## Assignment #2.4 – Chinook

Run a fluent dynamic query on the **Chinook.db** SQLite database.

The query comes from **stdin**, in the following format (sample):

```
Artists
OrderBy Name DESC
Where ArtistId % 10 == 0
Take 3
Select new (ArtistId, Name)
```

The first line contains the starting table.

Each of the next line contains the name of a LINQ operator, followed by a lambda given in the string format expected by dynamic LINQ.

The project will translate the above input into the following dynamic LINQ code:

```
var seq2 = db.Artists
   .OrderBy ("Name DESC")
   .Where ("ArtistId % 10 == 0")
   .Take (3)
   .Select ("new (ArtistId, Name)");
```

FYI, this corresponds to the following standard LINQ code:

```
var seq2 = db.Artists
   .OrderByDescending (id => Name)
   .Where (id => id.ArtistId % 10 == 0)
   .Take (3)
   .Select (id => new {id.ArtistId, id.Name};
```

The dynamic LINQ result, which here is a dynamic queryable sequence, will be further transformed and printed to **stdout** in JSON format:

```
var res = JsonSerializer.Serialize (seq2 .AsEnumerable () .ToList ());
```

Pretty printed result:

```
[
   {"ArtistId":150,"Name":"U2"},
   {"ArtistId":70,"Name":"Toquinho \u0026 Vin\u00EDcius"},
   {"ArtistId":200,"Name":"The Posies"}
]
```

**Submission**: to **automarker**, one single C# source file, containing your code and the chinook EF scaffold (cf. the skeleton)

**Additional packages**:

- Microsoft.EntityFrameworkCore.Sqlite
- System.Linq.Dynamic.Core

**Skelton folder** contains:

- a C# project file, including the additional packages
- a C# source file, including the scaffold
- a copy of **chinook.db**
- sample requests and responses
- bat and sh script files

**Tables**: ….

**Dynamic LINQ functions**: Select, Where, OrderBy, Skip, Take