

Лабораторна 4 - Робота з базовими функціями документо-орієнтованої БД на прикладі MongoDB

Установка:

- Встановіть MongoDB локально або на віртуальну машину (для наступних завдань необхідно буде запускати декілька екземплярів MongoDB)
<https://www.mongodb.org/downloads>
- [MongoDB Docker](#)
- В якості графічного клієнта по роботі з MongoDB пропонується використовувати Robomongo (<http://robomongo.org/>) або аналогічний клієнт
- Без установки з MongoDB можна працювати онлайн на сайті <http://try.mongodb.org/>

Завдання:

Вам необхідно змодельувати 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/>

- 1) Створіть декілька товарів з різним набором властивостей Phone/TV/Smart Watch/

Наприклад:

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

- 2) Напишіть запит, який виводить усі товари (відображення у JSON)
- 3) Підрахуйте скільки товарів у певної категорії
- 4) Підрахуйте скільки є різних категорій товарів
- 5) Виведіть список всіх виробників товарів без повторів
- 6) Напишіть запити, які вибирають товари за різними критеріям і їх сукупності:
 - а) категорія та ціна (в проміжку) - конструкція *\$and*,
 - б) модель чи одна чи інша - конструкція *\$or*,
 - в) виробники з переліку - конструкція *\$in*
- 7) Оновіть певні товари, змінивши існуючі значення і додайте нові властивості (характеристики) усім товарам за певним критерієм
- 8) Знайдіть товари у яких є (присутнє поле) певні властивості
- 9) Для знайдених товарів збільшіть їх вартість на певну суму

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](#)

Приклад моделювання:

<https://docs.mongodb.com/manual/tutorial/model-referenced-one-to-many-relationships-between-documents/>

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

- 1) Створіть кілька замовлень з різними наборами товарів, але так щоб один з товарів був у декількох замовленнях
- 2) Виведіть всі замовлення
- 3) Виведіть замовлення з вартістю більше певного значення
- 4) Знайдіть замовлення зроблені одним замовником
- 5) Знайдіть всі замовлення з певним товаром (товарами) (шукати можна по ObjectId)
- 6) Додайте в усі замовлення з певним товаром ще один товар і збільште існуючу вартість замовлення на деяке значення X
- 7) Виведіть кількість товарів в певному замовленні
- 8) Виведіть тільки інформацію про кастомера і номери кредитної карт, для замовлень вартість яких перевищує певну суму
- 9) Видаліть товар з замовлень, зроблених за певний період дат
- 10) Переіменуйте у всіх замовлення ім'я (прізвище) замовника
- 11) 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*).

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

- 1) Перевірте що при досягненні обмеження старі відгуки будуть затиратись

Вимогу до оформлення протоколу:

Завдання здається особисто без протоколу, або надсилається протокол який має містити:

- команди та результати їх виконання