

## Лабораторна робота

Тема: Введення в MongoDB.

Мета: Що таке MongoDB. Початок роботи з MongoDB. Графічний клієнт Compass.

### Хід роботи

#### Що таке MongoDB

**MongoDB** представляє найбільш популярну на даний момент документоорієнтовану систему управління базами даних. За різними оцінками входить до десятка найбільш використовуваних баз даних у світі.

Документи замість рядків

Якщо реляційні бази даних зберігають рядки, MongoDB зберігає документи. На відміну від рядків, документи можуть зберігати складну за структурою інформацію. Документ можна подати як сховище ключів та значень.

**Ключ** представляє просту мітку, з якою асоційовано певний шматок даних.

Однак за всіх відмінностей є одна особливість, яка зближує MongoDB та реляційні бази даних. У реляційних СУБД зустрічається таке поняття як **первинний ключ**. Це поняття описує стовпець, який має унікальні значення. У MongoDB для кожного документа є унікальний ідентифікатор, який має назву `_id`. І якщо явно не вказати його значення, то MongoDB автоматично згенерує для нього значення.

Кожному ключу порівнюється певне значення. Але тут також треба враховувати одну особливість: якщо в реляційних базах є чітко окреслена структура, де є поля, і якщо якесь поле не має значення, йому (залежно від налаштувань конкретної бд) можна надати значення NULL. У MongoDB все інакше. Якщо якомусь ключу не зіставлено значення, цей ключ легко опускається у документі і використовується.

## Колекції

Якщо традиційному світі SQL є таблиці, то світі MongoDB є колекції. І якщо в реляційних БД таблиці зберігають однотипні жорстко структуровані об'єкти, то в колекції можуть містити різні об'єкти, що мають різну структуру і різний набір властивостей.

## Реплікація

Вся система MongoDB може представляти не лише одну базу даних, яка розміщується на одному фізичному сервері. Функціональність MongoDB дозволяє розташувати декілька баз даних на кількох фізичних серверах, і ці бази даних зможуть легко обмінюватись даними та зберігати цілісність.

Система зберігання даних MongoDB представляє набір реплік. У цьому наборі є основний вузол і може бути набір вторинних вузлів. Усі вторинні вузли зберігають цілісність та автоматично оновлюються разом із оновленням головного вузла. І якщо основний вузол з якихось причин виходить з ладу, то один із вторинних вузлів стає головним.

## Формат даних у MongoDB

Одним із популярних стандартів обміну даними та їх зберігання є JSON (JavaScript Object Notation). JSON ефективно описує складні структури дані. Спосіб зберігання даних MongoDB у цьому плані схожий на JSON, хоча формально JSON не використовується. Для зберігання MongoDB застосовується формат, який називається  **BSON**  (БіСон) або скорочення від binary JSON.

BSON дозволяє працювати з даними швидше: швидше виконується пошук та обробка. Хоча треба зазначити, що BSON на відміну від зберігання даних у форматі JSON має невеликий недолік: загалом дані в JSON-форматі займають менше місця, ніж у форматі BSON, з іншого боку, цей недолік з лишком окупається швидкістю.

## Кросплатформеність

MongoDB написана на C++, тому її легко портувати на різні платформи. MongoDB може бути розгорнуто на платформах Windows, Linux,

MacOS, Solaris. Можна також завантажити вихідний код і самому скомпілювати MongoDB, але рекомендується використовувати бібліотеки з офсайту.

Простота у використанні

Відсутність жорсткої схеми бази даних та у зв'язку з цим потреби при найменшій зміні концепції зберігання даних перестворювати цю схему значно полегшують роботу з базами даних MongoDB та подальшим їх масштабуванням. Крім того, заощаджується час розробників. Їм більше не треба думати про перестворення бази даних та витратити час на побудову складних запитів.

Але, навіть з огляду на всі недоліки традиційних баз даних та переваги MongoDB, важливо розуміти, що завдання бувають різні та методи їх вирішення бувають різні. У якійсь ситуації MongoDB дійсно покращить продуктивність вашої програми, наприклад, якщо треба зберігати складні структури дані. В іншій ситуації краще буде використовувати традиційні реляційні бази даних. Крім того, можна використовувати змішаний підхід: зберігати один тип даних у MongoDB, а інший тип даних – у традиційних БД.

GridFS

Однією із проблем під час роботи з будь-якими системами баз даних є збереження даних великого розміру. Можна зберігати дані у файлах, використовуючи різні мови програмування. Деякі СУБД пропонують спеціальні типи даних для зберігання бінарних даних БД (наприклад, BLOB в MySQL).

