Лабораторна робота №5

ТЕМА: Оновлення даних. Видалення даних

Джерела: https://metanit.com/nosql/mongodb/2.3.php

Як і інші системи управління базами даних, MongoDB надає можливість оновлення даних. Для цього є ряд функцій.

replaceOne

Якщо нам потрібно повністю замінити один документ іншим, може використовуватись функція replaceOne:

db.collection.replaceOne(filter, update, options)

filter: приймає запит на вибірку документа, який потрібно оновити

update: представляє новий документ, який замістить старий під час оновлення

options: визначає додаткові параметри під час оновлення документів, основним з яких є параметр upsert.

Якщо параметр upsert має значення true, mongodb буде оновлювати документ, якщо він знайдений, і створювати новий, якщо такого документа немає. Якщо він має значення false, то mongodb не буде створювати новий документ, якщо запит на вибірку не знайде жодного документа.

Наприклад:

```
db.users.replaceOne({name: "Bob"}, {name: "Bob", age: 25})
```

В даному випадку знаходимо документ, в якому name = "Bob", та замінюємо його документом {name: "Bob", age: 25}

updateOne та updateMany

Часто не потрібно оновлювати весь документ, а лише значення однієї або кількох його властивостей. Для цього використовуються функції updateOne() - вона оновлює лише один документ і updateMany() - дозволяє оновити безліч документів.

Для оновлення окремих полів у цих функціях використовується оператор $set. Якщо документ не містить оновлюване поле, воно створюється.

```
> db.users.updateOne({name : "Tom", age: 25}, {$set: {age : 28}})
```

Тут ми шукаємо документ з name="Tom" та age=25 та встановлюємо для його властивості age значення 28

Якщо оновлюваного поля в документі немає, воно додається:

```
> db.users.updateOne({name : "Tom", age: 28}, {$set: {salary : 300}})
```

Якщо потрібно оновити значення кількох полів, то вони передаються оператору $set через кому:

```
> db.users.updateOne({name : "Tom"}, {$set: {name: "Tomas", age : 25}})
```

Для простого збільшення значення числового поля на певну кількість одиниць застосовується оператор $inc. Якщо документ не містить оновлюване поле, воно створюється. Цей оператор застосовується лише до числових значень.

```
> db.users.updateOne({name : "Tom"}, {$inc: {age:2}})
```

Якщо необхідно оновити всі документи, що відповідають певному критерію, застосовується функція updateMany():

```
> db.users.updateMany({name : "Tom"}, {$set: {salary : 560}})
```

Видалення поля

Для видалення окремого ключа використовується оператор $unset:

```
> db.users.updateOne({name : "Tom"}, {$unset: {salary: 1}})
```

Якщо раптом такого ключа в документі не існує, то оператор не має жодного впливу. Також можна видаляти одразу кілька полів:

```
> db.users.update({name : "Tom"}, {$unset: {salary: 1, age: 1}})
```

Оновлення масивів

Оператор $push

Оператор $push дозволяє додати ще одне значення до існуючого. Наприклад, якщо ключ як значення зберігає масив:

```
> db.users.updateOne({name : "Tom"}, {$push: {languages: " ukraine "}})
```

Вище використовується функція updateOne, але цей оператор також застосовується і у функції updateMany

```
> db.users.updateMany({name : "Tom"}, {$push: {languages: "ukraine"}})
```

Якщо ключ, для якого ми хочемо додати значення, не є масивом, ми отримаємо помилку Cannot apply $push/$pushAll modifier to non-array.

Використовуючи оператор $each, можна додати відразу кілька значень:

```
> db.users.updateOne({name : "Tom"}, {$push: {languages: {$each: ["українська", "spanish", "italian"]}}})
```

Ще пара операторів дозволяє настроїти вставку. Оператор $position задає позицію у масиві для вставки елементів, а оператор $slice вказує, скільки елементів залишити у масиві після вставки.

```
> db.users.updateOne({name : "Tom"}, {$push: {languages: {$each: ["german", "spanish", "italian"], $position:1, $slice:5} }})
```

В даному випадку елементи ["german", "spanish", "italian"] будуть вставлятися в масив languages з 1-го індексу, і після вставки в масиві залишаться тільки 5 перших елементів.

