# МІНІСТЕРСТВО ОСВІТИ І НАУКИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



## Лабораторна робота №14

## З дисципліни «Організація баз даних та знань»

***Виконав:***

*студент групи КН-210*

*Кухар Ярослав*

***Перевірив:***

*Кандидат тех. наук, ст. викладач Мельникова Н. І.*

Львів – 2020

**Тема:** Розробка бази даних типу NoSQL.

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

**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)

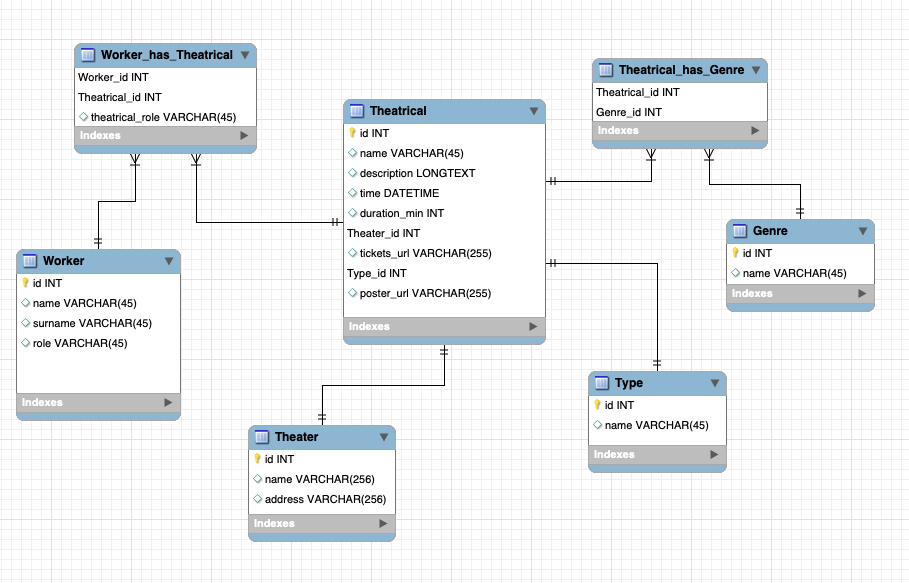
> db.persons.find().sort({name: 1})

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

**Хід роботи**

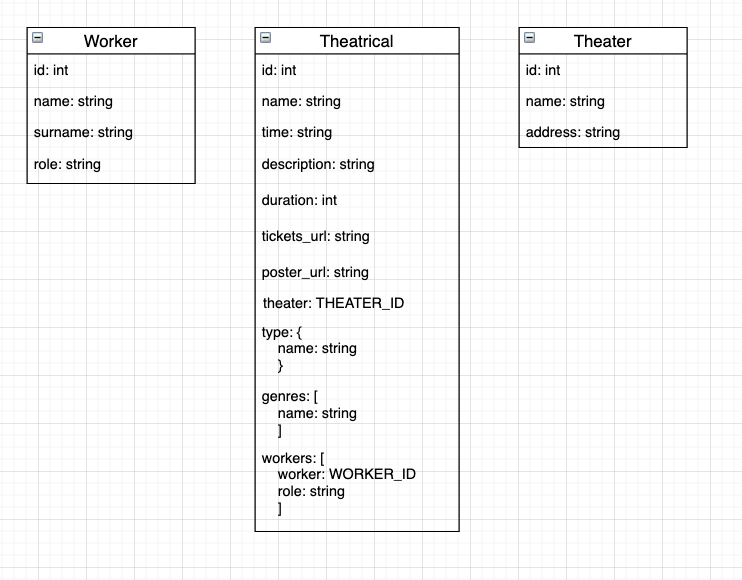
1. Розробити схему бази даних на основі предметної області з лабораторної роботи No1 у спосіб, що застосовується в СУБД MongoDB..
2. Перетворити сутності діаграми БД, розробленої для лабораторної роботи No1, у структури, прийнятні для обробки в MongoDB.
3. Забезпечити реалізацію функцій редагування, додавання та вилучення інформації в «сутність».