На відміну від реляційних СУБД MongoDB дозволяє зберігати різні документи з різним набором даних, проте розмір документа обмежується 16 мб. Але MongoDB пропонує рішення - спеціальну технологію **GridFS**, яка дозволяє зберігати дані за розміром більше 16 мб.

Система GridFS складається із двох колекцій. У першій колекції, яка називається files, зберігаються імена файлів, а також їх метадані, наприклад

розмір. А в іншій колекції, яка називається `chunks`, у вигляді невеликих сегментів зберігаються дані файлів зазвичай сегментами по 256 кб.

Для тестування GridFS можна використовувати спеціальну утиліту **`mongofiles`**, яка йде у пакеті `mongodb`.

## **Встановлення та початок роботи з MongoDB**

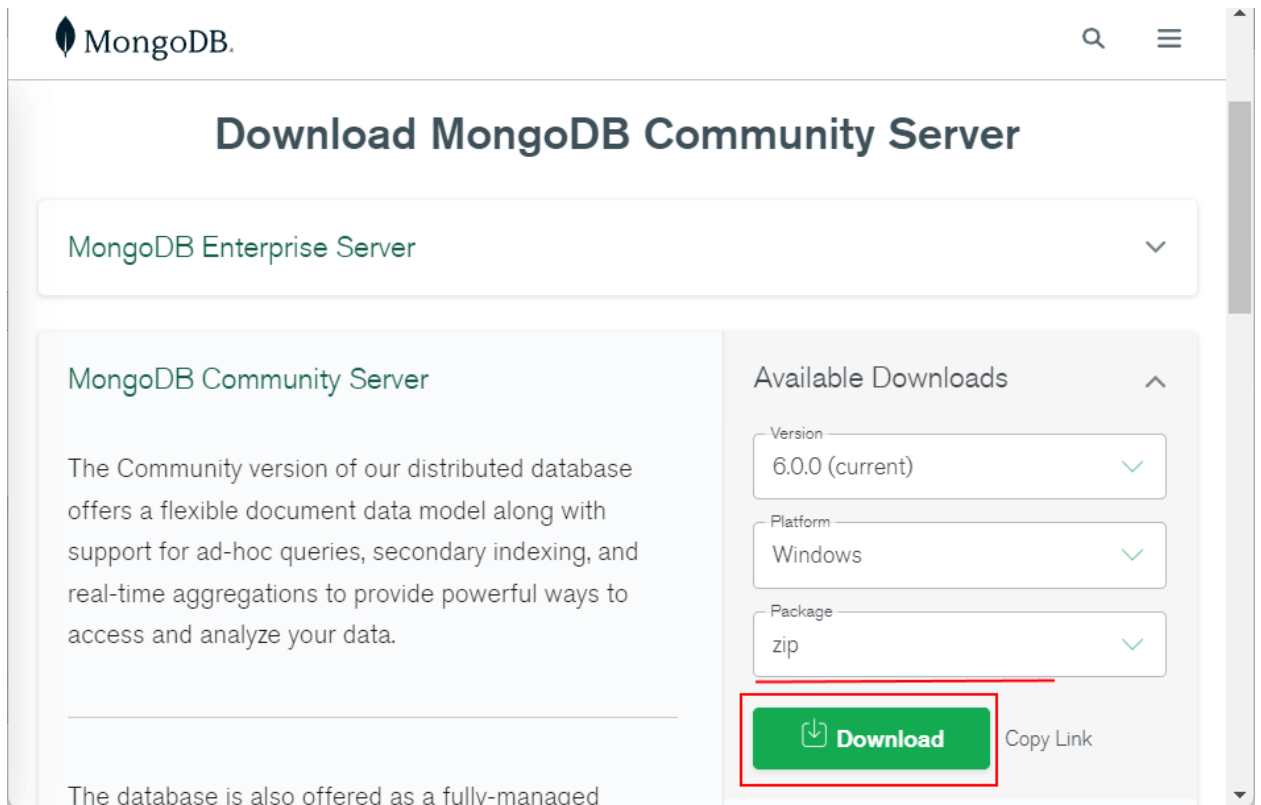
Встановлення сервера MongoDB

Офіційний сайт пропонує пакети дистрибутивів для різних платформ: Windows, Linux, MacOS, Solaris. І кожній платформі є кілька дистрибутивів. Причому є два види серверів – безкоштовний Community та платний Enterprise. У цьому посібнику будемо використовувати безкоштовну версію Community.

Для встановлення MongoDB завантажимо один пакет з офіційного сайту <https://www.mongodb.com/try/download/community>.

Для завантаження всіх необхідних файлів виберемо потрібну операційну систему та відповідний тип пакета. Розглянемо з прикладу установки на ОС Windows.

