

## Task 3 - Робота з базовими функціями документо-орієнтованої БД на прикладі MongoDB

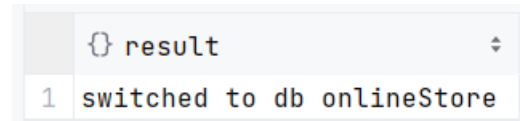
Виконав: Потужний Богдан (ФІ-03)

### Завдання:

Вам необхідно змоделювати інтернет-магазин який торгує найрізноманітнішими товарами (тобто у яких різний набір властивостей) - items.

А так же навчитися виконувати базові запити до MongoDB: <http://docs.mongodb.org/manual/core/read-operations-introduction/>

```
use onlineStore
```



### Частина 1:

#### 1. Створіть декілька товарів з різним набором властивостей Phone/TV/Smart Watch/

....

Наприклад:

```
{
  "category": "Phone",
  "model": "iPhone 6",
  "producer": "Apple",
  "price": 600
}
```

```
db.items.insertMany([
```

```
{
  "category": "Phone",
  "model": "iPhone 6",
  "producer": "Apple",
  "price": 600
},
{
  "category": "TV",
  "model": "Samsung QLED",
  "producer": "Samsung",
  "price": 1200
},
{
  "category": "Smart Watch",
  "model": "Apple Watch Series 3",
  "producer": "Apple",
  "price": 300
}
])
```

D)

filter {}	×	sort							
🔍 _id	↕	🔍 category	↕	🔍 model	↕	🔍 price	↕	🔍 producer	↕
1	6501d8afb322dd5e2124d8f9	Phone		iPhone 6		600		Apple	
2	6501d8afb322dd5e2124d8fa	TV		Samsung QLED		1200		Samsung	
3	6501d8afb322dd5e2124d8fb	Smart Watch		Apple Watch Series 3		300		Apple	

## 2. Напишіть запит, який виводить усі товари (відображення у JSON)

```
db.items.find().pretty()

1 [
2   {
3     "_id": {"$oid": "6501d8afb322dd5e2124d8f9"},
4     "category": "Phone",
5     "model": "iPhone 6",
6     "price": 600,
7     "producer": "Apple"
8   },
9   {
10    "_id": {"$oid": "6501d8afb322dd5e2124d8fa"},
11    "category": "TV",
12    "model": "Samsung QLED",
13    "price": 1200,
14    "producer": "Samsung"
15  },
16  {
17    "_id": {"$oid": "6501d8afb322dd5e2124d8fb"},
18    "category": "Smart Watch",
19    "model": "Apple Watch Series 3",
```

## 3. Підрахуйте скільки товарів у певній категорії

```
db.items.find({"category": "Phone"}).count()

1 [
2   {
3     "result": 1
4   }
5 ]
```

## 4. Підрахуйте скільки є різних категорій товарів

```
db.items.distinct("category").length

1 [
2   {
3     "result": 3
4   }
5 ]
```

## 5. Виведіть список всіх виробників товарів без повторів

```
db.items.distinct("producer")

1 [
2   {
3     "result": "Apple"
4   },
5   {
6     "result": "Samsung"
7   }
8 ]
```

6. Напишіть запити, які вибирають товари за різними критеріям і їх сукупності:
- a. категорія та ціна (в проміжку) - конструкція *\$and*,

```
db.items.find({
  $and: [
    {"category": "Phone"},
    {"price": {$gte: 500, $lte: 700}}
  ]
}).pretty()
```

```
1 [
2   {
3     "_id": {"$oid": "6501d8afb322dd5e2124d8f9"},
4     "category": "Phone",
5     "model": "iPhone 6",
6     "price": 600,
7     "producer": "Apple"
8   }
9 ]
```

- b. модель чи одна чи інша - конструкція *\$or*,

```
db.items.find({
  $or: [
    {"model": "iPhone 6"},
    {"model": "Samsung QLED"}
  ]
}).pretty()
```

	_id	category	model	price	producer
1	6501d8afb322dd5e2124d8f9	Phone	iPhone 6	600	Apple
2	6501d8afb322dd5e2124d8fa	TV	Samsung QLED	1200	Samsung

