

The background features a complex, glowing digital network of interconnected nodes and lines in various colors (blue, purple, orange, green). In the center, there is a large, multi-layered cylindrical structure resembling a database or data storage system, with a grid-like pattern and glowing elements. The overall aesthetic is high-tech and futuristic.

Лекція 7

Документно-орієнтована модель даних MongoDB

Визначення та історія розвитку MongoDB

MongoDB – це система управління базами даних, «заточена» під веб-застосунки та інфраструктуру Інтернету. Модель даних спроектована для досягнення високої пропускної спроможності читання та запису та забезпечує просту масштабованість з автоматичним переходом на резервний ресурс у разі відмови.



Визначення та історія розвитку MongoDB

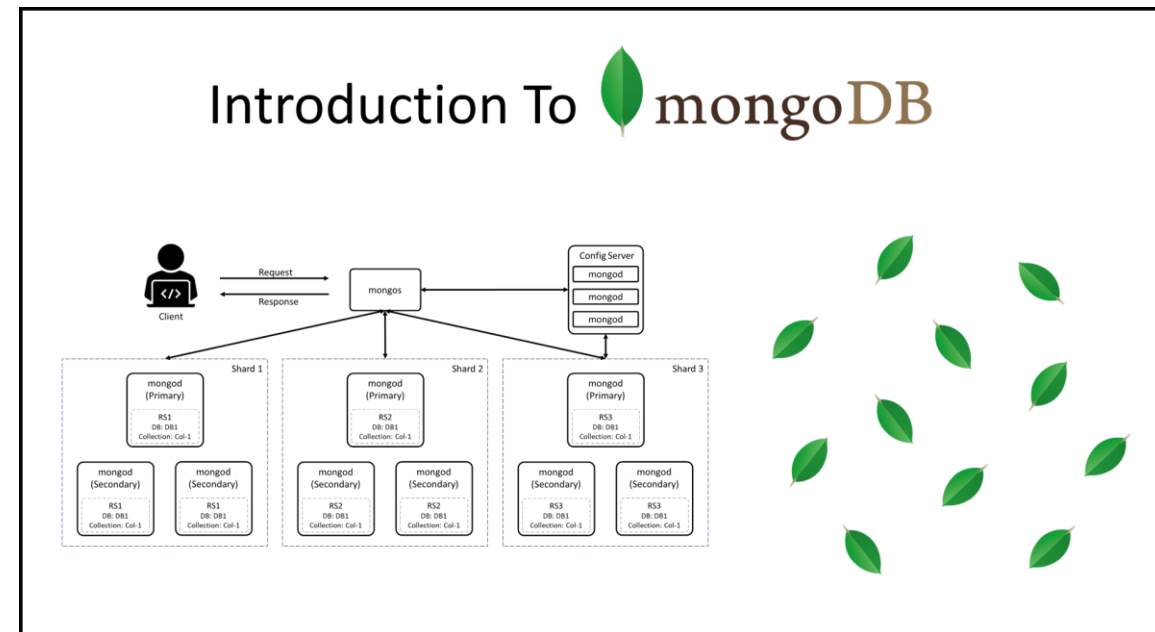
Історія MongoDB починається в середині 2007 року, коли компанія 10gen розпочала розробку проекту «програмна платформа як послуга». Ідея була в тому, щоб створити сервер додатків та базу даних, які могли б служити хостингом для веб-додатків, забезпечуючи масштабованість у міру потреби.



