

МІНІСТЕРСТВО ОСВІТИ І НАУКИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЛЬВІВСЬКА ПОЛІТЕХНІКА»



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

Виконав:
студент групи КН-210
Дойков Вадим

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».

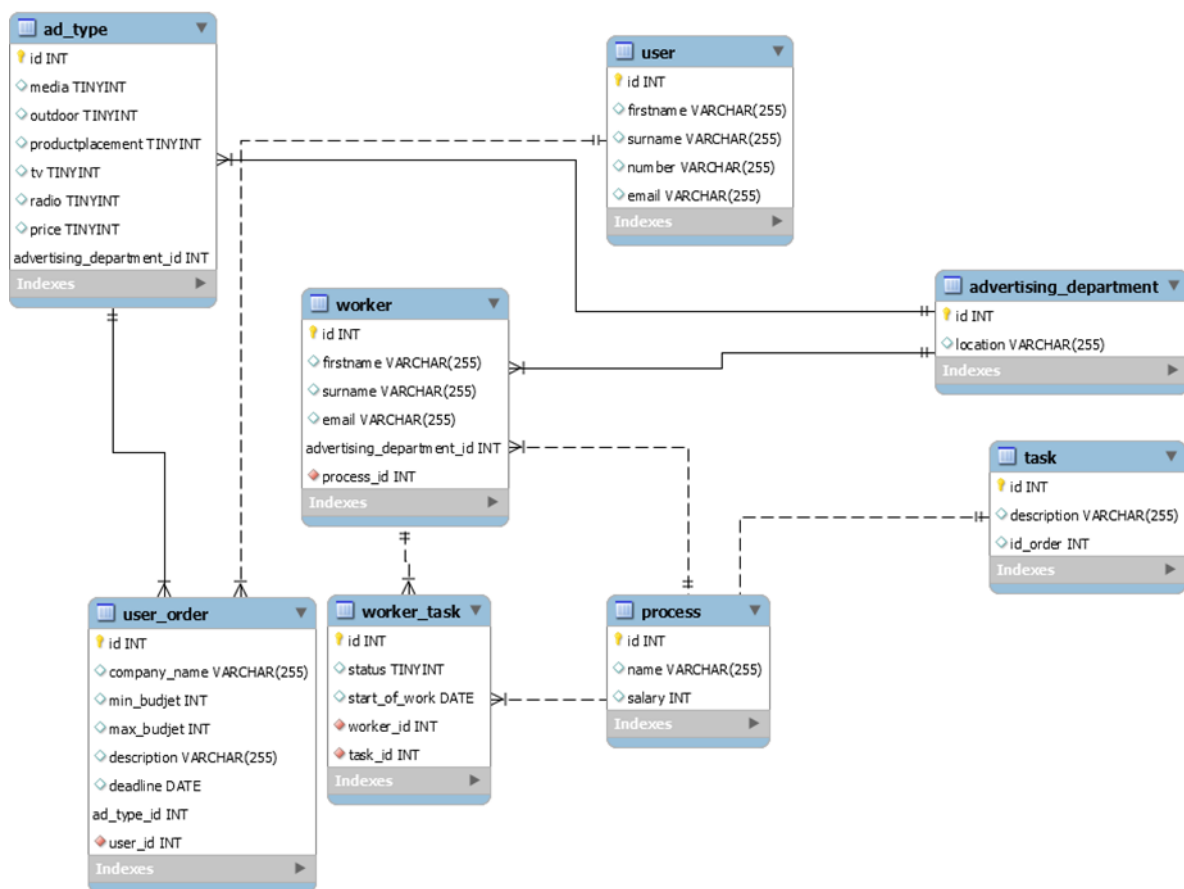


Рис 1. ER-діаграма

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.

