

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО  
КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

ЗВІТ З ЛАБОРАТОРНИХ РОБІТ  
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
**«СУЧАСНІ ПАРАДИГМИ ЗБЕРЕЖЕННЯ ДАНИХ»**

Виконав: студент групи КН-22-1(а)

Панцюк К.В.

Перевірив: асист. каф. Андреев П.І.

КРЕМЕНЧУК 2025

## Лабораторна робота №4

### Тема. Робота з індексами в MongoDB

**Мета роботи:** навчитися створювати індекси в MongoDB.

### Порядок виконання роботи

- 1.Отримати індивідуальний варіант завдання.
- 2.Для спроектованої схеми БД обрати поля, які часто використовуються у фільтрації або сортуванні, ігноруючи, які рідко використовуються та часто змінюються.
- 3.Обрати типи індексів для обраних полів.
- 4.Заміряти час виконання запитів фільтрації або сортування.
- 5.Створити обрані типи індексів.
- 6.Заміряти час виконання запитів після створення індексів.
- 7.Порівняти заміри та зробити висновок.

### Виконання роботи

- 1.Продовжуємо роботу з базою даних якою створили під час виконання 2-ї лабораторної роботи.

Після чого створимо таблицю де вказуємо назву колекції тип індексу та його опис .

	A	B	C	D	E
1	Колект	Поле	Тип індексу	Пояснення	
2	users	email	Унікальний індекс	Поле email часто використовується для автентифікації та пошуку користувачів. Унікальний індекс забезпечує, що кожен email є унікальним.	
3	products	categoryId	Одиничний індекс	Поле categoryId часто використовується для фільтрації товарів за категоріями.	
4	products	price	Одиничний індекс	Поле price часто використовується для сортування товарів за ціною.	
5	products	name, description, brand	Текстовий індекс	Ці поля часто використовуються для пошуку товарів за текстовими запитам.	
6	orders	userId, createdAt	Композитний індекс	Поле userId часто використовується для пошуку замовлень користувача, а createdAt для сортування за датою створення.	
7	reviews	productId, userId	Композитний унікальний індекс	Поле productId часто використовується для пошуку відгуків на товар, а userId для ідентифікації автора відгуку. Унікальний індекс забезпечує, що кожен користувач може залишити тільки один відгук на товар.	
8	categories	slug	Одиничний індекс	Поле slug часто використовується для пошуку категорій за URL-адресами.	
9	promotions	code	Одиничний індекс	Поле code часто використовується для пошуку промокодів.	
10	shipping	isActive	Одиничний індекс	Поле isActive часто використовується для фільтрації активних методів доставки.	
11					

Рисунок 4.1–Таблиця опису полів та індексів

3.Створимо код який буде демонструвати роботу з індексами і без вимірюючи час виконання.

```
// Виконання запитів без індексів та вимір часу
console.log('Виконання запитів без індексів...');
const usersCollection = db.collection('users');
const productsCollection = db.collection('products');
const ordersCollection = db.collection('orders');
const reviewsCollection = db.collection('reviews');
const categoriesCollection = db.collection('categories');
const promotionsCollection = db.collection('promotions');
const shippingCollection = db.collection('shipping');

// Запит для колекції users
const usersQueryWithoutIndex = { email: 'example@example.com' };
const usersStartTime = process.hrtime();
await usersCollection.findOne(usersQueryWithoutIndex);
const usersEndTime = process.hrtime(usersStartTime);
console.log(`Час виконання запиту для users без індексу: ${usersEndTime[0]}s ${usersEndTime[1] / 1000000}ms`);

// Запит для колекції products
const productsQueryWithoutIndex = { categoryId: 'someCategoryId' };
const productsStartTime = process.hrtime();
await productsCollection.findOne(productsQueryWithoutIndex);
const productsEndTime = process.hrtime(productsStartTime);
console.log(`Час виконання запиту для products без індексу: ${productsEndTime[0]}s ${productsEndTime[1] / 1000000}ms`);
```

Рисунок 4.2–Код для створення запитів без індексів

```
1 console.log('Індекси створено');
2
3 // Виконання запитів з індексами та вимір часу
4 console.log('Виконання запитів з індексами...');
5
6 // Запит для колекції users
7 const usersQueryWithIndex = { email: 'example@example.com' };
8 const usersStartTimeWithIndex = process.hrtime();
9 await usersCollection.findOne(usersQueryWithIndex);
10 const usersEndTimeWithIndex = process.hrtime(usersStartTimeWithIndex);
11 console.log(`Час виконання запиту для users з індексом: ${usersEndTimeWithIndex[0]}s ${usersEndTimeWithIndex[1] / 1000000}ms`);
12
13 // Запит для колекції products
14 const productsQueryWithIndex = { categoryId: 'someCategoryId' };
15 const productsStartTimeWithIndex = process.hrtime();
16 await productsCollection.findOne(productsQueryWithIndex);
17 const productsEndTimeWithIndex = process.hrtime(productsStartTimeWithIndex);
18 console.log(`Час виконання запиту для products з індексом: ${productsEndTimeWithIndex[0]}s ${productsEndTimeWithIndex[1] / 1000000}ms`);
19
20 // Запит для колекції orders
21 const ordersQueryWithIndex = { userId: 'someUserId', createdAt: { $gte: new Date('2023-01-01') } };
22 const ordersStartTimeWithIndex = process.hrtime();
23 await ordersCollection.findOne(ordersQueryWithIndex);
24 const ordersEndTimeWithIndex = process.hrtime(ordersStartTimeWithIndex);
25 console.log(`Час виконання запиту для orders з індексом: ${ordersEndTimeWithIndex[0]}s ${ordersEndTimeWithIndex[1] / 1000000}ms`);
26
27 // Запит для колекції reviews
28 const reviewsQueryWithIndex = { productId: 'someProductId', userId: 'someUserId' };
29 const reviewsStartTimeWithIndex = process.hrtime();
```