Визначення та історія розвитку MongoDB

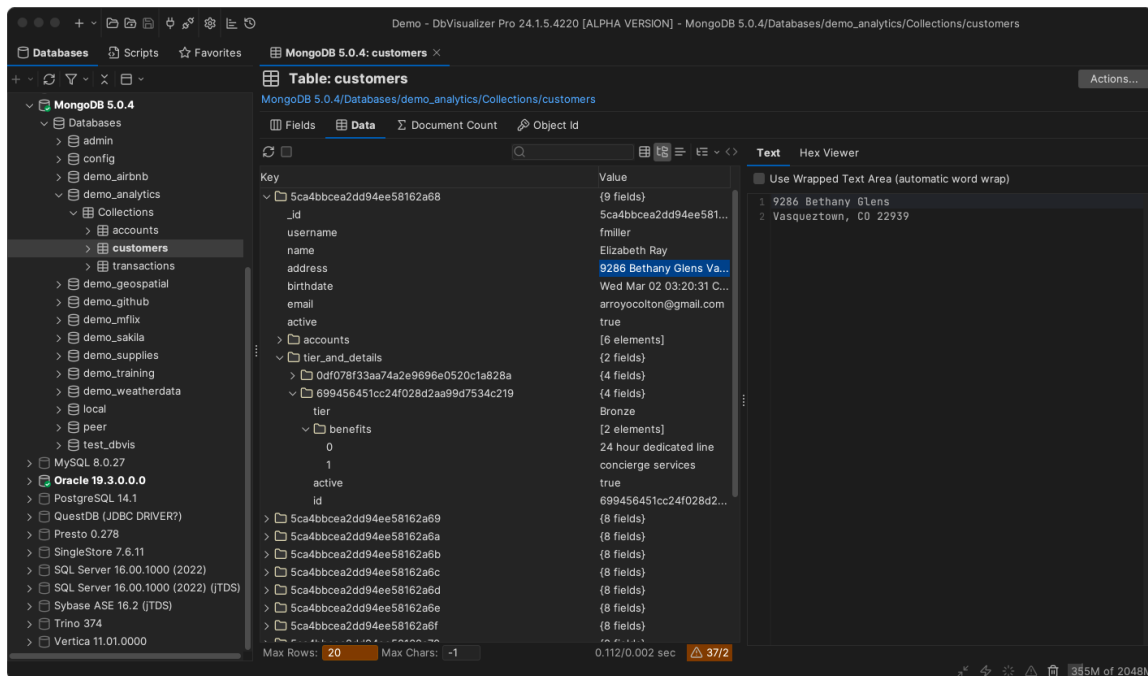
Основними особливостями MongoDB є:

- документно-орієнтоване сховище;
- повна підтримка індексів;
- реплікація даних;
- висока доступність даних;
- здатність до горизонтального масштабування;
- автошардинг;
- підтримка запитів;
- підтримка Map/Reduce;
- підтримка GridFS.



Визначення та історія розвитку MongoDB

Реплікація – це тиражування змін даних із головного сервера БД однією чи кількох залежних серверах.



Шардинг - поділ даних на рівні ресурсів, розбиття даних за якоюсь ознакою. Концепція шардингу полягає у логічному поділі даних з різних ресурсів.

