

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"  
Кафедра систем штучного інтелекту



ЗВІТ 14  
з курсу “ОБДЗ”  
на тему:  
**“ Розробка бази даних типу NoSQL ”**

**Виконала:**  
студентка групи КН-211  
Досяк Ірина  
**Викладач:**  
Якимишин Х.М.

## Лабораторна №14

**Мета роботи:** здобуття практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB

### Короткі теоретичні відомості.

Функціональні можливості:

- узгодженість даних
- транзакції
- доступність
- можливості запитів
- масштабування

Типи значень:

- String
- Array (масив)
- Binary data (двоичные данные)
- Boolean
- Date
- Double
- Integer
- JavaScript
- Min key/Max key
- Null
- Object
- ObjectID
- Regular expression
- Symbol
- Timestamp

Операції:

#### Перейменування колекції

> db.persons.renameCollection("нова\_назва")

## **Вибірка з БД**

- > db.persons.find()
- > db.persons.find({ name: "Tom", age: "32" })

## **Налаштування запитів і сортування**

- > db.persons.find().limit(3)
- > db.persons.find().skip(3)
- > db.persons.find().sort({ name: 1 })
- > db.persons.find().sort({ name: 1 }).skip(3).limit(3)

## **Команди групування Число елементів в колекції**

- > db.persons.count()
- > db.persons.find({ name: "Tom" }).count()
- > db.persons.find({ name: "Tom" }).skip(2).count(true)

## **Функція distinct**

- > db.persons.distinct("name")
- ["Tom", "Bill", "Bob"]

## **Метод group**

- > db.persons.group ({key: { name : true }, initial: {total : 0}, reduce : function (items,prev){prev.total += 1 } })

## **Умовні оператори \$gt (більше ніж) \$lt (менше ніж)**

## **\$gte (більше чи рівно) \$lte (менше чи рівно)**

- > db.persons.find ({age: { \$lt : 30 } })
- > db.persons.find ({age: { \$gt : 30 } })

## **Оновлення даних**

- > db.persons.save({"name": "Eugene", "age" : "29", languages: ["english", "german", "spanish"]})

## **Оновлення окремого поля**

- > db.persons.update({ name : "Eugene", age: "29" }, {"age": { \$set: "30" } })

## **Знищення поля**

- > db.persons.update({ name : "Tom" }, { \$unset: { salary: 1 } })
- > db.persons.update({ name : "Tom" }, { \$unset: { salary: 1, age: "" } })

## Оператор \$push

> db.persons.update({ name : "Tom" }, { \$push: { languages: " ukrainian " } })

## Оператор \$addToSet

> db.persons.update({ name : "Tom" }, { \$addToSet: { languages: " ukrainian " } })

## Знищення елемента з масиву

> db.persons.update({ name : "Tom" }, { \$pop: { languages: 1 } })

## Знищення даних

> db.persons.remove({ name : "Tom" })

## Знищення колекцій і баз даних

> db.persons.drop()

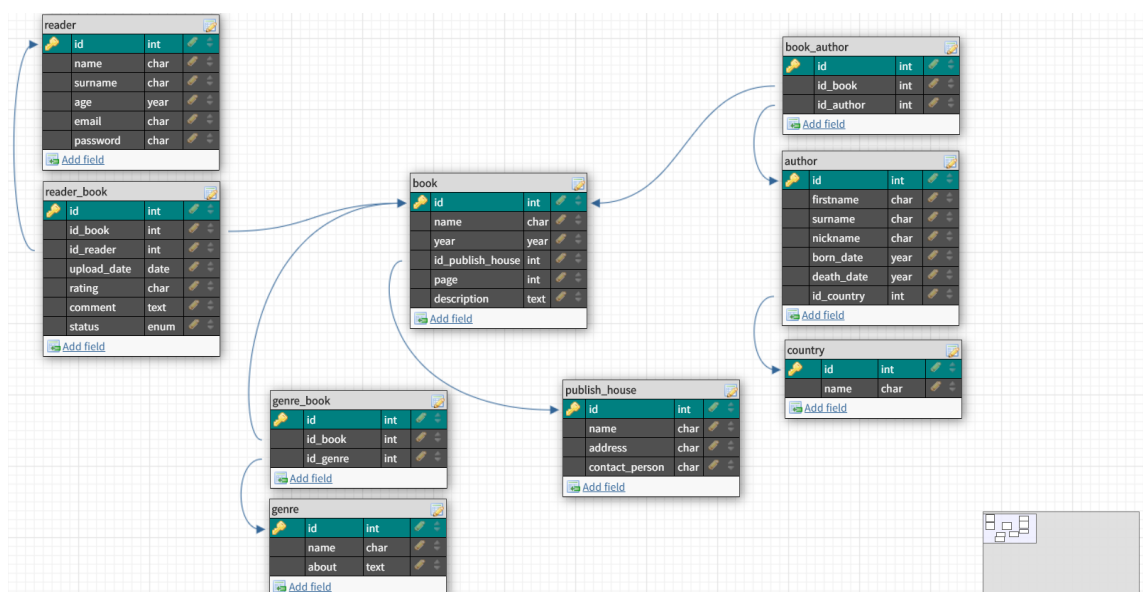
> db.dropDatabase()

## Завдання

1. Розробити схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB...
2. Забезпечити реалізацію функцій редагування, додавання та видалення інформації в «сутність».