MongoDB можна завантажити у низці варіантів. Так, для Windows доступне завантаження інсталятора `msi` і також доступне завантаження `zip`-пакета. Насправді нам достатньо завантажити `zip`-архів і розпакувати в потрібній нам папці. Тому виберемо цей варіант завантаження:



Після завантаження архівного пакета розпакуємо його до папки **C:\mongodb** . Якщо після встановлення ми відкриємо папку **bin** у розпакованому архіві ( **C:\mongodb\bin** ), то зможемо знайти там купу додатків, які виконують певну роль. Коротко розглянемо їх.

- **mongod** : сервер баз даних MongoDB. Він обробляє запити, керує форматом даних та виконує різні операції у фоновому режимі з управління базами даних
- **mongos** : служба маршрутизації MongoDB, яка допомагає обробляти запити та визначати розташування даних у кластері MongoDB

### Створення каталогу для БД та запуск MongoDB

Після встановлення потрібно створити на жорсткому диску каталог, у якому будуть бази даних MongoDB.

У Windows за промовчанням MongoDB зберігає бази даних на шляху **C:\data\db** , тому, якщо ви використовуєте Windows, вам потрібно створити відповідний каталог.

Якщо ж виникла потреба використовувати якийсь інший шлях до файлів, його можна передати при запуску MongoDB у прапорі **--dbpath**.

Отже, після створення каталогу зберігання БД можна запустити сервер MongoDB. Сервер представляє додаток **mongod**, який знаходиться в каталозі **bin** в папці сервера. Для цього запустимо термінал/командний рядок і там запровадимо відповідні команди. Для Windows це буде виглядати так:

```
Microsoft Windows [Version 10.0.22000.739]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\eugen>C:\mongodb\bin\mongod
{"t":{"$date":"2022-07-27T20:59:54.975+03:00"},"s":"I", "c":"CONTROL", "id":
:23285, "ctx":"-", "msg":"Automatically disabling TLS 1.0, to force-enable T
LS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-07-27T20:59:56.275+03:00"},"s":"I", "c":"NETWORK", "id":
:4915701, "ctx":"main", "msg":"Initialized wire specification", "attr":{"spec":
{"incomingExternalClient":{"minWireVersion":0, "maxWireVersion":17}, "incomingI
nternalClient":{"minWireVersion":0, "maxWireVersion":17}, "outgoing":{"minWireV
ersion":6, "maxWireVersion":17}, "isInternalClient":true}}}
{"t":{"$date":"2022-07-27T20:59:56.278+03:00"},"s":"I", "c":"NETWORK", "id":
:4648602, "ctx":"main", "msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-07-27T20:59:56.280+03:00"},"s":"I", "c":"REPL", "id":
:5123008, "ctx":"main", "msg":"Successfully registered PrimaryOnlyService", "at
tr":{"service":"TenantMigrationDonorService", "namespace":"config.tenantMigrat
ionDonors"}}
```

Командний рядок відобразить ряд службової інформації, наприклад, що сервер запускається на localhost на порту 27017.

І після успішного запуску сервера ми зможемо здійснювати операції з бд через клієнт.

### Встановлення клієнта Mongosh

Вище ми встановили сервер MongoDB. Однак для роботи з сервером нам потрібний клієнт. Найбільш простим клієнтом у цьому випадку є **MongoDB Shell** або **mongosh** – консольна оболонка для надсилання запитів до сервера, яка також надається безпосередньо компанією MongoDB.

Тут знову ж таки ми можемо вибрати версії клієнта для різних операційних систем. Для Windows пакет клієнта доступний у вигляді інсталятора msі та архіву zip. У цьому випадку виберемо архів zip.

Завантажений пакет розпакуємо в папку **C: Mongosh**.

Якщо в розпакованому архіві ми знайдемо в папку **bin** (тобто **C: \ mongosh \ bin**), то виявимо там консольну утиліту **mongosh**, яка буде застосовуватися для роботи з сервером MongoDB.

Підключення до сервера із клієнта

Використовуємо вище встановлену клієнтську програму `mongosh` для взаємодії з сервером `mongodb`. (Під час роботи з `mongosh` не варто забувати, що у нас повинен бути запущений сервер `mongod`). Отже, запустимо файл **mongosh**, який розташовується у вище розглянутій папці установки:



The screenshot shows a terminal window titled "mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000". The prompt is "Please enter a MongoDB connection string (Default: mongodb://localhost/):". The user has entered the connection string "mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.5.1". The terminal displays the following information:

```
Current Mongosh Log ID: 62e17f15d4d68fcbe6224f24
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.5.1
Using MongoDB:      6.0.0
Using Mongosh:       1.5.1

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
test>
```

Two red arrows point to the connection string and the "test>" prompt. A red box with the text "press Enter here" is positioned over the connection string. Another red box with the text "connected to the 'test' database" is positioned over the "test>" prompt.

