

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



**Лабораторна робота №14**  
з курсу “Організація баз даних та знань”

**Виконала:**

студентка групи КН-208

Ріжко Марія

**Перевірила:**

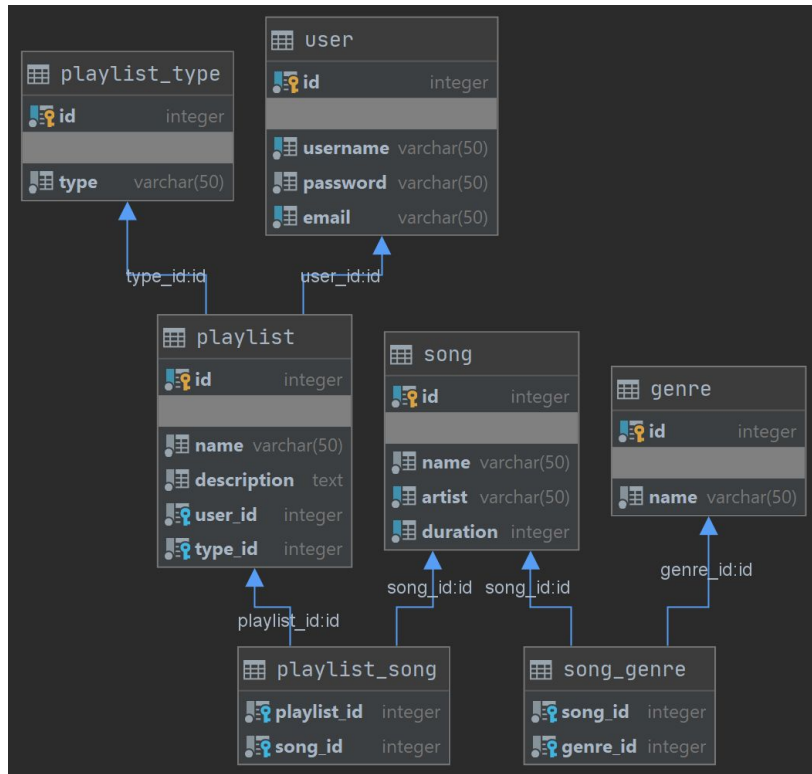
Якимишин Х.М.

Львів 2020 р.

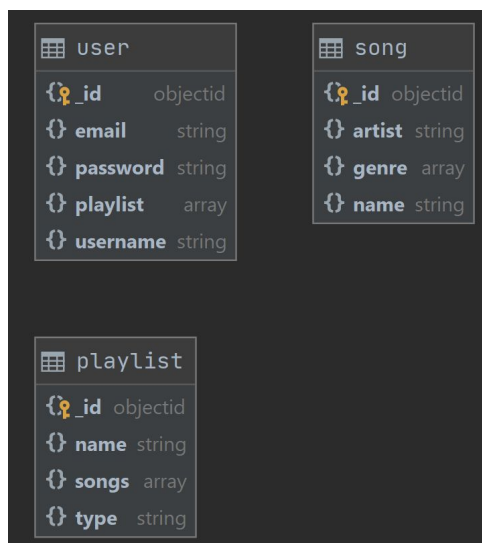
## Мета роботи:

Здобуття практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB.

SQL схема бд:

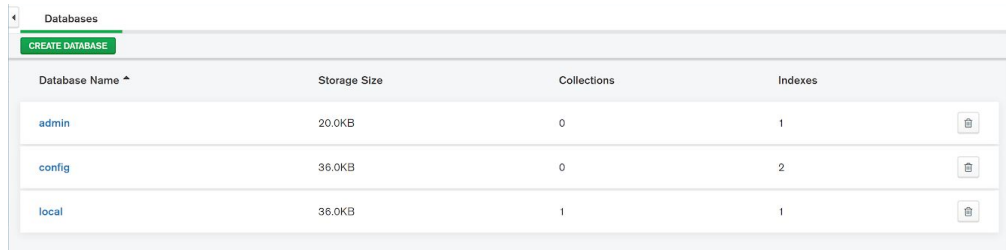


NoSQL схема бд:



