

МІНІСТЕРСТВО ОСВІТИ І НАУКИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



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

Виконав:
студент групи КН-210
Черещук Любомир

Перевірів:
Кандидат тех. наук, ст. викладач
Мельникова Н. І.

Львів – 2020

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"})
```

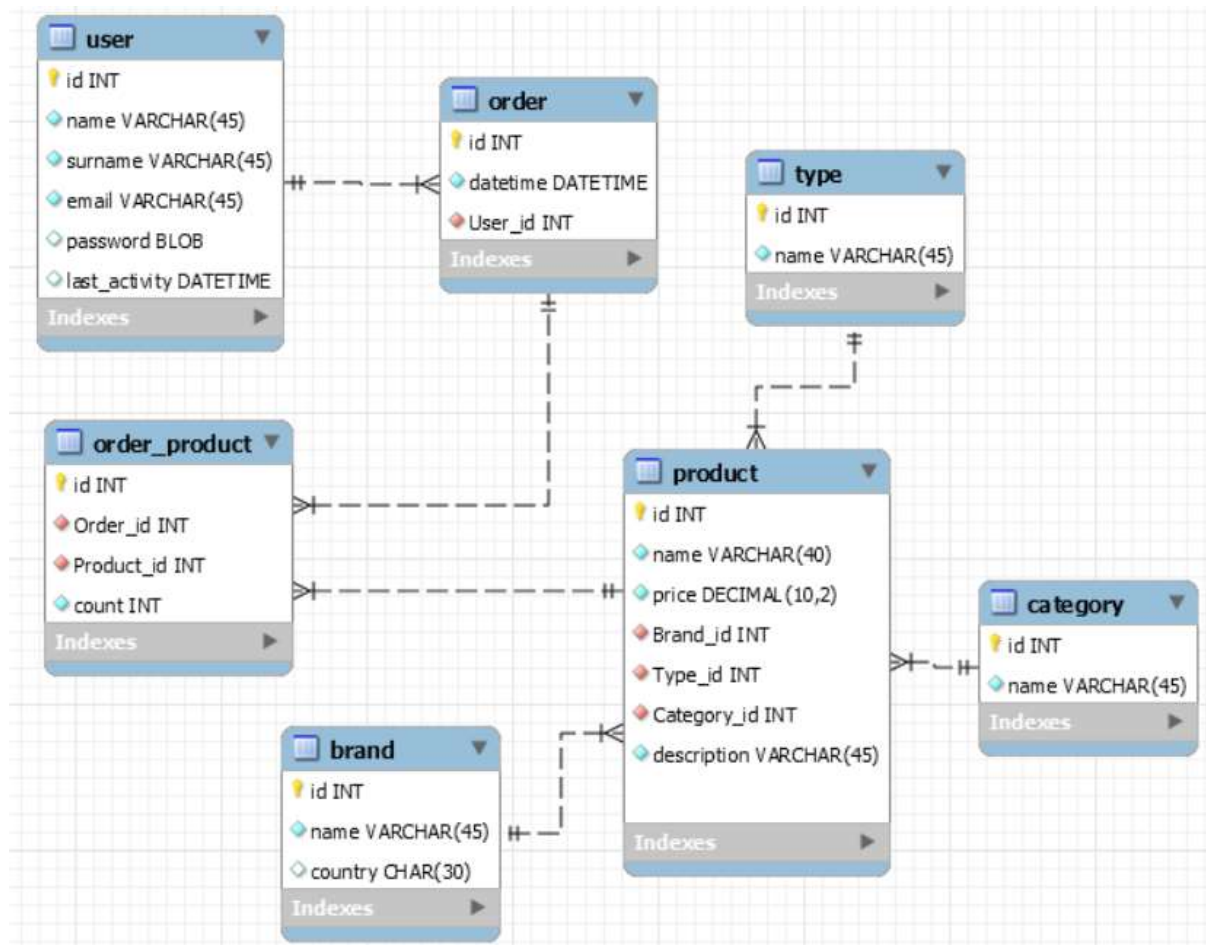
Налаштування запитів і сортування