При запуску програми `mongosh` спочатку вона запитає користувача, який рядок підключення використовуватиме підключення до сервера MongoDB. У цьому моменті просто натисніть на Enter, щоб використовувати рядок підключення до MongoDB за промовчанням. А за замовчуванням сервер `mongodb` запускається на порту 27017, а повний рядок підключення виглядає так: `mongodb://localhost:27017` або `mongodb://127.0.0.1:27017`

Після підключення консоль відобразить ряд службової інформації та підключиться до бази даних `test`.

Тепер зробимо якісь найпростіші дії. Введемо в консоль послідовно наступні команди і після кожної команди натиснемо на Enter:

```
1 db.users.insertOne( { name: "Tom" } )<font></font>
2 db.users.find()<font></font>
```

Перша команда **use test** встановлює як використовувану базу даних `test`. Навіть якщо такий бд немає, вона створюється автоматично.

За допомогою методу **db.users.insertOne()** до колекції `users` бази даних `test` додається об'єкт `{ name: "Tom" }`. Ідентифікатор **db** - це поточна база даних. У

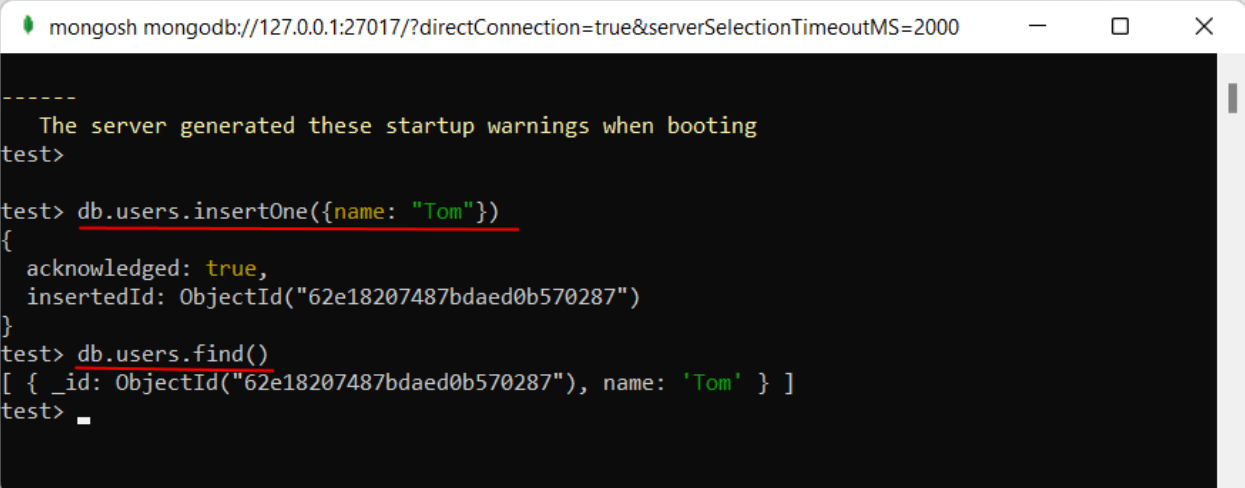
нашому випадку ми підключені до бази даних за промовчанням - тобто до бази даних `test`, відповідно `db` тут представляє базу даних `test`. При цьому не важливо, чи є така бд - якщо її немає, то вона створюється

Після `db` йде **`users`** - це колекція, в яку потім ми додаємо новий об'єкт. Якщо SQL нам треба створювати таблиці заздалегідь, то колекції MongoDB створює самостійно за її відсутності.

Опис об'єкта, що додається, визначається у форматі, з яким ви можливо знайомі, якщо мали справу з форматом JSON. Тобто в даному випадку об'єкт має один ключ `"name"`, якому зіставляється значення `"Tom"`. Тобто ми додаємо користувача з ім'ям Tom.

Якщо об'єкт був успішно доданий, консоль виведе результат операції, зокрема, ідентифікатор доданого об'єкта.

А третя команда **`db.users.find()`** виводить на екран усі об'єкти із бд `test`.



```
-----
The server generated these startup warnings when booting
test>
test> db.users.insertOne({name: "Tom"})
{
  acknowledged: true,
  insertedId: ObjectId("62e18207487bdaed0b570287")
}
test> db.users.find()
[ { _id: ObjectId("62e18207487bdaed0b570287"), name: 'Tom' } ]
test> _
```

З висновку ви можете побачити, що до початкових значень об'єкта було додано якесь незрозуміле поле **`ObjectId`**. Як ви пам'ятаєте, MongoDB як унікальний ідентифікатор документа використовує поле `_id`. І в даному випадку `ObjectId` таки представляє значення для ідентифікатора `_id`.