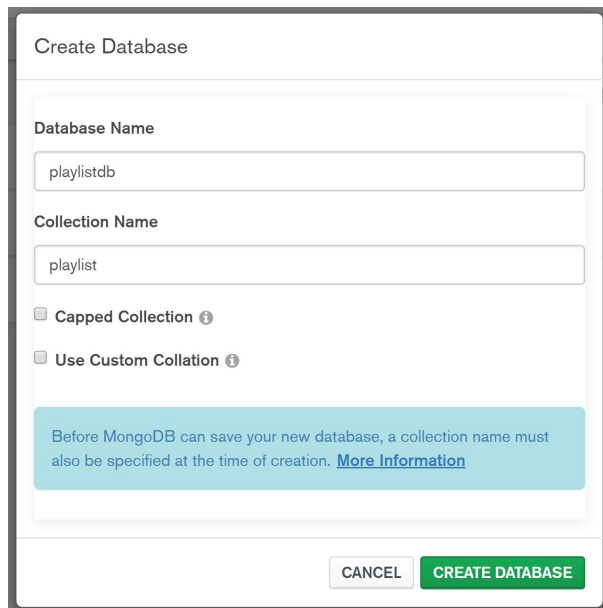
Створимо нову базу даних.

Список баз до створення:



Database Name ^	Storage Size	Collections	Indexes	
admin	20.0KB	0	1	
config	36.0KB	0	2	
local	36.0KB	1	1	

Створюємо нову бд playlistdb і її першу колекцію playlist:



Create Database

Database Name

playlistdb

Collection Name

playlist

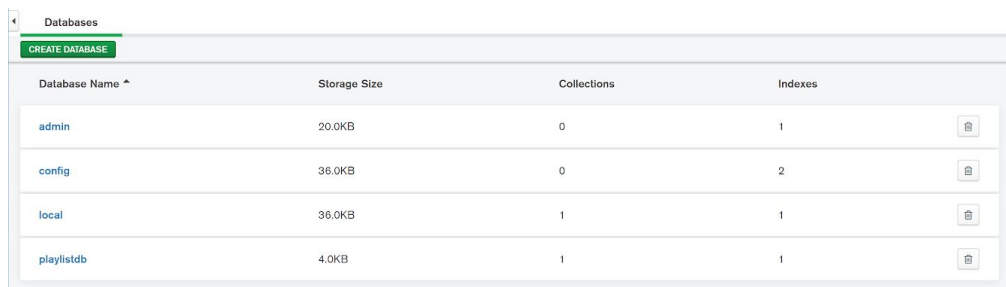
☐ Capped Collection ⓘ

☐ Use Custom Collation ⓘ

Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

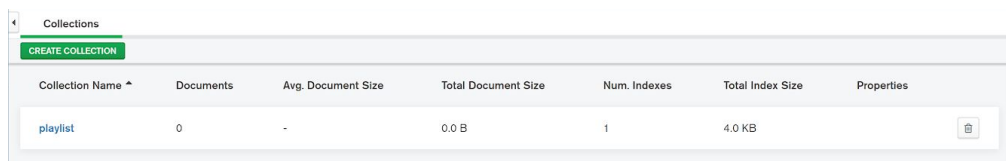
CANCEL CREATE DATABASE

В результаті маємо такі бд:



Database Name ^	Storage Size	Collections	Indexes	
admin	20.0KB	0	1	
config	36.0KB	0	2	
local	36.0KB	1	1	
playlistdb	4.0KB	1	1	

Відкриваємо playlistdb і бачимо всередині єдину колекцію playlist:



Collection Name ^	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
playlist	0	-	0.0 B	1	4.0 KB	

Створимо ще 2 колекції song і user:

Create Collection

Collection Name

song

☐ Capped Collection ⓘ  
☐ Use Custom Collation ⓘ

CANCEL

CREATE COLLECTION

Create Collection

Collection Name

user

☐ Capped Collection ⓘ  
☐ Use Custom Collation ⓘ

CANCEL

CREATE COLLECTION

В результаті отримуємо список з 3 колекцій:

Collections						
CREATE COLLECTION						
Collection Name ^	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
playlist	0	-	0.0 B	1	4.0 KB	
song	0	-	0.0 B	1	4.0 KB	
user	0	-	0.0 B	1	4.0 KB	

Наповнимо колекції даними.

