

Лабораторна робота №14
з курсу “ОБДЗ”
на тему:
“Розробка бази даних типу NoSQL”

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 для роботи з даними в середовищі проектування документних БД MongoDB

Додавання даних і створення колекцій

```
> db.persons.insert({"name": "Tom", "age": "28", languages: ["english", "spanish"]})  
> db.persons.find()  
> document=({"name": "Bill", "age": "32", languages: ["english", "french"]})  
> db.persons.insert(document)
```

Обмеження імен ключів:

Символ \$ не може бути першим символом в імені ключа

Ім'я ключа не може містити символ крапки.

Ім'я _id не рекомендується використовувати

Перейменування колекції

```
> db.persons.renameCollection("нова_назва")  
результат  
{ "ok" : 1 }
```

Явне створення колекції

```
> db.persons.createCollection("accounts")  
результат  
{ "ok" : 1 }
```

Обмеження колекції

```
> db.createCollection("profile", { capped:true, size:9500})  
{ "ok":1}  
> db.createCollection("profile", { capped:true, size:9500, max: 150})
```

Вибірка з БД

```
> db.persons.find()
> db.persons.insert({"name": "Tom", "age": "28", languages: ["english", "spanish"]})
> db.persons.insert({"name": "Bill", "age": "32", languages: ["english", "french"]})
> db.persons.insert({"name": "Tom", "age": "32", languages: ["english", "german"]})
> db.persons.find({ name: "Tom" })
> db.persons.find({ languages: "german" })
> db.persons.find({ name: "Tom", age: "32" })
> db.persons.find({ name: "Tom" }, { age: 1 })
> db.persons.find({ name: "Tom" }, { age: 0 })
```

Запит до вкладених об'єктів

```
> db.persons.insert({"name": "Alex", "age": "28", company: {"name": "microsoft", "country": "USA"}})
> db.persons.find({"company.name": "micriosoft"})
```

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)
```

Використання курсорів

```
> var cursor = db.persons.find()
> var cursor = db.persons.find()
> while(cursor.hasNext()){
... obj = cursor.next();
... print(obj["name"]);
... }
> var cursor = db.persons.find()
> cursor.forEach(function(obj){
... print(obj.name);
... })
```

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 } })
```

Оператор \$ne

```
> db.persons.find ({ age: { $ne : 22 } })
```

Пошук по масивам і оператори \$in, \$nin, \$all

```
> db.persons.find ({ age: { $in : [22, 32] } })
```

```
> db.persons.find ({ age: { $nin : [22, 32] } })
```

```
> db.persons.find ({ age: { $all : [22, 32] } })
```

```
> db.persons.find ({ age: { $all : [22] } })
```

```
> db.persons.find ({ languages: { $all : ["english", "french"] } })
```

Оператор \$or

```
> db.persons.find ({ $or : [{ name: "Tom"}, { age: "22" } ] })
> db.persons.find ({ name: "Tom", $or : [{ age: "22"}, { languages: "german" } ] })
```

Оператор \$size

```
> db.persons.find ({ languages: { $size: 2 } })
{ "name": "Tom", "age": "32", languages: ["english", "german"] }
```

Оператор \$exists

```
> db.persons.find ({ company: { $exists: true } })
```

Оновлення даних

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

Функція update. приймає три параметра:

```
> db.persons.update({ name : "Tom" }, { "name": "Tom", "age" : "25", "married" : false }, { upsert: true })
> db.persons.update({ name : "Tom" }, { "name": "Tom", "age" : "25", "married" : false }, { upsert: true,
multi:true })
```

Оновлення окремого поля

```
> db.persons.update({ name : "Eugene", age: "29" }, { "age": { $set: "30" } })
> db.persons.update({ name : "Tom" }, { $inc: { salary: 100 } })
```

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 " } })
> db.persons.update({ name : "Tom" }, { $addToSet: { languages: { $each: ["ukrainian", "spanish", "italian"] } } })
```

Знищення елемента з масиву

```
> db.persons.update({ name : "Tom" }, { $pop: { languages: 1 } })
> db.persons.update({ name : "Tom" }, { $pop: { languages: -1 } })
> db.persons.update({ name : "Tom" }, { $pull: { languages: "english" } })
> db.persons.update({ name : "Tom" }, { $pullAll: { languages: ["english", "german", "french"] } })
```

Знищення даних

```
> db.persons.remove({ name : "Tom" })
> db.persons.remove({ name : /T\w+/i })
> db.persons.remove({ age: { $lt : 30 } })
> db.persons.remove({ name : "Tom" }, true)
```

Знищення колекцій і баз даних

```
> db.persons.drop()
true
> db.dropDatabase()
```

Посилання в БД

Ручна установка посилань

```
> db.companies.insert({ "_id" : "microsoft", "year": 1974 })
> db.persons.insert({ "name": "Tom", "age": 28, company: "microsoft" })
> person = db.persons.findOne()
> db.companies.findOne({ _id: person.company })
```

Автоматичне зв'язування

```
> apple=({ "name" : "apple", "year": 1976 })
> db.companies.save(apple)
> steve = ({ "name": "Steve", "age": 25, company: new DBRef('companies', apple._id) })
> db.persons.save(steve)
> db.companies.findOne({ _id: steve.company.$id })
{ "$ref" : назва_колекції, "$id": значення [, "$db" : назва_бд ] }
```

Робота з індексами

```
> db.persons.ensureIndex({ "name" : 1 })
```

Налаштування індексів

```
> db.persons.ensureIndex({ "name" : 1 }, { "unique" : true })
> db.persons.ensureIndex({ "name" : 1, "age" : 1 }, { "unique" : true })
```

Керування індексами

```
> db.system.indexes.find()
> db.persons.dropIndex("name_1")
```

Хід роботи.

1. Розробити схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB..
2. Перетворити сутності діаграми БД, розробленої для лабораторної роботи №1, у структури, прийнятні для обробки в MongoDB.
3. Забезпечити реалізацію функцій редагування, додавання та вилучення інформації в «сутність».
4. Підготувати та захистити звіт до лабораторної роботи.

Контрольні запитання

1. Назвати основні типи баз даних NoSQL.
2. Назвати переваги та недоліки використання баз даних NoSQL.
3. Надати характеристику СУБД MongoDB.
4. Операції вставки даних.
5. Операції оновлення даних.
6. Операції знищення даних.
7. Умовні оператори.
8. Операції керування індексами.
9. Пошук даних.
10. Можливості документних БД.

Література

1. Official MongoDB Project Website
2. Eric Lai. (2009, July 1). No to SQL? Anti-database movement gains steam
3. <https://docs.mongodb.com/manual/release-notes/4.0/>
4. <https://docs.mongodb.com/manual/>
5. MongoDB articles on NoSQLDatabases.com
6. EuroPython Conference Presentation
7. Non-relational data persistence in Java using MongoDB - Software Engineer at MongoDB на YouTube
8. Interview with Mike Dirolf on The Changelog about MongoDB background and design decisions
9. MongoMvc - A MongoDB Demo App with ASP.NET MVC

Встановлення:

Щоб встановити mongoDB, перейдіть по цьому посиланню і натисніть відповідну ОС:

<http://www.mongodb.org/downloads>

Спочатку витягніть файли (переважно на диск C).

Створіть каталог даних на C: mongoDB для використання "md data", за якими слідують "md data\db"