

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту



Лабораторна робота #14

на тему:

«Розробка бази даних типу NoSQL»

з дисципліни

«Організація баз даних та знань»

Виконав:

студент групи КН-210

Максим Романьчук

Викладач:

Мельникова Н.І.

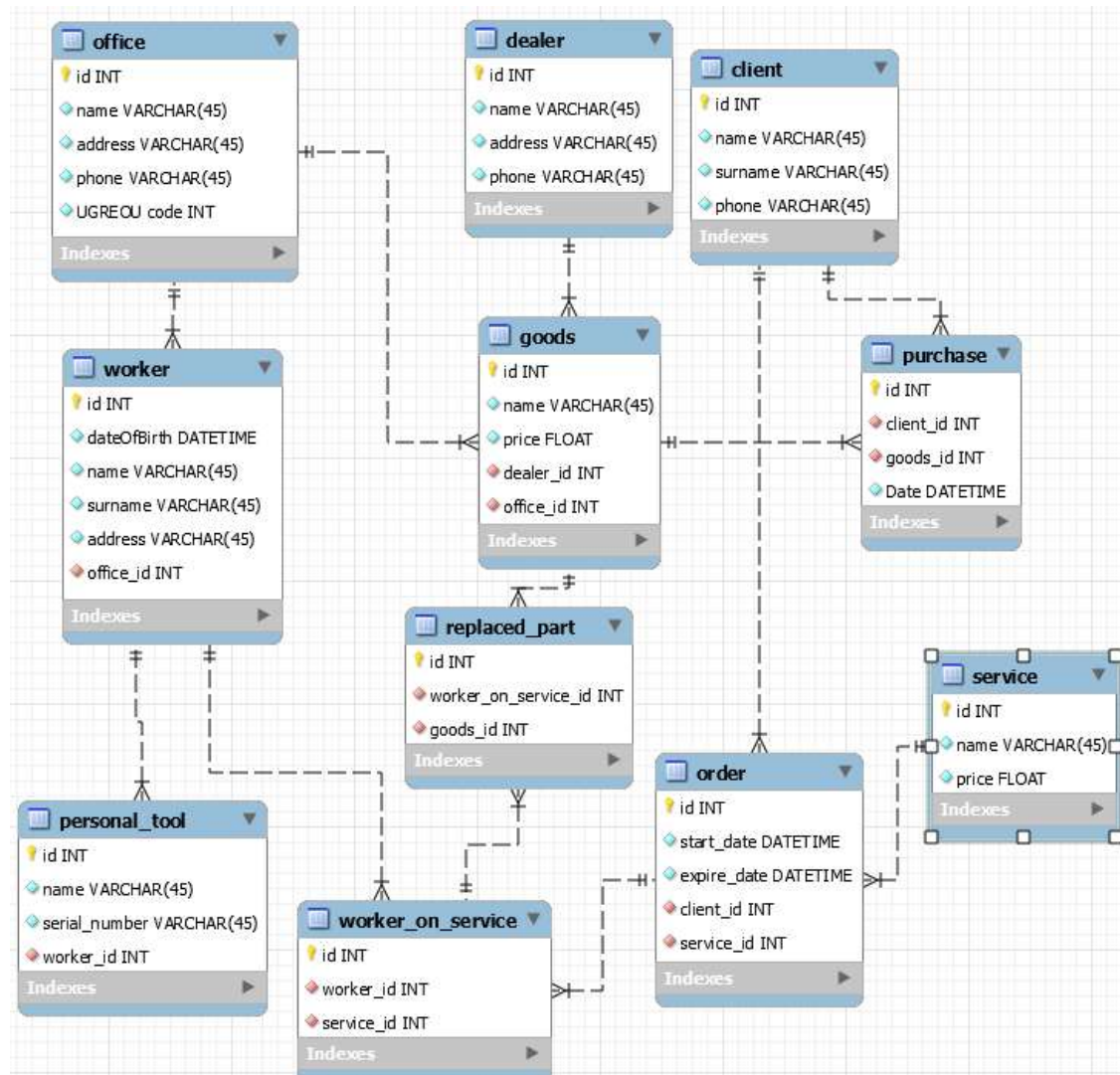
Львів – 2020 р.

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

Хід роботи:

1. Розробити схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB.

Тут зображена діаграма з лабораторної роботи №1:



2. Перетворити сутності діаграми БД, розробленої для лабораторної роботи №1, у структури, прийнятні для обробки в MongoDB.