Вибираємо колекцію song

playlistdb.song

DOCUMENTS 0

TOTAL SIZE 0B

AVG. SIZE 0B

INDEXES 1

TOTAL SIZE 4.0KB

AVG. SIZE 4.0KB

Documents

Aggregations

Explain Plan

Indexes

FILTER

OPTIONS

FIND

RESET

...

ADD DATA

VIEW

VIEW

VIEW

VIEW

Displaying documents 0 - 0 of N/A

REFRESH

І створюємо нову сутність:

Insert to Collection playlistdb.song

VIEW

{}

VIEW

```
1 { "artist": "artist1", "genre": "genre1", "name": "name1" }
2
```

CANCEL

INSERT

Або декілька за раз:

Insert to Collection playlistdb.song

VIEW { } ≡

```
1 [
2   { "artist": "artist1", "genre": "genre1", "name": "name2" },
3   { "artist": "artist1", "name": "name3" },
4   { "artist": "artist2", "genre": "genre2", "name": "name4" },
5   { "artist": "artist2", "genre": "genre1", "name": "name5" },
6   { "genre": "genre1", "name": "name6" },
7   { "artist": "artist1", "genre": "genre1", "name": "name7" },
8   { "artist": "artist3", "genre": ["genre1", "genre3", "genre4"], "name": "name8" },
9   { "artist": "artist1", "genre": "genre1", "name": "name9" },
10  { "artist": ["artist1", "artist3"], "genre": "genre1", "name": "name10" }
11 ]
```

CANCEL INSERT

В результаті отримуємо колекцію із 10 документами:

playlistdb.song

DOCUMENTS 10 TOTAL SIZE 782B AVG. SIZE 78B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Documents Aggregations Explain Plan Indexes

3 FILTER OPTIONS FIND RESET ...

ADD DATA VIEW ≡ { } ≡

Displaying documents 1 - 10 of 10 REFRESH

```
{
  "_id": {
    "$oid": "5ec4f15c84935121ccc230f9"
  },
  "artist": "artist1",
  "genre": "genre1",
  "name": "name1"
}

{
  "_id": {
    "$oid": "5ec4f1fb84935121ccc230fa"
  },
  "artist": "artist1",
  "genre": "genre1",
  "name": "name2"
}

{
  "_id": {
    "$oid": "5ec4f1fb84935121ccc230fb"
  },
  "artist": "artist1",
  "name": "name3"
}

{
  "_id": {
    "$oid": "5ec4f1fb84935121ccc230fc"
  },
  "artist": "artist2",
  "genre": "genre2",
  "name": "name4"
}

{
  "_id": {
    "$oid": "5ec4f1fb84935121ccc230fd"
  },
  "artist": "artist3",
  "genre": "genre3",
  "name": "name5"
}
```

Аналогічно заповнимо playlist та user.

Insert to Collection playlistdb.playlist

VIEW

```
1 [
2   { "name": "playlist1", "type": "public"},
3   { "name": "playlist2", "type": "private"},
4   { "name": "playlist3", "type": "private"},
5   { "name": "playlist4", "type": "public"},
6   { "name": "playlist5", "type": "public"}
7 ]
```

Insert to Collection playlistdb.user

VIEW

```
1 [
2   { "email": "ex@gmail.com", "password": "1234567890", "username": "MeriDK" },
3   { "email": "user@gmail.com", "password": "qwerty", "username": "topuser" }
4 ]
```

Маємо колекцію playlist з 5 документами:

Documents	Aggregations	Explain Plan	Indexes
FILTER			
ADD DATA VIEW			
Displaying documents 1 - 5 of 5			
<pre>{ "_id": ObjectId("Sec4f30684935121ccc23103"),   "name": "playlist1",   "type": "public" }</pre>			
<pre>{ "_id": ObjectId("Sec4f30684935121ccc23104"),   "name": "playlist2",   "type": "private" }</pre>			
<pre>{ "_id": ObjectId("Sec4f30684935121ccc23105"),   "name": "playlist3",   "type": "private" }</pre>			
<pre>{ "_id": ObjectId("Sec4f30684935121ccc23106"),   "name": "playlist4",   "type": "public" }</pre>			
<pre>{ "_id": ObjectId("Sec4f30684935121ccc23107"),   "name": "playlist5",   "type": "public" }</pre>			

Та колекцію юзерів із 2.

