# Exercises: ADO.NET

This document defines the **exercise** assignments for the [Databases Advanced - Entity Framework course @ SoftUni](https://softuni.bg/trainings/3966/entity-framework-core-february-2023)

**NOTE:** The example outputs should be the same as yours as long as you execute the queries one by one, following the order of the tasks. If you don't follow the order of the tasks or you execute one or more queries more than once, or you drop the database and start not from the beginning of the problems, you output will be different from the ones in this document.

**NOTE:** You will need the"**MinionsDB.sql**"file.The queries for creating the database and the tables, and inserting info in those tables, are already written in the file. You can use those queries or you can write your own, as long as they follow the objectives of the tasks. You should execute the queries step by step, as described in the"**MinionsDB.sql**"file.

## Initial Setup

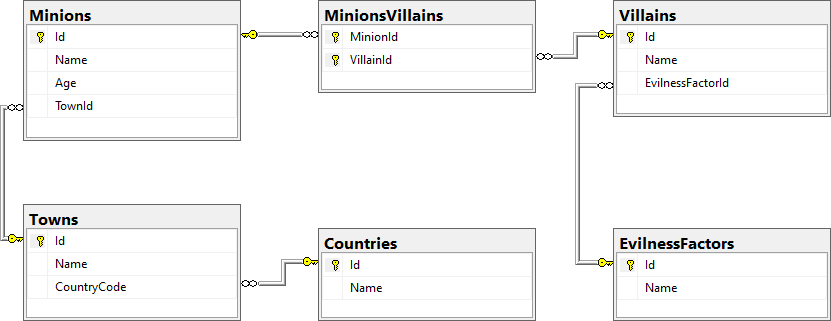
Write a program that connects to your **localhost** server. Create **a new database**called **MinionsDB** where we will keep information about our minions and villains.

For each **minion**, we should keep information about its **name**, **age** and **town**.

Each **town** has information about **the** **country**, where it is located. **Villains** have **name** and **evilness** **factors** (**super good**, **good**, **bad**, **evil**, **super** **evil**). Each **minion** can **serve** **several** **villains** and **each** **villain** can **have** **several** **minions** **serving** **him**. Fill all tables with at least 5 records each.

In the end, you should have the following tables:

* **Countries**
* **Towns**
* **Minions**
* **EvilnessFactors**
* **Villains**
* **MinionsVillains**



**NOTE:** You will need the"**MinionsDB.sql**"file.The queries for creating the database and the tables, and inserting info in those tables, are already written. You can use those queries or you can write your own, as long as they follow the objectives of the tasks.

## Villain Names

Write a program that prints on the console **all villains' names** and their **number of minions** of those who have more than 3 minions **ordered descending** by the **number of minions**.

### Example

|  |
| --- |
| **Output** |
| Jilly – 4 |

## Minion Names

Write a program that prints on the console **all minion names** and ages for a given **villain id**, ordered by **name** **alphabetically.**

If there is no villain with the given ID, print "**No villain with ID <VillainId> exists in the database.".**  
If the selected villain has no minions, print "**(no minions)**" on the second row.

### Example

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 1 | Villain: Gru  1. Becky 125  2. Bob 42  3. Kevin 1 |  | 2 | Villain: Victor  1. Bob 42  2. Simon 45 |  | 6 | Villain: Dimityr  (no minions) |

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8 | No villain with ID 8 exists in the database. |

## Add Minion

Write a program that **reads** **information** about a minion and its villain and **adds it to the database**. In case the town of the minion is not in the database, **insert** it as well. In case the villain is not present in the database, add him too with a default **evilness** **factor** of "evil". Finally set the new minion to be a servant of the villain. Print appropriate messages after each operation.