Визначення та історія розвитку MongoDB

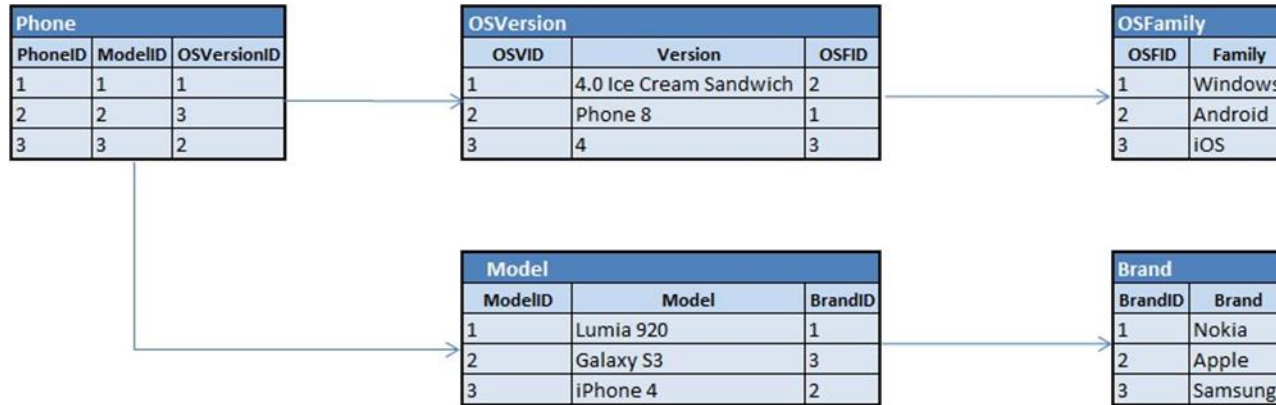


Схема бази даних у реляційному вигляді

База даних у вигляді NoSQL

Phone	
ID	Attributes
1	Brand:Nokia Model:Lumia 920 OSFamily:Windows OSVersion:8
2	Brand:Apple Model:iPhone 4 OSFamily:iOS OSVersion:4
3	Brand:Samsung Model:Galaxy S3 OSFamily:Android OSVersion:4.0 Ice Cream Sandwich Display:4.8 HD Super AMOLED

Визначення та історія розвитку MongoDB

Дані MongoDB представлені JSON-форматі. JSON (JavaScript Object Notation) - це текстовий формат обміну даними, заснований на JavaScript .

Значення JSON можуть бути різних типів даних:

- * Рядки (текст).
- * Числа.
- * Логічні значення (true або false).
- * nul` (порожнє значення).
- * Масиви.
- * Вкладені об'єкти.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

Визначення та історія розвитку MongoDB

Приклади JSON

Об'єкт JSON	Масив JSON	Більш складний приклад із вкладеними структурами:
<pre>json { "name": "John", "age": 30, "isStudent": true, "address": null }</pre>	<pre>json ["apple", "banana", "cherry"]</pre>	<pre>json { "name": "Eve", "age": 25, "address": { "street": "123 Main St", "city": "Anytown" }, "courses": ["Math", "English"] }</pre>

Архітектура MongoDB: Реплікація

Реплікація – це тиражування змін даних із головного сервера БД однією чи кількох залежних серверах



Робочий набір реплік



Початковий первинний вузол виходить з ладу, вторинний вузол призначається на роль первинного



Початковий первинний вузол відновлюється і стає вторинним

Архітектура MongoDB: Швидкодія та довговічність

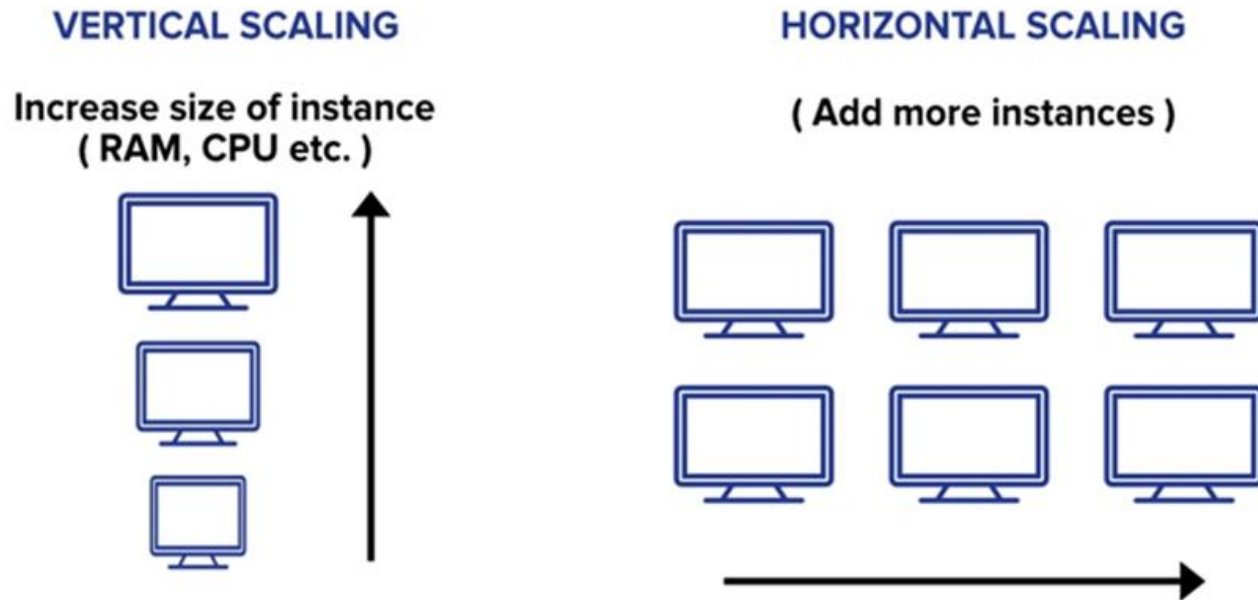
У світі даних існує зворотна залежність між швидкістю запису та довговічністю.

