

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



Лабораторна робота №12
з курсу:
“ОБДЗ”

Виконала :
студентка групи КН-210
Бурцьо Ольга

Перевірив:

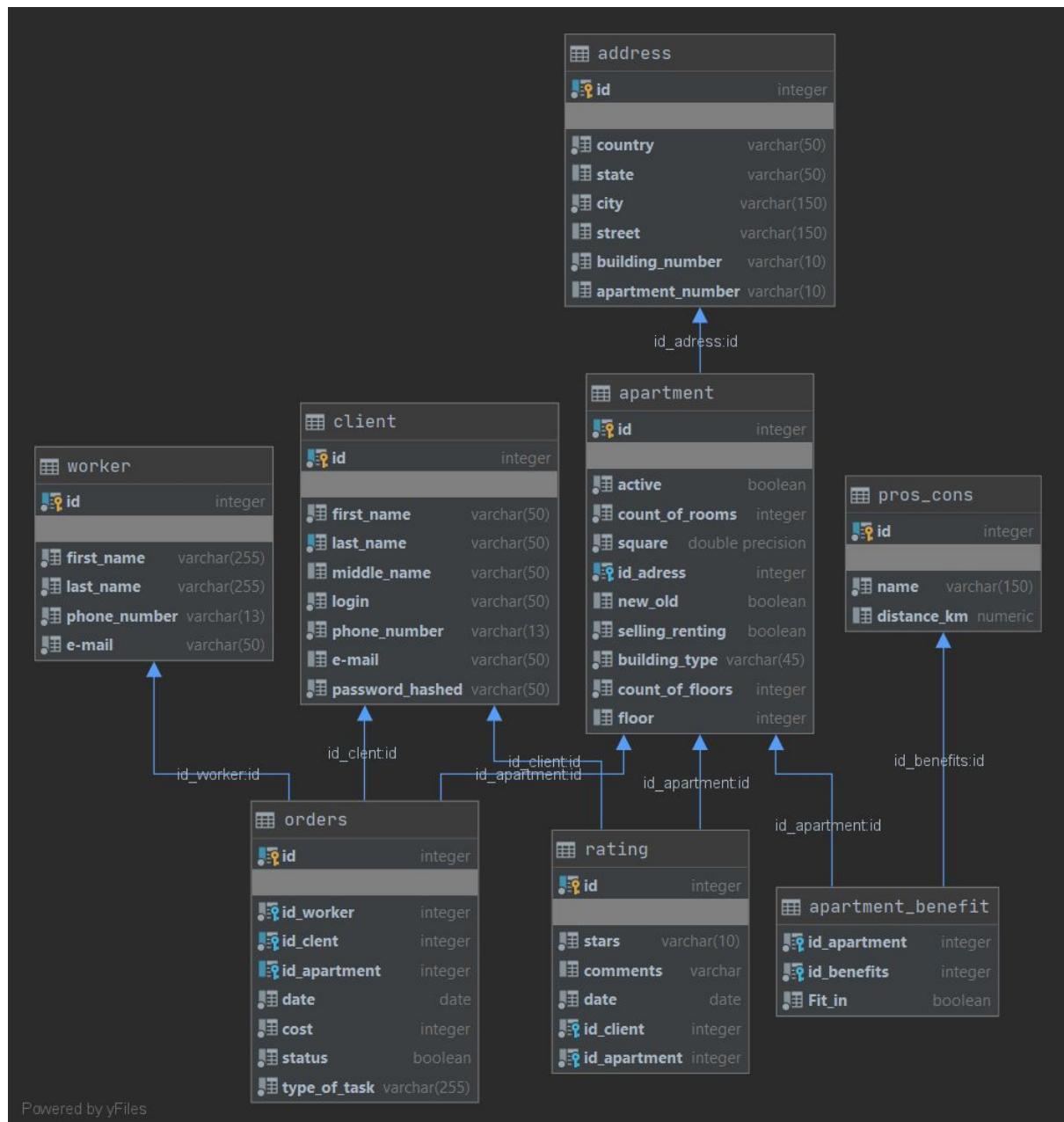
Мельникова Н.І.