ER-Diagram зараз виглядає наступним чином:

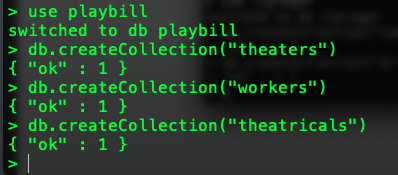


На основі цієї діаграми я будував вбудовану модель даних для MongoDB (Embedded Data Models).

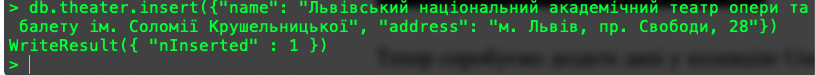
При побудові моделі MongoDB я використовував комбінований підхід до проектування. Це означає, що я використовую і вкладені об’єкти в одному документі, і поділ на декілька колекцій та посилання через ID.



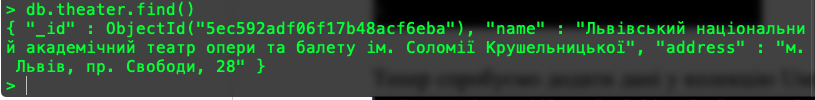
Запустимо MongoDB та створимо базу даних playbill та додамо до неї 3 колекції: Theaters, Theatricals, Workers.



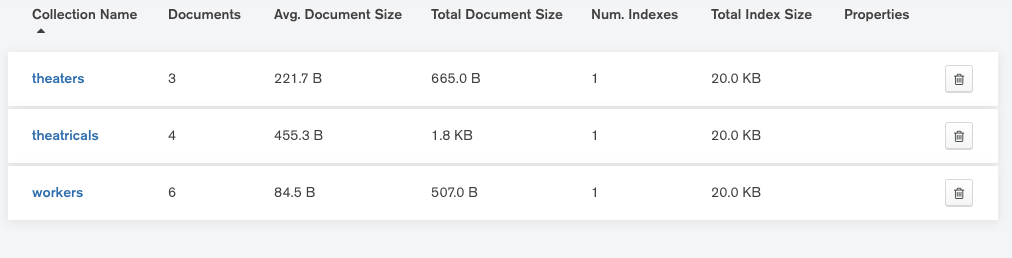
Тепер спробуємо додати дані у колекцію Theaters:



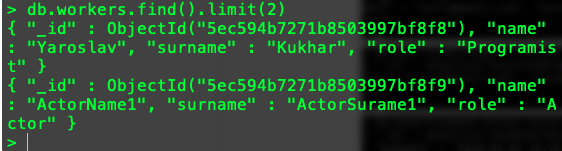
Перевіримо, чи дані додались:



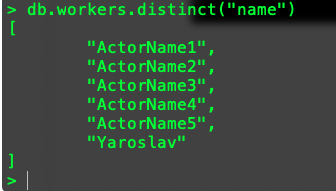
Дані додано успішно. За допомогою утиліти MongoDB Compass додаємо решту даних до бази даних.



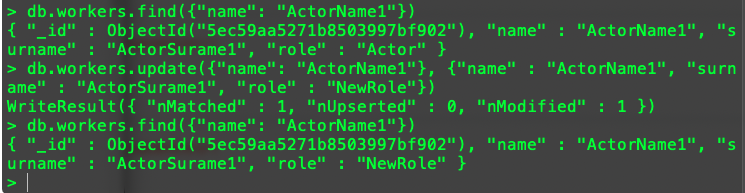
Тепер виконаємо кілька простих запитів:



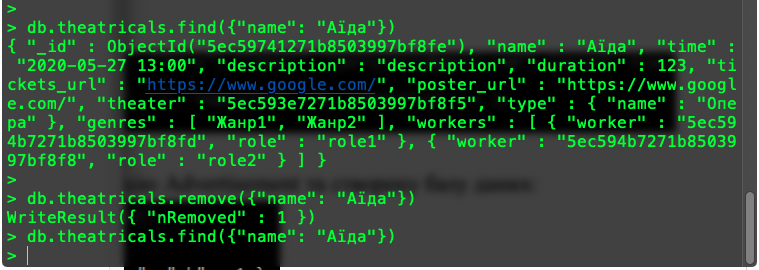




Оновлення запису:



Видалення запису



Можемо знищити колекцію : db.Колекція.drop() або базу даних : db.dropDatabase()

**Висновок**: я здобув практичні навички створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB.