

МІНІСТЕРСТВО ОСВІТИ І НАУКИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА  
ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



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

Виконав:  
студент групи КН-210  
Марій Павло

Викладач:  
Мельникова Наталя Іванівна

**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 (масив)
- 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)
```

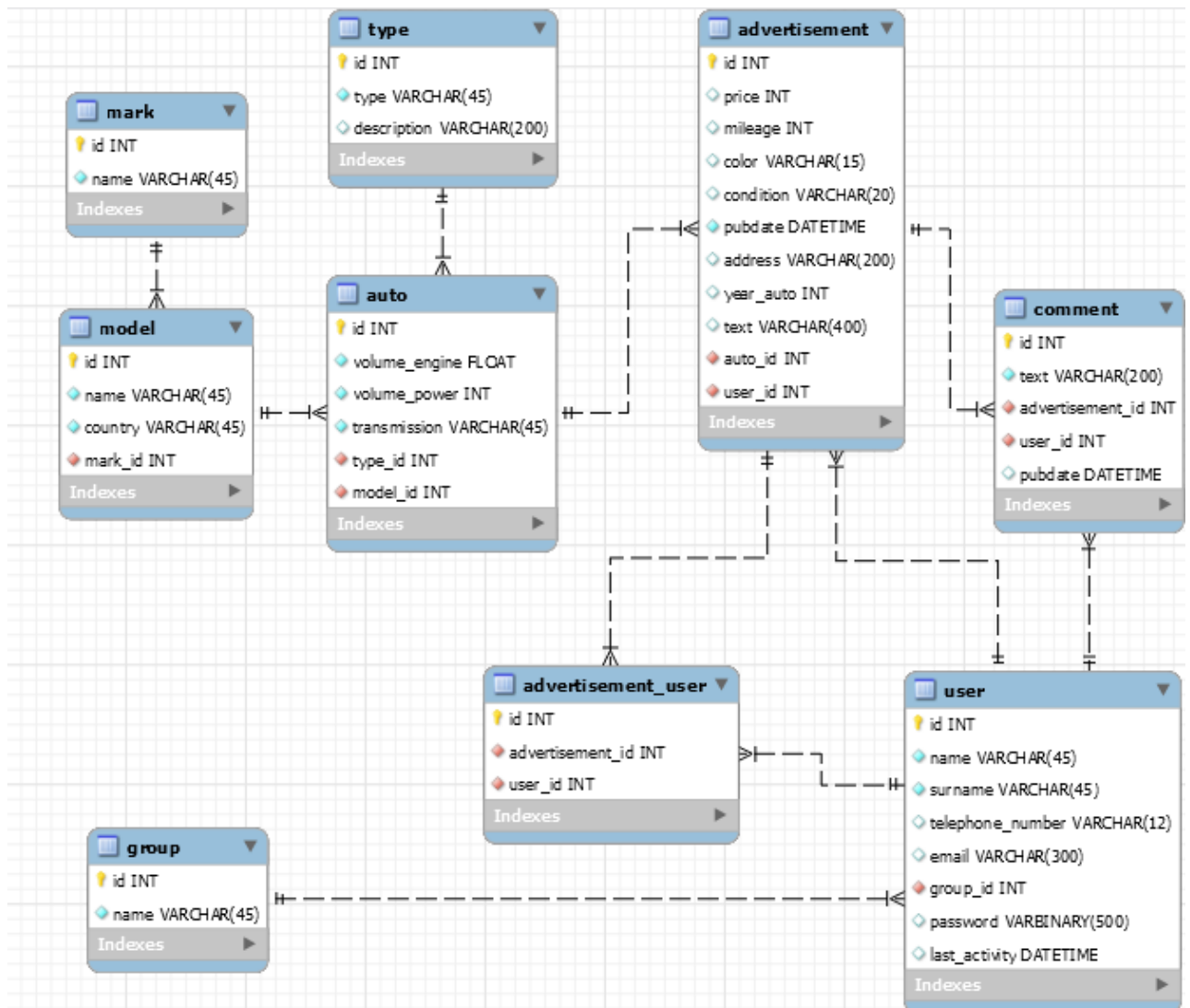
```
> db.persons.find().sort({name: 1})
```

