**Лабораторна робота №2**

**З дисципліни:** Бази даних та інформаційні системи  
 **Студента групи МІТ-31:** Якоба Р.В.

**Тема:** Практичне використання Aggregation Framework у MongoDB

**Мета роботи:** Закріпити знання про основні стадії Aggregation Framework. Навчитися будувати ефективні агрегаційні запити. Освоїти методи фільтрації, групування, сортування та обробки масивів у MongoDB. Практично працювати з $match, $group, $sort, $unwind, $lookup, $project. Аналізувати продуктивність агрегацій та оптимізувати запити.

**Завдання:**

1. Створити колеції orders, customers, та products

Код: db.orders.insertOne({

"orderId": "ORD001",

"customerId": ObjectId("65f1a3d5a123456789abcd01"),

"date": ISODate("2024-01-12"),

"items": [

{ "product": "Laptop", "quantity": 1, "price": 1200 },

{ "product": "Mouse", "quantity": 2, "price": 50 }

],

"status": "Completed"

})

db.customers.insertOne({

"\_id": ObjectId("65f1a3d5a123456789abcd01"),

"name": "John Doe",

"email": "john.doe@example.com",

"city": "New York",

"registeredAt": ISODate("2021-03-15")

})

db.products.insertOne({

"name": "Laptop",

"category": "Electronics",

"price": 1200,

"stock": 15

})

1. Частина 1. Базові агрегаційні операції

* Відфільтруйте замовлення за останні 3 місяці

Код: db.orders.aggregate([

{

$match: {

date: { $gte: new Date(new Date().setMonth(new Date().getMonth() - 3)) }

}

}

])

* Групування замовлень за місяцем

Код: db.orders.aggregate([

{

$group: {

\_id: { $month: "$date" },

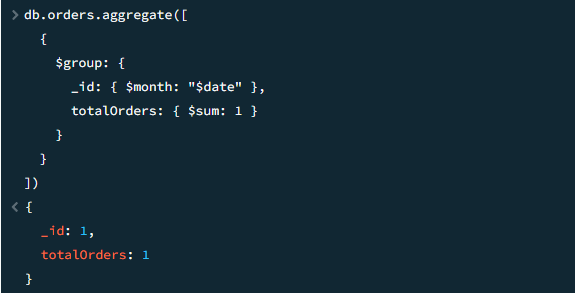
totalOrders: { $sum: 1 }

}

}

])

Результат:



* Сортування за сумою замовлення

Код: db.orders.aggregate([

{

$addFields: {

totalAmount: {

$sum: {

$map: {

input: "$items",

as: "item",

in: { $multiply: ["$$item.quantity", "$$item.price"] }

}

}

}

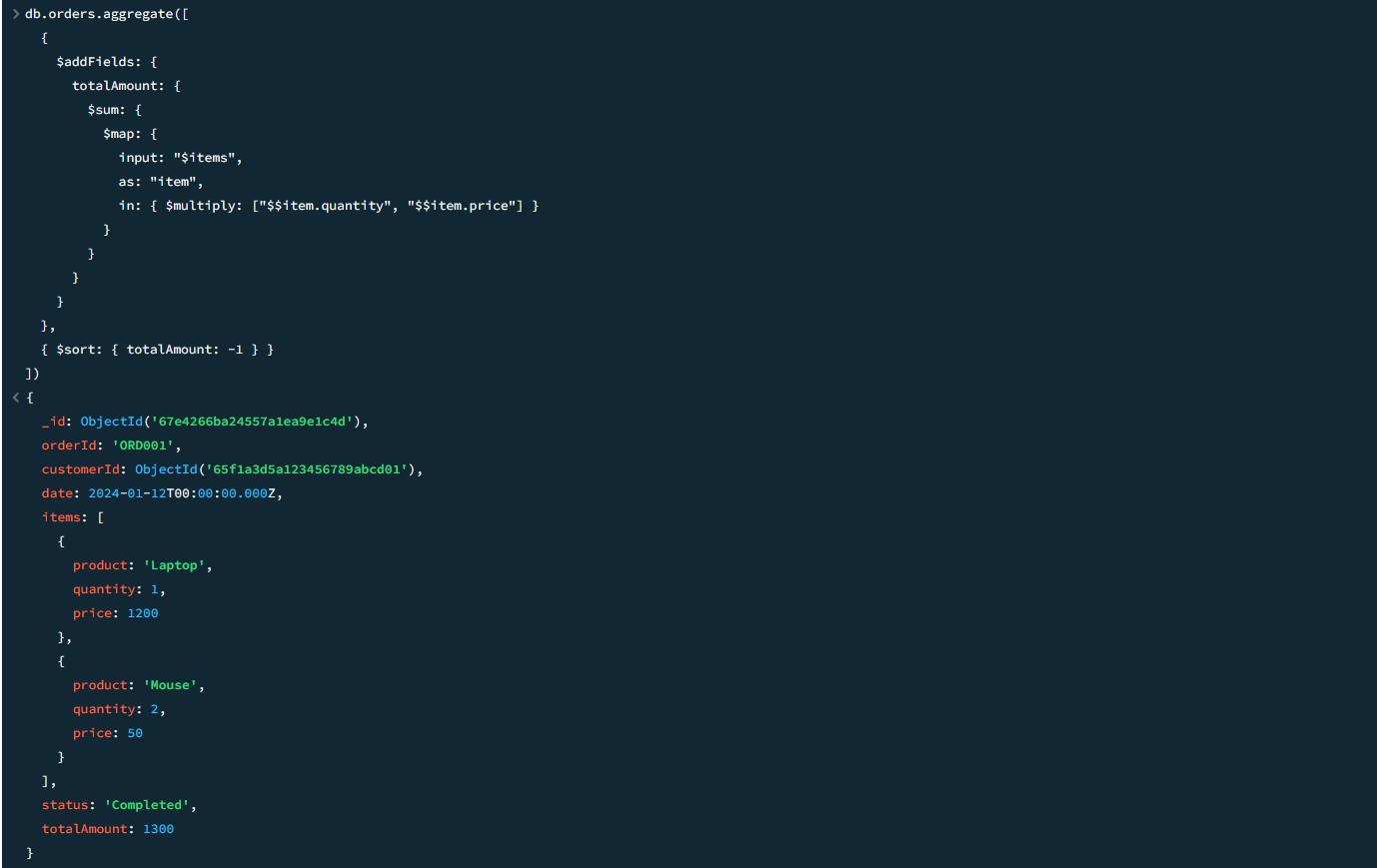
}

},

{ $sort: { totalAmount: -1 } }

])

Результат:



1. Частина 2: Робота з масивами

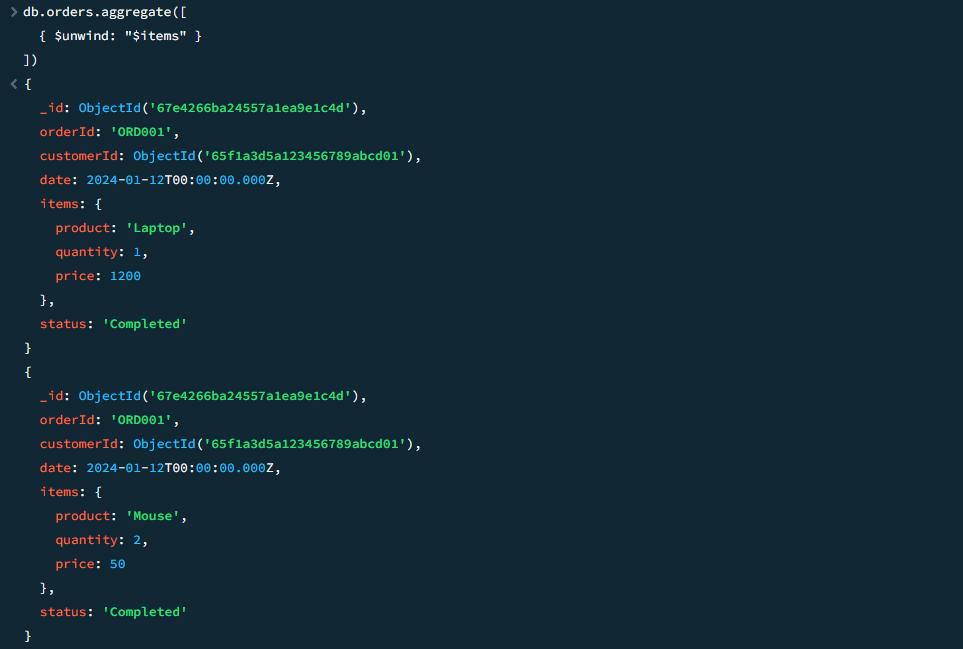
* Розгорніть масив items у замовленнях

Код: db.orders.aggregate([

{ $unwind: "$items" }

])

Результат:



* Підрахуйте кількість проданих одиниць товарів

Код: db.orders.aggregate([

{ $unwind: "$items" },

{

$group: {

\_id: "$items.product",

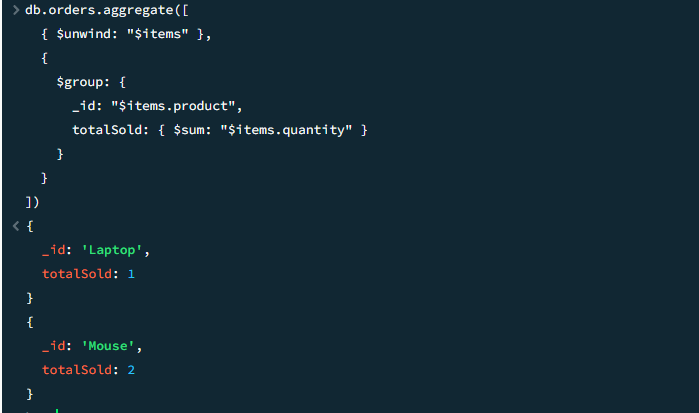
totalSold: { $sum: "$items.quantity" }

}

}

])

Результат:



1. Частина 3: З’єднання колекцій ($lookup)

* Отримання інформації про клієнтів у замовленнях

Код: db.orders.aggregate([

{

$lookup: {

from: "customers",

localField: "customerId",

foreignField: "\_id",

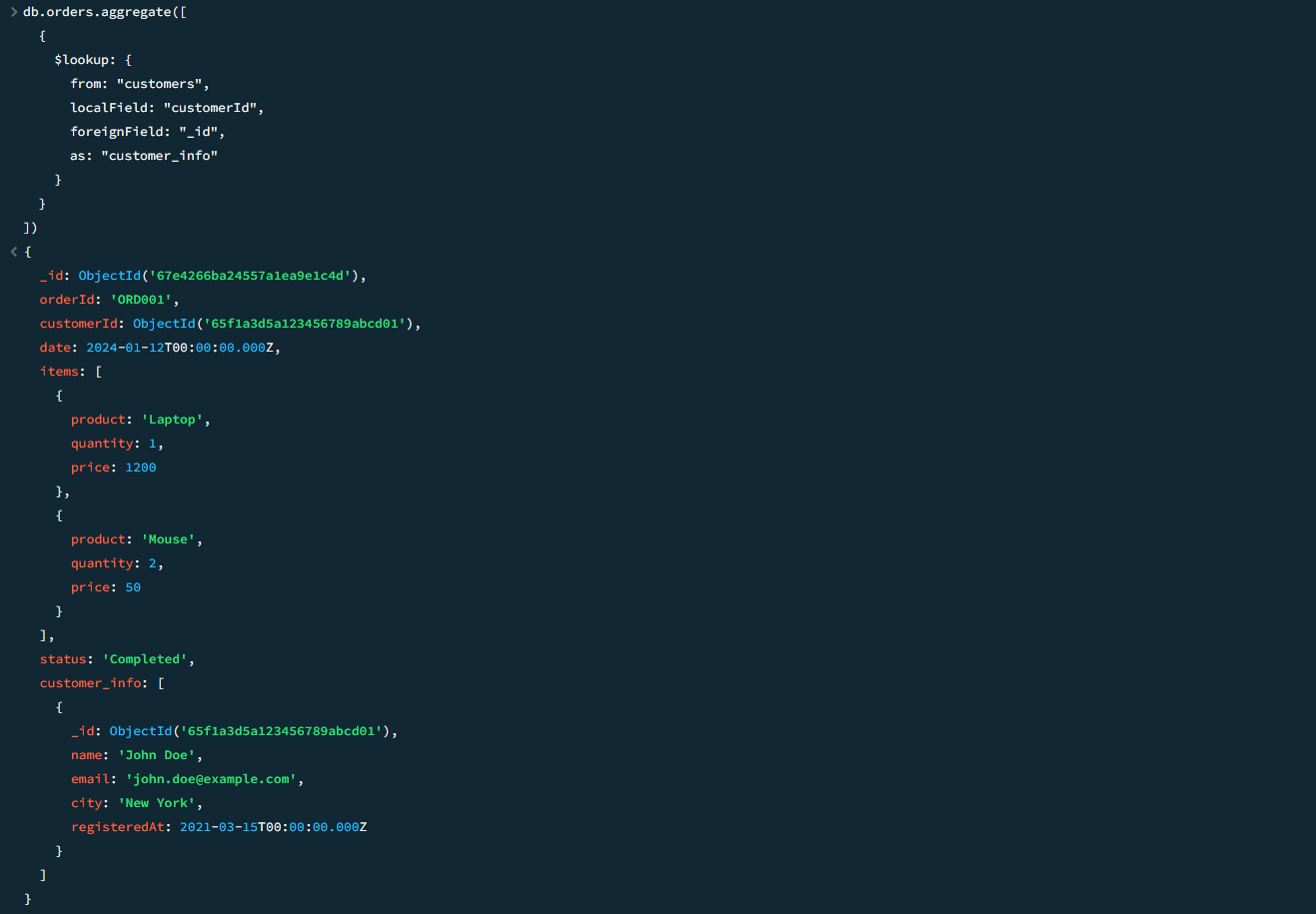
as: "customer\_info"

}

}

])

Результат:



* Визначте найбільш активних клієнтів

Код: db.orders.aggregate([

{

$group: {

\_id: "$customerId",

totalOrders: { $sum: 1 }

}

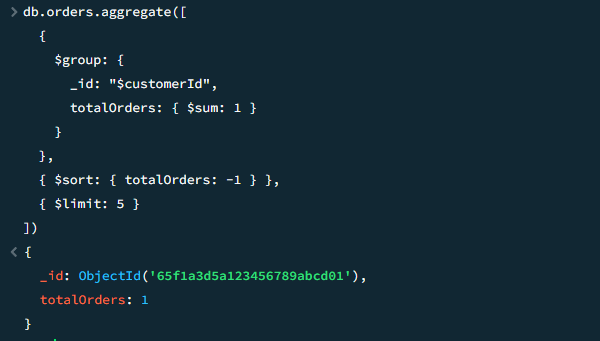
},

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

{ $limit: 5 }

])

Результат:



1. Частина 4: Оптимізація запитів

* Перевірте продуктивність запиту

Код: db.orders.explain("executionStats").aggregate([

{ $match: { status: "Completed" } }

])

Результат:



* Оптимізуйте агрегаційний запит

(Оптимізація через індексацію:)

Код: db.orders.createIndex({ date: 1 })

Результат:



1. Додаткові завдання

* Визначте категорії товарів із найбільшою кількістю продажів

Код: db.orders.aggregate([

{ $unwind: "$items" },

{

$lookup: {

from: "products",

localField: "items.product",

foreignField: "name",

as: "product\_info"

}

},

{ $unwind: "$product\_info" },

{

$group: {

\_id: "$product\_info.category",

totalSold: { $sum: "$items.quantity" }

}

},

{ $sort: { totalSold: -1 } }

])

Результат:



* Розрахуйте середню ціну товарів у кожній категорії

Код: db.products.aggregate([

{

$group: {

\_id: "$category",

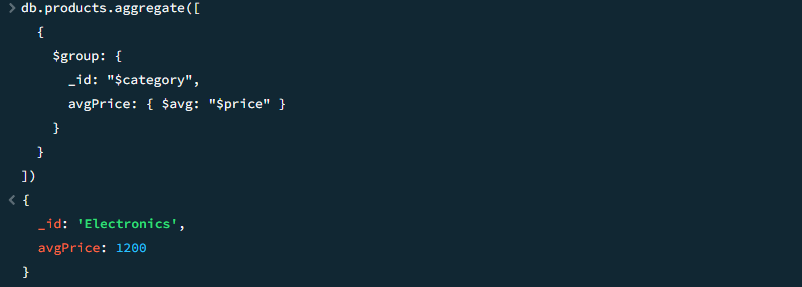
avgPrice: { $avg: "$price" }

}

}

])

Результат:



* Знайдіть користувачів, які зробили більше одного замовлення

Код: db.orders.aggregate([

{

$group: {

\_id: "$customerId",

orderCount: { $sum: 1 }

}

},

{ $match: { orderCount: { $gt: 1 } } }

])

**Висновки:**

Під час лабораторної роботи було досліджено операції з базою даних MongoDB, включаючи додавання, вибірку та агрегування даних. Також було виконано аналіз ефективності запитів за допомогою explain("executionStats"), що дозволило оцінити їх продуктивність. Виявлено, що використання індексів може значно покращити швидкість виконання запитів, мінімізуючи кількість перевірених документів у колекції.