

Бази даних та SQL

NoSQL
MongoDB

Бази даних та SQL

План уроку

- Введення в MongoDB
- Як розпочати роботу з MongoDB
- Базові запити
- Агрегуючі запити

Введення в MongoDB

Бази даних та SQL

Що таке MongoDB

MongoDB - кросплатформна документо-орієнтована система управління базами даних. Класифікована як база даних NoSQL, MongoDB відходить від традиційних основ реляційної структури бази даних на користь JSON-подібних документів з динамічними схемами (MongoDB називає цей формат BSON), що робить інтеграцію даних в певних видах застосунків простіше і швидше. MongoDB є безкоштовним програмним забезпеченням з відкритим вихідним кодом.

<https://www.mongodb.com/>

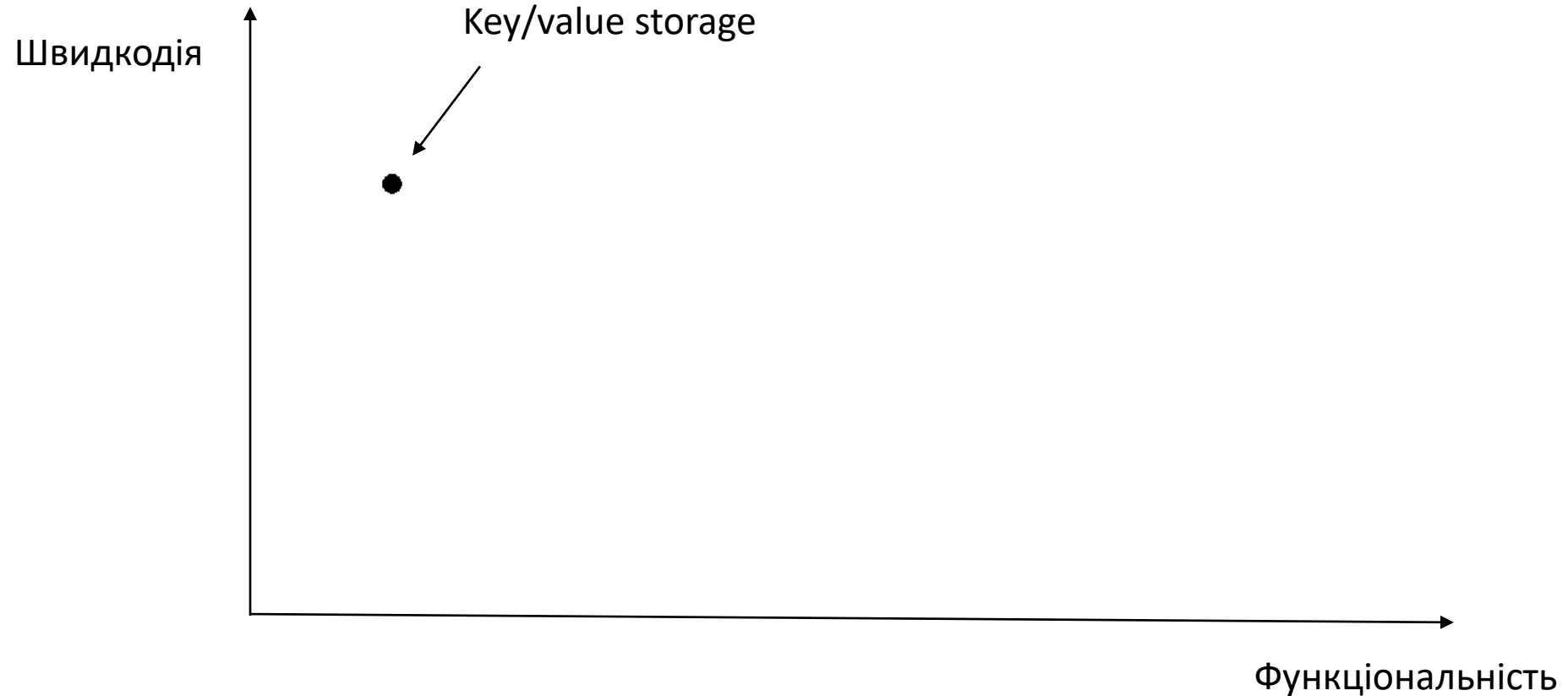
Бази даних та SQL

Місце MongoDB в екосистемі баз даних



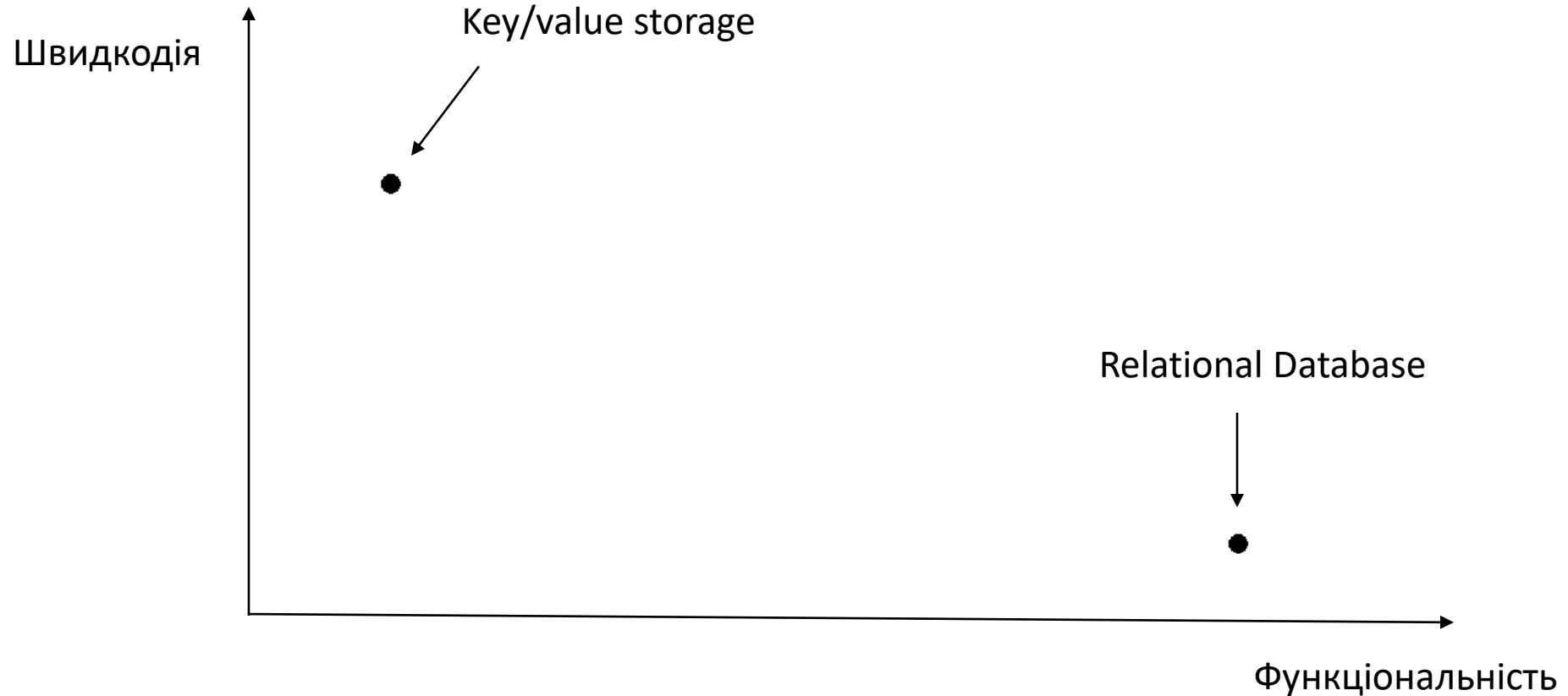
Бази даних та SQL

Місце MongoDB в екосистемі баз даних



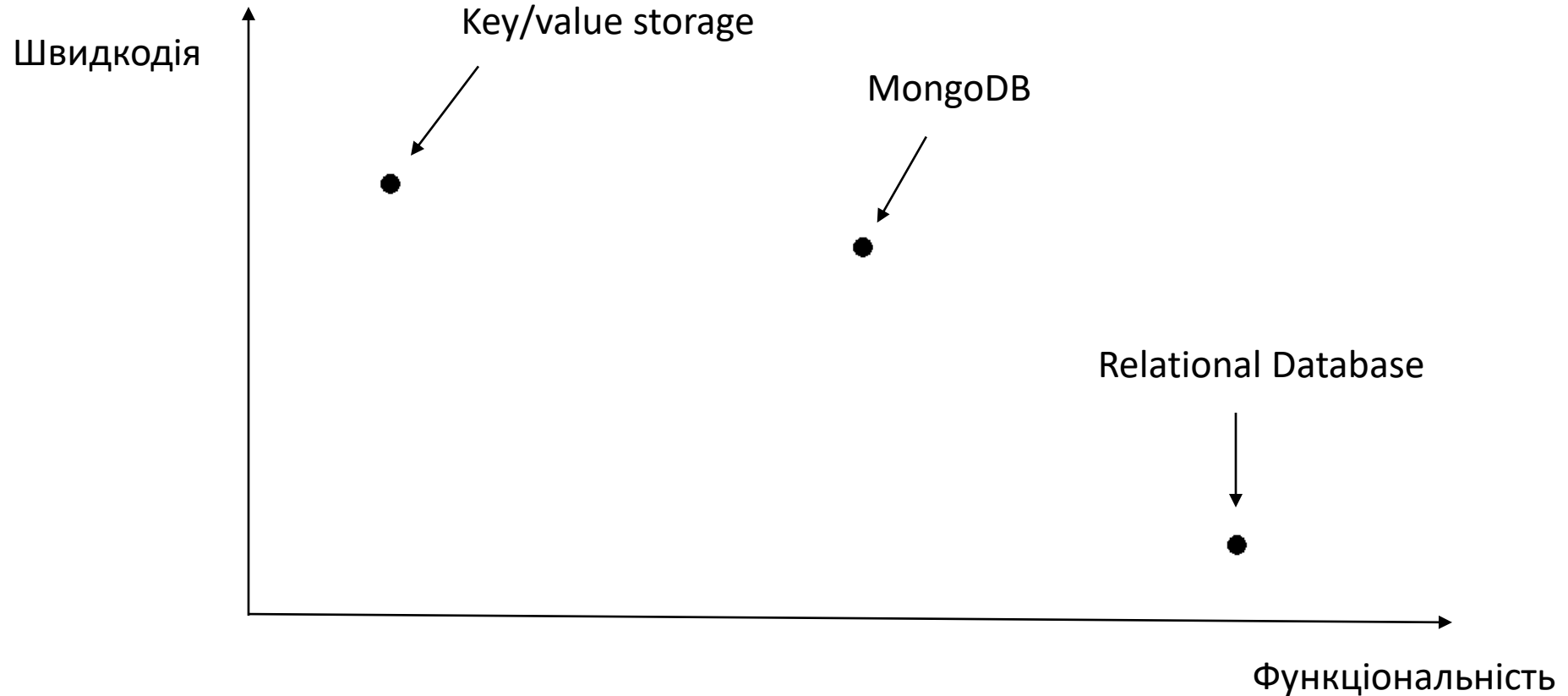
Бази даних та SQL

Місце MongoDB в екосистемі баз даних



Бази даних та SQL

Місце MongoDB в екосистемі баз даних



Бази даних та SQL

Переваги і недоліки

Переваги

- Швидкість розробки.
- Не потрібно синхронізувати схему в базі даних та застосунку.
- Легкість масштабування.
- Продуктивність.

Недоліки

- Менше гнучкості при запиті (наприклад, немає join).
- Не підтримуються транзакції.
- Відсутність ізоляції.
- Кожен документ має імена полів, які зберігаються у ньому (збільшує розмір на диску).

