

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра СШІ

ЗВІТ

До лабораторної роботи № 14

З дисципліни: *“Бази даних”*

Виконала:

ст. гр. КН-207

Данків Анастасія

Прийняв:

викладач

Мельникова Н.І.

Львів – 2019

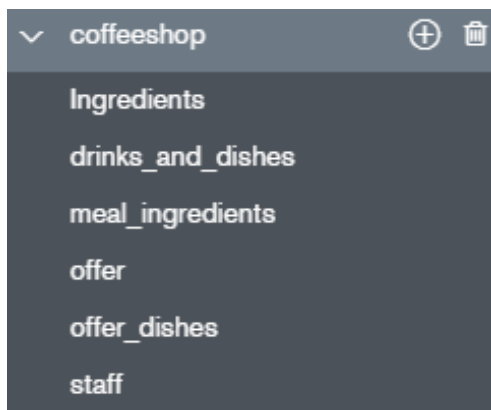
Мета роботи: здобуття практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB.

Хід роботи.

1. Розробити схему бази даних на основі предметної області з лабораторної роботи №1 у спосіб, що застосовується в СУБД MongoDB..
2. Перетворити сутності діаграми БД, розробленої для лабораторної роботи №1, у структури, прийнятні для обробки в MongoDB.
3. Забезпечити реалізацію функцій редагування, додавання та вилучення інформації в «сутність».

1. Створюю схеми предметної області 'Кафетерій'

```
use coffeeshop
switched to db coffeeshop
db.createCollection("Ingredients")
"ok" : 1 }
db.createCollection("meal_ingredients")
"ok" : 1 }
db.createCollection("drinks_and_dishes")
"ok" : 1 }
db.createCollection("offer_dishes")
"ok" : 1 }
db.createCollection("offer")
"ok" : 1 }
db.createCollection("staff")
"ok" : 1 }
```



2. Заповнюю дані колекції
db.ingredients.insert([

```
{
  "name": "coffee",
  "count": "5000",
  "type": "drink",
  "unit": "ml"
},
{
  "name": "milk",
  "count": "10000",
  "type": "drink",
  "unit": "ml"
}
```

])

```

> db.ingredients.insert([
...     {
...         "name": "coffee",
...         "count": "5000",
...         "type": "drink",
...         "unit": "ml"
...     },
...     {
...         "name": "milk",
...         "count": "10000",
...         "type": "drink",
...         "unit": "ml"
...     }
... ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})

```

```

> db.ingredients.find().pretty()
{
  "_id" : ObjectId("5cf59cf4e41eb48eef613d29"),
  "name" : "coffee",
  "count" : "5000",
  "type" : "drink",
  "unit" : "ml"
}
{
  "_id" : ObjectId("5cf59cf4e41eb48eef613d2a"),
  "name" : "milk",
  "count" : "10000",
  "type" : "drink",
  "unit" : "ml"
}
{
  "_id" : ObjectId("5cf59febe41eb48eef613d2b"),
  "name" : "water",
  "count" : "5000",
  "type" : "drink",
  "unit" : "ml"
}
{
  "_id" : ObjectId("5cf59febe41eb48eef613d2c"),
  "name" : "salat",
  "count" : "1000",
  "type" : "dish",
  "unit" : "gram"
}
{
  "_id" : ObjectId("5cf59febe41eb48eef613d2d"),
  "name" : "potato",
  "count" : "5000",
  "type" : "dish",
  "unit" : "gram"
}
{
  "_id" : ObjectId("5cf59febe41eb48eef613d2e"),
  "name" : "meat",
  "count" : "3000",
  "type" : "dish",
  "unit" : "gram"
}

```

```
_id: ObjectId("5cf5a3b6619cd22a00a9edea")
name: "coffee"
count: 5000
type: "drink"
unit: "ml"
```

```
_id: ObjectId("5cf5a4be31e6572a00ea8640")
name: "milk"
count: 10000
type: "drink"
unit: "ml"
```

```
name: "potato"
count: 1000
type: "dish"
unit: "gram"
_id: ObjectId("5cf5a50131e6572a00ea8641")
```

```
_id: ObjectId("5cf5a51b31e6572a00ea8642")
name: "salad"
count: 1000
type: "dish"
unit: "gram"
```

Drinks_and_dishes

```
_id: ObjectId("5cf5a6b031e6572a00ea8643")
name: "milk"
price: 20
size: 250
type: "drink"
unit: "ml"
```

```
_id: ObjectId("5cf5a6c931e6572a00ea8644")
name: "water"
price: 15
size: 500
type: "drink"
unit: "ml"
```

```
_id: ObjectId("5cf5a6e931e6572a00ea8645")
name: "potato"
price: 40
size: 300
type: "dish"
unit: "gram"
```

```
_id: ObjectId("5cf5a6fb31e6572a00ea8646")
name: "salad"
price: 45
```

Meal_ingredients

	_id ObjectId	id_ingredients ObjectId	count Int32	unit String	id_dd ObjectId
1	5cf5a7af619cd22a00a9edee	5cf59cf4e41eb48eef613d29	250	"ml"	5cf5a5d6619cd22a00a9eded
2	5cf5a8a331e6572a00ea8648	5cf59cf4e41eb48eef613d2a	250	"ml"	5cf5a6b031e6572a00ea8643

