

МІНІСТЕРСТВО ОСВІТИ І НАУКИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
«ЛЬВІВСЬКА ПОЛІТЕХНІКА»



**Лабораторна робота №14**  
**З дисципліни «Організація баз даних та знань»**

**Виконав:**  
студент групи КН-210  
Бікєєв Андрій

**Викладач:**  
Мельникова Н. І.

**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 видалення інформації в «сутність».

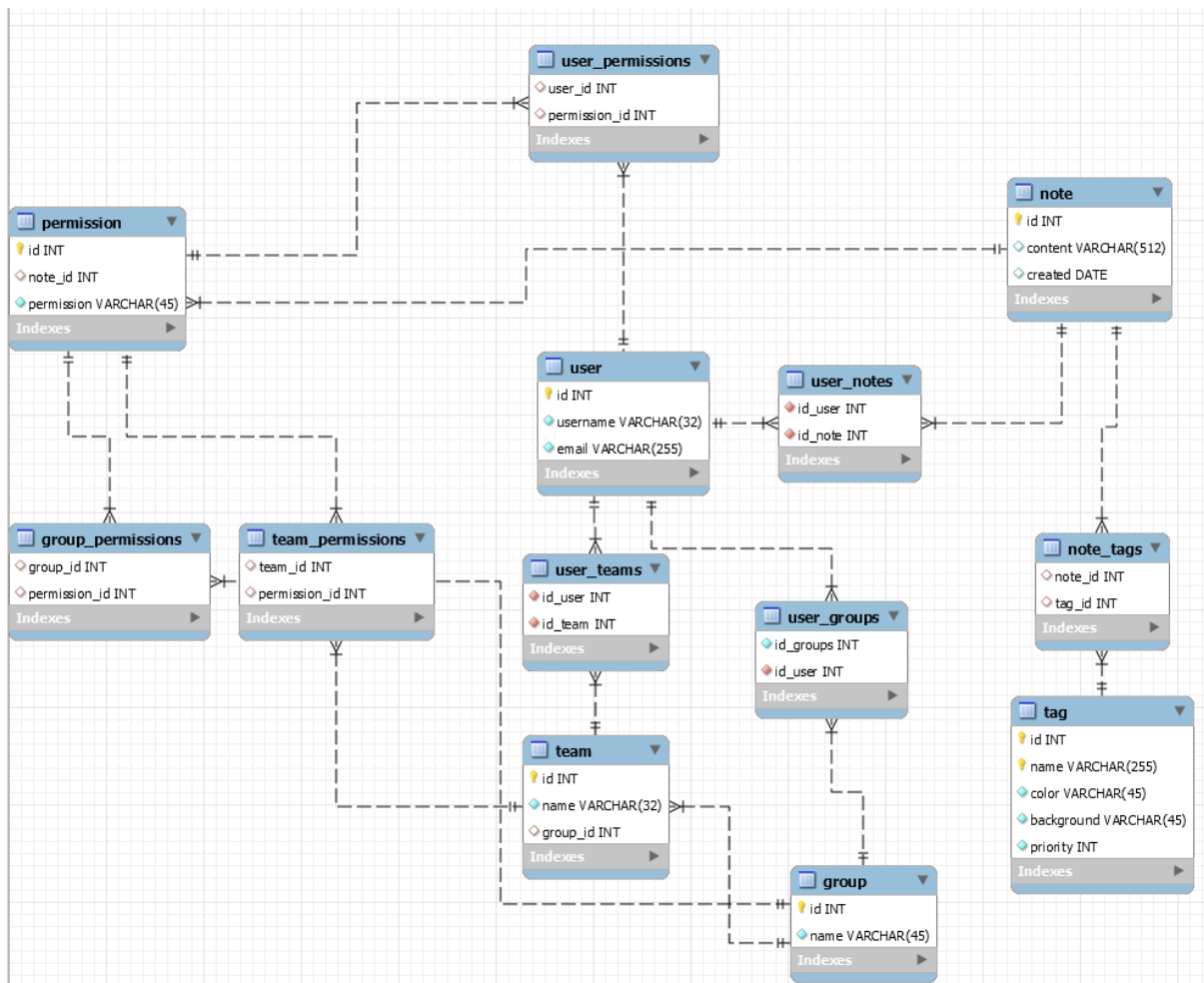


Рис 1. ER-діаграма

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.

