

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



Звіт до лабораторної роботи №14
з дисципліни:
“ОБДЗ”
на тему:
“Розробка бази даних типу NoSQL ”

Підготувала:
студентка групи КН-209
Дипко Олександра
Викладач:
Мельникова Н.І.

Львів 2020

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
- Min key/Max key
- Null
- Object
- ObjectID
- Regular expression
- Symbol
- Timestamp

Операції:

Перейменування колекції

```
В 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 Чило 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()

Хід роботи

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 вилучення інформації в «сутність».

1. Розробити схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB.

Схема баз даних в MySQL:

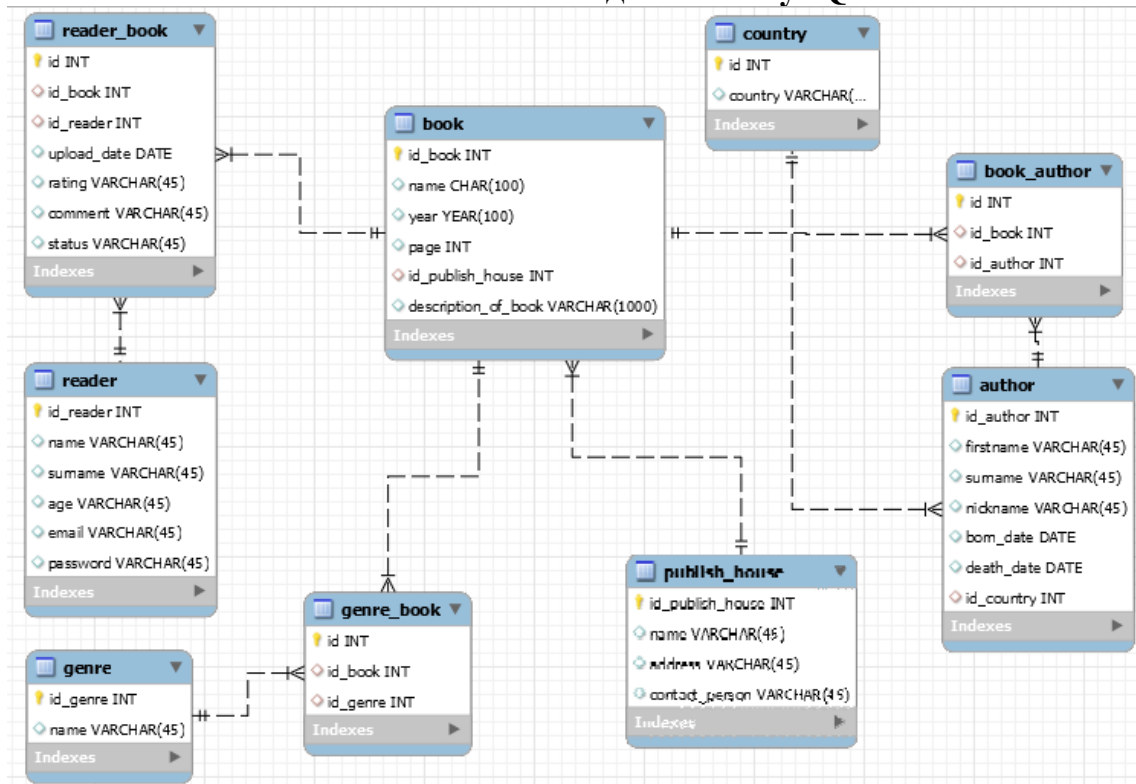


Схема баз даних в Mongoddb:

- Створюємо базу даних командою **use library** та колекції командою:
db.createCollection("назва")

```
> use library
switched to db library
> db.createCollection("book")
{ "ok" : 1 }
> db.createCollection("reader")
{ "ok" : 1 }
> db.createCollection("genre")
{ "ok" : 1 }
> db.createCollection("author")
{ "ok" : 1 }
> db.createCollection("country")
{ "ok" : 1 }
> db.createCollection("publish_house")
{ "ok" : 1 }
```

- Перевіряємо наявні колекції командою **show collections**

```
> show collections
author
book
country
genre
publish_house
reader
```

- Заповнюємо колекції даними командою **db.collection.insertOne()**
db.book.insertOne({"name":"Harry Potter and the Prisoner of Azkaban", "year":"2004", "page":"480", "publish_house":{""}, "author":""})

```
_id: ObjectId("5ebc10f94a1bcac052ca2e1a")
name: "Harry Potter and the Prisoner of Azkaban"
year: "2004"
page: "480"
author: ObjectId("5ebc122d4a1bcac052ca2e1c")
publish_house: ObjectId("5ebc133a4a1bcac052ca2e1f")
```

```
db.reader.insertOne({"name":"Oleksandra", "surname":"Dypko", "age":"18", "email":"oleksandra@gmail.com", "password":"1234567890", "genre":"","books":""})
```

```
_id: ObjectId("5ebc13794a1bcac052ca2e20")
name: "Oleksandra"
surname: "Dypko"
age: 18
email: "example@gmail.com"
password: "1234567890"
✓ books: Object
  _id: ObjectId("5ebc10f94a1bcac052ca2e1a")
```

```
db.author.insertOne({"firstname":"Joanne", "lastname":"Rowling", "nickname":"J.K.Rowling", "born_date":"1965-05-31", "death_date":"","country":""})
```

```
_id: ObjectId("5ebc122d4a1bcac052ca2e1c")
firstname: "Joanne"
surname: "Rowling"
nickname: "J.K.Rowling"
born_date: 1965-05-30T21:00:00.000+00:00
death_date: null
country: ObjectId("5ebc12bf4a1bcac052ca2e1d")
✓ books: Object
  _id: ObjectId("5ebc10f94a1bcac052ca2e1a")
```

```
db.country.insertOne({"name":"Great Britain", "authors":[]})
```

```
_id: ObjectId("5ebc12bf4a1bcac052ca2e1d")
name: "Great Britain"
> authors: Array
```

```
db.country.insertOne({"name":"USA", "authors":[]})
```

```
_id: ObjectId("5ebd75348b632670e0ed68b3")
name: "USA"
> authors: Array
```

2. Забезпечити реалізацію функцій редагування, додавання та вилучення інформації в «сутність».

Оновимо *country*, а саме *name*:

```
_id: ObjectId("5ebd75348b632670e0ed68b3")
name: "USA"
> authors: Array
```

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

```
_id: ObjectId("5ebd75348b632670e0ed68b3")
name: "United States"
authors: Array
```

Додамо нове поле в *country* (*\$set* створить вказане поле, якщо такого не існує):

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

```
_id: ObjectId("5ebd75348b632670e0ed68b3")
name: "United States"
authors: Array
city: "New York"
```

Видалимо створене поле:

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

```
_id: ObjectId("5ebd75348b632670e0ed68b3")
name: "United States"
> authors: Array
```

Контрольні запитання

1. Назвати основні типи баз даних NoSQL.
 - Бази даних виду «ключ-значення»;
 - Документо-орієнтовані 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

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».