

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра «Системи штучного інтелекту»



Лабораторна Робота №14
З предмету: «Організація баз даних та знань»

Виконав
студент групи КН-211
Турик Олександр
Прийняла :
Якимішин Х.М..

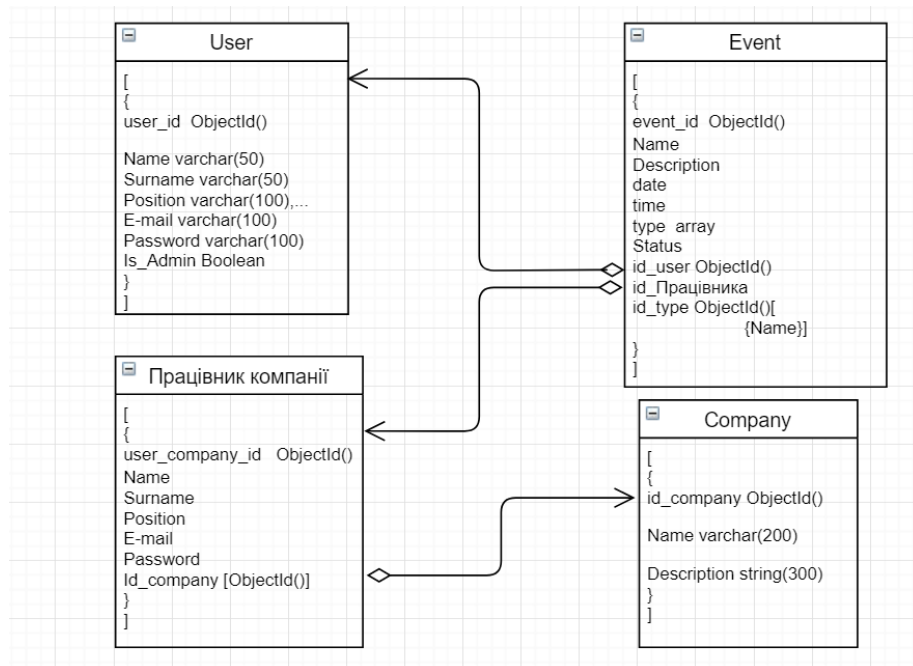
Львів-2020

Тема: “Розробка бази даних типу NoSQL”

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

Хід роботи

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



2. Створимо базу даних і заповнимо її даними:

1)Створення бази даних «db»

```
> use db
switched to db db
> ■
```

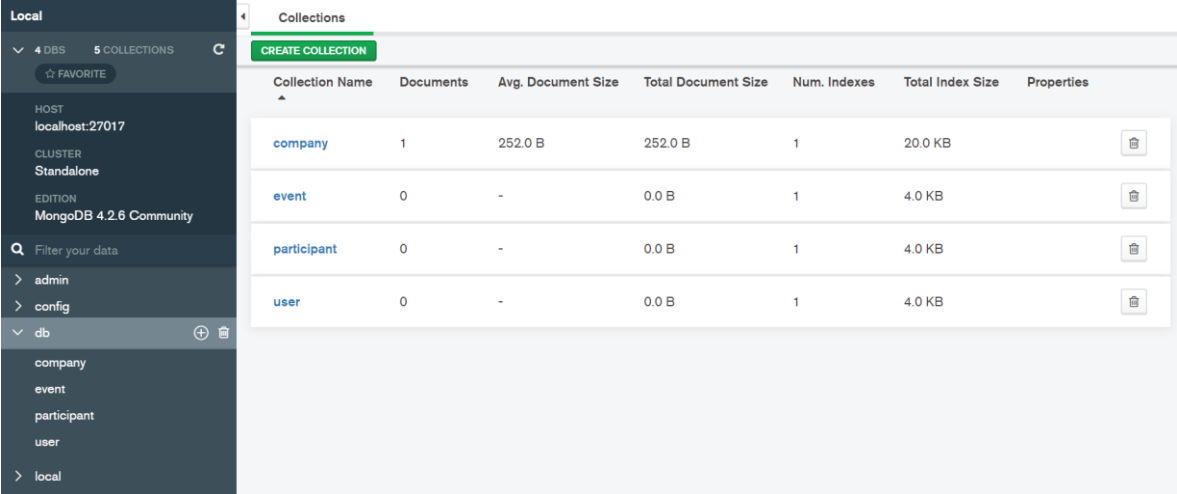
2)Створимо наші колекції

```
> db.createCollection("user")
{ "ok" : 1 }
> db.createCollection("event")
{ "ok" : 1 }
> db.createCollection("company")
{ "ok" : 1 }
> db.createCollection("participant")
{ "ok" : 1 }
```