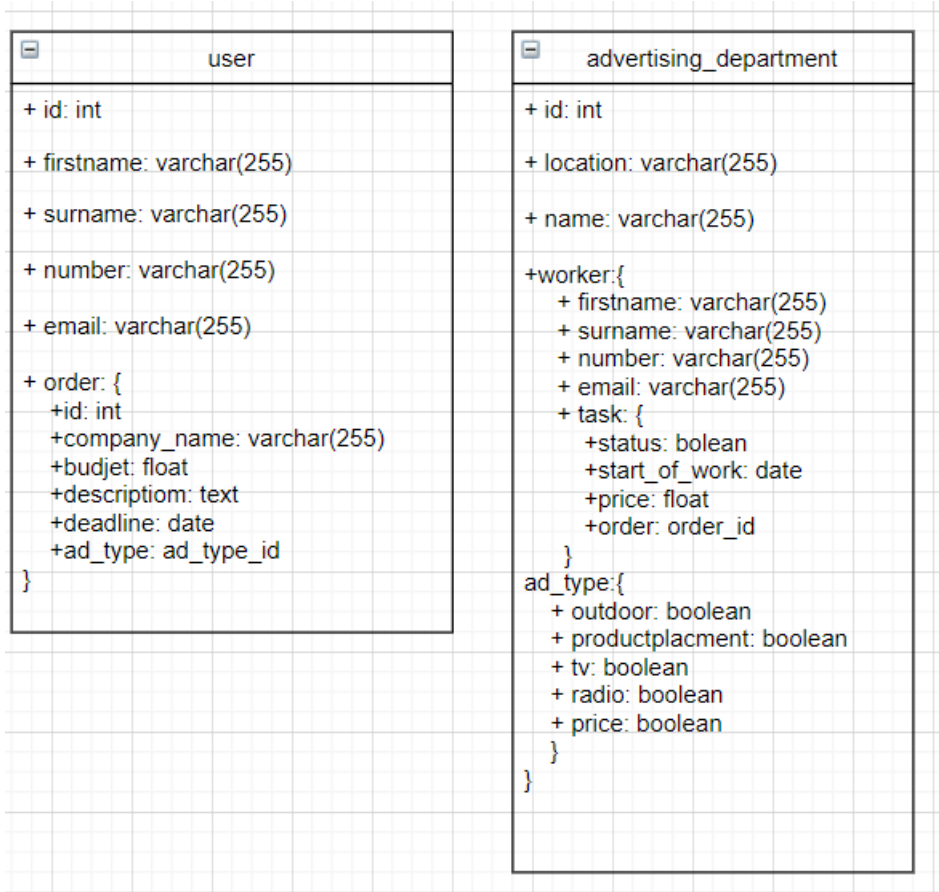


Рис 2. Модель даних

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

<pre> _id: ObjectId("5ec6911734ae9b1c84d55b54") firstname: "sanya" surname: "Shs" number: "22-22-4532" email: "shs@gmail.com" > order: Object </pre>	<pre> _id: ObjectId("5ec6905834ae9b1c84d55b50") location: "Хоткевича 48" name: "service" > worker: Object > ad_type: Object </pre>
<pre> _id: ObjectId("5ec6911734ae9b1c84d55b55") firstname: "Vitalya" surname: "Irxhovc" number: "325647-13" email: "tyw@gmail.com" > order: Object </pre>	<pre> _id: ObjectId("5ec6905834ae9b1c84d55b51") location: "Довженка 48" name: "web service" > worker: Object > ad_type: Object </pre>
<pre> _id: ObjectId("5ec6911734ae9b1c84d55b56") firstname: "Roman" surname: "Pan" number: "12313" email: "hellio@gmail.com" > order: Object </pre>	<pre> _id: ObjectId("5ec6905834ae9b1c84d55b52") location: "Петлюри 48" name: "web " > worker: Object > ad_type: Object </pre>
<pre> _id: ObjectId("5ec6911734ae9b1c84d55b57") firstname: "Olesya" surname: "Markiv" number: "217836-1236" email: "markul@gmail.com" > order: Object </pre>	<pre> _id: ObjectId("5ec6905834ae9b1c84d55b53") location: "Чукаріна 238" name: "company" > worker: Object > ad_type: Object </pre>

Рис 3. Заповнені колекції в середовищі MongoDB Compass

Тепер виконаємо кілька простих запитів:

Вивід обмеженої кількості полів:

```
> db.user.find().limit(1)
{ "_id" : ObjectId("5ec6911734ae9b1c84d55b54"), "firstname" : "sanya",
  "surname" : "Shs", "number" : "22-22-4532", "email" : "shs@gmail.com",
  "order" : { "company_name" : "porsche", "budget" : 222.1231, "description" : "reklama", "dealine" : ISODate("2022-10-01T21:00:00Z"), "ad_type" : 2 } }
```

Вивід поля за заданим аргументом:

```
> db.user.find({"firstname":"Vitalya"})
{ "_id" : ObjectId("5ec6911734ae9b1c84d55b55"), "firstname" : "Vitalya",
  "surname" : "Irxhovc", "number" : "325647-13", "email" : "tyw@gmail.com",
  "order" : { "company_name" : "toyota", "budget" : 3231.23, "description" : "my company hoche reklama", "dealine" : ISODate("2020-12-11T22:00:00Z"), "ad_type" : 3 } }
```

Виведення унікальних імен:

```
> db.user.distinct("firstname")
[ "Olesya", "Roman", "Vitalya", "sanya" ]
```

Виведення запису з бюджетом меншим за 1000:

```
{ "_id" : ObjectId("5ec6911734ae9b1c84d55b54"), "firstname" : "sanya", "surname" : "Shs", "number" : "22-22-4532", "email" : "shs@gmail.com", "order" : { "company_name" : "porsche", "budget" : 222.1231, "description" : "reklama", "dealine" : ISODate("2022-10-01T21:00:00Z"), "ad_type" : 2 } }
```

Використаємо курсори для виводу всіх імен в колекції user:

```
> while (cursor.hasNext()){obj = cursor.next(); print(obj["firstname"])}
sanya
Vitalya
Roman
Olesya
```

Видалення поля з колекції:

```
> db.user.remove({"firstname":"sanya"})
WriteResult({ "nRemoved" : 1 })
> db.user.find({"firstname":"sanya"})
>
```

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