Declare variables and fill data
List<Song> songs = SongRepository.GetAll();
int? value = 0;

// TODO: Write Your Query Here

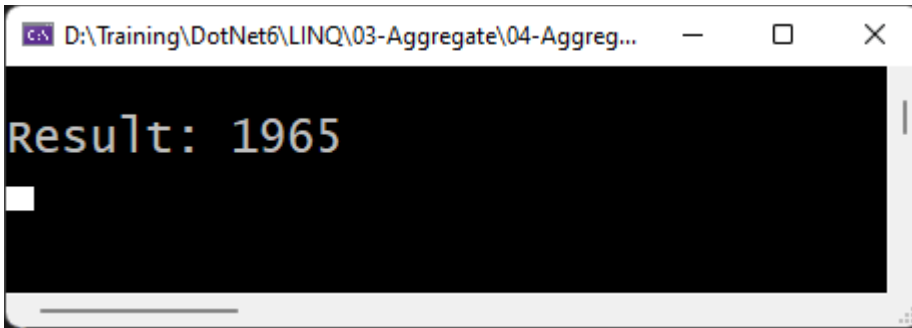
// Display Result
Console.WriteLine();
Console.WriteLine($"Result: {value}");

// Pause for Results
Console.ReadKey();
```

Write a query to count all the rows where the **GenreId** is equal to 35.

Try it Out

Run the application and the results should look like the following:

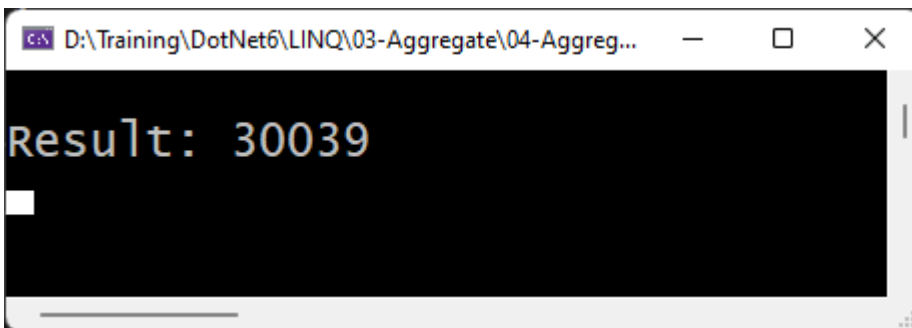


Lab 2: Sum() Method

Write a query that sums the value in the **Plays** property where the **GenreId** is equal to 35.

Try it Out

Run the application and the results should look like the following:



Lab 3: Average() Method

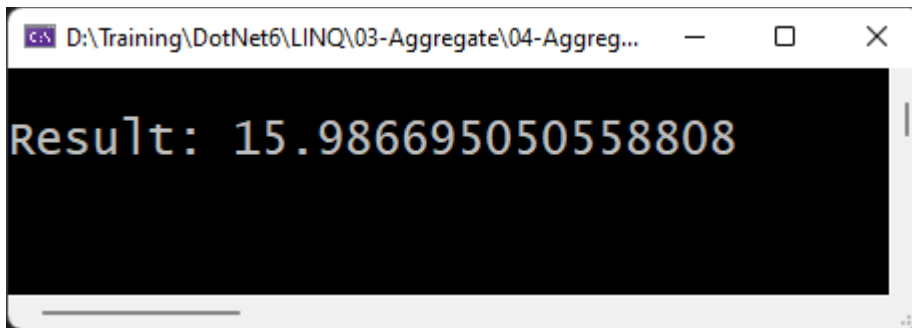
Write a query to calculate the average number of **Plays** where the **GenreId** is equal to 35.

Modify the **value** variable to be a **double?** instead of an **int?**.

```
double? value;
```

Try it Out

Run the application and the results should look like the following:



Lab 4: Min() Method

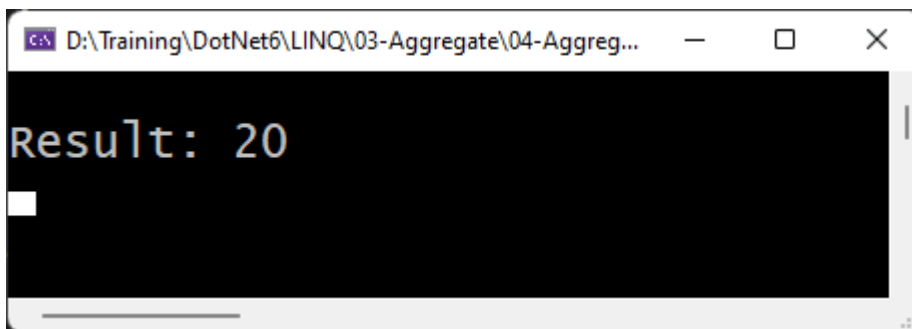
Write a query to retrieve the minimum **Rating** of **all** songs.

Modify the **value** variable to be a **int?** instead of an **double?**.