Оператор $addToSet

Оператор $addToSet подібно до оператора $push додає об'єкти в масив. Відмінність полягає в тому, що $addToSet додає дані, якщо їх ще немає в масиві (при додаванні через $push дані дублюються, якщо додаються елементи, які є в масиві):

```
> db.users.updateOne({name : "Tom"}, {$addToSet: {languages: "russian"}})
```

Видалення елемента з масиву

Оператор $pop дозволяє видаляти елемент із масиву:

```
> db.users.updateOne({name : "Tom"}, {$pop: {languages: 1}})
```

Вказуючи на ключ languages значення 1, ми видаляємо перший елемент з кінця. Щоб видалити перший елемент спочатку масиву, треба передати негативне значення:

```
> db.users.updateOne({name : "Tom"}, {$pop: {languages: -1}})
```

Дещо іншу дію передбачає оператор $pull. Він видаляє кожне входження елемента до масиву. Наприклад, через оператор $push ми можемо додати те саме значення в масив кілька разів. І тепер за допомогою $pull видалимо його:

```
> db.users.updateOne({name : "Tom"}, {$pull: {languages: "english"}})
```

Якщо ми хочемо видалити не одне значення, а відразу кілька, тоді ми можемо застосувати оператор $pullAll:

```
> db.users.updateOne({name : "Tom"}, {$pullAll: {languages: ["english", "german", "french"]}})
```

Видалення даних

Для видалення документів у MongoDB передбачені функції deleteOne() – видаляє один документ та deleteMany() – дозволяє видалити кілька документів. Як параметр у ці функції передається фільтр документів, що видаляються.

Наприклад, видалимо документ, у якому name="Tom":

```
> db.users.deleteOne({name : "Tom"})
```

У результаті перший знайдений документ із name=Tom буде видалено. Для видалення всіх документів, які відповідають фільтру, застосовується функція deleteMany():

```
> db.users.deleteMany({name : "Tom"})
```

Причому, як і у випадку з find, ми можемо задавати умови вибірки для видалення у різний спосіб (у вигляді регулярних виразів, у вигляді умовних конструкцій тощо):

```
> db.users.deleteOne({name : /^T\w+/i})
> db.users.deleteOne({age: {$lt : 30}})
```

Щоб видалити разом всі документи з колекції, треба залишити порожнім параметр запиту:

```
> db.users.deleteMany({})
```

Видалення колекцій та баз даних

Ми можемо видаляти не лише документи, а й колекції та бази даних. Для видалення колекцій використовується функція drop:

```
> db.users.drop()
```

І якщо видалення колекції пройде успішно, то консоль виведе:

```
true
```

Щоб видалити всю базу даних, треба скористатися функцією dropDatabase():

```
> db.dropDatabase()
```

Словник предметної області

1. Створити БД
   В базі даних зберігаються наступні відношення:

   1. Контрагенти
   2. Групи контрагентів
   3. Договора контрагентів
   4. Банківські рахунки
   5. Номенклатура
   6. Фізичні особи
   7. Працівники
   8. Контактні особи
   9. Країни
   10. Міста
   11. Вулиці
   12. Валюти
   13. Курси валют
   14. Банки
   15. Групи номенклатури
   16. Одиниці вимірювання
   17. Ціни номенклатури
   18. Рахунки
   19. Товари для рахунку

Відношення Контрагенти:

