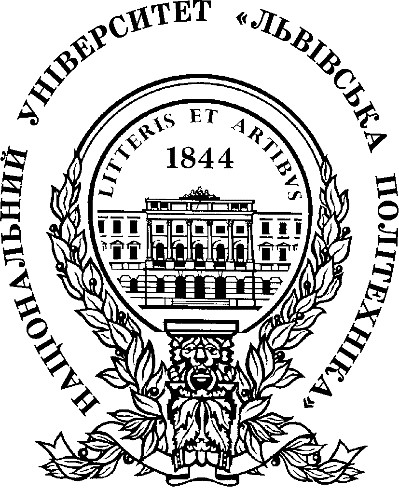
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту



ЗВІТ № 14 **з курсу “ОБДЗ”** на тему:

# «Розробка бази даних типу NoSQL»

**Виконав:**

студент групи КН-208

Фіняк М.В.

**Викладач:**

Якимишин Х.М.

# Лабораторна робота № 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 (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 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"]})

Запит до вкладених об’єктів

> db.persons.insert({"name": "Alex", "age": "28", company:

{"name":"microsoft", "country":"USA"}})

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)

> db.persons.find({"company.name": "micriosoft"})

**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.ingredient.insertMany("name"="Mikl","price\_for\_unit":"32.25", "quantity":"30l",supplier\_id: )

> 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"]}})

***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()

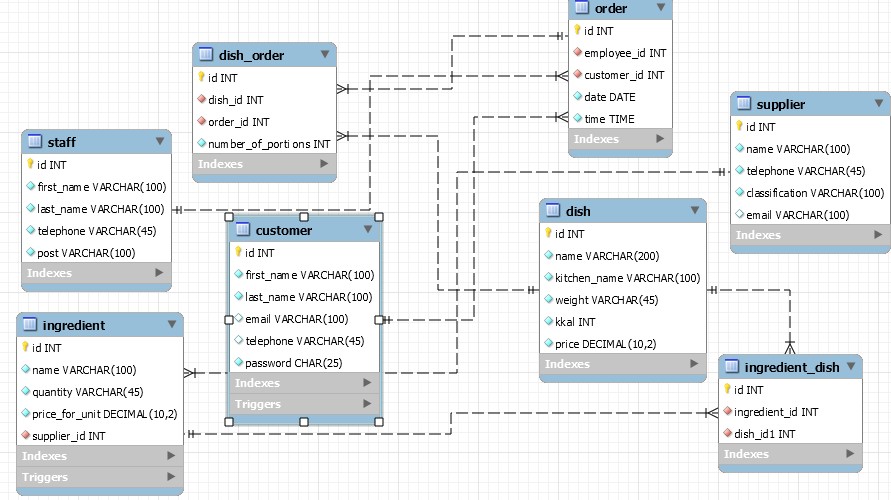
> db.persons.dropIndex("name\_1")

# 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 вилучення інформації в «сутність».

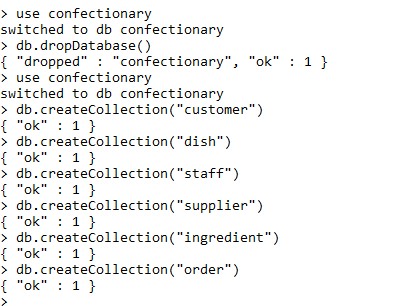
# Хід роботи

***Схема баз даних в MySQL***

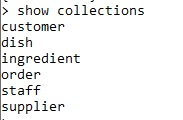


***Схема баз даних в Mongodb***

1. Створюємо базу даних командою *use restaurant* та відповідні колекції командою *db.createCollection("назва\_колекції").*

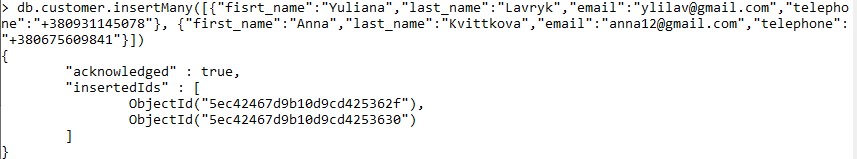


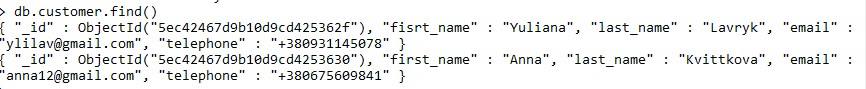
1. Перевіряємо наявні колекції за допомогою команди show collections.



