МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"
ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ
КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №14

Підготувала:

Студентка групи КН-209

Кульчицька Олена

Викладач:

Мельникова Н.І.

2020

Лабораторна робота №14
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 (массив)
• 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

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}})

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. прийнимає 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"]}})

Знщення 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».

Схема баз даних в MySQL

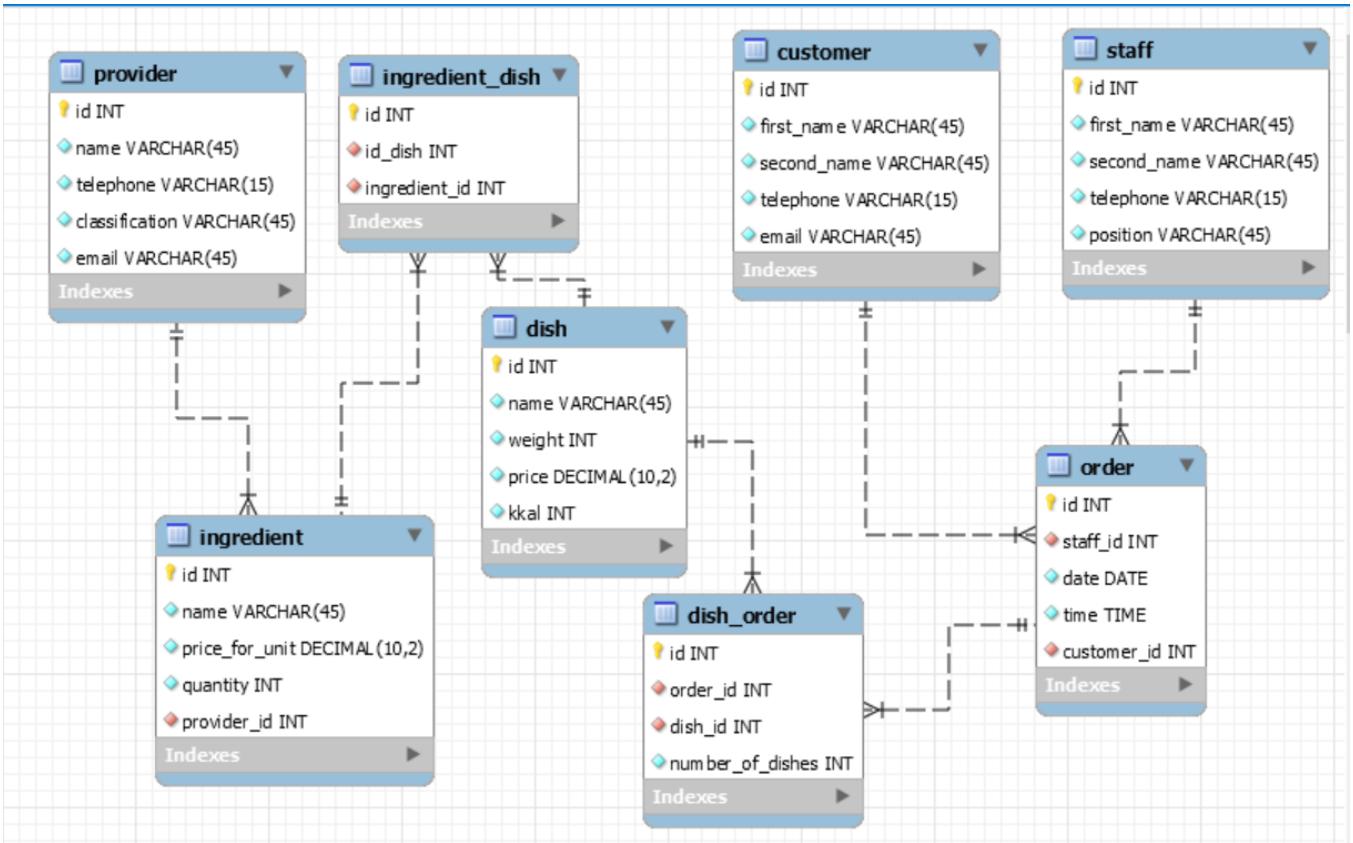

Схема баз даних в Mongodb:

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

```
> use confectionary
switched to db confectionary
> db.createCollection("dish")
{ "ok" : 1 }
> db.creteCollection("customer")
> db.createCollection("customer")
{ "ok" : 1 }
> db.createCollection("staff")
{ "ok" : 1 }
> db.createCollection("provider")
{ "ok" : 1 }
> db.createCollection("ingredient")
{ "ok" : 1 }
> db.createCollection("order")
{ "ok" : 1 }
```

- Перевіряємо наявні колекції командою *show collections.*

```
> show collections
customer
dish
ingredient
order
provider
staff
```