Documents	Aggregations	Explain Plan	Indexes
FILTER			
ADD DATA VIEW			
Displaying documents 1 - 2 of 2			
<pre>{ "_id": ObjectId("Sec4f35c84935121ccc23108"),   "email": "ex@gmail.com",   "password": "1234567890",   "username": "MerIDK" }</pre>			
<pre>{ "_id": ObjectId("Sec4f35c84935121ccc23109"),   "email": "user@gmail.com",   "password": "qwerty",   "username": "topuser" }</pre>			

Змінимо юзернейм користувача topuser на prostouser.

Вибираємо потрібний документ:

>	<pre>{ "_id": ObjectId("Sec4f35c84935121ccc23109"),   "email": "user@gmail.com",   "password": "qwerty",   "username": "topuser" }</pre>	   
---	--	---

Вибираємо Редагувати:

1	<pre>_id: ObjectId("Sec4f35c84935121ccc23109")</pre>	ObjectId
2	<pre>email : "user@gmail.com"</pre>	String
3	<pre>password : "qwerty"</pre>	String
4	<pre>username : "topuser"</pre>	String
CANCEL UPDATE		

Вносимо зміни:

1	<pre>_id: ObjectId("Sec4f35c84935121ccc23109")</pre>	ObjectId
2	<pre>email : "user@gmail.com"</pre>	String
3	<pre>password : "qwerty"</pre>	String
4	<pre>username : "prostouser"</pre>	String
Document Modified. CANCEL UPDATE		

Зберігаємо:

```
{
  "_id": "ObjectId(\"Sec4f30684935121ccc23109\")",
  "email": "user@gmail.com",
  "password": "qwerty",
  "username": "prostouser"
}
```

Видалимо плейлист з назвою playlist5.  
Колекція до видалення:

playlistdb.playlist

DOCUMENTS 5    TOTAL SIZE 297B    AVG. SIZE 59B    INDEXES 1    TOTAL SIZE 20.0KB    AVG. SIZE 20.0KB

Documents   Aggregations   Explain Plan   Indexes

FILTER

ADD DATA   VIEW   [List Icon]   [JSON Icon]   [Table Icon]

Displaying documents 1 - 5 of 5   [Previous] [Next]   C REFRESH

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23103\")", "name": "playlist1", "type": "public" }

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23104\")", "name": "playlist2", "type": "private" }

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23105\")", "name": "playlist3", "type": "private" }

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23106\")", "name": "playlist4", "type": "public" }

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23107\")", "name": "playlist5", "type": "public" }

Плейлист до видалення:

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23107\")", "name": "playlist5", "type": "public" }

[Edit] [Share] [Copy] [Delete]

Вибираємо Видалити і підтверджуємо:

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23107\")", "name": "playlist5", "type": "public" }

Document Flagged For Deletion.    CANCEL    DELETE

Колекція плейлистів після видалення:

playlistdb.playlist

DOCUMENTS 4    TOTAL SIZE 238B    AVG. SIZE 60B    INDEXES 1    TOTAL SIZE 20.0KB    AVG. SIZE 20.0KB

Documents   Aggregations   Explain Plan   Indexes

FILTER

ADD DATA   VIEW   [List Icon]   [JSON Icon]   [Table Icon]

Displaying documents 1 - 4 of 4   [Previous] [Next]   C REFRESH

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23103\")", "name": "playlist1", "type": "public" }

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23104\")", "name": "playlist2", "type": "private" }

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23105\")", "name": "playlist3", "type": "private" }

{ "\_id": "ObjectId(\"Sec4f30684935121ccc23106\")", "name": "playlist4", "type": "public" }



Виберемо пісні, які мають жанр genre1.

