# Exercises: LINQ

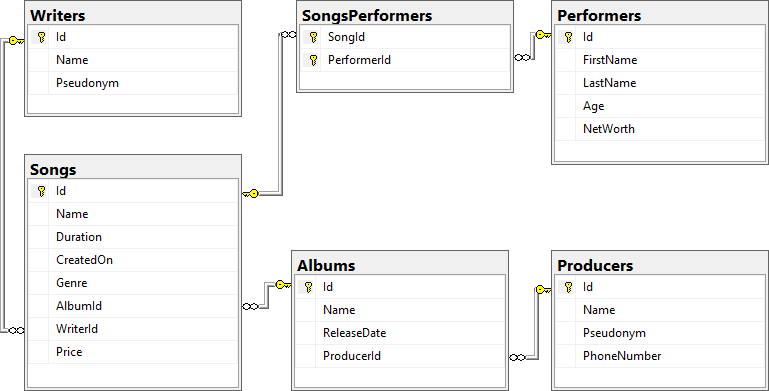
This document defines the **exercise assignments** for the [Databases Advanced - Entity Framework course @ SoftUni](https://softuni.bg/trainings/3966/entity-framework-core-february-2023)  
You can check your solutions in [Judge](https://judge.softuni.org/Contests/3919/LINQ)

# MusicHub

People love listening to music, but they see that YouTube is getting older and older. You want to make people happy and you've decided to make a better version of YouTube – **MusicHub**. It's time for you to start coding. Good luck and impress us.

## MusicHub Database

You must create a **database** for a **MusicHub**. It should look like this:



### Constraints

Your **namespaces** should be:

* MusicHub – for your **StartUp** class, if you have one
* MusicHub.Data – for your **DbContext**
* MusicHub.Data.Models – for your **Models**

Your **models** should be:

#### Song

* **Id** – **integer**, **Primary Key**
* **Name** – **text** with **max length 20** (**required**)
* **Duration** – **TimeSpan** (**required**)
* **CreatedOn** – **date** (**required**)
* **Genre** ­– genre **enumeration** with possible values**:** **"****Blues, Rap, PopMusic, Rock, Jazz" (required)**
* **AlbumId** – **integer**, **Foreign key**
* **Album** –the **Song**'s **Album**
* **WriterId** – **integer, Foreign key (required)**
* **Writer** –the **Song**'s **Writer**
* **Price** – **decimal** (**required**)
* **SongPerformers** –a **c**ollection of type **SongPerformer**

#### Album

* **Id** – **integer**, **Primary Key**
* **Name** – **text** with **max length 40** (**required**)
* **ReleaseDate** – **date** (**required**)
* **Price** – **calculated property** (the sum of all song prices in the album)
* **ProducerId** – **integer**, **foreign key**
* **Producer** – **the Album's Producer**
* **Songs** – a collection of all **Songs** in the **Album**

#### Performer

* **Id** – **integer**, **Primary Key**
* **FirstName** – **text** with **max length 20** (**required)**
* **LastName** – **text** with **max length 20** (**required)**
* **Age** – **integer** (**required**)
* **NetWorth** **–** **decimal** (**required**)
* **PerformerSongs** – a collection of type **SongPerformer**

#### Producer

* **Id** – **integer**, **Primary Key**
* **Name** – **text** with **max length 30** **(**required**)**
* **Pseudonym** – **text**
* **PhoneNumber** – **text**
* **Albums** – a collection of type **Album**

#### Writer

* **Id** – **integer**, **Primary Key**
* **Name** – **text** with **max length 20** (required**)**
* **Pseudonym** – **text**
* **Songs** – a collection of type **Song**

#### SongPerformer

* **SongId** – **integer**, **Primary Key**
* **Song** – the performer's **Song** (**required**)
* **PerformerId** – **integer**, **Primary Key**
* **Performer** – the **Song**'s **Performer (required)**

### Table relations

* **One Song** can have **many Performers**
* **One Permormer** canhave **many Songs**
* **One Writer** can have **many Songs**
* **One Album** can have **many Songs**
* **One Producer** can have **many Albums**

**NOTE:** You will need a constructor, accepting **DbContextOptions** to test your solution in **Judge**!

## All Albums Produced by Given Producer

You need to write method **string ExportAlbumsInfo(MusicHubDbContext context, int producerId)** in the **StartUp** class that receives a **ProducerId**. Export **all albums** which are **produced by** the provided **ProducerId**. For each **Album**, get the **Name**, **ReleaseDate** in format the "**MM/dd/yyyy**", **ProducerName**, the **Album Songs** with each **Song Name**, **Price** (**formatted to the second digit**) and the **Song WriterName**. **Sort** the **Songs** by **Song** **Name** (**descending**) and by **Writer** (**ascending**). At the end export **the Total Album Price** with exactly **two digits after the decimal place**. **Sort** the **Albums** by their **Total** **Price** (**descending**).

### Example

|  |
| --- |
| **Output (producerId = 9)** |
| -AlbumName: Devil's advocate  -ReleaseDate: 07/21/2018  -ProducerName: Evgeni Dimitrov  -Songs:  ---#1  ---SongName: Numb  ---Price: 13.99  ---Writer: Kara-lynn Sharpous  ---#2  ---SongName: Ibuprofen  ---Price: 26.50  ---Writer: Stanford Daykin  -AlbumPrice: 40.49  … |

## Songs Above Given Duration

You need to write method **string ExportSongsAboveDuration(MusicHubDbContext context, int duration)** in the **StartUp** class that receives **Song** duration(**integer, in seconds**). Export the songs which are **above** the given duration. For each **Song**, export its **Name**, **Performer Full Name**, **Writer Name**, **Album** **Producer** and **Duration** (**in format**("**c**")). **Sort** the **Songs** by their **Name** (**ascending**), and then by **Writer** (**ascending**). If a **Song** has more than one **Performer**, export all of them and sort them (**ascending**). If there are no **Performers** for a given song, don't print the "**---Performer**" line at all.

### Example

|  |
| --- |
| **Output (duration = 4)** |
| -Song #1  ---SongName: Away  ---Writer: Norina Renihan  ---Performer: Lula Zuan  ---AlbumProducer: Georgi Milkov  ---Duration: 00:05:35  -Song #2  ---SongName: Bentasil  ---Writer: Mik Jonathan  ---Performer: Zabrina Amor  ---AlbumProducer: Dobromir Slavchev  ---Duration: 00:04:03 … |