```
> 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

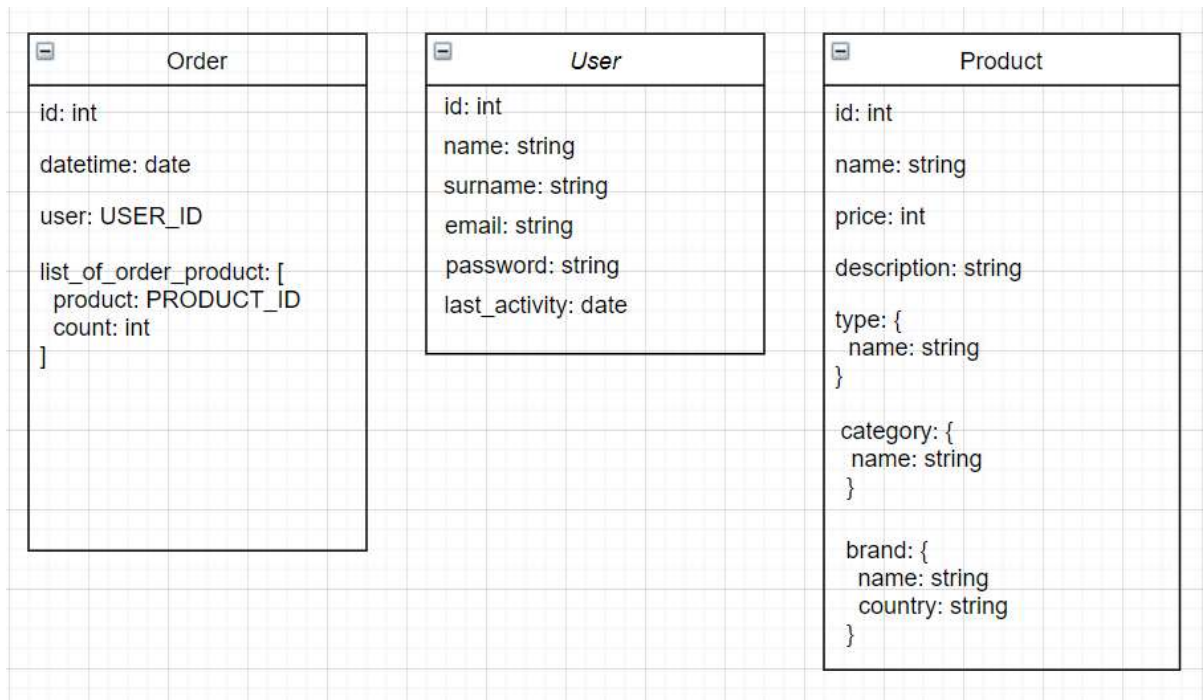
1. Розробити схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB.
2. Перетворити сутності діаграми БД, розробленої для лабораторної роботи №1, у структури, прийнятні для обробки в MongoDB.

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.



Запустимо MongoDB та створимо базу даних sportproducts та додамо до неї 3 колекції: Замовлення, Продукт, Користувач.

```
> use sportproducts
switched to db sportproducts
> db.createCollection("Order")
{ "ok" : 1 }
> db.createCollection("User")
{ "ok" : 1 }
> db.createCollection("Product")
{ "ok" : 1 }
>
```




Тепер спробуємо додати дані у колекцію User:

```
> db.User.insertOne({
...   "name": "John",
...   "surname": "Dark",
...   "email": "johndark@gmail.com",
...   "password": "pass1",
...   "last_activity": new Date('2020-05-16')
... })
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5ec0273cbe56bf465be1f7cf")
}
```

Перевіримо, чи дані додалися:

```
> db.User.find()
{ "_id" : ObjectId("5ec0273cbe56bf465be1f7cf"), "name" : "John", "surname" : "Dark", "email" : "johndark@gmail.com", "password" : "pass1", "last_activity" : ISODate("2020-05-16T00:00:00Z") }
>
```

Дані додано успішно. За допомогою утиліти MongoDB Compass додаємо решту даних до бази даних.

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
Order	8	160.5 B	1.3 KB	1	36.0 KB	
Product	8	189.4 B	1.5 KB	1	36.0 KB	
User	10	129.8 B	1.3 KB	1	36.0 KB	