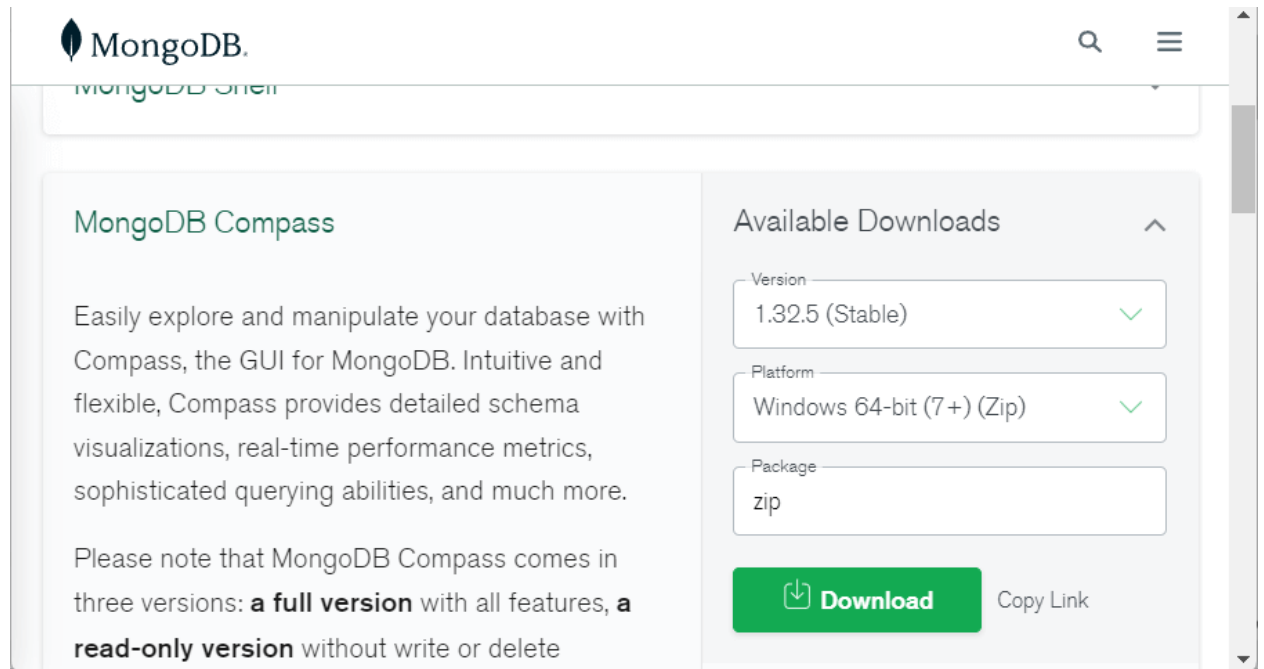
## Графічний клієнт Compass

Для роботи з MongoDB також можна використовувати офіційний графічний клієнт **MongoDB Compass**. Для його завантаження перейдемо за адресою <https://www.mongodb.com/try/download/compass>. На цій сторінці ми