Документо-орієнтована модель даних у MongoDB

```
{
  "name": "Bill",
  "surname": "Gates",
  "age": 48,
  "company": {
    "name" : "microsoft",
    "year" : 1974,
    "price" : 300000,
    "employees" : [ { "location" : "USA", "number" :5400},
                     { "location" : "Europe", "number" :1340} ]
  }
}
```

Документо-орієнтована модель даних у MongoDB

Документ представляє набір пар ключ-значення. Ключі є рядками. Значення можуть відрізнятися за типом даних. Усього є наступні типи значень:

String: рядковий тип даних, як у наведеному вище прикладі (для рядків використовується UTF-8).

Array (масив): тип даних для зберігання масивів елементів.

Binary data (двійкові дані): тип для зберігання даних у бінарному форматі.

Boolean: булевий тип даних, що зберігає логічні значення TRUE або FALSE.

Date: зберігає дату у форматі часу Unix.

Double: числовий тип даних для зберігання чисел з плаваючою крапкою.

Integer: використовується для збереження цілочисельних значень.

JavaScript: тип даних для зберігання коду JavaScript.

Min key / Max key: використовуються для порівняння значень із найменшим/найбільшим елементом BSON.

Null: тип даних для зберігання значення Null.

Object: тип даних для зберігання вкладених документів.

ObjectId: тип даних для зберігання id документа.

Regular expression: використовується для зберігання регулярних виразів.

Symbol: тип даних, ідентичний рядковому. Використовується переважно для тих мов, які мають спеціальні символи.

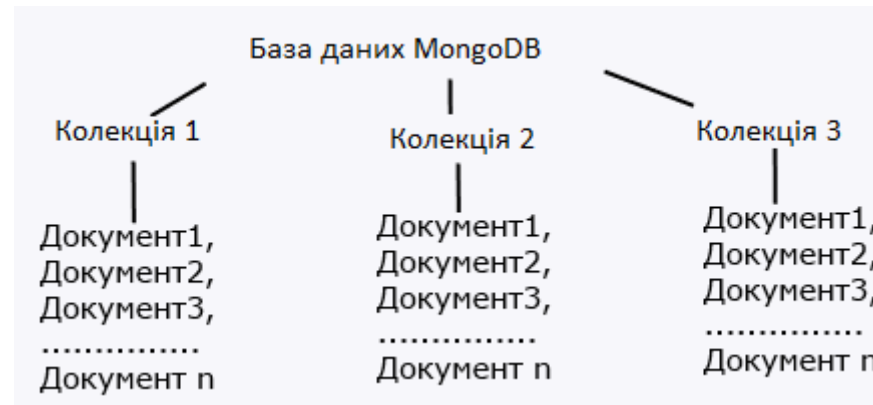
Timestamp: застосовується для зберігання часу.

Бази даних та SQL

Колекції

Якщо в реляційних СУБД вміст складають таблиці, то MongoDB база даних складається з колекцій.

Кожна колекція має своє унікальне ім'я - довільний ідентифікатор, що складається з не більше ніж 128 різних алфавітно-цифрових символів та знаку підкреслення.