Тепер виконаємо кілька простих запитів:

```
> db.User.find().limit(4)
{ "_id" : ObjectId("5ec0273cbe56bf465be1f7cf"), "name" : "John", "surname" : "Dark", "email" : "johndark@gmail.com", "password" : "pass1", "last_activity" : ISODate("2020-05-16T00:00:00Z") }
{ "_id" : ObjectId("5ec029ac07a05124604d0280"), "name" : "Ivan", "surname" : "Ron", "email" : "ivanronan@gmail.com", "password" : "pass2", "last_activity" : ISODate("2020-05-13T00:00:00Z") }
{ "_id" : ObjectId("5ec029dc07a05124604d0281"), "name" : "Bill", "surname" : "Fetch", "email" : "billfetchn@gmail.com", "password" : "pass3", "last_activity" : ISODate("2020-05-15T00:00:00Z") }
{ "_id" : ObjectId("5ec029f507a05124604d0282"), "name" : "Anna", "surname" : "Dell", "email" : "annadell@gmail.com", "password" : "pass4", "last_activity" : ISODate("2020-05-14T00:00:00Z") }
>
```

```
> db.User.find({"surname" : "Fetch"})
{ "_id" : ObjectId("5ec029dc07a05124604d0281"), "name" : "Bill", "surname" : "Fetch", "email" : "billfetchn@gmail.com", "password" : "pass3", "last_activity" : ISODate("2020-05-15T00:00:00Z") }
>
```

```
> db.Product.find({}, { "_id" : 0 }).sort({ "price" : 1 })
{ "name" : "ball", "price" : 400, "description" : "for football", "type" : { "name" : "balls" }, "category" : { "name" : "football" }, "brand" : { "name" : "adidas", "country" : "usa" } }
{ "name" : "shorts", "price" : 500, "description" : "for running", "type" : { "name" : "clothes" }, "category" : { "name" : "running" }, "brand" : { "name" : "nike", "country" : "usa" } }
{ "name" : "grif", "price" : 500, "description" : "for crossfit", "type" : { "name" : "equipment" }, "category" : { "name" : "crossfit" }, "brand" : { "name" : "oly", "country" : "usa" } }
{ "name" : "protein", "price" : 600, "description" : "for bodybuilding", "type" : { "name" : "sport nutrition" }, "category" : { "name" : "bodybuilding" }, "brand" : { "name" : "gold nutrition", "country" : "usa" } }
{ "name" : "pants", "price" : 600, "description" : "for running", "type" : { "name" : "clothes" }, "category" : { "name" : "running" }, "brand" : { "name" : "puma", "country" : "usa" } }
{ "name" : "t-short", "price" : 700, "description" : "for running", "type" : { "name" : "clothes" }, "category" : { "name" : "running" }, "brand" : { "name" : "zara", "country" : "turkey" } }
{ "name" : "shoes", "price" : 900, "description" : "for running", "type" : { "name" : "clothes" }, "category" : { "name" : "running" }, "brand" : { "name" : "nike", "country" : "usa" } }
{ "name" : "bike", "price" : 1500, "description" : "for biking", "type" : { "name" : "bikes" }, "category" : { "name" : "athletics" }, "brand" : { "name" : "wanday", "country" : "usa" } }
>
```



```
> db.Product.find({"brand.name": "adidas"}, {_id: 0 })
{ "name" : "ball", "price" : 400, "description" : "for football", "type" : { "name" : "balls" }, "category" : { "name" : "football" }, "brand" : { "name" : "adidas", "country" : "usa" } }
```

```
> db.Order.find( { "datetime": { $gte: new Date('2020-05-18') } })
{ "_id" : ObjectId("5ec0433507a05124604d0292"), "datetime" : ISODate("2020-05-19T00:00:00Z"), "user" : "5ec029f507a05124604d0282", "list_of_order_product" : [ { "product" : "5ec03a9d07a05124604d028c", "count" : 1 } ] }
{ "_id" : ObjectId("5ec0445507a05124604d0296"), "datetime" : ISODate("2020-05-19T00:00:00Z"), "user" : "5ec02a1707a05124604d0283", "list_of_order_product" : [ { "product" : "5ec03af607a05124604d028e", "count" : 1 } ] }
```

```
> db.Order.find( { "list_of_order_product.0.count": 1 }).limit(5)
{ "_id" : ObjectId("5ec0424cbe56bf465belf7d1"), "datetime" : ISODate("2020-05-16T00:00:00Z"), "user" : "5ec0273cbe56bf465belf7cf", "list_of_order_product" : [ { "product" : "5ec02c05be56bf465belf7d0", "count" : 1 } ] }
{ "_id" : ObjectId("5ec042b807a05124604d0290"), "datetime" : ISODate("2020-05-16T00:00:00Z"), "user" : "5ec029ac07a05124604d0280", "list_of_order_product" : [ { "product" : "5ec03a2407a05124604d0289", "count" : 1 } ] }
{ "_id" : ObjectId("5ec0430807a05124604d0291"), "datetime" : ISODate("2020-05-17T00:00:00Z"), "user" : "5ec029dc07a05124604d0281", "list_of_order_product" : [ { "product" : "5ec03a2407a05124604d0289", "count" : 1 } ] }
{ "_id" : ObjectId("5ec0433507a05124604d0292"), "datetime" : ISODate("2020-05-19T00:00:00Z"), "user" : "5ec029f507a05124604d0282", "list_of_order_product" : [ { "product" : "5ec03a9d07a05124604d028c", "count" : 1 } ] }
{ "_id" : ObjectId("5ec0437407a05124604d0293"), "datetime" : ISODate("2020-05-15T00:00:00Z"), "user" : "5ec02a4607a05124604d0284", "list_of_order_product" : [ { "product" : "5ec03af607a05124604d028e", "count" : 1 } ] }
```

```
> db.Product.find({ $and: [{ "price": { $gt: 700 } }, {"brand.name": "nike"}] })
{ "_id" : ObjectId("5ec03ab207a05124604d028d"), "name" : "shoes", "price" : 900, "description" : "for running", "type" : { "name" : "clothes" }, "category" : { "name" : "running" }, "brand" : { "name" : "nike", "country" : "usa" } }
```

Використаємо курсори:

```
> let cursorItem = db.User.find()
> while (cursorItem.hasNext()){obj = cursorItem.next(); print(obj["name"])}
John
Ivan
Bill
Anna
Andy
Mark
Lila
Katya
Steve
Tony
```

Тепер виконаємо певні команди групування:

```
> db.Product.find({"price" : {$gt: 500}}).count()
5
>
```

Функція distinct

```
> db.Product.distinct("brand.name")
[ "adidas", "gold nutrition", "nike", "oly", "puma", "wanday", "zara" ]
>
```

```
> db.Product.updateMany({"name": "grif"}, {$set: {"brand.name": "adidas"}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }  
>
```

Можемо знищити колекцію : `db.Колекція.drop()` або базу даних :
`db.dropDatabase()`

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