### Input

The input comes in two lines:

* On the first line, you will receive the **minion** **information** in the format "**Minion: <Name> <Age> <TownName>**"
* On the second – the **villain** **information** in the format "**Villain: <Name>**"

### Output

After completing an operation, you must print one of the following messages:

* After adding a new **town** to the database: "**Town <TownName> was added to the database.**"
* After adding a new **villain** to the database: "**Villain <VillainName> was added to the database.**"
* Finally, after successfully adding the **minion** to the database and making it a **servant** of a villain: "**Successfully added <MinionName> to be minion of <VillainName>.**"

**\*Bonus task:** Make sure all operations are executed successfully. In case of an error do not change the database.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| Minion: Bob 14 Berlin  Villain: Gru | Successfully added Robert to be minion of Gru. |
| Minion: Cathleen 20 Liverpool  Villain: Gru | Successfully added Cathleen to be minion of Gru. |
| Minion: Mars 23 Sofia  Villain: Poppy | Villain Poppy was added to the database.  Successfully added Mars to be minion of Poppy. |
| Minion: Carry 20 Eindhoven  Villain: Jimmy | Town Eindhoven was added to the database.  Villain Jimmy was added to the database.  Successfully added Carry to be minion of Jimmy. |

## Change Town Names Casing

Write a program that **changes all town names to uppercase** for a given country.

You will receive one line of input with the **name** of the country.

**Print the** **number of towns that were changed** in the format **"<ChangedTownsCount> town names were affected."**. On a second line, **print** the **names that were changed**, separated by a comma and a space.

If **no** **towns** were affected (the country does not exist in the database or has no cities connected to it), **print** "**No town names were affected.**".

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| Bulgaria | 4 town names were affected.  [PLOVDIV, VARNA, BURGAS, SOFIA] |
| Germany | No town names were affected. |

## \*Remove Villain

Write a program that receives the **ID** of a villain, **deletes him from the database** and **releases his minions** from serving to him. Print on **two** **lines** the name of the deleted villain in the format "<**Name**> **was** **deleted**." and the number of minions released in format "<**MinionCount**> **minions** **were** **released**.". Make sure all operations go as planned, otherwise do not make any changes in the database.

If there is no villain in the database with the given ID, print "**No such villain was found**.".

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | Gru was deleted.  5 minions were released. |
| 3 | Jilly was deleted.  4 minions were released. |
| 6 | Dimityr was deleted.  0 minions were released. |
| 101 | No such villain was found. |

## Print All Minion Names

Write a program that **prints all minion names** from the minions table **in the following order:** first record, last record, first + 1, last - 1, first + 2, last - 2 … first + n, last - n.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 15 | 2 | 14 | 3 | 13 | 4 | 12 | 5 | 11 | 6 | 10 | 7 | 9 | 8 |

### Example

|  |  |
| --- | --- |
| **Original Order** | **Output** |
| Bob  Kevin  Bob  Simon  Cathleen  Carry  Becky  Mars  Misho  Zoe  Json  Bob  Cathleen  Mars  Carry | Bob  Carry  Kevin  Mars  Bob  Cathleen  Simon  Bob  Cathleen  Json  Carry  Zoe  Becky  Misho  Mars |

## Increase Minion Age

Read from the console minion IDs separated by space.**Increment the age** of those minions **by 1** and make the first letter of their name**lower case**. Finally, **print the name and the age of all minions** in the database, each on a new row in the format **"<Name> <Age>**".

### Example

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 2 5 4 | Bob 42  kevin 2  Bob 32  simon 46  cathleen 12  Carry 50  Becky 125  Mars 21  Misho 5  Zoe 125  Json 21  Bob 14  Cathleen 20  Mars 23  Carry 20 |  | 1 | bob 43  kevin 2  Bob 32  simon 46  cathleen 12  Carry 50  Becky 125  Mars 21  Misho 5  Zoe 125  Json 21  Bob 14  Cathleen 20  Mars 23  Carry 20 |

## Increase Age Stored Procedure

Create stored procedure **usp\_GetOlder** (**directly in the database** using **Management Studio** or any other similar tool) that receives **MinionId** and **increases that minion's age by 1**. Write a program that **uses that stored procedure to increase the age** of a minion whose id will be given as input from the console. After that **print the name and the age** of that minion.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | bob – 44 years old |
| 3 | bob – 33 years old |
| 5 | cathleen – 13 years old |