# Exercises: Advanced Querying

This document defines the **exercise assignments** for the ["Databases Advanced – EF Core" course @ Software University.](https://softuni.bg/trainings/3492/entity-framework-core-october-2021)

# BookShop System

For the following tasks, use the [BookShop](http://svn.softuni.org/admin/svn/csharp-databases/2019-Jan/2.%20DB-Advanced-EF-Core/07.%20DB-Advanced-Advanced-Querying/BookShop.zip) database. You can download the complete project or create it yourself in **task 0**,but you should still use the pre-defined Seed() method from the project to have the same **sample** data.

## Book Shop Database

You must create a **database** for a **book** **shop** **system**. It should look like this:



### Constraints

Your **namespaces** should be:

* BookShop – for your **StartUp** class
* BookShop.Data – for your DbContext
* BookShop.Models – for your models
* BookShop.Models.**Enums** – for your models

Your **models** should be:

* BookShopContext – your DbContext
* Author:
  + AuthorId
  + FirstName (up to 50 characters, unicode, not required)
  + LastName (up to 50 characters, unicode)
* Book:
  + BookId
  + Title (up to 50 characters, unicode)
  + Description (up to 1000 characters, unicode)
  + ReleaseDate (not required)
  + Copies (an integer)
  + Price
  + EditionType – enum (Normal, Promo, Gold)
  + AgeRestriction – enum (Minor, Teen, Adult)
  + Author
  + BookCategories
* Category:
  + CategoryId
  + Name (up to 50 characters, unicode)
  + CategoryBooks
* BookCategory – mapping class

For the following tasks, you will be creating methods that accept a BookShopContext as a parameter and use it to run some queries. Create those methods inside your **StartUp** class and upload your whole solution to **Judge**.

## Age Restriction

**NOTE**: You will need method public static string **GetBooksByAgeRestriction**(BookShopContext context, string command) and public StartUp class.

Return in a **single** **string**allbook **titles**, each on a **new line,**that have**an age** **restriction**, equal to the **given** **command**. Order the titles **alphabetically**.

Read **input** from the console in your **main** **method**, and call your **method** with the **necessary** **arguments**. Print the **returned** **string** to the console. **Ignore** the casing of the input.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| miNor | A Confederacy of Dunces  A Farewell to Arms  A Handful of Dust  … |
| teEN | A Passage to India  A Scanner Darkly  A Swiftly Tilting Planet  … |

## Golden Books

**NOTE**: You will need method public static string **GetGoldenBooks**(BookShopContext context) and public StartUp class.

Return in a **single** string **title of the golden edition books** that have **less than 5000 copies**,each on a **new line**. Order them by **book** **id** ascending.

Call the **GetGoldenBooks**(BookShopContext context) method in your **Main()** and print the returned string to the console.

### Example

|  |
| --- |
| **Output** |
| Lilies of the Field  Look Homeward  The Mirror Crack'd from Side to Side  … |

## Books by Price

**NOTE**: You will need method public static string **GetBooksByPrice**(BookShopContext context) and public StartUp class.

Return in a single string all **titles and prices** **of books** with a **price higher than 40**, each on a **new** **row** in the **format** given below. Order them by **price** descending.

### Example

|  |
| --- |
| **Output** |
| O Pioneers! - $49.90  That Hideous Strength - $48.63  A Handful of Dust - $48.63  … |

## Not Released In

**NOTE**: You will need method public static string **GetBooksNotReleasedIn**(BookShopContext context, int year) and public StartUp class.

Return in a **single** string all **titles of books** that are **NOT released** in a given year. Order them by **book** **id** ascending.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2000 | Absalom  Nectar in a Sieve  Nine Coaches Waiting  … |
| 1998 | The Needle's Eye  No Country for Old Men  No Highway  … |

## Book Titles by Category

**NOTE**: You will need method public static string **GetBooksByCategory**(BookShopContext context, string input) and public StartUp class.

Returnin a single string the **titles of books** by a given **list of categories**. The list of **categories** will be given in a single line separated by one or more spaces. Ignore casing. Order by **title** alphabetically.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| horror mystery drama | A Fanatic Heart  A Farewell to Arms  A Glass of Blessings  … |

## Released Before Date

**NOTE**: You will need method public static string **GetBooksReleasedBefore**(BookShopContext context, string date) and public StartUp class.

Return **the title, edition type, and price of all books that are released before a given date. The date will be a string in the format dd-MM-yyyy.**

**Return all of the rows in a single string, ordered by release date descending.**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 12-04-1992 | If I Forget Thee Jerusalem - Gold - $33.21  Oh! To be in England - Normal - $46.67  The Monkey's Raincoat - Normal - $46.93  … |
| 30-12-1989 | A Fanatic Heart - Normal - $9.41  The Curious Incident of the Dog in the Night-Time - Normal - $23.41  The Other Side of Silence - Gold - $46.26  … |

## Author Search

**NOTE**: You will need method public static string **GetAuthorNamesEndingIn**(BookShopContext context, string input) and public StartUp class.

Return the **full** **names** of **authors**, whose **first** **name** ends with a **given** **string**.

Return all **names** in a **single** **string**, each on a **new** **row** ordered alphabetically.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| e | George Powell  Jane Ortiz |
| dy | Randy Morales |

## Book Search

**NOTE**: You will need method public static string **GetBookTitlesContaining**(BookShopContext context, string input) and public StartUp class.

Return the **titles** of **the book**, which contain a **given** **string**. Ignore casing.

Return all **titles** in a **single** **string**, each on a **new** **row** ordered alphabetically.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| sK | A Catskill Eagle  The Daffodil Sky  The Skull Beneath the Skin |
| WOR | Great Work of Time  Terrible Swift Sword |