Під *швидкістю запису* розуміється сукупний обсяг операцій вставки, оновлення та видалення, які база даних здатна обробити за одиницю часу.

Під *довговічністю* розуміється міра впевненості у цьому, що результат операцій запису збережеться.

Архітектура MongoDB: Масштабування

Найпростіший спосіб масштабування більшості баз даних – модернізація устаткування. Така методика називається *вертикальним масштабуванням*

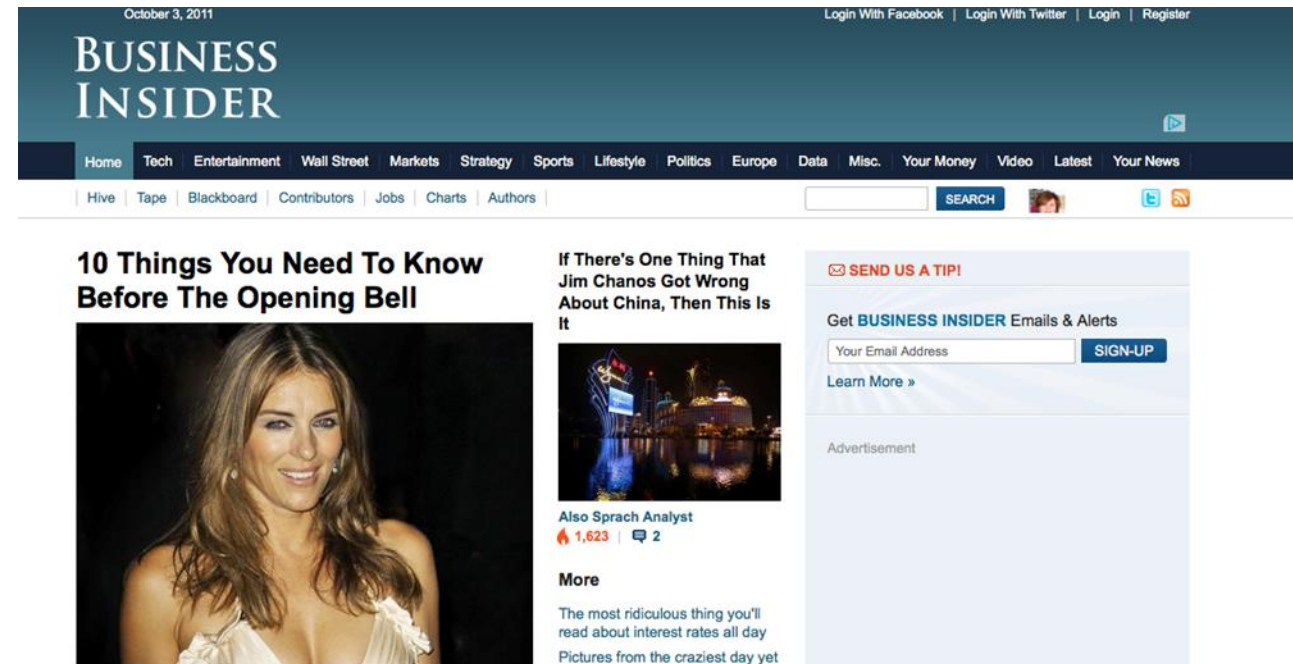


Під час *горизонтального масштабування* потужності розподіляються на кількох комп'ютерах.

Архітектура MongoDB: Переваги MongoDB

MongoDB підходить для основного сховища даних для веб-додатків.