- c. виробники з переліку - конструкція *\$in*

```
db.items.find({
  "producer": {$in: ["Apple", "Samsung"]}
}).pretty()
```

	_id	category	model	price	producer
1	6501d8afb322dd5e2124d8f9	Phone	iPhone 6	600	Apple
2	6501d8afb322dd5e2124d8fa	TV	Samsung QLED	1200	Samsung
3	6501d8afb322dd5e2124d8fb	Smart Watch	Apple Watch Series 3	300	Apple

7. Оновить певні товари, змінивши існуючі значення і додайте нові властивості (характеристики) усім товарам за певним критерієм

```
db.items.updateMany(
  {"category": "Phone"},
  {
    $set: {"feature": "Water Resistant"},
    $inc: {"price": 50}
  }
)
```

	_id	category	feature	model	price	producer
1	6501d8afb322dd5e2124d8f9	Phone	Water Resistant	iPhone 6	650	Apple
2	6501d8afb322dd5e2124d8fa	TV	<unset>	Samsung QLED	1200	Samsung
3	6501d8afb322dd5e2124d8fb	Smart Watch	<unset>	Apple Watch Series 3	300	Apple

8. Знайдіть товари у яких є (присутнє поле) певні властивості

```
db.items.find({"feature": "Water Resistant"}).pretty()
```

	_id	category	feature	model	price	producer
1	6501d8afb322dd5e2124d8f9	Phone	Water Resistant	iPhone 6	650	Apple

## 9. Для знайдених товарів збільшіть їх вартість на певну суму

```
db.items.updateMany(  
  {"feature": "Water Resistant"},  
  {  
    $inc: {"price": 100}  
  }  
)
```

```
db.items.find({"feature": "Water Resistant"}).pretty()
```

	_id	category	feature	model	price	producer
1	6501d8afb322dd5e2124d8f9	Phone	Water Resistant	iPhone 6	750	Apple

## Частина 2:

Товари ви додаєте в замовлення - orders, яке містити вартість, ім'я замовника, і адресу доставки.

Товари (items) в замовленні (order) повинні бути представлені у вигляді [references](#), а замовник (customer) у вигляді [embed](#)

```
{  
  "order_number" : 201513,  
  "date" : ISODate("2015-04-14"),  
  "total_sum" : 1923.4,  
  "customer" : {  
    "name" : "Andrii",  
    "surname" : "Rodionov",  
    "phones" : [ 9876543, 1234567],  
    "address" : "PTI, Peremohy 37, Kyiv, UA"  
  },  
  "payment" : {  
    "card_owner" : "Andrii Rodionov",  
    "cardId" : 12345678  
  },  
  "order_items_id" : [  
    {  
      "$ref" : "items",  
      "$id" : ObjectId("552bc0f7bbcdf26a32e99954")  
    },  
    {  
      "$ref" : "items",  
      "$id" : ObjectId("552bc285bbcdf26a32e99957")  
    }  
  ]  
}
```

1. Створіть кілька замовлень з різними наборами товарів, але так щоб один з товарів був у декількох замовленнях

```
db.orders.insertMany([
  {
    "order_number": 201513,
    "date": "2015-04-14",
    "total_sum": 2250,
    "customer": {
      "name": "Andrii",
      "surname": "Rodinov",
      "phones": [9876543, 1234567],
      "address": "PTI, Peremohy 37, Kyiv, UA"
    },
    "payment": {
      "card_owner": "Andrii Rodionov",
      "cardId": 12345678
    },
    "order_items_id": [
      {
        "$ref": "items",
        "$id": ObjectId("6501d8afb322dd5e2124d8f9")
      },
      ...
      {
        "$ref": "items",
        "$id": ObjectId("6501d8afb322dd5e2124d8f9")
      }
    ]
  }
])
```

2. Виведіть всі замовлення

```
db.orders.find().pretty()
```

	_id	customer	date	order_items_id
1	6501de7db322dd5e2124d900	{"name": "Andrii", "surname": "Rodinov", "	2015-04-14	["{ \"\$ref\" : \"items\", \"\$id\"
2	6501de7db322dd5e2124d901	{"name": "Bohdan", "surname": "Potuzhnyi",	2015-04-15	["{ \"\$ref\" : \"items\", \"\$id\"
3	6501de7db322dd5e2124d902	{"name": "Test", "surname": "Testov", "pho	2015-04-16	["{ \"\$ref\" : \"items\", \"\$id\"