## Виконання

1. Схема баз даних в MySQL:



## Схема баз даних в MongoDB:

*use library*

*show collections*

```
> use library
switched to db library
> show collections
author
book
country
genre
publish_house
reader
```

*db.name.findOne()*

```
> db.author.findOne()
{
  "_id" : ObjectId("5eaf02df9c405ced7700d01e"),
  "firstname" : "Elizabeth",
  "surname" : "Gilbert",
  "nickname" : "Elizabeth Gilbert",
  "born_date" : ISODate("1986-09-17T20:00:00Z"),
  "id_country" : {
    "_id" : ObjectId("5eaf0ceb9c405ced7700d020"),
    "name" : "USA"
  },
  "book" : [
    ObjectId("5eaf11fb9c405ced7700d023")
  ]
}
> db.book.findOne()
{
  "_id" : ObjectId("5eaf11fb9c405ced7700d023"),
  "name" : "The city of girls",
  "year" : 2019,
  "page" : 496,
  "id_publish_house" : {
    "_id" : ObjectId("5eaf13599c405ced7700d024"),
    "name" : "A-ba-ba-ha-la-ma-ha"
  },
  "description_of_book" : "Told from the perspective of an older woman as she looks back on her youth with both pleasure and regret.",
  "genre" : [
    ObjectId("5eaf11179c405ced7700d021"),
    ObjectId("5eaf11479c405ced7700d022")
  ],
  "author" : [
    ObjectId("5eaf02df9c405ced7700d01e")
  ],
  "reader" : [
    ObjectId("5eaf16459c405ced7700d025")
  ]
}
```

```
> db.publish_house.findOne()
{
  "_id" : ObjectId("5eaf13599c405ced7700d024"),
  "name" : "A-ba-ba-ha-la-ma-ha",
  "adress" : "Kiev,st.Swimming pools, building 1/2",
  "contact" : "client@gmail.com",
  "book" : [
    ObjectId("5eaf11fb9c405ced7700d023")
  ]
}
> db.genre.findOne()
{
  "_id" : ObjectId("5eaf11179c405ced7700d021"),
  "name" : "Historical fiction",
  "about" : "A story about real person or event in past",
  "book" : [
    ObjectId("5eaf11fb9c405ced7700d023")
  ]
}
> db.country.findOne()
{
  "_id" : ObjectId("5eaf0ceb9c405ced7700d020"),
  "name" : "USA",
  "authors" : [
    ObjectId("5eaf02df9c405ced7700d01e")
  ]
}
```

2. Оновимо country, а саме name:

```
db.country.updateOne({name: 'USA'}, {$set: {name: 'United States'}})
```

```
_id: ObjectId("5eaf0ceb9c405ced7700d020")
name: "United States"
✓ author: Array
  0: ObjectId("5eaf02df9c405ced7700d01e")
```

Додамо нове поле в country:

```
db.country.updateOne({name: 'United States'}, {$set: {city: 'New York'}})
```

*\$set створить вказане поле, якщо такого не існує*

```
_id: ObjectId("5eaf0ceb9c405ced7700d020")
name: "United States"
> author: Array
city: "New York"
```

Видалимо створене поле:

```
db.country.update({}, {$unset: {city: 'New York'}});
```

```
_id: ObjectId("5eaf0ceb9c405ced7700d020")
name: "United States"
> author: Array
```

Отже, використовуючи MongoDB можна:

- створювати документи, не задаючи їм структуру для збереження заздалегідь;
- кожен документ може мати власну структуру;
- у кожній базі даних може бути власний синтаксис;
- можна додавати поля під час роботи з даними.

## Контрольні запитання

1. Назвати основні типи баз даних NoSQL.

- Базы даних виду «ключ-значення»;
- документоорієнтовані бази даних;
- графові бази даних;
- колоноподібні бази даних.

## 2. Назвати переваги та недоліки використання баз даних NoSQL.

### Переваги:

- простота роботи;
- простіший синтаксис запитів;
- кожен документ може мати власну структуру;
- можна додавати нові поля під час роботи з даними;

### Недоліки:

- Обмежена ємність вбудованої мови запитів;
- Низька цінність і вузькопрофільність знань.

## 3. Надати характеристику СУБД MongoDB.

MongoDB — документо-орієнтована система керування базами даних. Вона підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, підтримує реплікацію і побудову відмовостійких конфігурацій.

## 4. Операції вставки даних.

`insert()` або `save()`

## 5. Операції оновлення даних.

`update()` , `save()`

## 6. Операції знищення даних.

`update()`, `remove()`

## 7. Умовні оператори.

- `$gt` (більше ніж)
- `$lt` (менше ніж)
- `$gte` (більше чи рівно)
- `$lte` (менше чи рівно)

## 8. Операції керування індексами.

`db.system.indexes.find()`

```
db.name.dropIndex("name")
```

9. Пошук даних.

```
find(), findOne(), findAndModify()
```

10. Можливості документних БД.

- Документо-орієнтоване сховище
- Досить гнучка мова для формування запитів
- Динамічні запити
- Повна підтримка індексів
- Профілювання запитів
- Швидкі оновлення
- Ефективне зберігання даних великих обсягів, наприклад, фото та відео
- Журналювання операцій, що модифікують дані в БД

**Висновок:** на лабораторній роботі я здобула практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB. Розробила схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB, і забезпечила реалізацію функцій редагування, додавання та вилучення інформації в «сутність».