3)Додамо дані до однієї з колекцій

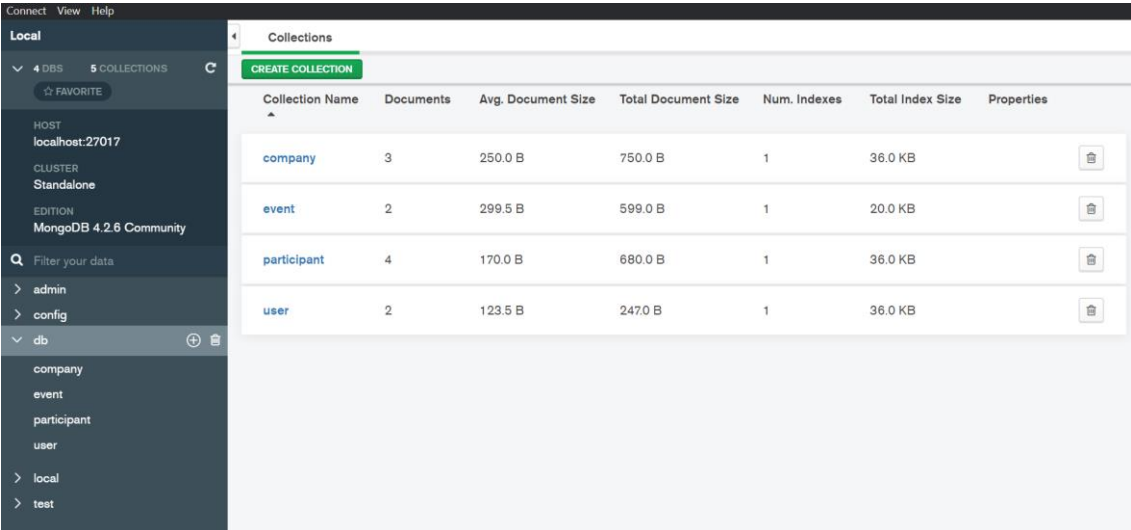
```
> db.company.insertOne({
... _id:1,
... "name":"Softserve",
... "Description":"SoftServe є однією з найбільших компаній-розробників програмного забезпечення у Центральній та Східній Європі"})
{ "acknowledged" : true, "insertedId" : 1 }
> ■
```

