

# Лабораторна робота 10

## Документи та індекси в MongoDB

**Мета роботи:** відпрацювати навички наповнення бази даних контентом, створення індексів та отримання необхідної інформації з бази.

### ЗАВДАННЯ:

Виконати всі завдання на лабораторну роботу.

### ЗМІСТ ЗВІТУ:

Скріншот виконання завдань на лабораторну роботу.

## ПОРЯДОК ВИКОНАННЯ

**Завдання 1.** Створіть базу даних, з якої ви надалі працюватимете

```
use laptopsShop;
```

**Завдання 2.** Створіть колекцію, в якій у вас будуть зберігатися документи .

```
db.createCollection("laptops");
```

**Завдання 3.** Наповніть колекцію документами. Для виконання даного пункту використовувався скрипт, результат якого знаходиться у файлі *lab\_4\_laptops\_docs.json* на ПНС.

**Завдання 4.** Створіть складовий та повнотекстовий індекси для документів у вашій колекції .

Запит на створення складового індексу на основі полів *specs.manufacturer* та *in\_stock*

```
db.laptops.ensureIndex(  
  {  
    "specs.manufacturer": 1,  
    "in_stock": 1  
  }  
);
```

Запит на створення повнотекстового індексу

```
db.laptops.ensureIndex(  
  {  
    "in_stock": 1  
  }  
);
```

Виведення індексів для колекції *laptops*:

```
db.laptops.getIndexes ();
```

**Завдання 5.** Створіть запит, що виробляє оновлення будь-якого поля документа та запит, що додає до нього нові поля.

Запит на зміну значення поля *in\_stock* для ноутбука з *\_id=ObjectId('63f2690611153ea7202a9ee6')*:

```
db.laptops.updateOne({_id:  
ObjectId('63f2690611153ea7202a9ee6')}), {$set:  
{in_stock: false}});
```

Запит на додавання поля *bestseller* для ноутбука з *\_id=ObjectId('63f2690611153ea7202a9ee6')*:

```
db.laptops.updateOne({_id:  
ObjectId('63f2690611153ea7202a9ee6')}), {$set:  
{bestseller: true}});
```

**Завдання 6.** Створіть запит для отримання всіх документів колекції та визначте, використовується чи для нього індекс .

Запит для отримання всіх документів у колекції *laptops*:

```
db.laptops.find ( );
```

Команда для отримання інформації про обробку запиту

```
db.laptops.find().explain();
```

Виконавши запит `db.laptops.find({in_stock: true}).explain();` ми можемо побачити, що для нього вже використовується один індекс, створений нами раніше `indexName: 'in_stock_1'`

**Завдання 7.** Створіть пару простих запитів для вибірки даних з БД

- Запит на вибірку ноутбуків , виробник яких "HP":

```
db.laptops.find ({" specs.manufacturer " : "HP"})
```

- Запит на вибірку наявних ноутбуків:

```
db.laptops.find ({" in_stock " : true })
```

**Завдання 8.** Створіть складні запити з кожним з перерахованих модифікаторів

- Запит на вибірку ноутбуків , у яких виробник " Apple " і він є в наявності

```
db.laptops.find({"specs.manufacturer": "Apple",  
"in_stock": true})
```

- Запит на вибірку ноутбуків, у яких виробник "Apple" або "Samsung"

```
db.laptops.find({$or: [{"specs.manufacturer":  
"Apple"}, {"specs.manufacturer": "Samsung"}]})
```

- Запит на вибірку кількості ноутбуків до 10000.0 грн

```
db.laptops.find ({"price": {$lt: 10000.0}})
```

- Запит на вибірку кількості ноутбуків від 15000.0 грн

```
db.laptops.find ({"price": {$gt: 15000.0}})
```

- Запит на вибірку кількості ноутбуків НЕ від " Apple "

```
db.laptops.find({"specs.manufacturer": {$ne: "Apple"}})
```

- Запит на вибірку кількості ноутбуків , у яких виробник "Apple", "HP" або " Samsung "

```
db.laptops.find ({" specs.manufacturer ": {$in: ["Apple", "HP", "Samsung"]}})
```

**Завдання 9.** Створіть запити з використанням методів сортування, обмеження та пропуску даних

- Запит на вибірку всіх унікальних виробників ноутбуків, в алфавітному порядку (за зростанням )

```
db.laptops.distinct("specs.manufacturer")
```

- Запит на вибірку топ 5 найдорожчих ноутбуків (за спаданням)

```
db.laptops.find({}, {"name": 1, "price": 1}).sort({"price": -1}).limit(5);
```

- Запит на вибірку топ 5 самих дешевих ноутбуків, пропустивши 10 самих дешевих

```
db.laptops.find({}, {"name": 1, "price": 1}).sort({"price": 1}).skip(10).limit(5);
```

**Завдання 10.** Створіть кілька запитів для вставки даних у масив

- Запит на додавання кольору "orange" для всіх ноутбуків від Apple

```
db.laptops.updateMany({"specs.manufacturer":  
"Apple"}, { $push: {colors: 'orange'}});
```

- Запит на додавання кольору titan для всіх ноутбуків від 10000.0 руб. і вище

```
db.laptops.updateMany({"price": {$gte: 10000.0}},  
{$push: {colors: 'titan'}});
```

**Завдання 11.** Створіть запити, що проводять оновлення даних у масиві: як за позицією елемента в масиві, так і за його значенням

- Запит на оновлення значення "red" у масиві "colors" на "sky blue" для всіх ноутбуків

```
db.laptops.updateMany({"colors": "red"}, {$set:  
{"colors.$": "sky blue"}});
```

- Запит на оновлення другого значення (index = 1) у масиві "colors" на "orange" для всіх ноутбуків

```
db.laptops.updateMany({}, {$set: {"colors.1":  
"orange"}});
```

**Завдання 12.** Створіть запити, що видаляють елементи з масиву: за позицією елемента в масиві та за його значенням.

- Запит на видалення значення "sky blue" у масиві "colors" для всіх ноутбуків

```
db.laptops.updateMany({}, { $pull: { "colors": "sky  
blue" } });
```

- Запит на видалення першого значення (index = 1) у масиві  
"colors" для всіх ноутбуків

```
db.laptops.updateMany({}, { $unset: { "colors.0": 1 } });  
db.laptops.updateMany({}, { $pull: { "colors": null } });
```