3. Виведіть замовлення з вартістю більше певного значення

```
db.orders.find({"total_sum": {$gt: 1500}}).pretty()
```

	_id	customer	date	order_items_id
1	6501de7db322dd5e2124d900	{"name": "Andrii", "surname": "Rodinov", "	2015-04-14	["{ \"\$ref\" : \"items\", \"\$id\"
2	6501de7db322dd5e2124d901	{"name": "Bohdan", "surname": "Potuzhnyi",	2015-04-15	["{ \"\$ref\" : \"items\", \"\$id\"

4. Знайдіть замовлення зроблені одним замовником

```
db.orders.find({"customer.name": "Andrii"}).pretty()
```

	_id	customer	date	order_items_id
1	6501de7db322dd5e2124d900	{"name": "Andrii", "surname": "Rodinov", "	2015-04-14	["{ \"\$ref\" : \"items\", \"\$id\"

5. Знайдіть всі замовлення з певним товаром (товарами) (шукати можна по ObjectId)

```
db.orders.find({"order_items_id.$id": ObjectId("6501d8afb322dd5e2124d8fa")}).pretty()
```

	_id	customer	date	order_items_id
1	6501de7db322dd5e2124d900	{"name": "Andrii", "surname": "Rodinov", "	2015-04-14	["{ \"\$ref\" : \"items\", \"\$id\"
2	6501de7db322dd5e2124d901	{"name": "Bohdan", "surname": "Potuzhnyi",	2015-04-15	["{ \"\$ref\" : \"items\", \"\$id\"

6. Додайте в усі замовлення з певним товаром ще один товар і збільште існуючу вартість замовлення на деяке значення X

```
order_items_id
[{"$ref": "items", "$id": "6501d8afb322dd5e2124d8f9"}, {"$ref": "items", "$id": "6501d8afb322dd5e2124d8fa"}, {"$ref": "items", "$id": "6501d8afb322dd5e2124d8fb"}, {"$ref": "items", "$id": "6501d8afb322dd5e2124d8fc"}]
```

```
db.orders.updateMany(
  {"order_items_id.$id": ObjectId("6501d8afb322dd5e2124d8fa")},
  {
    $push: {"order_items_id": {"$ref": "items", "$id":
ObjectId("6501d8afb322dd5e2124d8fb")}},
    $inc: {"total_sum": 300}
  }
)
```

```
order_items_id
1 {"$ref": "items", "$id": "6501d8afb322dd5e2124d8f9"}, {"$ref": "items", "$id": "6501d8afb322dd5e2124d8fa"}, {"$ref": "items", "$id": "6501d8afb322dd5e2124d8fb"}, {"$ref": "items", "$id": "6501d8afb322dd5e2124d8fc"}]
```

7. Виведіть кількість товарів в певному замовленні

```
db.orders.find({"order_number": 201513}).toArray()[0].order_items_id.length
```

```
result
1 5
```

```
"order_items_id": [{" $ref": "items", "$id": "6501d8afb322dd5e2124d8f9"}, {" $ref": "items", "$id": "6501d8afb322dd5e2124d8fa"}, {" $ref": "items", "$id": "6501d8afb322dd5e2124d8fb"}, {" $ref": "items", "$id": "6501d8afb322dd5e2124d8fc"}],
"order_number": 201513,
```

8. Виведіть тільки інформацію про кастомера і номери кредитної карт, для замовлень вартість яких перевищує певну суму

```
db.orders.find(
  {"total_sum": {"$gt": 1500}},
  {"customer": 1, "payment.cardId": 1, "_id": 0}
).pretty()
```

```
customer payment
1 {"name": "Andrii", "surname": "Rodinov", "cardId": new NumberInt("12345678")}
2 {"name": "Bohdan", "surname": "Potuzhnyi", "cardId": new NumberInt("123456789")}
```

