

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ  
ТАРАСА ШЕВЧЕНКА  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ПРОГРАМНИХ СИСТЕМ І ТЕХНОЛОГІЙ**

**Лабораторна робота №5**

з дисципліни «Архітектура та проектування баз  
даних»

**Варіант  
№4**

**«Сховище, в якому зберігається інформація про художні  
фільми»**

Виконав студент  
групи: ІПЗ-23,  
Гоша Давід

Перевірила:  
Духновська Ксенія  
Костянтинівна

## Тема:

Оновлення даних. Видалення даних

## Завдання №1

За своїм варіантом та за прикладом створити словник своєї предметної області.

### Словник предметної області:

*Відношення Фільми:*

| Назва           | Тип, довжина  | Ключ, обмеження               |
|-----------------|---------------|-------------------------------|
| Id              |               | PK                            |
| Назва фільму    | Nvarchar(100) | FK                            |
| Країна виробник | Nvarchar(50)  | FK                            |
| Жанр            | Nvarchar(MAX) | може мати декілька жанрів     |
| Тривалість      | Nchar(12)     | hour + min, разом 12          |
| Рік створення   | Nchar(4)      | тільки рік, без дня та місяця |

*Відношення Назва фільму:*

| Назва             | Тип даних    | Ключ        |
|-------------------|--------------|-------------|
| Назва фільму      | Nvarchar(50) | PK          |
| Коротенький опис  | String       | До 100 слів |
| Назва англійською | Nvarchar(50) |             |

*Відношення Країни:*

| Назва                          | Тип даних    | ключ                                 |
|--------------------------------|--------------|--------------------------------------|
| Назва країни                   | Nvarchar(50) | FK країни                            |
| Мова країни-виробника          | Nvarchar(50) | FK мови                              |
| Мови перекладу                 | Nvarchar(50) | Може бути декілька мов               |
| Рекомендовані фільми цих країн | *            | Посилання на інші фільми цьої країни |

### Відношення Купівлі:

| Назва     | Тип, довжина | Ключ, обмеження                 |
|-----------|--------------|---------------------------------|
| Id товару | Int          | PK                              |
| Ціна      | Money        | FK, обмеження по uint_t         |
| Власник   | Nvarchar(50) | Тільки авторизовані користувачі |

## Завдання №2

Написати запити, за прикладами наведеними в теоретичній частині.

1) Функція `replaceOne()`:

`db.Type.replaceOne({FilmName: "Cruella"}, {name: "One at Home", Year: "1999"})`

В даному випадку знаходимо документ, в якому `FilmName`: "Cruella", та замінюємо його документом `{name: "One at Home", Year: "1999"}`.

Повідомлення після написання запиту :

```
db.Type.replaceOne({FilmName: "Cruella"}, {name: "One atHome", Year: "1999"})
"acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

До застосування функції `replaceOne()`:

```
{
  "_id" : ObjectId("6238c63455f4ebf6313acf7e"),
  "FilmName" : "Cruella",
  "Country" : "USA",
  "Actors" : [
    "Gabriel Ashton Young",
    "Richard Brandon White",
    "Madison Amelia Bailey"
  ],
  "Languge" : "French ",
  "Year" : 2008,
  "Genre" : "Multfilm"
}
```

Результат після застосування функції `replaceOne()`:

```
{
  "_id" : ObjectId("6238c63455f4ebf6313acf7e"),
  "name" : "One atHome",
  "Year" : "1999"
}
```

2) Функція `updateOne()`:

`db.Type.updateOne({"name" : "One atHome"}, {$set: {name: "updateOne", Year: "2000"}})`

Тут ми шукаємо документ з `name`: "One atHome" та встановлюємо для його властивості `name` значення `updateOne` та для `Year` значення `2000`.

Повідомлення після написання запиту:

```
db.Type.updateOne({ "name" : "One atHome"}, {$set: {name: "updateOne", Year : "2000"}})
{"acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

До застосування функції updateOne():

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "One atHome",
"Year" : "1999"
```

Результат після застосування функції updateOne():

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000"
```

### 3) Оператор \$set:

```
db.Type.updateOne({name : "updateOne", year: "2000"}, {$set: {seed: 100/50}})
```

Для оновлення окремих полів у цих функціях використовується оператор \$set. Якщо документ не містить оновлюване поле, воно створюється. В нашому випадку утворилося поле seed.