```
int? value;
```

Try it Out

Run the application and the results should look like the following:



Lab 5: MinBy() Method

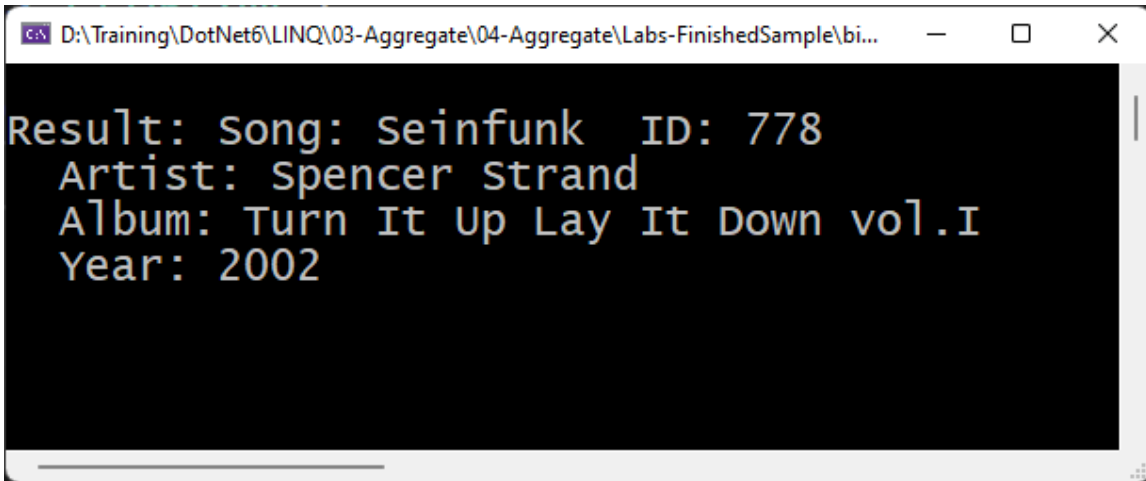
Write a query to retrieve the song that has minimum **Rating** of all songs.

Modify the **value** variable to be a **Song?** instead of a **int?**.

```
Song? value;
```

Try it Out

Run the application and the results should look like the following:



Lab 6: Aggregate() Method

Write a query to select all rows and order the songs by the **DateAdded** property.

Take the first 5 songs.

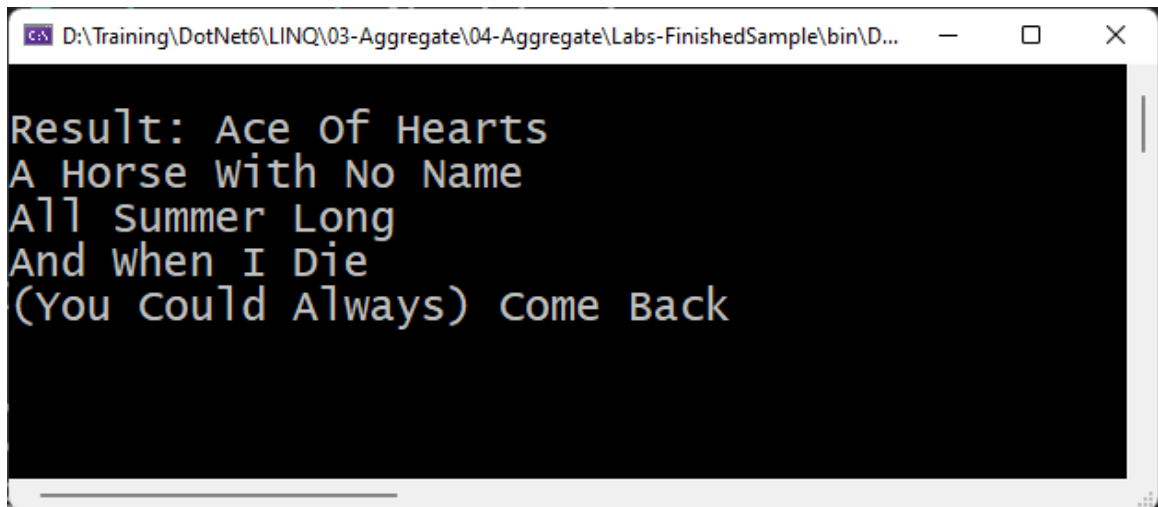
Add an `Aggregate()` method to concatenate the song names together separated by `Environment.NewLine`.

Modify the **value** variable to be a **string** instead of a **Song?**.

```
string value;
```

Try it Out

Run the application and the results should look like the following:



```
D:\Training\DotNet6\LINQ\03-Aggregate\04-Aggregate\Labs-FinishedSample\bin\D...  
Result: Ace Of Hearts  
A Horse With No Name  
All Summer Long  
And When I Die  
(You Could Always) Come Back
```