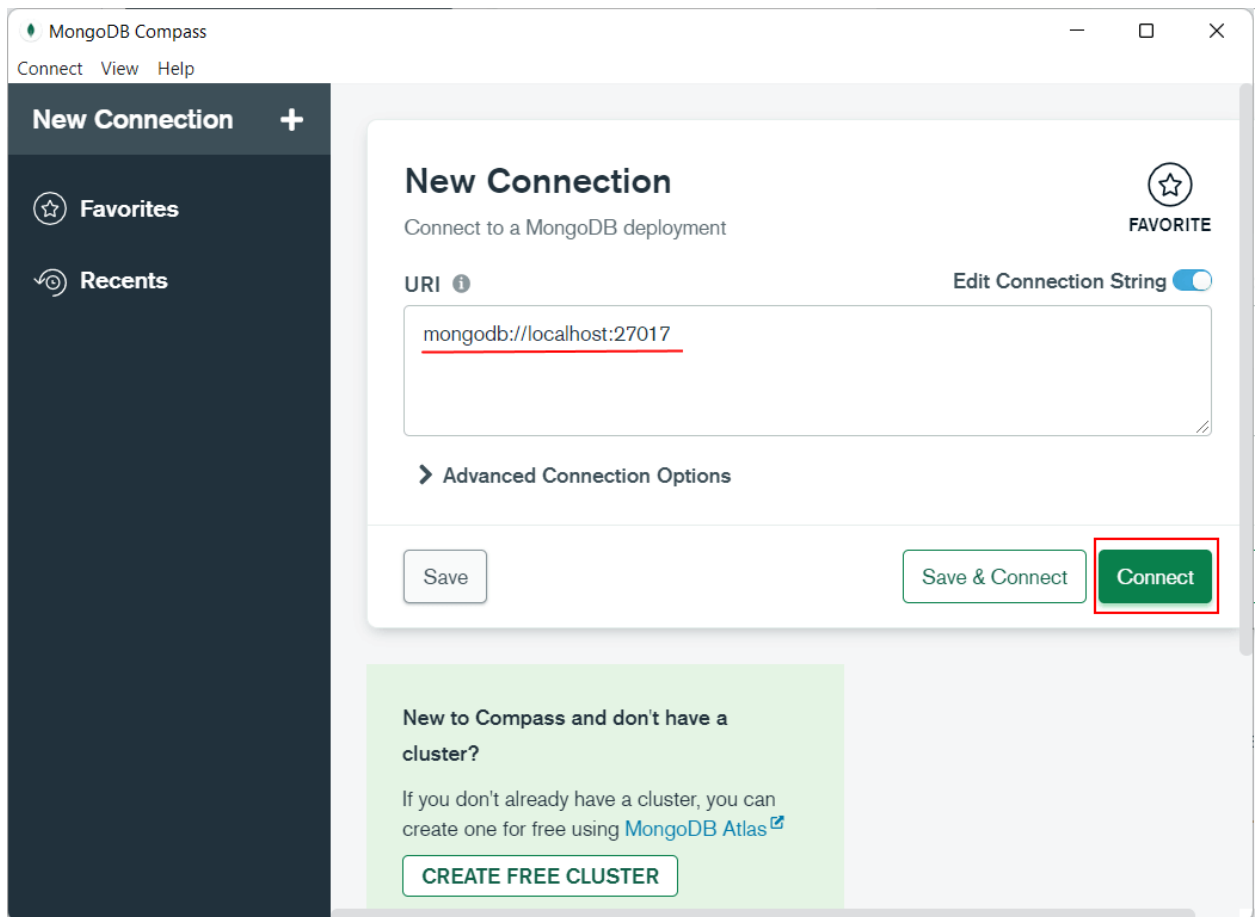
можемо вибрати опції для завантаження - версію Compass та цільову операційну систему. Розглянемо з прикладу установки на Windows.

Для Windows нам доступні дві опції: завантаження як установника msi і завантаження архіву zip. Насправді нам достатньо завантажити архів і розпакувати його в потрібному місці. Тому виберемо цю опцію:

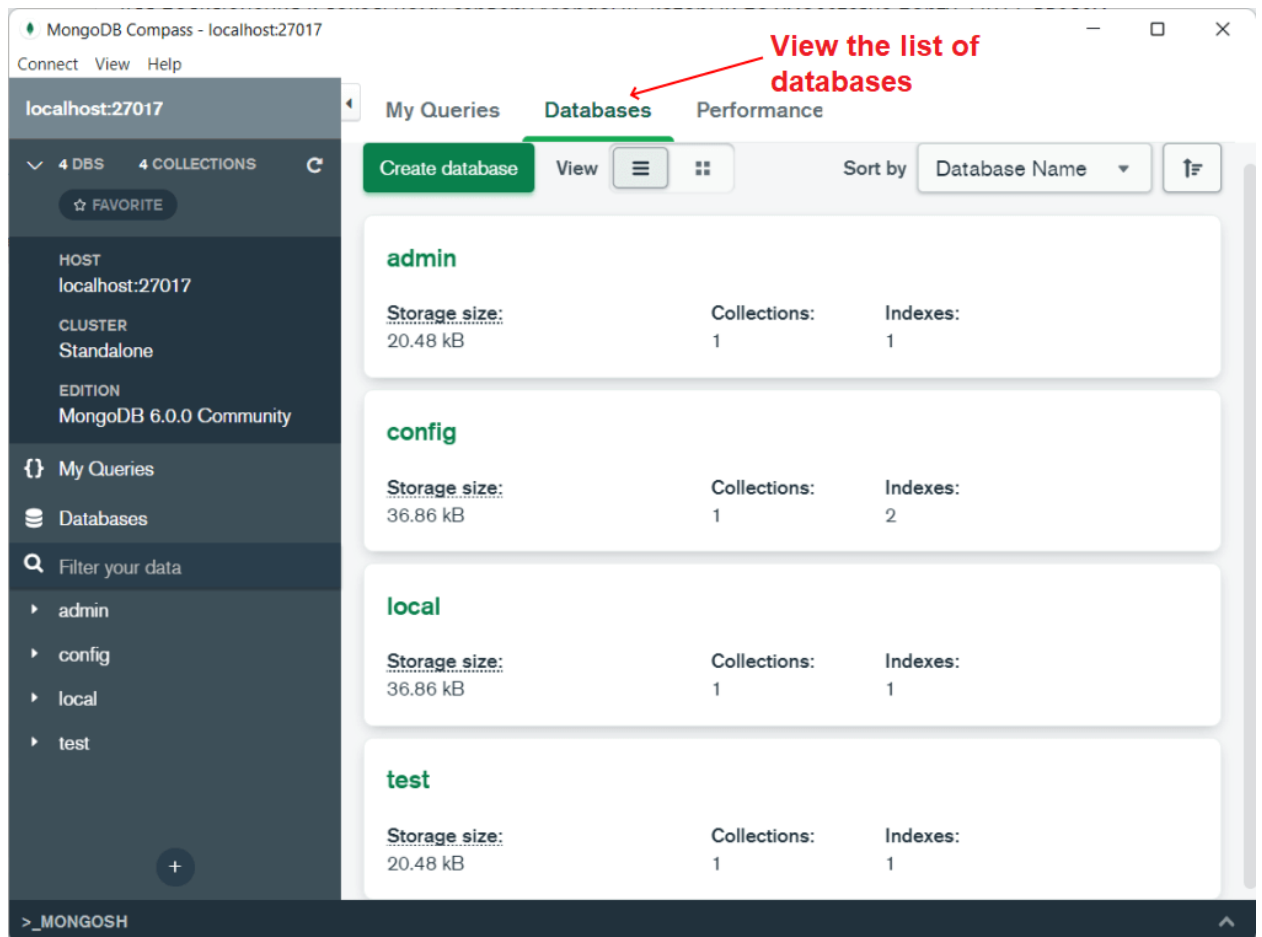


Після завантаження zip-архіву розпакуємо його та перейдемо до розпакованої папки.

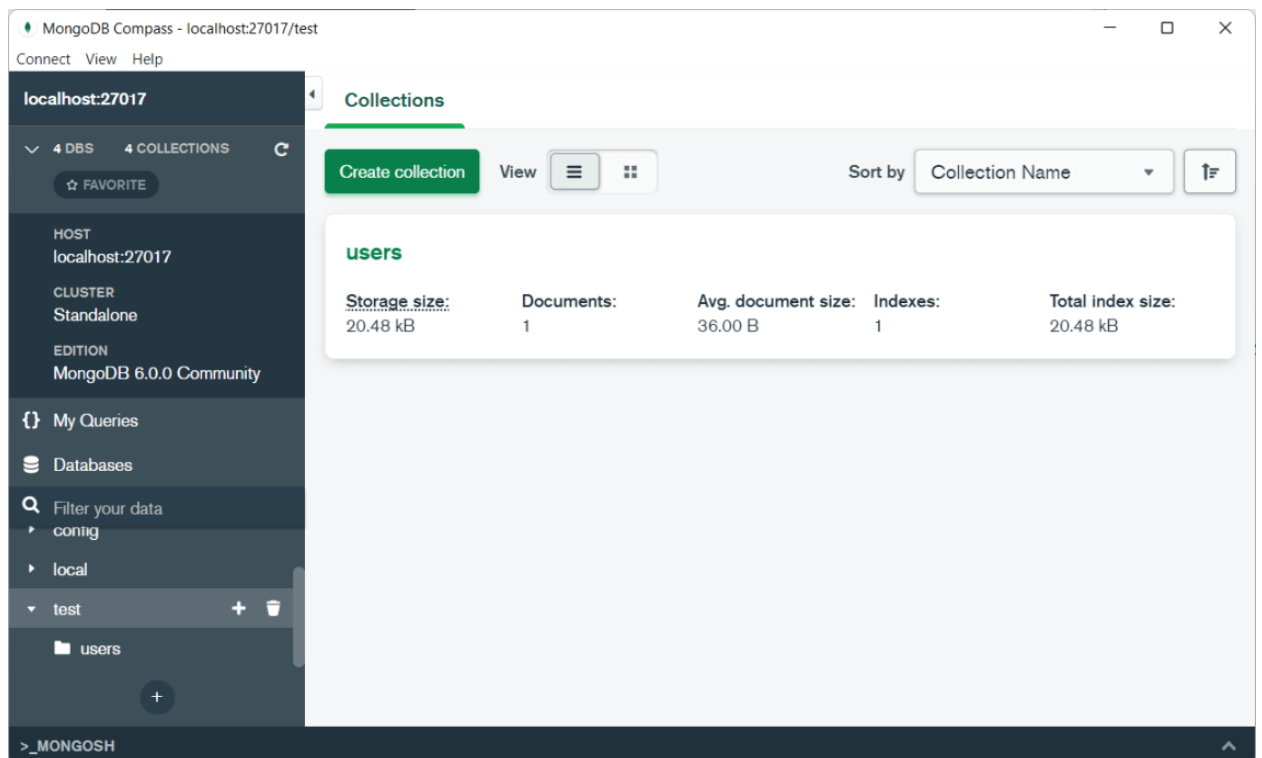
Тут ми можемо знайти файл **MongoDBCompass.exe** . Саме він і представляє виконуваний файл програми. Запустимо його:



За замовчуванням відкриється вікно для створення нового підключення. Воно має лише одне поле - **URI**, куди треба ввести рядок підключення. За замовчуванням тут може з'явитися рядок підключення до локального сервера - **mongodb://localhost:27017**. Рядок підключення може відрізнятися залежно від місця запуску сервера MongoDB (локально на поточному комп'ютері або десь на іншому мережевому ресурсі), використовуваного логіну та паролю та інших налаштувань. У разі ми розглянемо підключення до локально запусченому серверу `mongodb`, установка якого було розглянуто у минулій темі. Тому залишимо в ньому значення **mongodb://localhost:27017** (або введемо цей рядок, якщо поле порожнє) та натиснемо на кнопку **Connect**. Після успішного підключення нам відкриється вміст сервера, де ми можемо переглянути різну інформацію, наприклад, список баз даних, які є на сервері:

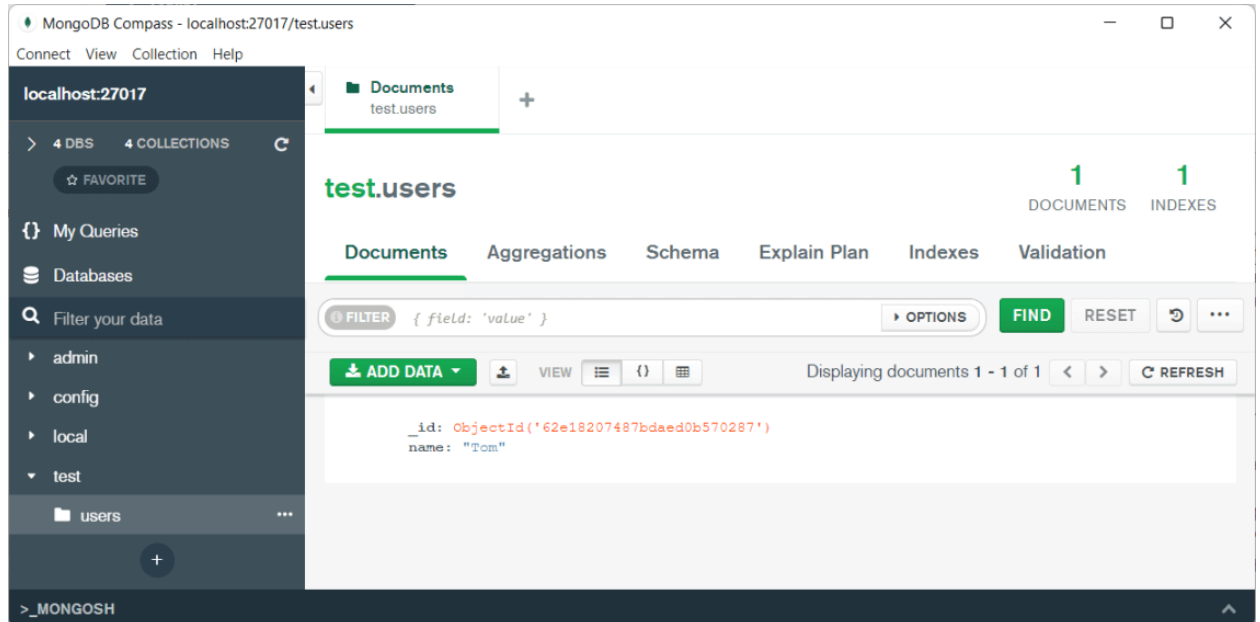


Ми можемо вибрати певну базу даних і отримати інформацію, зокрема, побачити набір колекцій в бд, скільки вони займають даних.



В даному випадку на скріншоті відображається список колекцій бази даних test, створених у минулій темі за допомогою mongosh оболонки.

Натиснувши на певну колекцію, можна побачити графічно всі дані, які є в колекції:



Використовуючи графічний інтерфейс Compass, ми можемо керувати цими даними, додавати, змінювати, видаляти їх.