MongoDB підтвердила здатність справлятися з усіма аспектами програми – від основних завдань предметної галузі до таких допоміжних аспектів, як протоколювання та аналітика у реальному часі. Наступна позитивна риса MongoDB полягає у його здатності розробляти програми швидше, ніж при використанні реляційних СУБД



The screenshot shows the Business Insider website interface. At the top, the date "October 3, 2011" is displayed on the left, and login options "Login With Facebook", "Login With Twitter", "Login", and "Register" are on the right. The main header features the "BUSINESS INSIDER" logo. Below the header is a navigation menu with categories: Home, Tech, Entertainment, Wall Street, Markets, Strategy, Sports, Lifestyle, Politics, Europe, Data, Misc., Your Money, Video, Latest, and Your News. A secondary menu includes Hive, Tape, Blackboard, Contributors, Jobs, Charts, and Authors. A search bar and social media icons are also present. The main content area displays a featured article titled "10 Things You Need To Know Before The Opening Bell" with a photo of a woman. To its right is another article snippet: "If There's One Thing That Jim Chanos Got Wrong About China, Then This Is It" with a photo of a building at night and engagement stats (1,623 likes, 2 comments). On the far right, there is a "SEND US A TIP!" section and a sign-up form for "BUSINESS INSIDER Emails & Alerts" with a "SIGN-UP" button and a "Learn More" link. Below the sign-up form is an "Advertisement" placeholder.

Налаштування MongoDB

Для встановлення MongoDB та MongoDB Compass необхідно завантажити файл установки з офіційного сайту MongoDB:

<https://www.mongodb.com/try/download/community>.

Встановлюємо 6-ту версію як найбільш працездатну.

Налаштування MongoDB

0. Запускаємо термінал від імені адміністратора

1. Створюємо директорію *mongo_db \ data*:

```
mkdir C:\data\db
```

2. Далі переходимо в директорію, де встановлено додаток mongod:

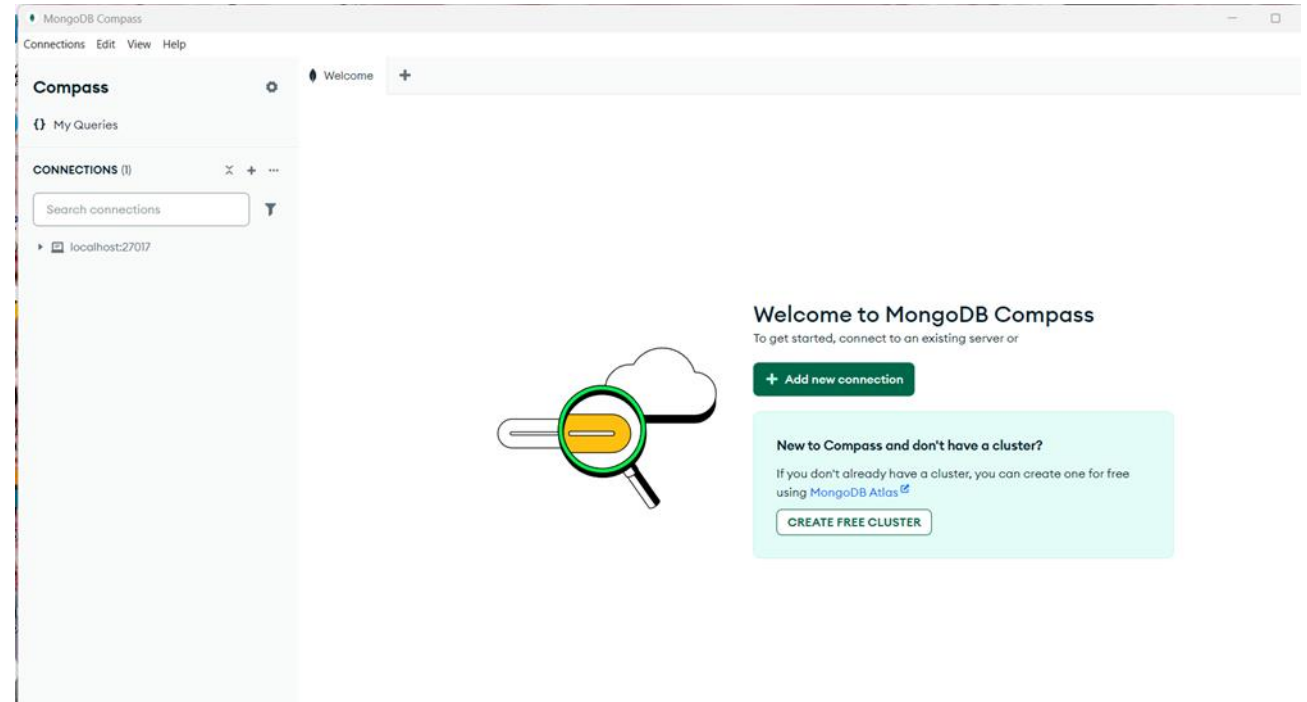
```
cd "C:\Program Files\MongoDB\Server\6.0\bin"
```