- Заповнимо колекції даними за допомогою команди *db.колекція.insertOne()*

>db.customer.insertOne({"cus_first_name":"Ben","cus_sec_name":"Heine","phone":2763454,"email":"bheine@gmail.com"})

```
_id: ObjectId("5eb99baa310bc535f853bfa9")
cus_first_name: "Ben"
cus_sec_name: "Heine"
phone: 2763454
email: "bheine@gmail.com"
```

>db.staff.insertOne({"st_first_name":"Caren","st_sec_name":"Smitt","phone":2768495,"position":"waiter"})

```
_id: ObjectId("5eb99c44310bc535f853bfaa")
st_first_name: "Caren"
st_sec_name: "Smitt"
phone: 2768495
position: "waiter"
```

>db.provider.insertOne({"name":"Roshen","phone":2768954,"classification":"sweets","email":"roshen@gmail.com"})

```
_id: ObjectId("5eb99d1d310bc535f853bfab")
name: "Roshen"
phone: 2768954
classification: "sweets"
email: "roshen@gmail.com"
```

- Дадаємо поле за допомогою автоматичного зв’язування командою new DBRef().

>molokiya=({"name":"molokiya","phone":2768354,"classification":"milk","email":"molokiya@gmail.com"})

> db.provider.save(molokiya)

```
_id: ObjectId("5eb9a171310bc535f853bfac")
name: "molokiya"
phone: 2768354
classification: "milk"
email: "molokiya@gmail.com"
```

> ingredient=({"name":"milk","price_for_unit":20,"quantity":10,id_provider: new DBRef('provider',molokiya._id)})

> db.ingredient.save(ingredient)

```
_id: ObjectId("5eb9a271310bc535f853bfad")
name: "milk"
price_for_unit: 20
quantity: 10
id_provider: DBRef(provider, 5eb9a171310bc535f853bfac, undefined)
```

>olena=({"cus_first_name":"Olena","cus_sec_name":"Klein","phone":9963454,"email":"olena@gmail.com"})
> db.customer.save(olena)

```
_id: ObjectId("5eb9a468310bc535f853bfae")
cus_first_name: "Olena"
cus_sec_name: "Klein"
phone: 9963454
email: "olena@gmail.com"
```

>steve=({"st_first_name":"Steve","st_sec_name":"Clark","phone":2000495,"position":"cook"})
> db.staff.save(steve)

```
_id: ObjectId("5eb9a4e4310bc535f853bfaf")
st_first_name: "Steve"
st_sec_name: "Clark"
phone: 2000495
position: "cook"
```

>apple_pie = ({"name":"apple_pie","weight":350,"price":150,"kkal":1000, ingredient: new DBRef('ingredient',ingredient._id)})
> db.dish.save(apple_pie)

```
_id: ObjectId("5eb9a70a310bc535f853bfb1")
name: "apple_pie"
weight: 350
price: 150
kkal: 1000
ingredient: DBRef(ingredient, 5eb9a271310bc535f853bfad, undefined)
```

> order01 =({"date_time":new Date('2020-05-10T15:00:00Z'),id_staff: new DBRef('staff',steve._id),id_customer: new DBRef('customer',olena._id),id_dish: new DBRef('dish',apple_pie._id)})
> db.order.save(order01)

```
_id: ObjectId("5eb9a7c1310bc535f853bfb2")
date_time: 2020-05-10T15:00:00.000+00:00
id_staff: DBRef(staff, 5eb9a4e4310bc535f853bfaf, undefined)
id_customer: DBRef(customer, 5eb9a468310bc535f853bfae, undefined)
id_dish: DBRef(dish, 5eb9a70a310bc535f853bfb1, undefined)
```

- Оновимо поле cus_first_name в колекції customer командою:

> db.customer.updateOne({cus_first_name: 'Ben'}, {$set: {cus_first_name: 'Ben-Mark'}})

```
_id: ObjectId("5eb99baa310bc535f853bfa9")
cus_first_name: "Ben-Mark"
cus_sec_name: "Heine"
phone: 2763454
email: "bheine@gmail.com"
```


- Додамо нове поле до колекції командою:

>db.customer.updateOne({cus_first_name: 'Ben-Mark'}, {$set: {country_birth: 'USA'}})


```
_id: ObjectId("5eb99baa310bc535f853bfa9")
cus_first_name: "Ben-Mark"
cus_sec_name: "Heine"
phone: 2763454
email: "bheine@gmail.com"
country_birth: "USA"
```


- Видалимо створене поле з колекції командою:

>db.customer.updateOne({}, {$unset: {country_birth: 'USA'}})


```
_id: ObjectId("5eb99baa310bc535f853bfa9")
cus_first_name: "Ben-Mark"
cus_sec_name: "Heine"
phone: 2763454
email: "bheine@gmail.com"
```


Висновок: на лабораторній роботі я здобула практичних навичок створення та обробки бази даних типуNoSQLнаприкладі 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».