Рисунок 4.3–Код для замірів з створеними індексами

```

PS D:\Mongo_db_course\Mongo_db_course\mongo-project> node mongoIndexPerformance.js
З'єднання з MongoDB закрите
З'єднання з MongoDB закрите
Підключено до MongoDB сервера
Виконання запитів без індексів...
Час виконання запиту для users без індексу: 0s 14.1885ms
Підключено до MongoDB сервера
Виконання запитів без індексів...
Час виконання запиту для users без індексу: 0s 14.1885ms
Час виконання запиту для products без індексу: 0s 3.5577ms
Виконання запитів без індексів...
Час виконання запиту для users без індексу: 0s 14.1885ms
Час виконання запиту для products без індексу: 0s 3.5577ms
Час виконання запиту для users без індексу: 0s 14.1885ms
Час виконання запиту для products без індексу: 0s 3.5577ms
Час виконання запиту для orders без індексу: 0s 15.3637ms
Час виконання запиту для reviews без індексу: 0s 3.3003ms
Час виконання запиту для products без індексу: 0s 3.5577ms
Час виконання запиту для orders без індексу: 0s 15.3637ms
Час виконання запиту для reviews без індексу: 0s 3.3003ms
Час виконання запиту для categories без індексу: 0s 2.3695ms
Час виконання запиту для orders без індексу: 0s 15.3637ms
Час виконання запиту для reviews без індексу: 0s 3.3003ms
Час виконання запиту для categories без індексу: 0s 2.3695ms
Час виконання запиту для reviews без індексу: 0s 3.3003ms
Час виконання запиту для categories без індексу: 0s 2.3695ms
Час виконання запиту для categories без індексу: 0s 2.3695ms
Час виконання запиту для promotions без індексу: 0s 3.046ms
Час виконання запиту для promotions без індексу: 0s 3.046ms
Час виконання запиту для shipping без індексу: 0s 3.2838ms
Час виконання запиту для shipping без індексу: 0s 3.2838ms
Створення індексів...
Індекси створено
Індекси створено
Виконання запитів з індексами...
Час виконання запиту для users з індексом: 0s 1.0195ms
Час виконання запиту для products з індексом: 0s 1.6127ms
Час виконання запиту для products з індексом: 0s 1.6127ms
Час виконання запиту для orders з індексом: 0s 1.194ms
Час виконання запиту для reviews з індексом: 0s 1.4502ms
Час виконання запиту для categories з індексом: 0s 1.7844ms

```

Рисунок 4.4—Результати замірів без індексів і з ними

## Відповіді на контрольні питання

### Що таке індекс у MongoDB і для чого він потрібен?

Індекс у MongoDB – це спеціальна структура даних, яка зберігає впорядковані значення одного чи кількох полів документа. Він потрібен для прискорення пошуку, сортування та фільтрації, зменшуючи кількість документів, які потрібно переглядати.

### Які переваги і недоліки створення індексів у базі даних?

Переваги: значне прискорення виконання запитів, оптимізація пошуку, сортування та фільтрації. Недоліки: уповільнення операцій запису та оновлення, додаткове використання пам'яті, ризик зниження продуктивності при неправильному виборі індексів.

### Чим відрізняються індекс за одним полем та композитні індекси?

Індекс за одним полем охоплює лише одне поле документа, наприклад `db.users.createIndex({ age: 1 })`. Композитний індекс охоплює кілька полів, наприклад `db.users.createIndex({ age: 1, city: -1 })`, і використовується для запитів,

які одночасно фільтрують або сортують за цими полями. Порядок полів у композитному індексі має значення.

### **Що таке унікальний індекс і коли його варто використовувати?**

Унікальний індекс гарантує, що значення в індексованому полі не повторюються. Його варто використовувати для полів, які мають бути унікальними, наприклад email або username. Приклад: `db.users.createIndex({ email: 1 }, { unique: true })`.

### **Які існують спеціальні типи індексів у MongoDB, окрім базових?**

Текстовий індекс для пошуку по словах у текстових полях. Геопросторовий індекс (2dsphere, 2d) для координат і геолокації. Hashed індекс для рівномірного розподілу даних, зокрема при шардінгу. TTL індекс для автоматичного видалення документів після певного часу. Wildcard індекс для індексації всіх полів у вкладених документах.

### **Як створити індекс у MongoDB та які параметри можна при цьому вказати?**

Індекс створюється командою `db.collection.createIndex({ field: 1 })`, де 1 означає сортування за зростанням, а -1 за спаданням. Параметри: `unique: true` для унікальності, `sparse: true` для індексації лише документів з цим полем, `expireAfterSeconds` для TTL індексу, `partialFilterExpression` для індексації частини документів.