3. Викликаємо команду для вказівки додатку папки для зберігання даних:

```
.\mongod.exe - dbpath = "C:\data\db"
```

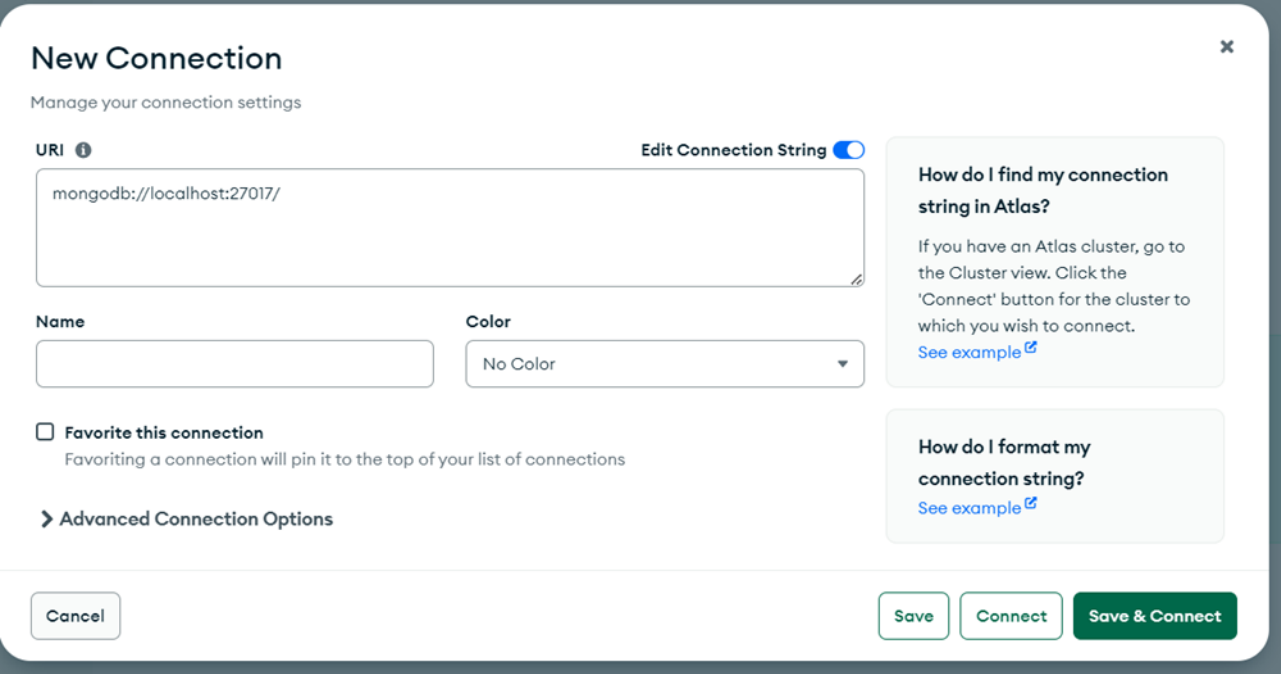
Підключення до сервера MongoDB за допомогою MongoDB Compass

1. Для підключення до сервера запустіть програму MongoDB Compass.



Підключення до сервера MongoDB за допомогою MongoDB Compass

2. Використовуємо рядок підключення за замовчуванням (mongodb://localhost:27017) та натискаємо Connect.