Назва	Тип, довжина	Ключ, обмеження
Id		PK
Назва	Nvarchar(100)	
Група	Int	FK (Групи контрагентів)
Повна назва	Nvarchar(100)	
Вид (юридична особа, фізична особа)	bool	
Код ЄДРПОУ	Varchar(8)	Число із 8 цифр
Основний банківській рахунок	Nchar(29)	UA+цифри, разом 29
Основний договір	Int	FK, Договора
Контактна особа	Int	FK, контактні особи
Юридична адреса	Nvarchar(MAX)	
Фактична адреса	Nvarchar(MAX)	
Поштова адреса	Nvarchar(MAX)	
Керівник	int	FK, Працівники
Головний бухгалтер	int	FK, Працівники
Контактна особа від контрагента	int	FK, контактні особи від контрагента
ПІБ(для фізичної особи)	int	FK, фізичні особи
Документ (для фізичної особи)	Nvarchar(12)	

Відношення групи контрагентів

Назва	Тип, довжина	Ключ, обмеження
№	int	PK
Назва	Nvarchar(30)	

Відношення договору контрагентів

Назва	Тип, довжина	Ключ, обмеження
Id	int	PK
Вид договору	Int	<= 5
Номер	Int	
Дата	date	
Валюта	int	FK, валюти

Відношення Банківські рахунки

Назва	Тип, довжина	Ключ, обмеження
Код	int	PK
Номер рахунку	Int	
МФО банку	Int	FK, banks
Дата відкриття	Date	
Дата закриття	Date	

Назва рахунку	Nvarchar(50)	
Валюта	int	

Відношення Номенклатура

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
Група номенклатури	Int	FK, групи номенклатури
Назва номенклатури	Nvarchar(50)	
Базова одиниця вимірювання	Int	FK, одиниці вимірювання
Артикул	Nvarchar(10)	
Облік ПДВ	bool	
Акцизний податок	Real(10.2)	

Відношення фізичні особи

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
ПІБ	Nvarchar(50)	
ІПН	Varchar(10)	Число із 10 цифр

Відношення працівники

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
Фізична особа	Int	FR, фізичні особи

Відношення контактні особи

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
Працівник	int	FK, працівникі

Відношення Країни

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
Назва країни	Nvarchar(50)	

Відношення Міста

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
Назва міста	Nvarchar(50)	
Країна	Int	FK, країни

Відношення вулиці

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK

Назва	Nvarchar(50)	
Тип вулиці	int	Число менше 6

Відношення валюти

Назва	Тип, довжина	Ключ, обмеження
Код	Varchar(4)	PK
Назва	Nvarchar(20)	

Відношення курси валют

Назва	Тип, довжина	Ключ, обмеження
Id	int	PK
Валюта	Varchar(4)	FK валюти
Дата	date	
Курс	Real(10.2)	

Відношення Банки

Назва	Тип, довжина	Ключ, обмеження
МФО	Int	PK, число із 6 цифр
Назва банку	Nvarchar(50)	

Відношення Групи номенклатури

Назва	Тип, довжина	Ключ, обмеження
Код	int	PK
Назва групи	Nvarchar(30)	

Відношення одиниці вимірювання

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
назва	Nvarchar(20)	

Відношення ціни номенклатури

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
Дата	Date	
Номенклатура	Int	FK, номенклатура
Ціна	Real(10.2)	

Відношення рахунки

Назва	Тип, довжина	Ключ, обмеження
Код	int	PK
Номер	Varchar(10)	
Дата	Date	
Контрагент	Int	FK, контрагенти
Договір	Int	FK, договори
Банківський рахунок	Int	FK, банківські рахунки
Скідка на суму товарів	Real(10.2)	

Відношення товари для рахунку

Назва	Тип, довжина	Ключ, обмеження
Код	Int	PK
Рахунок	Int	FK, рахунки
Товар	Int	FK, номенклатура
Кількість	Real(10.2)	
Скідка	Real(10.2)	

ЗАВДАННЯ

1. За своїм варіантом та за прикладом створити словник своєї предметної області.
2. Написати запити (для своїх варіантів), за прикладами наведеними в теоретичній частині.