## 9. Видаліть товар з замовлень, зроблених за певний період дат

	÷	{ date	÷	{ order_items_id
1	:	"Rodinov", "phones": [	2015-04-14	[{"{"\$ref\" : \"items\", \"\$id\" : \"6501d8afb322dd5e2124d8f9\" }\", \"{ \"\$ref\" : \"items\", \"\$
2	:	"Potuzhnyi", "phones":	2015-04-15	[{"{"\$ref\" : \"items\", \"\$id\" : \"6501d8afb322dd5e2124d8f9\" }\", \"{ \"\$ref\" : \"items\", \"\$
3	:	'Testov', "phones": [new	2015-04-16	[{"{"\$ref\" : \"items\", \"\$id\" : \"6501d8afb322dd5e2124d8f9\" }"]

```
db.orders.updateMany(
  {
    "date": { $gte: "2015-04-15", $lte: "2015-04-20" }
  },
  {
    $pull: { "order_items_id": {"$id":
ObjectId("6501d8afb322dd5e2124d8f9")} }
  }
)
```

{ date	÷	{ order_items_id
2015-04-14	:	[{"{"\$ref\" : \"items\", \"\$id\" : \"6501d8afb322dd5e2124d8f9\" }\", \"{ \"\$ref\" : \"items\",
2015-04-15	:	[{"{"\$ref\" : \"items\", \"\$id\" : \"6501d8afb322dd5e2124d8fa\" }\", \"{ \"\$ref\" : \"items\",
2015-04-16	:	[ ]

## 10. Переіменуйте у всіх замовлення ім'я (прізвище) замовника

```
db.orders.updateMany(
  { "customer.surname": "Rodinov" },
  { $set: { "customer.surname": "Smith" } }
)
```

	<code>{ _id</code>	<code>{ customer</code>	<code>{ date</code>
1	6501de7db322dd5e2124d900	<code>{"name": "Andrii", "surname": "Smith", "phones": [ne</code>	2015-04
2	6501de7db322dd5e2124d901	<code>{"name": "Bohdan", "surname": "Potuzhnyi", "phones":</code>	2015-04
3	6501de7db322dd5e2124d902	<code>{"name": "Test", "surname": "Testov", "phones": [new</code>	2015-04

## 11. (+2 бали)\* Знайдіть замовлення зроблені одним замовником, і виведіть тільки інформацію про кастомера та товари у замовленні підставивши замість ObjectId("\*\*\*\*") назви товарів та їх вартість (аналог join-а між таблицями *orders* та *items*).

```
db.orders.aggregate([
  {
    $lookup: {
      from: "items",
      localField: "order_items_id.$id",
      foreignField: "_id",
      as: "products"
    }
  },
  {
    $project: {
      "customer": 1,
      "products.model": 1,
      "products.price": 1
    }
  }
]).pretty()
```

	{ _id	{ customer	{ products
1	6501de7db322dd5e2124d900	{"name": "Andrii", "surname": "Smith", "ph	[{"model": "iPhone 6", "price": new Number
2	6501de7db322dd5e2124d901	{"name": "Bohdan", "surname": "Potuzhnyi",	[{"model": "Samsung QLED", "price": new Nu
3	6501de7db322dd5e2124d902	{"name": "Test", "surname": "Testov", "pho	[]

### Частина 3:

Створіть [Capped collection](#) яка б містила 5 останніх відгуків на наш інтернет-магазин. Структуру запису визначіть самостійно.

```
db.createCollection("reviews", {capped: true, size: 1000000, max: 5})
```

1. Перевірте що при досягненні обмеження старі відгуки будуть затиратись

```
for (let i = 1; i <= 6; i++) {
  db.reviews.insertOne({"review": "Review " + i})
}
```

//Гіпотетично відгук 1 має бути відсутнім

```
db.reviews.find().pretty()
```

	{ _id	{ review
1	6501e5cab322dd5e2124d907	Review 2
2	6501e5cab322dd5e2124d908	Review 3
3	6501e5cab322dd5e2124d909	Review 4
4	6501e5cab322dd5e2124d90a	Review 5
5	6501e5cab322dd5e2124d90b	Review 6

Так і є, відгук 1 не було виведено.