Створюю аналогічну БД, але без полів-ключів, бо вони створюються автоматично:

```
> show collections
client
dealer
goods
office
order
personal_tool
purchase
replaced_part
service
worker
worker_service
```

Заповнюю її початковими даними:

Таблицю покупок

```
> db.purchase.insert({ date: Date(), client: "client1", goods: "design book"})
WriteResult({ "nInserted" : 1 })
> db.purchase.insert({ date: Date(), client: "client2", goods: "coding book"})
WriteResult({ "nInserted" : 1 })
> db.purchase.insert({ date: Date(), client: "client2", goods: "keyboard"})
WriteResult({ "nInserted" : 1 })
```

Таблицю товарів

```
> db.goods.insert({
... name: "keyboard",
... price: 3500.00,
... dealer: "dealer1",
... office: "office1"})
WriteResult({ "nInserted" : 1 })
> db.goods.insert({ name: "mouse", price: 250.00, dealer: "dealer1", office: "office2"})
WriteResult({ "nInserted" : 1 })
> db.goods.insert({ name: "USB cable", price: 175.00, dealer: "dealer1", office: "office3"})
WriteResult({ "nInserted" : 1 })
> db.goods.insert({ name: "speakers", price: 500.00, dealer: "dealer1", office: "office4"})
WriteResult({ "nInserted" : 1 })
> db.goods.insert({ name: "coding book", price: 700.00, dealer: "dealer2", office: "office1"})
WriteResult({ "nInserted" : 1 })
> db.goods.insert({ name: "typing book", price: 650.00, dealer: "dealer2", office: "office2"})
WriteResult({ "nInserted" : 1 })
> db.goods.insert({ name: "design book", price: 1300.00, dealer: "dealer2", office: "office3"})
WriteResult({ "nInserted" : 1 })
> db.goods.insert({ name: "analytics book", price: 1050.00, dealer: "dealer2", office: "office4"})
```

Таблицю інструментів

```
> db.personal_tool.insert({
... name: "wrench",
... serial_num: "0A",
... worker: "Tom Sanders"})
WriteResult({ "nInserted" : 1 })
> db.personal_tool.insert({ name: "key", serial_num: "0B", worker: "Sam Mandes"})
WriteResult({ "nInserted" : 1 })
> db.personal_tool.insert({ name: "axe", serial_num: "0C", worker: "Tim Burton"})
WriteResult({ "nInserted" : 1 })
> db.personal_tool.insert({ name: "showel", serial_num: "0D", worker: "Jean Lee"})
WriteResult({ "nInserted" : 1 })
> db.personal_tool.insert({ name: "toolkit", serial_num: "0E", worker: "Sean Smith"})
WriteResult({ "nInserted" : 1 })
```


І так у всі таблиці (вміст деяких буде продемонстровано далі).

3. Забезпечити реалізацію функцій редагування, додавання та видалення інформації в «сутність».

Оновлення цін одного та кількох сервісів.

```
> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "replace", "price" : 450 }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 450 }

> db.service.update({ price: {$lt: 1000}}, { $inc: { price: 200 } })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "replace", "price" : 650 }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 450 }

> db.service.update({ price: {$lt: 500}}, { $inc: { price: 200 } })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "replace", "price" : 650 }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 650 }

> db.service.updateMany({ price: {$lt: 1000}}, { $inc: { price: 200 } })
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }

> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "replace", "price" : 850 }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 850 }
```

Заміна значень полів.

```
> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "replace", "price" : 850 }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 850 }

> db.service.update({name: "replace"}, {
... $set: {
... name: "repair",
... price: 700 }})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "repair", "price" : 700 }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 850 }
```

Додавання і видалення полів.

```
> db.service.update({name: "repair"}, { $set: { universal: true } })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "repair", "price" : 700, "universal" : true }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 850 }

> db.service.update({name: "repair"}, { $unset: { universal: "" } })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.service.find()
{ "_id" : ObjectId("5ec6238a33ccc4d4f2d06f2b"), "name" : "repair", "price" : 700 }
{ "_id" : ObjectId("5ec6239833ccc4d4f2d06f2c"), "name" : "refill", "price" : 850 }
```

Висновок: Виконуючи цю лабораторну роботу, я дослідив такі базові операції командної стрічки MongoDB, як створення колекцій та баз даних і створення та оновлення даних в них