Встановимо фільтр { "genre": "genre1" } та отримаємо список з 8 документів потрібних нам пісень:

The screenshot shows the MongoDB Compass interface. At the top, the database is 'playlistdb' and the collection is 'song'. The filter bar shows the query: { "genre": "genre1" }. The results pane displays 8 documents. The first document is expanded, showing its structure: { "\_id": { "\$oid": "5ec4f15c84935121ccc230f9" }, "artist": "artist1", "genre": "genre1", "name": "name1" }. The other documents follow a similar pattern with different IDs and names.

При чому пошук відбувається не тільки для пісень, в яких це єдиний жанр.

```
{
  "_id": {
    "$oid": "5ec4f15c84935121ccc230f9"
  },
  "artist": "artist1",
  "genre": "genre1",
  "name": "name1"
}
```

Але й для тих, в яких цей жанр є один із списку.

```
{
  "_id": {
    "$oid": "5ec4f1fb84935121ccc23100"
  },
  "artist": "artist3",
  "genre": ["genre1", "genre3", "genre4"],
  "name": "name3"
}
```

Порахуємо час виконання запитів за допомогою EXPLAIN

Для цього експортуємо дані з SQL таблиці:

```
postgres_public_son...
File Edit Format View Help
20140,song_name20,artist5,240
20141,song_name21,artist6,220
20142,song_name22,artist6,210
20143,song_name23,artist6,215
20144,song_name24,artist6,120
20145,song_name25,artist6,130
20146,song_name26,artist6,140
20147,song_name27,artist7,150
20148,song_name28,artist7,160
20149,song_name29,artist7,170
20150,song_name30,artist7,180
20151,song_name31,artist7,190
20152,song_name32,artist7,154
20153,song_name33,artist7,173
20154,song_name34,artist7,193
20155,song_name35,artist8,145
20156,song_name36,artist8,139
20157,song_name37,artist8,102
20158,song_name38,artist9,156
20159,song_name39,artist9,166
20160,song_name40,artist10,165
100% Unix (LF) UTF-8
```

І загрузимо її в нову базу даних.

ADD DATA => Import File =>

Import To Collection playlistdb.song

Select File

X:\projects\songdb\postgres\_public\_song.csv

BROWSE

Select Input File Type

JSON

CSV

Options

Select delimiter

COMMA

☒ Ignore empty strings

☐ Stop on errors

Specify Fields and Types

	<input checked="" type="checkbox"/> 1 String	<input checked="" type="checkbox"/> 120 String	<input checked="" type="checkbox"/> song_name1 String	<input checked="" type="checkbox"/> artist1 String
1	2	130	song_name2	artist1
2	3	140	song_name3	artist1
3	4	150	song_name4	artist1
4	5	160	song_name5	artist1
5	6	170	song_name6	artist1
6	7	180	song_name7	artist1
7	8	125	song_name8	artist2
8	9	135	song_name9	artist2
9	10	145	song_name10	artist2
10	11	155	song_name11	artist3

CANCEL

IMPORT

Import completed

20,159 (100%)

DONE

playlistdb.song

Documents

Aggregations

Explain Plan

Indexes

DOCUMENTS 20.2k

TOTAL SIZE 1.9MB

AVG. SIZE 96B

INDEXES 1

TOTAL SIZE 36.0KB

AVG. SIZE 36.0KB

Зробимо пошук пісень з часом меншим за 120 с використовуючи функцію EXPLAIN

EXPLAIN

VIEW DETAILS AS VISUAL TREE RAW JSON

Query Performance Summary

Documents Returned: 2016	Actual Query Execution Time (ms): 12
Index Keys Examined: 0	Sorted in Memory: no
Documents Examined: 20169	No index available for this query.