Повідомлення після написання запиту:

```
> db.Type.updateOne({name : "updateOne", Year: "2000"}, {$set: {seed : 100/50}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

До застосування оператора \$set:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000"
```

Результат після застосування оператора \$set:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"seed" : 2
```

### 4) Функція updateMany():

```
db.Type.updateMany({year : "2000"}, {$set: {country : "Italy"}})
```

Тут оновлюємо всі документи, котрі відповідають певному критерію (year : "2021"), та встановлюємо, наприклад, іншу країну виробника {\$set: {country : "Italy"}}.

Повідомлення після написання запиту:

```
> db.Type.updateMany({Year : "2000"}, {$set: {Country : "Italy"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> □
```

До застосування функції updateMany():

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"seed" : 2
```

Результат після застосування функції updateMany():

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"seed" : 2,
"Country" : "Italy"
```

5) Оператор \$unset:

```
db.Type.update({name : "updateOne"}, {$unset: {seed : 2}})
```

Цей оператор призначений для видалення окремого ключа. В нашому випадку - поле, яке ми створили раніше (seed) - видаляється.

Повідомлення після написання запити:

```
> db.Type.update({name : "updateOne"}, {$unset: {seed : 2}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

До застосування оператора \$unset:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"seed" : 2,
"Country" : "Italy"
```

Результат після застосування оператора \$unset:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy"
```

5) Оператор \$push:

```
db.Type.updateOne({name : "updateOne"}, {$push: {Genre: "Thriller"}})
```

Оператор \$push дозволяє додати ще одне значення до існуючого. В нашому випадку до фільму Don`t look up додався жанр Thriller.

Повідомлення після написання запити:

```
> db.Type.updateOne({name : "updateOne"}, {$push: {Genre: "Thriller"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

До застосування оператора \$push:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy"
```

Результат після застосування оператора \$push:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Thriller"
]
```

6) Оператори \$each, \$position, \$slice:

```
db.Type.updateOne({name : "updateOne"}, {$push: {Genre: {$each:
["Action", "Comedy"], $position:1, $slice:4}}})
```

Використовуючи оператор \$each, можна додати відразу кілька значень. Оператор \$position задає позицію у масиві для вставки елементів, а оператор \$slice вказує, скільки елементів залишити у масиві після вставки. В нашому випадку ми додали одразу додаткові два жанри ["Action", "Comedy"], розташували їх посередині, вивели 4 цих жанри.

Повідомлення після написання запиту:

```
db.Type.updateOne({name : "updateOne"}, {$push: {Genre: {$each: ["Action", "Comedy"],
"acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

До застосування операторів \$each, \$position, \$slice:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Thriller"
]
```

Результат після застосування операторів \$each, \$position, \$slice:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Thriller",
  "Action",
  "Comedy"
]
```

7) Оператор \$addToSet:

```
db.Type.update({name : "updateOne"}, {$addToSet: {Genre: "Adventures"}})
```



Оператор `$addToSet` додає об'єкти в масив. Відмінність від `$push` полягає в тому, що `$addToSet` додає дані, якщо їх ще немає в масиві. В нашому випадку ми спочатку хотіли додати жанр, який вже є в масиві до цього фільму (Adventures), а його повторно не додало. Повідомлення після написання запиту:

```
> db.Type.update({name : "updateOne"}, {$addToSet: {Genre: "Adventures"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

До застосування оператора `$addToSet`:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Thriller",
  "Action",
  "Comedy"
]
```

Результат після застосування оператора `$addToSet`:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Thriller",
  "Action",
  "Comedy",
  "Adventures"
]
```

8) Оператор `$pop`:

`db.Type.updateOne({name : "updateOne"}, {$pop: {Genre: -1}})`

Оператор `$pop` дозволяє видаляти елемент з масиву. Вказавши на ключ `genre: 1`, ми видалили останній елемент. Щоб видалити перший елемент, треба написати `-1`.

Повідомлення після написання запиту:

```
> db.Type.updateOne({name : "updateOne"}, {$pop: {genre: 1}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 0 }
```

До застосування оператора `$pop`:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Thriller",
  "Action",
  "Comedy",
  "Adventures"
]
```

