# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ТАРАСА ШЕВЧЕНКА



## ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**Кафедра прикладних інформаційних систем**

**Звіт до лабораторної роботи №5**

# з курсу

**«Бази даних»**

*Студента 2 курсу*

*групи ПП-21 спеціальності 122 «Комп'ютерні науки» ОП «Прикладне програмування»*

%username%

*Викладач:*

Краснощок В.М.

## Київ – 202

**1.Назва роботи**

Проектування постреляційних баз даних.

1. **Тема роботи**

Проектування постреляційних баз даних.

1. **Мета роботи**

Набуття навичок по проектуванню постреляційних баз даних. Створення власної бази даних в СУБД MongoDB. Отримання даних з використанням запитів.

1. **Умова завдання**

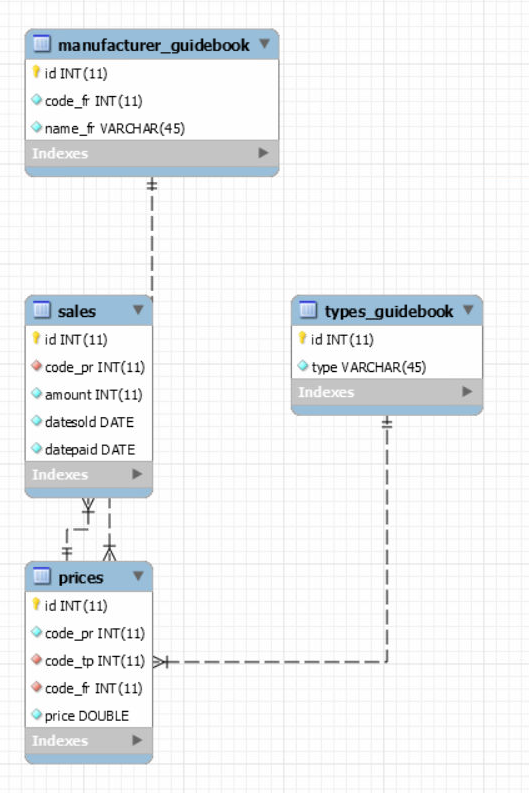
Відповідно до обраного варіанту та індивідуального завдання спроектувати 2 схеми документів для заповнення бази даних в СУБД MongoDB (на 2 та 3 колекції). Написати скрипти по перенесенню даних в СУБД. Отримати інформацію про колекції та базу даних.

Отримати інформацію про найбільш вживаних методів роботи з об’єктами бази даних та методів для роботи з колекціями.

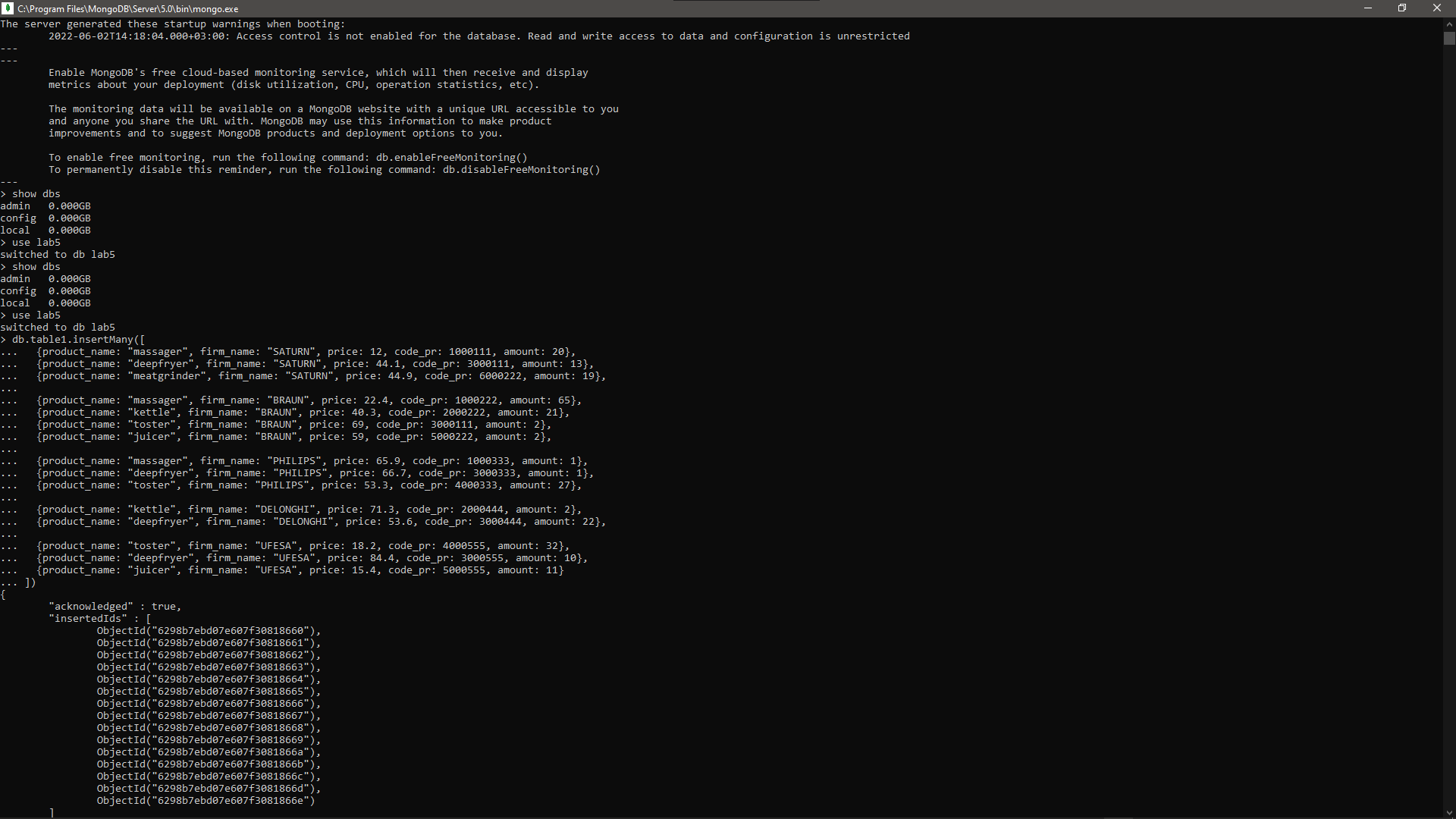
Виконати запити до однії (довільної) схеми відповідно до завдання та отримати інформацію про використовувані методи на JavaScript

1. **Рішення**

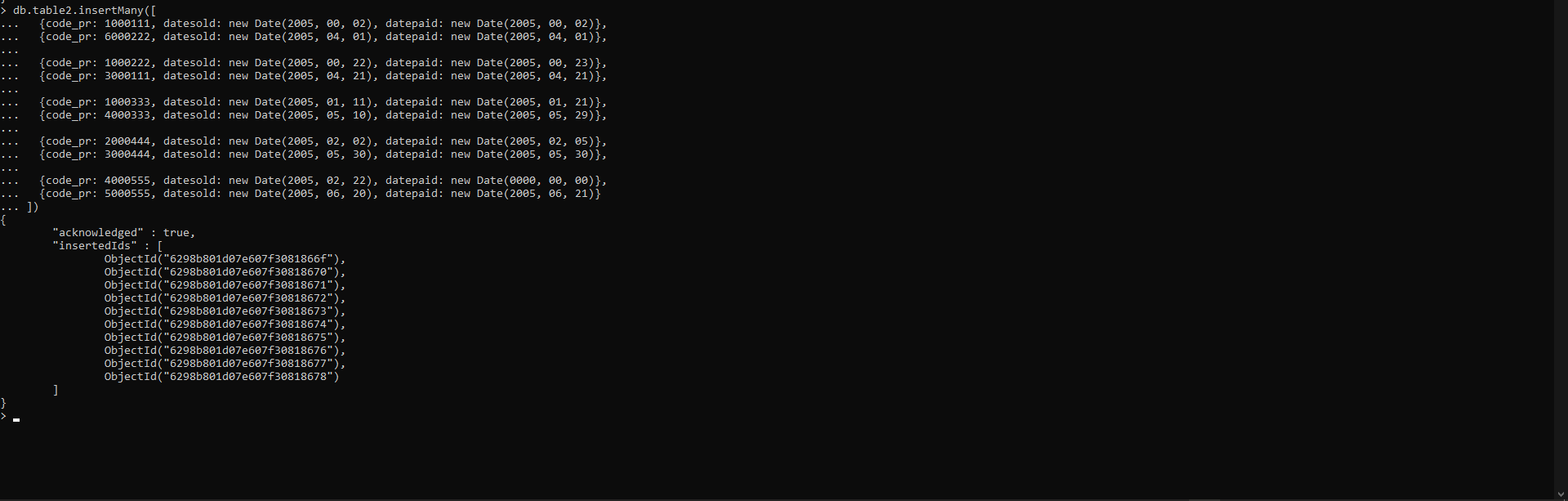
Діаграма:



Створення коллекцій:

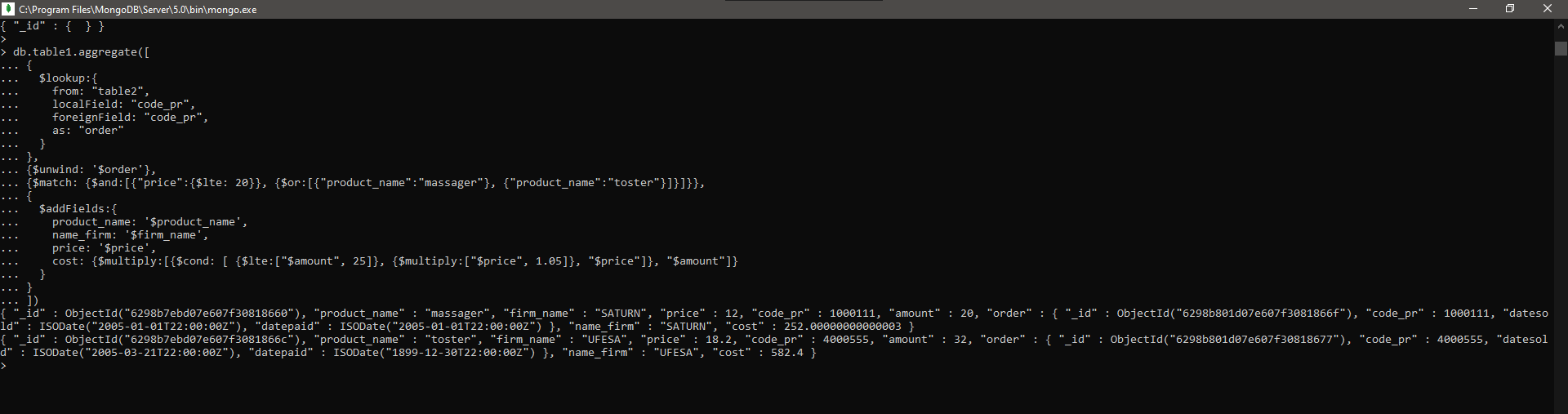


Перша коллекція (таблиці prices, manufacturer\_guidebook та types\_guidebook)



Друга коллекція (таблиця sales)

Запит 1.1:



db.table1.aggregate([

{

$lookup:{

from: "table2",

localField: "code\_pr",

foreignField: "code\_pr",

as: "order"

}

},

{$unwind: '$order'},

{$match: {$and:[{"price":{$lte: 20}}, {$or:[{"product\_name":"massager"}, {"product\_name":"toster"}]}]}},

{

$addFields:{

product\_name: '$product\_name',

name\_firm: '$firm\_name',

price: '$price',

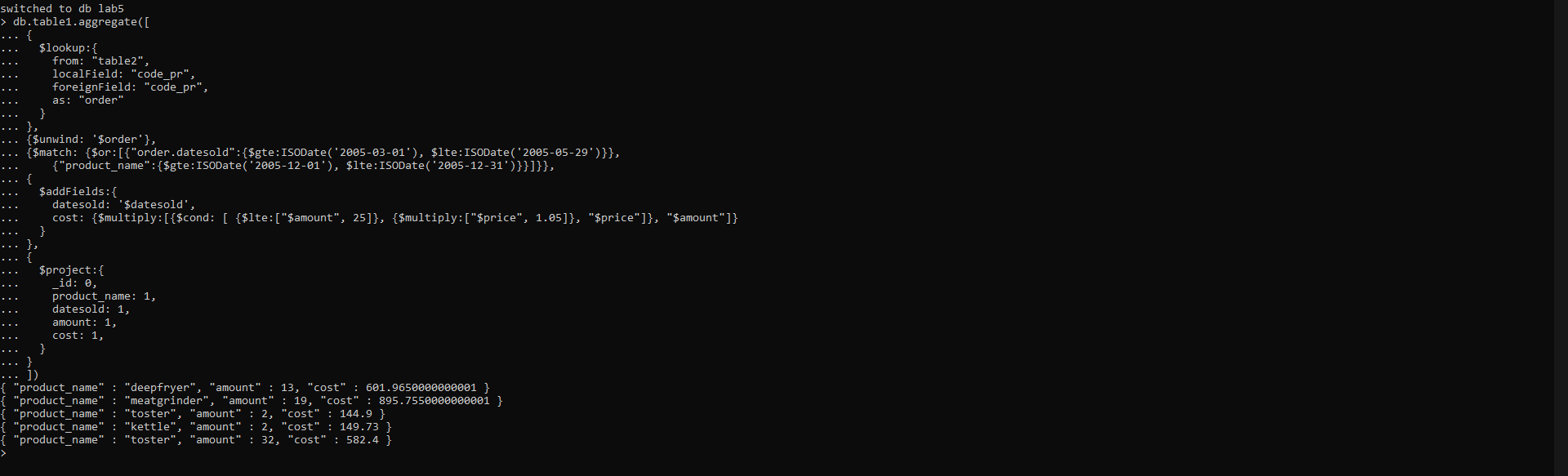
cost: {$multiply:[{$cond: [ {$lte:["$amount", 25]}, {$multiply:["$price", 1.05]}, "$price"]}, "$amount"]}

}

}

])

Запит 1.2:



db.table1.aggregate([

{

$lookup:{

from: "table2",

localField: "code\_pr",

foreignField: "code\_pr",

as: "order"

}

},

{$unwind: '$order'},

{$match: {$or:[{"order.datesold":{$gte:ISODate('2005-03-01'), $lte:ISODate('2005-05-29')}},

{"product\_name":{$gte:ISODate('2005-12-01'), $lte:ISODate('2005-12-31')}}]}},

{

$addFields:{

datesold: '$datesold',

cost: {$multiply:[{$cond: [ {$lte:["$amount", 25]}, {$multiply:["$price", 1.05]}, "$price"]}, "$amount"]}

}

},

{

$project:{

\_id: 0,

product\_name: 1,

datesold: 1,

amount: 1,

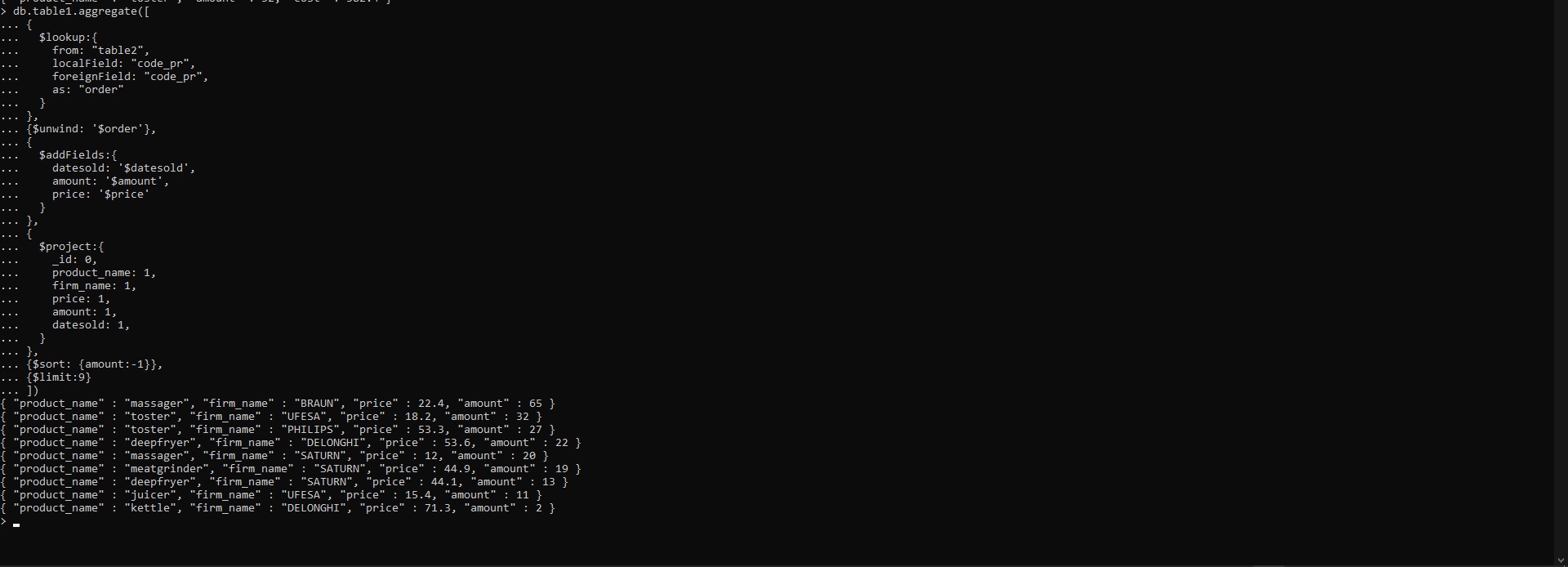
cost: 1,

}

}

])

Запит 1.3:



db.table1.aggregate([

{

$lookup:{

from: "table2",

localField: "code\_pr",

foreignField: "code\_pr",

as: "order"

}

},

{$unwind: '$order'},

{

$addFields:{

datesold: '$datesold',

amount: '$amount',

price: '$price'

}

},

{

$project:{

\_id: 0,

product\_name: 1,

firm\_name: 1,

price: 1,

amount: 1,

datesold: 1,

}

},

{$sort: {amount:-1}},

{$limit:9}

])

**Контрольні питання**

1.Основні характеристики СУБД MongoDB.

MongoDB - документоорієнтована система управління базами даних, яка не потребує опису схеми таблиць. Вважається одним з класичних прикладів NoSQL-систем, використовує JSON-подібні документи і схему бази даних. Написана на мові C++. Застосовується в веб-розробці, зокрема, в рамках JavaScript-орієнтованого стека MEAN.

2.Операції CRUD в MongoDB.

Для додавання використовують команду insertOne або insertMany

Для читання використовують команду find

Для оновлення інформації update, updateMany i replaceOne

Для видалення використовують drop або remove

3.Значення команди find().

Команда дозволяє виводити інформацію про колекцію. За бажанням, можна додати певні умови та виводити тільки необхідні поля.

4.Правила формування складних запитів.

Складний запит можна формувати у декілька способів. Якщо він до однієї колекції, то ми прописуємо у першому документі необхідні умови, а у другому налаштовуємо відображення полів. Якщо запит потребує інформацію з іншої колекції – зберігаємо результат запиту в змінну та використовуємо його в іншому запиті як умову.

Також, можна додатково опрацьовувати інформацію додатковими командами після find. Варто зазначити, що у такому випадку порядок таких команд важливий

5.Побудова індексів та команда explain().

db.firms.ensureIndex({name\_firm:1}) – створити,

db.firms.ensureIndex({name\_firm:1},{unique:true}) – створити унікальний,

db.firms.getIndexes() – переглянути.

6.Отримання довідки в MongoDB.

Команда db.help() виводить список найбільш вживаних методів роботи з об’єктами бази даних. Список методів для роботи з колекціями виводить команда db.foo.help().

db.foo.get + TAB

виводить список методів роботи з колекціями, імена яких починаються словом get.

7.Принципи проектування схеми бази даних в MongoDB.

Процес проектування БД являє собою послідовність переходів від неформального мовного опису інформаційної структури предметної області до формалізованого опису об'єктів предметної області в термінах деякої моделі.

Проектування БД складається з таких етапів:

•системний аналіз предметної області;

•концептуальне проектування;

•логічне проектування;

•фізичне проектування.

Системний аналіз передбачає мовний опис реальних об'єктів предметної області, визначення зв'язків між об'єктами, дослідження характеристик об'єктів і зв'язків. Результати дослідження використовуються при концептуальному проектуванні БД.

Концептуальне проектування полягає в створенні концептуальної моделі, яку відображає концептуальна схема БД. На цьому етапі визначаються об'єкти, зв'язки між об'єктами, атрибути, ключові атрибути.

Логічне проектування полягає в створенні логічної моделі на основі вибраної моделі даних. На цьому етапі необхідно вже знати яка СУБД буде застосовуватися в системі (ієрархічна, мережна, реляційна, об'єктно-орієнтована). Для перевірки вірності логічної моделі застосовується нормалізація. Крім того логічна модель перевіряється на умову забезпечення всіх транзакцій користувачів (для РСУБД).

Фізичне проектування полягає в описі засобів фізичної реалізації логічного проекту БД. Фізичні моделі визначають засоби розміщення даних в середовищі зберігання і засоби доступу до цих даних, які підтримуються на фізичному рівні.

8.Принципи встановлення ключових полів в MongoDB.

При додаванні даних поле \_id створюється автоматично (це ключ в кожній колекції) (його можна задавати і самостійно)

У старших чотирьох байтах знаходиться стандартна часова мітка Unix, що містить кількість секунд від початку "епохи". Наступні три байта - ідентифікатор комп’ютера, а наступні два - ідентифікатор процесу. Останні три байта - локальний для процесу лічильник, який збільшується при генерації кожного наступного ідентифікатора об’єкта.

1. **Висновки**

В результаті виконання даної лабораторної роботи я набув навички по проектуванню постреляційних баз даних. Створив власну базу даних в СУБД MongoDB. Отримав дані, використовуючи запити. Вважаю дану лабораторну роботу виконаною в повному обсязі.