The screenshot shows the 'New Connection' dialog box in MongoDB Compass. The dialog has a title bar with a close button (X) and the text 'New Connection' and 'Manage your connection settings'. The main content area includes:

- A 'URI' field with a help icon (i) and an 'Edit Connection String' toggle switch. The text 'mongodb://localhost:27017/' is entered in the field.
- A 'Name' input field.
- A 'Color' dropdown menu currently set to 'No Color'.
- A checkbox labeled 'Favorite this connection' with the subtext 'Favoriting a connection will pin it to the top of your list of connections'.
- A link for '> Advanced Connection Options'.

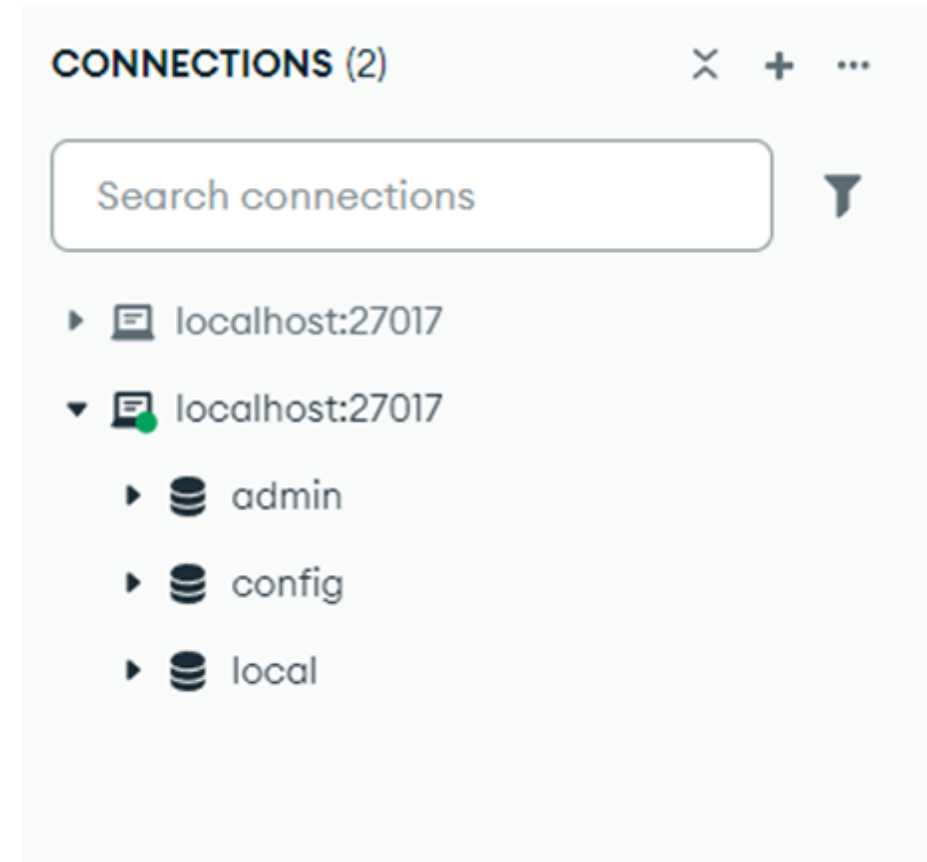
On the right side, there are two informational boxes:

- 'How do I find my connection string in Atlas?' with instructions: 'If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect. See example'.
- 'How do I format my connection string?' with a link 'See example'.

At the bottom, there are three buttons: 'Cancel', 'Save', and 'Save & Connect'.

Підключення до сервера MongoDB за допомогою MongoDB Compass

3. Після чого ми отримуємо доступ до бази даних і маємо бачити 3 бази за замовчуванням: admin, config, local.



Створення бази даних

У терміналі створюємо БД під назвою “My_DB”.

```
use My_DB
```

Створення колекції

У MyMongoDB_21_1 створити колекцію *users*

```
db.createCollection ("users")
```

Додавання даних до колекції

Додавання одного документа:

```
db.users.insertOne(  
  {name:"John",  
    email:"admin@gmail",  
    age: 42,  
    car: true,  
    colors: ["red", "blue", "green"],  
    child: {"name":"Jack","age":5}  
  }  
)
```

Додання довільних даних

```
db.users.insertMany ([
  {
    Fname: "Орина",
    Lname: "Багач",
    group: "186.10.21.1",
    tema: "Розроблення айдентики для освітньої платформи",
    points:95
  },
  {
    Fname: "Марія",
    Lname: "Бідаш",
    group: "186.10.21.1",
    tema: "Розроблення рівнів та елементів оточення гейміфікованого проекту доповненої
реальності",
    points:88
  }
])
```

Додання довільні дані до бази даних із використанням MongoDB Compass.