```
> db.persons.find().sort({name: 1}).skip(3).limit(3)
```

## Хід роботи

1. Розробити схему бази даних на основі предметної області з лабораторної роботи №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.

Після всіх змін, які я вніс в базу даних під час виконання лабораторних робіт, її ER-Diagram зараз виглядає наступним чином:



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.

Advertisement	User	Auto
id: int price: int mileage: int color: string condition: string pubdate: datetime address: string year_auto: int text: string author: USER_ID auto: AUTO_ID list_of_users_who_save_this_adv: [ user: USER_ID ] list_of_comments: [ text: string, pubdate: datetime, author: USER_ID ]	id: int name: string surname: string telephone: string email: string password: string last_activity: datetime group: { name: string }	id: int volume_engine: float volume_power: float transmission: string type: { name: string, description: string } model: { name: string, country: string, mark: { name: string } }

Запустимо MongoDB та створимо базу даних autosale та додамо до неї 3 колекції: Оголошення, Авто, Користувач.

```
> use autosale
switched to db autosale
> db.createCollection("Advertisement")
{ "ok" : 1 }
> db.createCollection("User")
{ "ok" : 1 }
> db.createCollection("Auto")
{ "ok" : 1 }
>
```

Тепер спробуємо додати дані у колекцію User:

```
> db.User.insert({"name": "Oliver", "surname": "Smith", "telephone": "0987654321", "email": "oliver@gmail.com", "password": "password", "last_activity": "2020-03-26 21:21:25", "group": {"name": "Guest"}})
WriteResult({ "nInserted" : 1 })
>
```

Перевіримо, чи дані додалися:

```
> db.User.find()
{ "_id" : ObjectId("5ea7fe1d8c160dc68d280675"), "name" : "Oliver", "surname" : "Smith", "telephone" : "0987654321", "email" : "oliver@gmail.com", "password" : "password", "last_activity" : "2020-03-26 21:21:25", "group" : { "name" : "Guest" } }
>
```

Дані додано успішно. За допомогою MongoDB Compass для зручності, додаємо решту даних до бази даних.

Collection Name ^	Documents	Avg. Document Size	Total Document Size	Num. Indexes
<b>Advertisement</b>	15	595.1 B	8.9 KB	1
<b>Auto</b>	20	210.9 B	4.2 KB	1
<b>User</b>	20	203.2 B	4.1 KB	1

Тепер виконаємо кілька простих запитів:

```
> db.User.find().limit(3)
{ "_id" : ObjectId("5ea7fe1d8c160dc68d280675"), "name" : "Oliver", "surname" : "Smith", "telephone" : "0987654321", "email" : "oliver@gmail.com", "password" : "password", "last_activity" : "2020-03-26 21:21:25", "group" : { "name" : "Guest" } }
{ "_id" : ObjectId("5ea8032212ec1607e0a04348"), "name" : "Jack", "surname" : "Johnson", "telephone" : "0987654321", "email" : "jack@gmail.com", "password" : "password", "last_activity" : "2020-03-26 21:21:25", "group" : { "name" : "Logged" } }
{ "_id" : ObjectId("5ea8032212ec1607e0a04349"), "name" : "Harry", "surname" : "Williams", "telephone" : "0987654321", "email" : "harry@gmail.com", "password" : "password", "last_activity" : "2020-03-25 21:21:25", "group" : { "name" : "VIP1" } }
>

> db.User.find({"name": "Oliver"})
{ "_id" : ObjectId("5ea7fe1d8c160dc68d280675"), "name" : "Oliver", "surname" : "Smith", "telephone" : "0987654321", "email" : "oliver@gmail.com", "password" : "password", "last_activity" : "2020-03-26 21:21:25", "group" : { "name" : "Guest" } }
>