Результат після застосування оператора \$pop:

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Action",
  "Comedy",
  "Adventures"
]
```

9) Функція deleteOne():  
db.Type.deleteOne({name : "updateOne"})

Функція deleteOne() передбачена для видалення документа, в якому name = "updateOne". Щоб видалити кілька документів, можна використати deleteMany().

Повідомлення після написання запити:

```
> db.Type.deleteOne({name : "updateOne"})
{ "acknowledged" : true, "deletedCount" : 1 }
```

До застосування функції deleteOne():

```
"_id" : ObjectId("6238c63455f4ebf6313acf7e"),
"name" : "updateOne",
"Year" : "2000",
"Country" : "Italy",
"Genre" : [
  "Action",
  "Comedy",
  "Adventures"
]
```

Після застосування функції deleteOne():



```

    "_id" : ObjectId("6238c63455f4ebf6313acf7d"),
    "FilmName" : "Blue Bayou",
    "Country" : "Canada",
    "Actors" : [
      "Jayden Tyler Ross",
      "Luis Sebastian Butler",
      "Alex Elijah Ward"
    ],
    "Languge" : "English",
    "Year" : 2019,
    "Genre" : "Historical"

    "_id" : ObjectId("6238c63455f4ebf6313acf7f"),
    "FilmName" : "My Salinger Year",
    "Country" : "Brazil",
    "Actors" : [
      "Tyler Jack Campbell",
      "Ryan John Reed",
      "Molly Makayla Russell"
    ],
    "Languge" : "Portuguese ",
    "Year" : 2021,
    "Genre" : "Triller"

```

10) Видалення за допомогою регулярних виразів:

*db.films.deleteOne({name : /^A\w+/i})*

Можливо видалити документ, в якому назва фільму починається з будь-якої літери (в нашому випадку = A).

Повідомлення після написання запиту:

```

[> db.films.deleteOne({name : /^A\w+/i})
{ "acknowledged" : true, "deletedCount" : 1 }

```

До видалення, за допомогою регулярних виразів:

```

db.typ.find().pretty()

  "_id" : ObjectId("6238c63455f4ebf6313acf7c"),
  "FilmName" : "Palmer",
  "Country" : "USA",
  "Actors" : [
    "Luke Zachary Brooks",
    "Alden Devin Ross",
    "Cameron Juan Kelly"
  ],
  "Languge" : "English",
  "Year" : 2012,
  "Genre" : "Horror"

  "_id" : ObjectId("6238c63455f4ebf6313acf7d"),
  "FilmName" : "Blue Bayou",
  "Country" : "Canada",
  "Actors" : [
    "Jayden Tyler Ross",
    "Luis Sebastian Butler",
    "Alex Elijah Ward"
  ],
  "Languge" : "English",
  "Year" : 2019,
  "Genre" : "Historical"

  "_id" : ObjectId("6238c63455f4ebf6313acf7f"),
  "FilmName" : "My Salinger Year",
  "Country" : "Brazil",
  "Actors" : [
    "Tyler Jack Campbell",
    "Ryan John Reed",
    "Molly Makayla Russell"
  ],
  "Languge" : "Portuguese ",
  "Year" : 2021,
  "Genre" : "Triller"

```

Після видалення, за допомогою регулярних виразів:

```
> db.fyp.find().pretty()
{
  "_id" : ObjectId("6238c63455f4ebf6313acf7d"),
  "FilmName" : "Blue Bayou",
  "Country" : "Canada",
  "Actors" : [
    "Jayden Tyler Ross",
    "Luis Sebastian Butler",
    "Alex Elijah Ward"
  ],
  "Language" : "English",
  "Year" : 2019,
  "Genre" : "Historical"
},
{
  "_id" : ObjectId("6238c63455f4ebf6313acf7f"),
  "FilmName" : "My Salinger Year",
  "Country" : "Brazil",
  "Actors" : [
    "Tyler Jack Campbell",
    "Ryan John Reed",
    "Molly Makayla Russell"
  ],
  "Language" : "Portuguese ",
  "Year" : 2021,
  "Genre" : "Triller"
}
```

## Висновок :

Виконавши лабораторну роботу № 5 на тему: «Оновлення даних. Видалення даних» було створено словник своєї предметної області та написано запити, за прикладами наведеними в теоретичній частині.