Offer

meal_ingredients				
	_id ObjectId	number_order Int32	date Date	id_staff ObjectId
1	5cf5a9fd619cd22a00a9edf2	1	2019-04-24T21:00:00.000+03:00	5cf5a98e31e6572a00ea864a
2	5cf5acfe31e6572a00ea864e	2	2019-05-24T21:00:00.000+03:00	5cf5a98e31e6572a00ea864a
3	5cf5ad1a31e6572a00ea864f	3	2019-05-25T21:09:00.000+03:00	5cf5a98e31e6572a00ea864a
4	5cf5ad3331e6572a00ea8650	4	2019-01-25T20:09:00.000+02:00	5cf5a98e31e6572a00ea864a
5	5cf5ad4e31e6572a00ea8651	5	2019-09-09T20:09:00.000+03:00	5cf5a9b431e6572a00ea864c
6	5cf5ad6531e6572a00ea8652	6	2019-10-09T20:09:00.000+03:00	5cf5a9b431e6572a00ea864c

Staff

meal_ingredients				
	_id ObjectId	name String	age Int32	position String
1	5cf5a944619cd22a00a9edf1	"Anastasia"	18	"admin"
2	5cf5a98e31e6572a00ea864a	"Edik"	19	"waiter"
3	5cf5a99f31e6572a00ea864b	"Mark"	25	"waiter"
4	5cf5a9b431e6572a00ea864c	"Eva"	21	"barista"
5	5cf5a9cb31e6572a00ea864d	"Galya"	32	"chef"

Вставляю новые значения в таблицу Staff

```
Командный рядок - mongo
>
> db.staff.insert([
...   {
...     "name": "Kate",
...     "age": "22",
...     "position": "barista",
...   }
... ])
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 1,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
> db.staff.find().pretty()
{
  "_id" : ObjectId("5cf5a944619cd22a00a9edf1"),
  "name" : "Anastasia",
  "age" : 18,
  "position" : "admin"
}
{
  "_id" : ObjectId("5cf5a98e31e6572a00ea864a"),
  "name" : "Edik",
  "age" : 19,
  "position" : "waiter"
}
{
  "_id" : ObjectId("5cf5a99f31e6572a00ea864b"),
  "name" : "Mark",
  "age" : 25,
  "position" : "waiter"
}
{
  "_id" : ObjectId("5cf5a9b431e6572a00ea864c"),
  "name" : "Eva",
  "age" : 21,
  "position" : "barista"
}
```

```
Командний рядок - mongo
    "nModified" : 0,
    "nRemoved" : 0,
    "upserted" : [ ]
  })
> db.staff.find().pretty()
{
  "_id" : ObjectId("5cf5a944619cd22a00a9edf1"),
  "name" : "Anastasia",
  "age" : 18,
  "position" : "admin"
}
{
  "_id" : ObjectId("5cf5a98e31e6572a00ea864a"),
  "name" : "Edik",
  "age" : 19,
  "position" : "waiter"
}
{
  "_id" : ObjectId("5cf5a99f31e6572a00ea864b"),
  "name" : "Mark",
  "age" : 25,
  "position" : "waiter"
}
{
  "_id" : ObjectId("5cf5a9b431e6572a00ea864c"),
  "name" : "Eva",
  "age" : 21,
  "position" : "barista"
}
{
  "_id" : ObjectId("5cf5a9cb31e6572a00ea864d"),
  "name" : "Galya",
  "age" : 32,
  "position" : "chef"
}
{
  "_id" : ObjectId("5cf5b095e41eb48eef613d2f"),
  "name" : "Kate",
  "age" : 22,
  "position" : "barista"
}
}
```

Вибірка даних з таблиці

```
> db.staff.find({"name":"Anastasia"})
{ "_id" : ObjectId("5cf5a944619cd22a00a9edf1"), "name" : "Anastasia", "age" : 18, "position" : "admin" }
```

Функція distinct

```
db.staff.distinct("name")
"Anastasia", "Edik", "Mark", "Eva", "Galya", "Kate" ]
```

Видалення інформації

```
}
> db.staff.remove({name:"Galya"})
WriteResult({ "nRemoved" : 1 })
> db.staff.find().pretty()
{
  "_id" : ObjectId("5cf5a944619cd22a00a9edf1"),
  "name" : "Anastasia",
  "age" : 18,
  "position" : "admin"
}
{
  "_id" : ObjectId("5cf5a98e31e6572a00ea864a"),
  "name" : "Edik",
  "age" : 19,
  "position" : "waiter"
}
{
  "_id" : ObjectId("5cf5a99f31e6572a00ea864b"),
  "name" : "Mark",
  "age" : 25,
  "position" : "waiter"
}
{
  "_id" : ObjectId("5cf5a9b431e6572a00ea864c"),
  "name" : "Eva",
  "age" : 21,
  "position" : "barista"
}
{
  "_id" : ObjectId("5cf5b095e41eb48eef613d2f"),
  "name" : "Kate",
  "age" : "22",
  "position" : "barista"
}
> _
```

Висновок: я здобула практичних навичок створення та обробки бази даних типу NoSQL на прикладі СУБД MongoDB.