> db.Advertisement.find().sort({"price": 1}).limit(3)
{ "_id" : ObjectId("5ea862dc79435d25f8e6e934"), "price" : 2000, "mileage" : 10000, "color" : "blue", "condition" : "bad", "pubdate" : "2020-01-01 10:10:10", "address" : "77088", "year_auto" : 2001, "text" : "Some text", "author" : "5ea8032212ec1607e0a0434e", "auto" : "5ea807b012ec1607e0a0435e", "list_of_users_who_save_this_adv" : [ "5ea7fe1d8c160dc68d280675", "5ea8032212ec1607e0a04348", "5ea8032212ec1607e0a04349" ], "list_of_comments" : [ { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a0434f" }, { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a04350" } ] }
{ "_id" : ObjectId("5ea862dc79435d25f8e6e935"), "price" : 2000, "mileage" : 40000, "color" : "yellow", "condition" : "good", "pubdate" : "2020-02-02 10:10:10", "address" : "77089", "year_auto" : 2002, "text" : "Some text", "author" : "5ea8032212ec1607e0a0434e", "auto" : "5ea807b012ec1607e0a0435e", "list_of_users_who_save_this_adv" : [ "5ea7fe1d8c160dc68d280675", "5ea8032212ec1607e0a04348", "5ea8032212ec1607e0a04349" ], "list_of_comments" : [ { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a0434f" }, { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a04350" } ] }
{ "_id" : ObjectId("5ea862dc79435d25f8e6e935"), "price" : 3000, "mileage" : 20000, "color" : "black", "condition" : "good", "pubdate" : "2020-02-02 10:10:10", "address" : "77089", "year_auto" : 2002, "text" : "Some text", "author" : "5ea7fe1d8c160dc68d280675", "auto" : "5ea807b012ec1607e0a0435e", "list_of_users_who_save_this_adv" : [ "5ea8032212ec1607e0a0434e", "5ea8032212ec1607e0a04348", "5ea8032212ec1607e0a04349" ], "list_of_comments" : [ { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a0434f" }, { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a04350" } ] }
>

> db.User.find({"group.name": "VIP3"}).limit(2)
{ "_id" : ObjectId("5ea8032212ec1607e0a0434b"), "name" : "Charlie", "surname" : "Jones", "telephone" : "0987654321", "email" : "charlie@gmail.com", "password" : "password", "last_activity" : "2020-03-26 21:21:25", "group" : { "name" : "VIP3" } }
{ "_id" : ObjectId("5ea8032212ec1607e0a04351"), "name" : "James", "surname" : "Hernandes", "telephone" : "0987654321", "email" : "james@gmail.com", "password" : "password", "last_activity" : "2020-01-26 21:21:25", "group" : { "name" : "VIP3" } }
>
```

Використаємо курсори:

```
> var cursor = db.User.find()
> while(cursor.hasNext()){obj = cursor.next(); print(obj["name"])}
Oliver
Jack
Harry
Jacob
Charlie
Thomas
George
Oscar
James
William
James
John
Robert
Michael
William
David
Richard
Jack
John
Jerry
>
```

Тепер виконаємо певні команди групування:

```
> db.Advertisement.find({"price": 20000}).count()
1
> _
```

```
> db.User.distinct("name")
[
  "Charlie",
  "David",
  "George",
  "Harry",
  "Jack",
  "Jacob",
  "James",
  "Jerry",
  "John",
  "Michael",
  "Oliver",
  "Oscar",
  "Richard",
  "Robert",
  "Thomas",
  "William"
]
>
```

```
> db.Advertisement.find({"price": {$gte: 50000}}).limit(2)
{ "_id" : ObjectId("5ea862dc79435d25f8e6e93a"), "price" : 60000, "mileage" : 10000,
  "color" : "white", "condition" : "good", "pubdate" : "2020-02-02 10:10:10", "address" : "77089", "year_auto" : 2002, "text" : "Some text", "author" : "5ea8032212ec1607e0a0434e", "auto" : "5ea807b012ec1607e0a0435e", "list_of_users_who_save_this_adv" : [ "5ea7fe1d8c160dc68d280675", "5ea8032212ec1607e0a04348", "5ea8032212ec1607e0a04349" ], "list_of_comments" : [ { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a0434f" }, { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a04350" } ] }
{ "_id" : ObjectId("5ea862dc79435d25f8e6e93c"), "price" : 50000, "mileage" : 10000,
  "color" : "black", "condition" : "good", "pubdate" : "2020-02-02 10:10:10", "address" : "77089", "year_auto" : 2002, "text" : "Some text", "author" : "5ea8032212ec1607e0a0434e", "auto" : "5ea807b012ec1607e0a0435e", "list_of_users_who_save_this_adv" : [ "5ea7fe1d8c160dc68d280675", "5ea8032212ec1607e0a04348", "5ea8032212ec1607e0a04349" ], "list_of_comments" : [ { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a0434f" }, { "text" : "Some text", "pubdate" : "2020-01-01 10:10:10", "author" : "5ea8032212ec1607e0a04350" } ] }
```

Оновлення документа:

```
> db.User.update({"name": "Jerry"}, {"email": "jerryjerry@gmail.com"}, {upsert: true})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> _
```

Тепер знищимо колекцію Advertisement та створену базу даних:

```
> db.Advertisement.drop()
true
> db.dropDatabase()
{ "dropped" : "autosale", "ok" : 1 }
> _
```

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