Львів – 2020

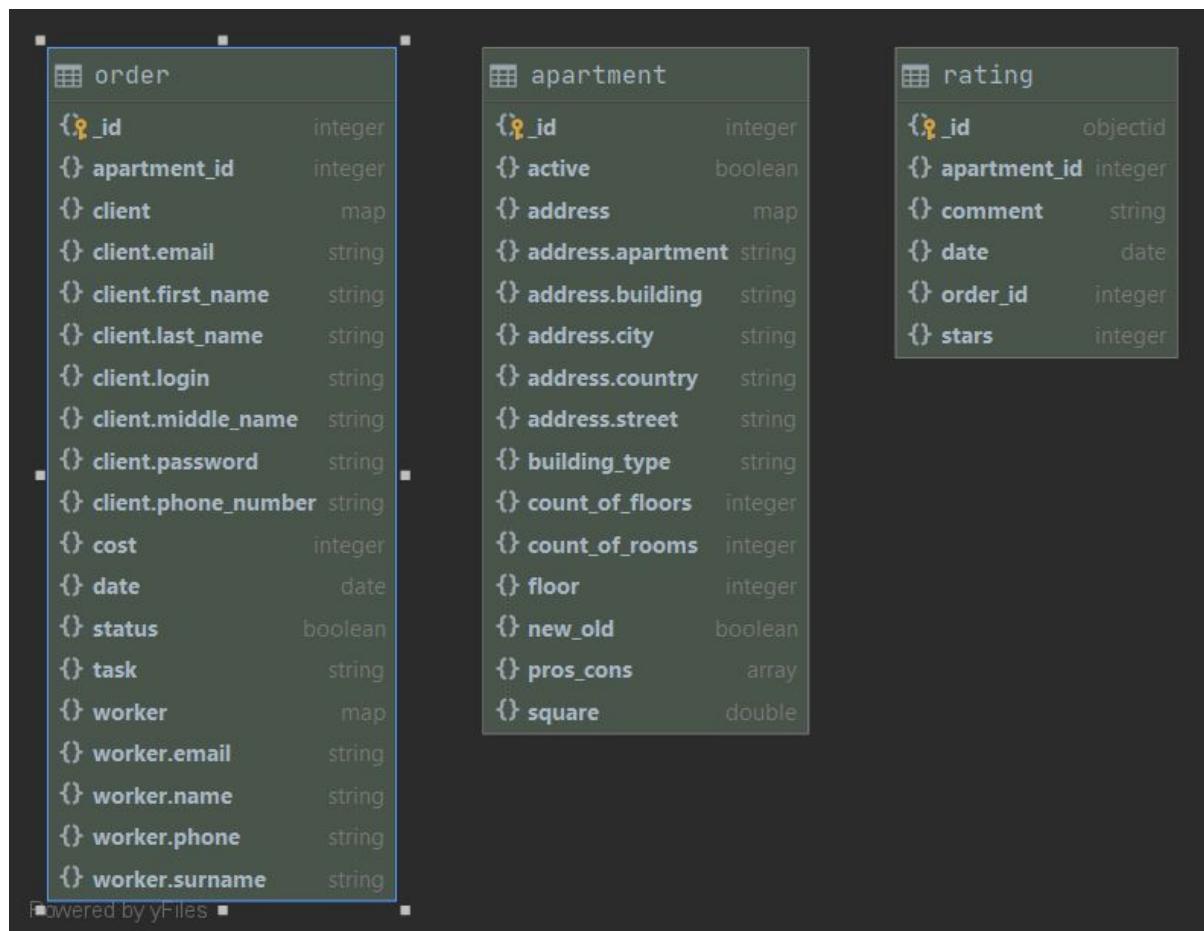
Мета роботи: здобуття практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB.

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.

1. ER - діаграма з попередніх лабораторних робіт



2. ER-диаграма для NoSQL



3. Створюємо БД agency

```
use agency
```

4. Створюємо колекції

```
db.createCollection('order', {autoIndexId: true, size: 5000})
db.createCollection('rating', {autoIndexId: true, size: 1000})
db.createCollection('apartment', {autoIndexId: true, size: 5000})
db.createCollection('client', {autoIndexId: true, size: 1000})
```

```
show collections
```

```
{> COLLECTION_NAME      <
1 rating
2 apartment
3 client
4 order
```

5. Видаляємо колекцію client.

```
| db.client.drop()
```

```
{> COLLECTION_NAME      <
1 rating
2 apartment
3 order
```

6. Заповнюємо колекції

```
db.rating.insertMany(
  [
    {
      stars: 10,
      comment:'So cool',
      date: ISODate('2020-01-21'),
      order_id: 2,
      apartment_id: 3
    }
  ]
)
```

```

db.apartment.insertMany(
  [
    {
      _id: 2,
      active: true,
      count_of_rooms: 4,
      square: 210.2,
      new_old: false,
      building_type: 'house',
      floor: 2,
      pros_cons: ['garage', 'garden', 'park', 'swimming pool', 'school'],
      address: {
        country: 'USA',
        state: 'Washington',
        city: 'Los Angeles',
        street: 'Washington',
        building: '345'
      }
    },
    {
      _id: 3,
      active: true,
      count_of_rooms: 3,
      square: 147,
      new_old: true,
      building_type: 'penthouse',
      count_of_floors: 20,
      floor: 20,
    }
  ]
)

```

7. Перевіряємо чи успішно додалось

The terminal window displays three MongoDB commands:

- `db.apartment.find()`
- `db.order.find()`
- `db.rating.find()`

Below the commands is a table comparing two apartment documents (1 and 2) across various fields.