4)Перевіримо зміни у програмі MongoDB Compass

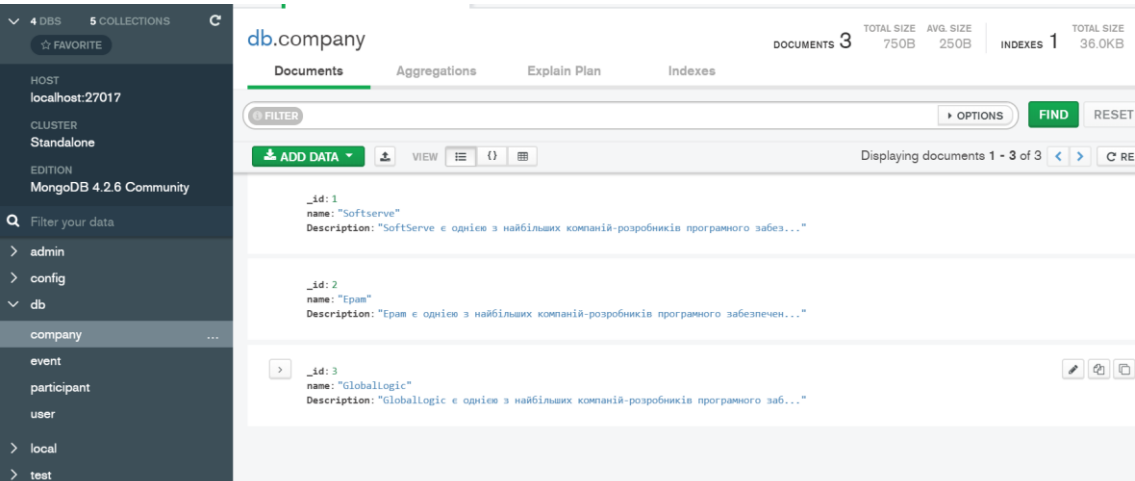


Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
company	1	252.0 B	252.0 B	1	20.0 KB	
event	0	-	0.0 B	1	4.0 KB	
participant	0	-	0.0 B	1	4.0 KB	
user	0	-	0.0 B	1	4.0 KB	

Заповнивши у кожну таблицю дані, отримаємо такий вигляд колекцій:



Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
company	3	250.0 B	750.0 B	1	36.0 KB	
event	2	299.5 B	599.0 B	1	20.0 KB	
participant	4	170.0 B	680.0 B	1	36.0 KB	
user	2	123.5 B	247.0 B	1	36.0 KB	

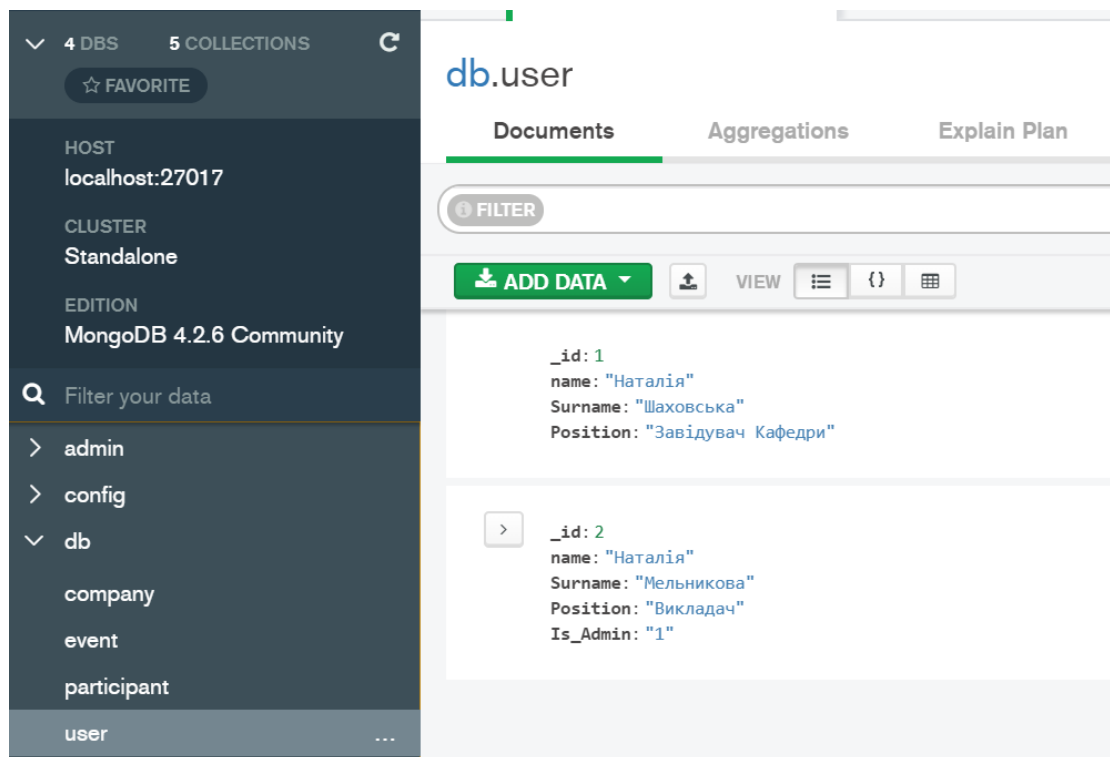
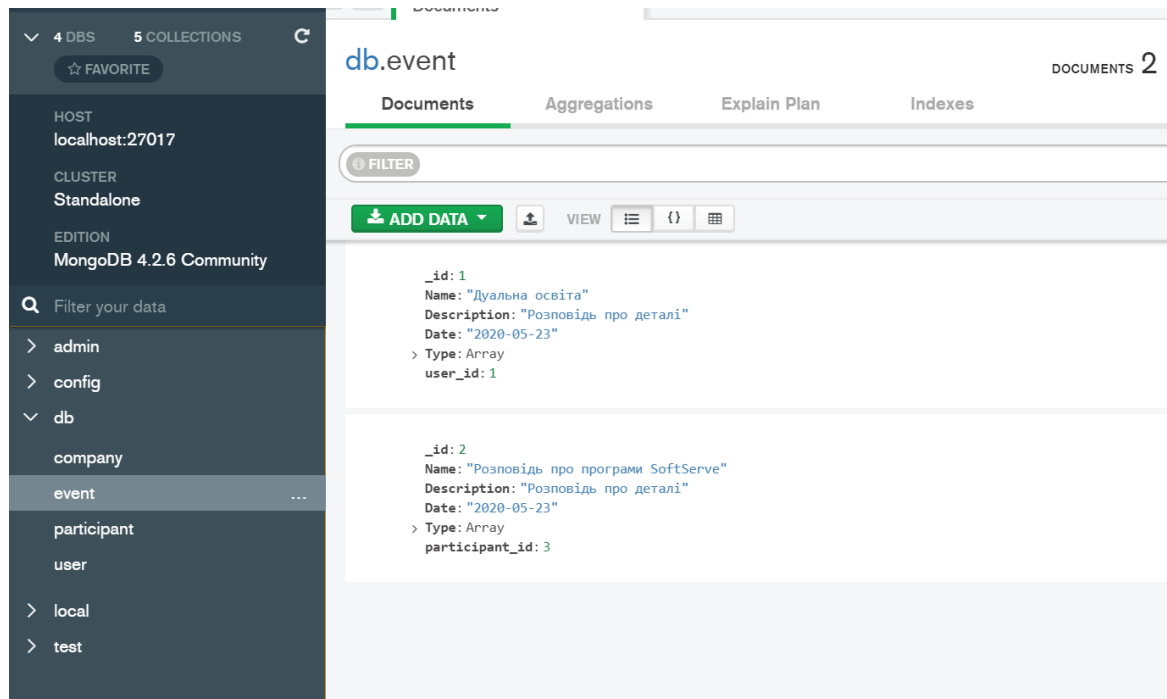