Бази даних та SQL

Колекції

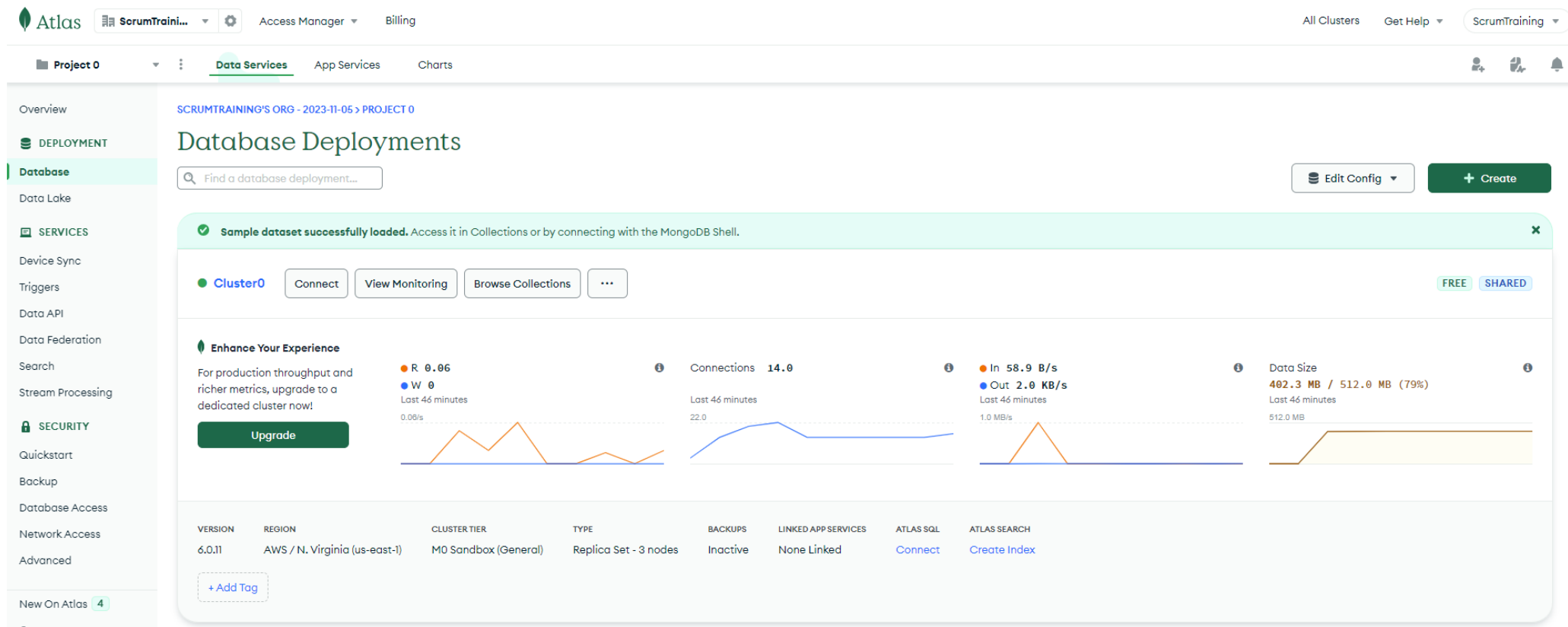
```
{
  "name": "Bill",
  "surname": "Gates",
  "age": 48,
  "company": {
    "name" : "microsoft",
    "year" : 1974,
    "price" : 300000,
    "employees" : [ { "location" : "USA", "number" :5400},
                     { "location" : "Europe", "number" :1340} ]
  }
},
{
  "name": "Steve",
}
```