ADD DATA ▾

EXPORT DATA ▾

UPDATE

DELETE

Import JSON or CSV file

Insert document

```
_id: ObjectId('67a6f6d6c5df125bdc4d7947')  
first_name : "Орина"  
last_name  : "Багач"  
group      : "186.10.21.1"  
тема       : "Розроблення айдентики для освітньої платформи"
```

Выборка данных из коллекции

`db.users.find()` – виводить усі записи з бази даних.

`db.users.find().limit(2)` – відображає лише два записи з колекції

`db.users.find({}, {_id:0}).limit(2)` – заборона на виведення певних полів

`db.users.find({}, {_id:0}).sort({group:-1})` – сортування даних: 1 – за зростанням; -1 - за спаданням

`db.users.find({group:"186.10.21.1"}, {_id:0})` – виведення даних за умовою (виводить усіх здобувачів першої групи).

Условные операторы

\$eq (равно)

\$ne (не равно).

\$gt (больше чем).

\$lt (меньше чем).

\$gte (больше или равно).

\$lte (меньше или равно).

\$in определяет массив значений, одно из которых должно иметь поле документа.

\$nin определяет массив значений, которые не должны иметь поле документа.

Логические операторы

$\$or$: соединяет два условия, и документ должен соответствовать одному из этих условий.

$\$and$: соединяет два условия, и документ должен соответствовать обоим условиям.

$\$not$: документ должен НЕ соответствовать условию.

$\$nor$: соединяет два условия, и документ должен НЕ соответствовать обоим условиям.

Операторы для работы с массивами

`$all`: определяет набор значений, которые должны иметься в массиве.

`$size`: определяет количество элементов, которые должны быть в массиве.

`$elemMatch`: определяет условие, которым должны соответствовать элементы в массиве.

Оператор \$exists

\$exists позволяет извлечь только те документы, в которых определенный ключ присутствует или отсутствует.

Оператор \$type

\$type извлекает только те документы, в которых определенный ключ имеет значение определенного типа, например, строку или число.

Розглянемо приклади використання цих операторів.

```
db.users.find({$or: [{group: "186.10.21.3"}, {name: "John"}]},  
{_id:0}) – використання логічного оператора АБО
```

```
db.users.find({points: {$gt: 70}}, {_id:0}) – використання  
умовного оператора
```

```
db.users.find({points: {$in: [95, 90]}}, {_id:0}) – виведення  
масиву значень
```

```
db.users.find({child: {$exists: true}}, {_id:0}) – вибір об'єкта  
щодо наявності поля
```

Оновлення та видалення даних

`db.users.updateOne({name:"John"}, {$set:{age:30}})` – **оновлення даних в одному полі**

`db.users.updateMany({group:"186.10.21.1"}, {$set:{group: "перша група"}})` – **оновлення даних у кількох полях**

`db.users.updateMany({tema:{$exists:true}}, {$set:{form: "бакалавр"}})` – **додавання нових даних**

`db.users.deleteOne({name:"John"})` – **видалення одного запису**

`db.users.deleteMany({name:{$in:["Jack", "Smit2"]}})` – **видалення кількох записів**

Поиск на совпадение в тексте

```
db.users.createIndex({тема:"text",  
annotation:"text"}) - створення індексів на поля  
«тема» і «annotation»
```

```
db.users.find({$text:{$search:"Доповнена  
реальність"}}) - пошук тексту за словами  
"Доповнена реальність"
```

Агрегування даних

`db.users.count({points:{$lt: 80}})` – визначає кількість студентів у яких бал менше 80

`db.users.distinct("points")` – виведення оригінальних даних по заданому полю