db.company DOCUMENTS 3 TOTAL SIZE 750B AVG. SIZE 250B INDEXES 1 TOTAL SIZE 36.0KB

Documents Aggregations Explain Plan Indexes

Displaying documents 1 - 3 of 3

<input type="checkbox"/>	<pre>{ "_id": 1, "name": "Softserve", "Description": "SoftServe є однією з найбільших компаній-розробників програмного забезпечення у Центральній та Східній Європі"} </pre>
<input type="checkbox"/>	<pre>{ "_id": 2, "name": "Ерам", "Description": "Ерам є однією з найбільших компаній-розробників програмного забезпечення у Центральній та Східній Європі"} </pre>
<input type="checkbox"/>	<pre>{ "_id": 3, "name": "Globallogic", "Description": "Globallogic є однією з найбільших компаній-розробників програмного забезпечення у Центральній та Східній Європі"} </pre>



3. Забезпечити реалізацію функцій редагування, додавання та видалення інформації в «сутність».

1) Додавання

```
db.user.insertOne({
  _id: 3,
  "name": "Роман",
  "Surname": "Гасько",
  "Position": "Викладач",
  "Is_Admin": 0,
  "Mail": "r.t.gasko@gmail.com"
})
```

```

> db.user.insertOne({
...   _id:3,
...   "name":"Роман",
...   "Surname":"Гасько",
...   "Position":"Викладач",
...   "Is_Admin":0,
...   "Mail":"r.t.gasko@gmail.com"
... })
{ "acknowledged" : true, "insertedId" : 3 }

```

db.user

Documents Aggregations Exp

FILTER

ADD DATA

VIEW

Filter your data

- admin
- config
- db
 - company
 - event
 - participant
 - user**
- local
- test

Document 1:

```

_id: 1
name: "Наталія"
Surname: "Шаховська"
Position: "Завідувач Кафедри"

```

Document 2:

```

_id: 2
name: "Наталія"
Surname: "Мельникова"
Position: "Викладач"
Is_Admin: "1"

```

Document 3:

```

_id: 3
name: "Роман"
Surname: "Гасько"
Position: "Викладач"
Is_Admin: 0
Mail: "r.t.gasko@gmail.com"

```

2) Оновлення документу

```

db.user.updateOne({_id:1},{ $set: {"Is_Admin":1}}, {upsert:true} )
> db.user.updateOne({_id:1},{ $set: {"Is_Admin":1}}, {upsert:true} )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

```

db.user

Documents Aggregations Explain Plan Indexes

FILTER

ADD DATA

VIEW

Filter your data

- admin
- config
- db
 - company
 - event
 - participant
 - user**
- local
- test

Document 1:

```

_id: 1
name: "Наталія"
Surname: "Шаховська"
Position: "Завідувач Кафедри"
Is_Admin: 1

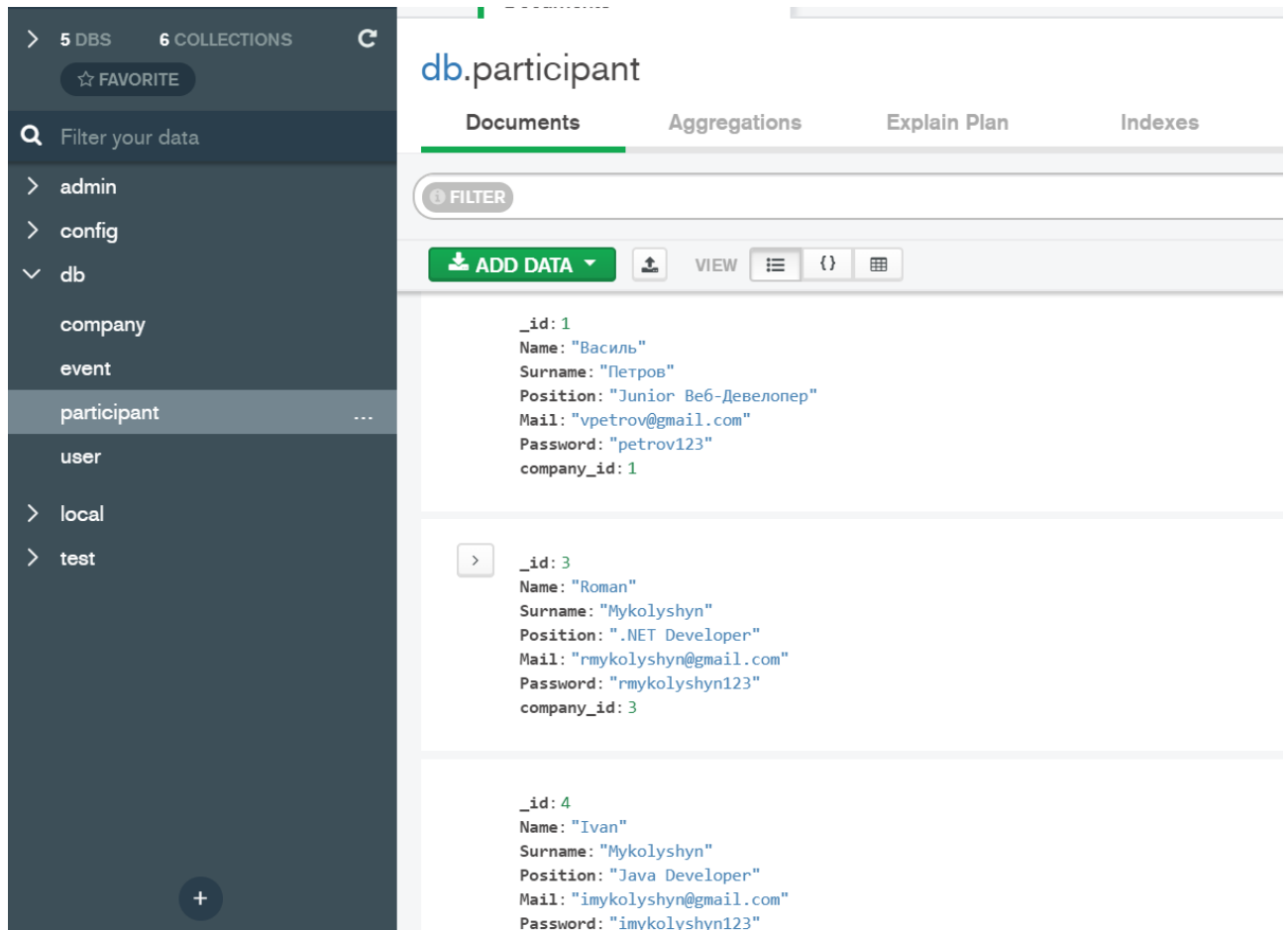
```