Встановлення та початок роботи з MongoDB

Бази даних та SQL

Встановлення MongoDB

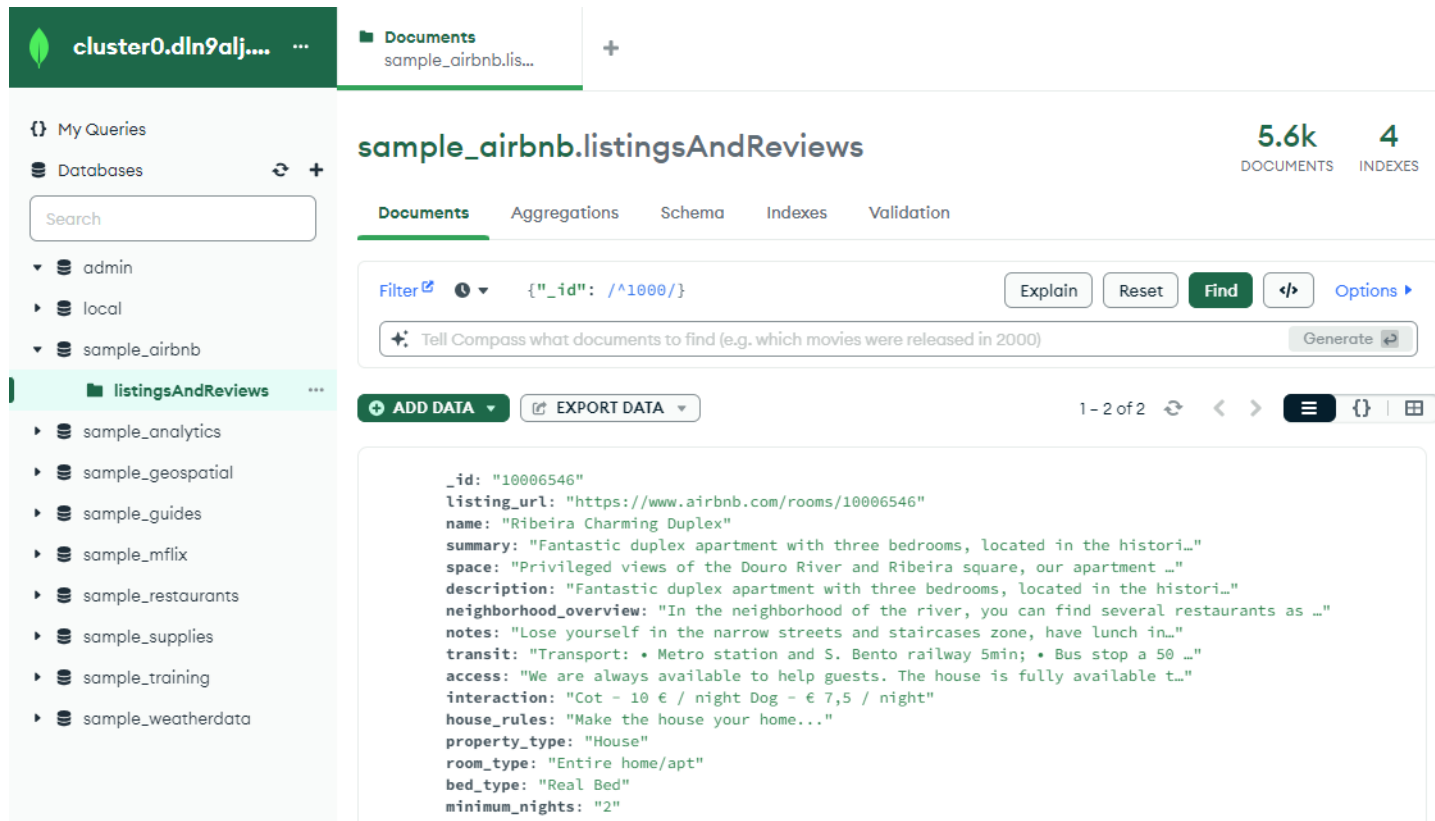
MongoDB Atlas - Повністю керований MongoDB у хмарі



Бази даних та SQL

Графічний клієнт MongoDB Compass

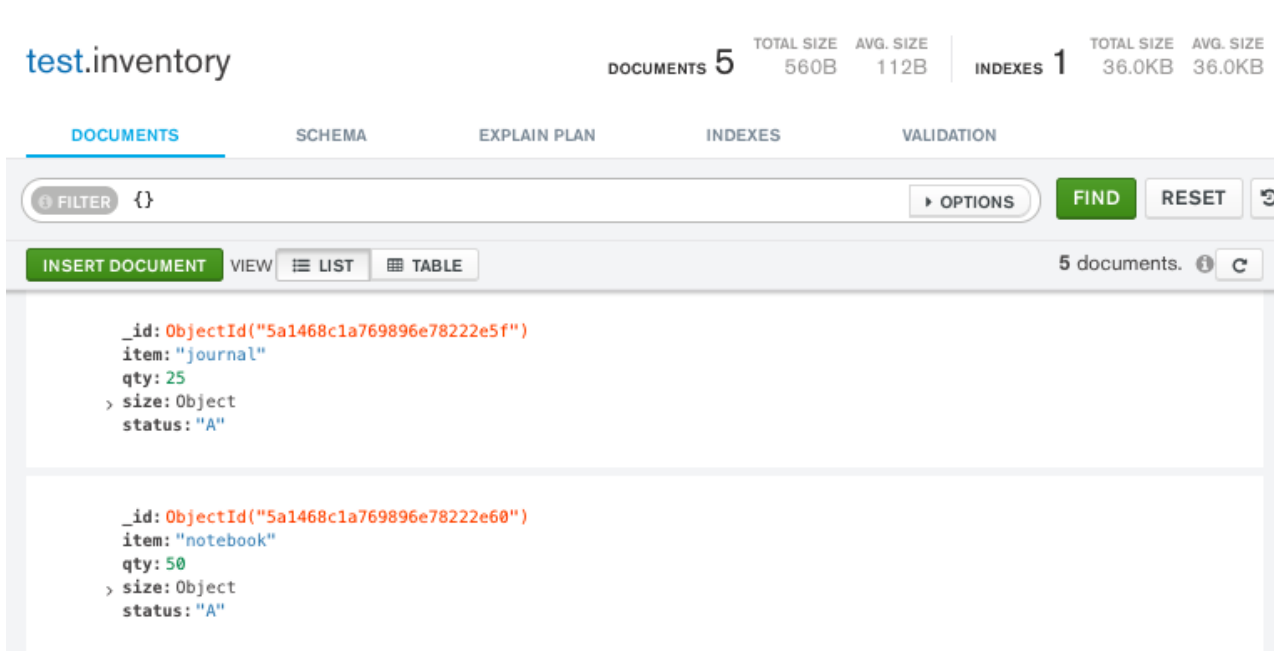
MongoDB Compass – графічний клієнт, який дозволяє виконувати адміністрування бази даних без введення команд та виклику методів.