Створимо новий індекс. Список індексів до:

playlistdb.song

Documents

Aggregations

Explain Plan

Indexes

DOCUMENTS 20.2k

TOTAL SIZE 1.9MB

AVG. SIZE 96B

INDEXES 1

TOTAL SIZE 36.0KB

AVG. SIZE 36.0KB

CREATE INDEX					
Name and Definition ^	Type	Size	Usage	Properties	Drop
<div><div>_id_</div><div><div>_id_</div></div></div>	REGULAR ⓘ	36.9 KB	6 since Wed May 20 2020	UNIQUE ⓘ	

Створюємо індекс

Create Index

Choose an index name

duration\_idx

Configure the index definition

120

Select a t...

+

ADD ANOTHER FIELD

> Options

CANCEL

CREATE INDEX

Індекси для song тепер

playlistdb.song

DOCUMENTS 20.2k TOTAL SIZE 1.9MB AVG. SIZE 96B INDEXES 2 TOTAL SIZE 332.0KB AVG. SIZE 166.0KB

Documents Aggregations Explain Plan **Indexes**

CREATE INDEX

Name and Definition ^	Type	Size	Usage	Properties	Drop
<code>_id_</code> <code>_id</code>	REGULAR ⓘ	213.0 KB	7 since Wed May 20 2020	UNIQUE ⓘ	
<code>duration_idx</code> <code>120</code>	REGULAR ⓘ	127.0 KB	1 since Wed May 20 2020		

Бачимо що пошук відбувся використовуючи новий індекс, час скоротився у 4 рази.

playlistdb.song

DOCUMENTS 20.2k TOTAL SIZE 1.9MB AVG. SIZE 96B INDEXES 2 TOTAL SIZE 332.0KB AVG. SIZE 166.0KB

Documents Aggregations **Explain Plan** Indexes

FILTER { "120": { "\$lt": "120" } } OPTIONS EXPLAIN RESET ...

VIEW DETAILS AS VISUAL TREE RAW JSON

**Query Performance Summary**

- Documents Returned: 2016
- Index Keys Examined: 2016
- Documents Examined: 2016
- Actual Query Execution Time (ms): 3
- Sorted in Memory: no
- Query used the following index: 120

**FETCH**  
nReturned: 2016 Execution Time: 0 ms  
DETAILS

**IXSCAN**  
nReturned: 2016 Execution Time: 0 ms  
Index Name: duration\_idx  
Multi Key Index: no  
DETAILS

Контрольні питання:

1. Назвати основні типи баз даних NoSQL.

- Ключ-значення
- Документоорієнтовані
- Графові
- Bigtable-подібні бази даних

2. Назвати переваги та недоліки використання баз даних NoSQL.

Переваги:

- динамічна схема
- масштабування
- легка в управлінні
- швидкість
- гнучкість

Недоліки:

- нова
- сумісність мов програмування
- обмеження запитів
- кадри

### 3. 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.

### 4. Операції вставки даних.

```
> db.food.insert({"fruit" : ["peach", "plum", "pear"]})  
> db.food.find({"fruit" : "pear"})
```

### 5. Операції оновлення даних.

```
> db.persons.update({name : "Tom"},{$pop:{languages: 1}})
```

### 6. Операції знищення даних.

```
> db.persons.remove({name : "Tom"})
```

### 7. Умовні оператори.

\$eq – рівно  
\$ne – нерівно  
\$gt – більше  
\$lt – менше  
\$gte – більше або рівно  
\$lte – менше або рівно  
\$in – в масиві значень  
\$nin – поза масивом значень

### 8. Операції керування індексами.

```
> db.persons.ensureIndex({"name" : 1})  
> db.system.indexes.find()  
> db.persons.dropIndex("name_1")
```

## 9. Пошук даних.

```
db.users.find({"address.state" : "NY"})
```

## 10. Можливості документних БД

База даних не виділяє окремий формат або схему.

Кожен документ може мати свою внутрішню структуру.

Документні БД є хорошим вибором для швидкої розробки.

В будь-який момент можна змінювати властивості даних, не змінюючи структуру або самі дані.

## Висновок

На лабораторній роботі я ознайомила та навчилася працювати з NoSQL базами даних.