Як можна побачити, додалося значення Is_Admin до документу, id якого=1.

3)Видалення даних

```
db.participant.remove({'_id':2})
```

```
> db.participant.remove({'_id':2})
WriteResult({ "nRemoved" : 1 })
> ■
```



Бачимо, що відбулося видалення елемента у якого _id=2.

Відповіді на контрольні питання

1. Назвати основні типи баз даних NoSQL

Бази даних можна розділити на такі типи:

- База даних ключ-значення,
- Графова база даних
- Об'єктно-орієнтована
- Документно-орієнтована

2. Назвати переваги та недоліки використання баз даних NoSQL.

- Гнучкість роботи з даними.
- Висока продуктивність при виконанні простих запитів.

- Можна додавати поля або колонки для певних таблиць які не мають бути в інших без зміни структури.
 - Немає обмежень на типи даних, що дозволяє зберігати великий обсяг неструктурованої інформації.
 -
- Основні недоліки:
- Гірша зв'язність даних
 - При неправильному написі поля/колекції у MongoDB створить нове замість відображення помилки.
 - Кожна база даних може мати свій синтаксис що може погіршувати роботу з нею

3. Надати характеристику СУБД MongoDB.

- Надає всі переваги і недоліки нереляційних бд.
- Без чіткої структури можуть виникати непорозуміння щодо роботи з бд.
- Підтримує зберігання даних у JSON форматі
- Має гнучку мову для формування запитів

4. Операції вставки даних.

```
db.collection.insert/insertOne/insertMany({}/{},{},{},{},{})
```

collection – ім'я колекції.

```
db.collection.update( { }, { $set/$pop/$pull/$push/$addToSet: { } } )
```

5.Операції оновлення даних.

```
db.collection.save({ })
```

```
db.collection.update({ },{ })
```

6.Операції знищення даних.

```
db.persons.update({ } { $unset: { } })
```

```
db.persons.remove({ })
```

```
db.persons.drop()
```

```
db.dropDatabase()
```

7. Умовні оператори.

\$eq - співпадає

\$ne – не співпадає

\$gt – більше ніж

\$lt – менше ніж

\$gte – більше рівне

\$lte – менше рівне

\$in – масив значень, одне з яких мусить містити поле документа

\$nin – масив значень які мусить не бути в полі документа

8. Операції керування індексами.

```
db.collection.ensureIndex({ })
```

```
db.system.indexes.find()
```

```
db.collection.dropIndex();
```

collection – назва колекції.

9. Пошук даних.

```
db.persons.find({ { } })
```

10. Можливості документних БД.

- ❖ Узгодження даних
- ❖ Транзакції
- ❖ Доступність
- ❖ Можливості запитів
- ❖ Масштабування
- ❖ Гнучкість

Висновок: на даній лабораторній роботі я ознайомився з MongoDB, написав у командному клієнті запити для створення бд, заповнив її даними, пробував виконати основні запити та ознайомився з принципом документо-орієнтованої бази даних.