	1	2
<code>_id</code>	1	2
<code>active</code>	✓	✓
<code>address</code>	{ "country": "Ukraine", "city": "Lviv", "street": "Patona", "building": "345" }	{ "country": "USA", "state": "Washington", "city": "Los Angeles", "street": "Washington", "building": "345" }
<code>building_type</code>	flat	house
<code>count_of_floors</code>	15	<unset>
<code>count_of_rooms</code>	2	4
<code>floor</code>	5	2
<code>new_old</code>	✓	✗
<code>pros_cons</code>	["sport club", "garage", "garden", "supermarket", "park"]	["garage", "garden", "park", "swimming pool", "school"]
<code>square</code>	107.5	210.2

8. Знайдемо житла, яке знаходиться в Україні.

```
db.apartment.find({"address.country": "Ukraine"})
```

	1	2	3
_id	1	3	
active	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
address	{"country": "Ukraine", "city": "Lviv"}	{"country": "Ukraine", "city": "Kyiv"}	
building_type	flat	penthouse	
count_of_floors	15	20	
count_of_rooms	2	3	
floor	5	20	
new_old	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
pros_cons	["sport club", "garage", "garden"]	["sport club", "garage", "garden"]	
square	107.5	147	

9. Знайдемо житло, яке або продається, або здається в оренду

```
db.order.find({$or:[{task:"selling"},{task:"purchase"}]})
```

	1	2	3
_id	1	3	
apartment_id	1	3	
client	{"first_name": "Mariia", "last_name": "Pavlenko"}	{"first_name": "Andriy", "last_name": "Shevchenko"}	
cost	80500	5500	
date	2020-03-13T00:00:00.000Z	2019-03-19T00:00:00.000Z	
status	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
task	selling	purchase	
worker	{"name": "Oleh", "surname": "Pavlenko"}	{"name": "Olha", "surname": "Shevchenko"}	

10. Знайдемо житла, де є поблизу супермаркет

```
db.apartment.find({pros_cons: "supermarket"})
```

_id	1	3
active	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
address	{"country": "Ukraine", "ci	{"country": "Ukraine", "ci
building_type	flat	penthouse
count_of_floors	15	20
count_of_rooms	2	3
floor	5	20
new_old	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
pros_cons	["sport club", "garage", "	["sport club", "garage", "
square	107.5	147

11. Знайдемо 2 записи, але пропускаючи 1

```
db.order.find().limit(2).skip( e: 1)
```

_id	2	3
client	{"first_name": "Denys", "l	{"first_name": "Andriy", "
cost	500	5500
date	2020-03-13T00:00:00.000Z	2019-03-19T00:00:00.000Z
status	<input type="checkbox"/>	<input checked="" type="checkbox"/>
task	notarius service	purchase
worker	{"name": "Olena", "surname	{"name": "Olha", "surname"
apartment_id	<unset>	3

12. Посортуємо угоди за ціною в порядку зростання

```
db.order.find().sort({cost:1})
```

_id	2	3	1
client	{"first_name": "Denys", "l	{"first_name": "Andriy", "l	{"first_name": "Ma
cost	500	5500	80500
date	2020-03-13T00:00:00.000Z	2019-03-19T00:00:00.000Z	2020-03-13T00:00:00.000Z
status	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
task	notarius service	purchase	selling
worker	{"name": "Olena", "surname	{"name": "Olha", "surname"	{"name": "Oleh", "
apartment_id	<unset>	3	1

13. Посортуємо по спаданню за ціною перші 2 записи

```
db.order.find().sort({cost:-1}).limit(2)
```

	1	2
_id	1	3
apartment_id	1	3
client	{"first_name": "Mariia", "last_name": "Svyyrynska"}	{"first_name": "Andriy", "last_name": "Svyyrynskyi"}
cost	80500	5500
date	2020-03-13T00:00:00.000Z	2019-03-19T00:00:00.000Z
status	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
task	selling	purchase
worker	{"name": "Oleh", "surname": "Svyyrynskyi"}	{"name": "Olha", "surname": "Svyyrynska"}

14. Створюємо унікальні індекси на логін клієнта та одночасне написання ім'я та прізвища

```
db.order.createIndex( name: {"client.login": 1}, keyPath: {unique: true})
db.order.createIndex( name: {"client.first_name": 1, "client.last_name": 1}, keyPath: {unique:true})
```

```
db.order.getIndexes()
```

15. Переглядаємо наявні індекси

key	name	ns	v	unique
{"_id": 1}	_id_	agency.order	2 <unset>	
{"client.login": 1}	client.login_1	agency.order	2	<input checked="" type="checkbox"/>
{"client.first_name": 1, "client.last_name": 1}	client.first_name_1_client.last_name_1_1	agency.order	2	<input checked="" type="checkbox"/>

При дублюванні вказаних даних вибиває таку помилку

```
Bulk write operation error on server cluster0-shard-00-01-fzlcj.mongodb.net:27017. Write errors: [BulkWriteError(index=0, code=11000, message='E11000 duplicate key error collection: agency.order index: client.login_1 dup key: { client.login: "svyyryr" }', details={})],
```

16. Видалимо 2 індекс, оскільки він не потрібний

```
db.order.dropIndex({"client.first_name": 1, "client.last_name": 1})
```

key	name	ns	v	unique
{"_id": 1}	_id_	agency.order	2 <unset>	
{"client.login": 1}	client.login_1	agency.order	2	<input checked="" type="checkbox"/>

Висновок: здобула практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB.