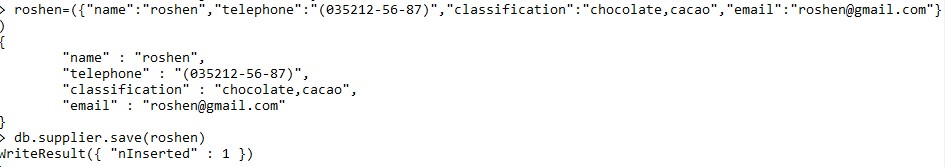
1. Заповнюємо колекцію supplier даними за допомогою команди db.назва\_колекції.insertOne() та колекцію customer за допомогою команди db.назва\_колекції.insertMany() . Перевіряємо чи дані внесені командою db.назва\_колекції.find().



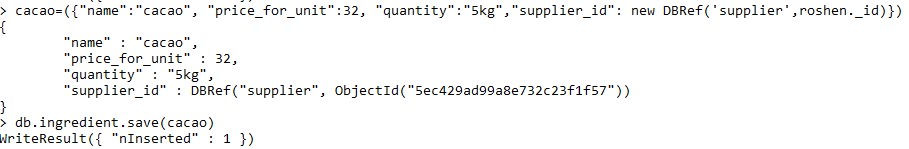




1. Визначаємо новий документ і вносимо його в колекцію supplier за допомогою команд назва\_документа=({...}) та db.назва\_колекції.save(назва\_документа).

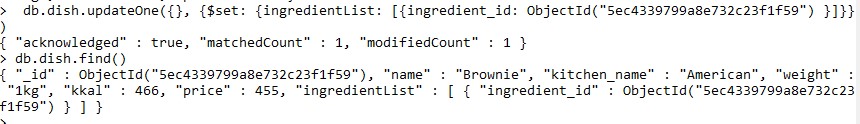


1. Визначаємо новий документ для таблиці ingredient та проводимо автоматичне зв’язування створеного документа та документа колекції supplier, використовуючи опцію new DBRef().



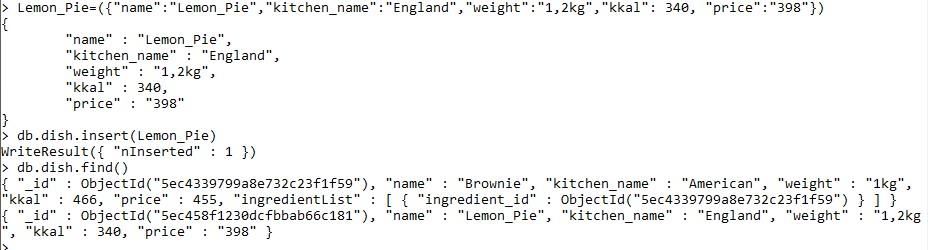
1. Додаємо нове поле в колекцію dish, яке буде зберігати значення індексів інгредієнтів (колекція ingredient), які входять до складу страви (колекція dish) за допомогою команди db.назва\_колекції.updateOne() та заповнимо його.

За допомогою команди db.назва\_колекції.find() перевіримо наявність нового поля в колекції dish.



1. Додаємо дані в інші колекції .

**Колекція dish:**



**Колекція staff:**



**Колекція оrder:**



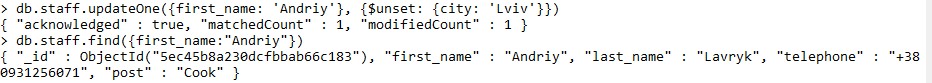
1. Оновимо поле first\_name в колекції staff командою db.назва\_колекції.updateOne() та перевіримо виконання оновлення за допомогою команди db.назва\_колекції.findOne().



1. Додамо нове поле до колекції staff .



Перевіряємо :



1. Видалимо створене поле з колекції staff та зробимо перевірку.

# Контрольні запитання

1. Назвати основні типи баз даних NoSQL.
   * + Бази 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;
  + Труднощі швидкого переходу з однієї нереляційної бази даних в іншу;
  + Обмежена ємність вбудованої мови запитів.

1. Надати характеристику СУБД MongoDB.

***MongoDB*** — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

1. Операції вставки даних. insert() або save()
2. Операції оновлення даних. update() , save() 6. Операції знищення даних. update(), remove()
3. Умовні оператори.
   * $gt (більше ніж)
   * $lt (менше ніж)
   * $gte (більше чи рівно)
   * $lte (менше чи рівно)
4. Операції керування індексами. db.system.indexes.find() db.name.dropIndex("name")
5. 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:** під час виконання даної лабораторної роботи я здобув практичні навички створення та редагування бази даних типуNoSQLнаприкладі СУБД MongoDB. Розробив схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB, і

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