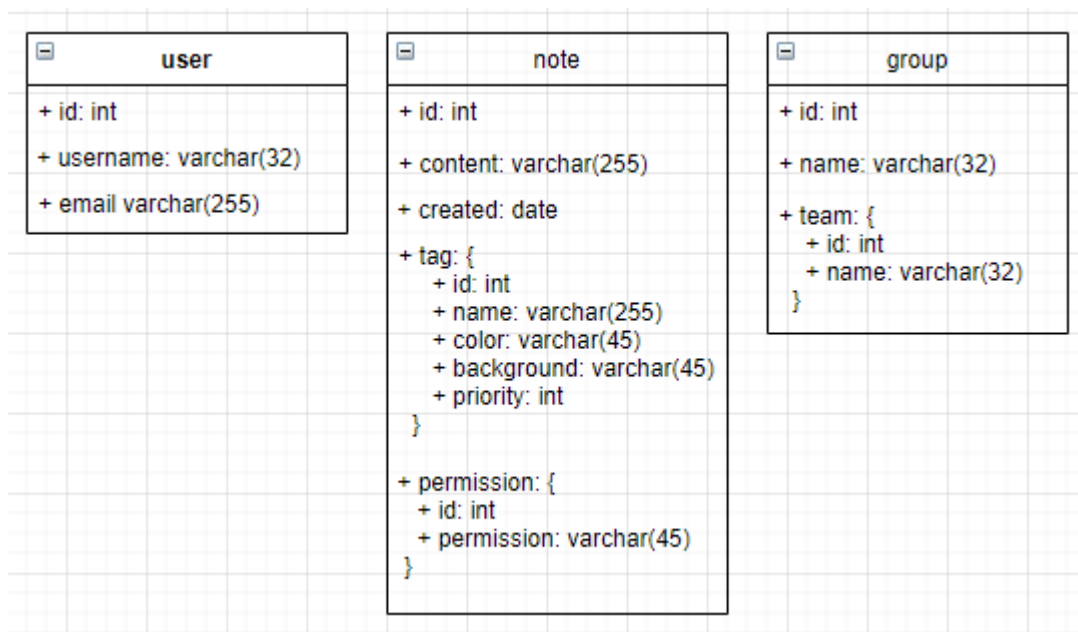


Рис 2. Модель даних

Запустимо MongoDB та створимо базу даних lab14 та додамо до неї 2 колекції: user, advertising\_department. Після чого наповнимо їх даними.

```
_id: ObjectId("5ecf766f46b9d9f323c22082")
username: "Luduk"
email: "luduk@gmail.com"

_id: ObjectId("5ecf76a846b9d9f323c22083")
username: "vadimDoiDoi"
email: "vadim@hotmail.com"

_id: ObjectId("5ecf76d646b9d9f323c22085")
username: "Not-a-Filler"
email: "fillMeUpDeadMan@gmail.com"

_id: ObjectId("5ecf770a46b9d9f323c22086")
username : "A-Bot-Huh"
email : "itsanemail@email.domain"

_id: ObjectId("5ecf7bef46b9d9f323c22087")
content: "This is a written note or something."
created: "2020-05-28"
tag: Object
  _id: ObjectId("5ecf7c2f46b9d9f323c22088")
  name: "Anime"
  color: "#0f0f0f"
  background: "#ffffff"
  priority: "1"
permission: Object
  _id: ObjectId("5ecf7c7246b9d9f323c2208a")
  permission: "read"

_id: ObjectId("5ecf7c9946b9d9f323c2208b")
name: "Group_1"
team: Array
  0: Object
    _id: ObjectId("5ecf7cc446b9d9f323c2208c")
    name: "Team_1"
```

Рис 3. Заповнені колекції в середовищі MongoDB Compass

Тепер виконаємо кілька простих запитів:  
Вивід обмеженої кількості полів:

```
> db.user.find().limit(1);
{ "_id" : ObjectId("5ecf766f46b9d9f323c22082"), "username" : "Luduk", "email" : "luduk@gmail.com" }
```

Вивід поля за заданим аргументом:

```
> db.user.find({"username": "VadimDoiDoi"});
{ "_id" : ObjectId("5ecf76a846b9d9f323c22083"), "username" : "VadimDoiDoi", "email" : "vadim@hotmail.org" }
```

Виведення унікальних юзернеймів:

```
> db.user.distinct("username")
[ "A-Bot-Huh", "Luduk", "Not-a-Filler", "VadimDoiDoi" ]
```

Використаємо курсори для виводу всіх імен в колекції user:

```
> cursor = db.user.find(); while (cursor.hasNext()) {obj = cursor.next(); print(obj["username"])}
Luduk
VadimDoiDoi
Not-a-Filler
A-Bot-Huh
```

Видалення поля з колекції:

```
> db.user.remove({"username": "A-Bot-Huh"});
WriteResult({ "nRemoved" : 1 })
> db.user.find({"username": "A-Bot-Huh"});
>
```

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