Базові запити

Бази даних та SQL

Вибераємо усі документи в колекції



The screenshot shows the MongoDB Compass interface for the 'test.inventory' collection. At the top, it indicates 5 documents and 1 index. The 'DOCUMENTS' tab is selected. A filter bar shows an empty filter '{}'. Below the filter, there are buttons for 'INSERT DOCUMENT', 'VIEW', 'LIST', and 'TABLE'. The 'VIEW' button is active. The main area displays two documents in a list view:

```
{ "_id": ObjectId("5a1468c1a769896e78222e5f"),  
  "item": "journal",  
  "qty": 25,  
  "size": Object,  
  "status": "A"  
}
```

```
{ "_id": ObjectId("5a1468c1a769896e78222e60"),  
  "item": "notebook",  
  "qty": 50,  
  "size": Object,  
  "status": "A"  
}
```

Ця операція використовує предикат фільтра {}, який відповідає такому оператору SQL:

```
SELECT * FROM inventory
```



Бази даних та SQL

Визначаємо умову порівняння

The screenshot shows the MongoDB Compass interface for the `test.inventory` collection. At the top, it displays statistics: 5 documents (560B total size, 112B avg size) and 1 index (36.0KB total size, 36.0KB avg size). The `DOCUMENTS` tab is selected. A filter bar contains the query `{ status: "D" }`. Below the filter, there are buttons for `INSERT DOCUMENT`, `VIEW`, `LIST`, and `TABLE`. The results section shows 2 documents. The first document is a paper with a quantity of 100, and the second is a planner with a quantity of 75. Both have a status of "D".

```
{ status: "D" }
```

```
{
  "_id": ObjectId("5a1468c1a769896e78222e61"),
  "item": "paper",
  "qty": 100,
  "size": Object,
  "status": "D"
}
```

```
{
  "_id": ObjectId("5a1468c1a769896e78222e62"),
  "item": "planner",
  "qty": 75,
  "size": Object,
  "status": "D"
}
```

Формат запиту наступний

```
{ <field1>: <value1>, ... }
```

Ця операція використовує предикат фільтра `{ status: "D" }`, який відповідає такому оператору SQL:

```
SELECT * FROM inventory
WHERE status = "D"
```



Бази даних та SQL

Оператори запитів

The screenshot shows the MongoDB Compass interface for a database named 'test.inventory'. The top bar indicates 5 documents and 1 index. The 'FILTER' tab is active, showing a query: `{ status: { $in: ["A", "D"] } }`. Below the query, three documents are displayed in a list view:

- `{ "_id": ObjectId("5a1468c1a769896e78222e5f"), "item": "journal", "qty": 25, "size": Object, "status": "A" }`
- `{ "_id": ObjectId("5a1468c1a769896e78222e60"), "item": "notebook", "qty": 50, "size": Object, "status": "A" }`
- `{ "_id": ObjectId("5a1468c1a769896e78222e61"), "item": "paper", "qty": 100, "size": Object, "status": "D" }`

Формат запиту наступний –

`{ <field1>: { <operator1>: <value1> }, ... }`

Ця операція використовує предикат фільтра

`{ status: { $in: ["A", "D"] } }`, який відповідає такому оператору SQL:

`SELECT * FROM inventory`

`WHERE status in ("A", "D")`



Бази даних та SQL

Усі умовні оператори запитів

Ім'я	опис
<code>\$eq</code>	Відповідає значенням, які дорівнюють вказаному значенню.
<code>\$gt</code>	Відповідає значенням, які перевищують вказане значення.
<code>\$gte</code>	Відповідає значенням, які є більшими або дорівнюють вказаному значенню.
<code>\$in</code>	Відповідає будь-якому зі значень, указаних у масиві.
<code>\$lt</code>	Відповідає значенням, меншим за вказане значення.
<code>\$lte</code>	Відповідає значенням, меншим або рівним вказаному значенню.
<code>\$ne</code>	Відповідає всім значенням, які не дорівнюють вказаному значенню.
<code>\$nin</code>	Не відповідає жодному зі значень, указаних у масиві.

Бази даних та SQL

Умовні оператори запитів. AND

The screenshot shows the MongoDB Compass interface for a database named 'test.inventory'. At the top, it displays statistics: 5 documents (560B total size, 112B avg. size) and 1 index (36.0KB total size, 36.0KB avg. size). Below this, there are tabs for DOCUMENTS, SCHEMA, EXPLAIN PLAN, INDEXES, and VALIDATION. The DOCUMENTS tab is active. A filter bar contains the query: `{ status: "A", qty: { $lt: 30 } }`. To the right of the filter bar are buttons for OPTIONS, FIND, RESET, and a refresh icon. Below the filter bar, there is an 'INSERT DOCUMENT' button and a 'VIEW' section with 'LIST' and 'TABLE' options. The 'LIST' view is selected, showing 1 document. The document details are: `{ _id: ObjectId("5a1468c1a769896e78222e5f"), item: "journal", qty: 25, size: Object, status: "A" }`.

Формат запиту наступний –

`{ <field1>: { <operator1>: <value1> }, ... }`

Ця операція використовує предикат фільтра

`{ status: "A", qty: { $lt: 30 } }`, який відповідає такому оператору SQL:

`SELECT * FROM inventory`

`WHERE status = "A"`

`AND qty < 30`



Бази даних та SQL

Умовні оператори запитів. OR

The screenshot shows the MongoDB Compass interface for a database named 'test.inventory'. The 'DOCUMENTS' tab is active. At the top, statistics show 5 documents (560B total size, 112B avg size) and 1 index (36.0KB total size, 36.0KB avg size). The filter bar contains the query: `{ $or: [{ status: "A" }, { qty: { $lt: 30 } }] }`. Below the filter, there are buttons for 'INSERT DOCUMENT', 'VIEW' (with 'LIST' and 'TABLE' sub-buttons), and '3 documents'. The results are displayed in a list view, showing three documents:

- `{ "_id": ObjectId("5a1468c1a769896e78222e5f"), "item": "journal", "qty": 25, "size": Object, "status": "A" }`
- `{ "_id": ObjectId("5a1468c1a769896e78222e60"), "item": "notebook", "qty": 50, "size": Object, "status": "A" }`
- `{ "_id": ObjectId("5a1468c1a769896e78222e63"), "item": "postcard", "qty": 45, "size": Object, "status": "A" }`

Формат запиту наступний –

`{ <field1>: { <operator1>: <value1> }, ... }`

Ця операція використовує предикат фільтра

`{ $or: [{ status: 'A' }, { qty: { $lt: 30 } }] }`, який

відповідає такому оператору SQL:

`SELECT * FROM inventory`

`WHERE status = "A"`

`OR qty < 30`



Бази даних та SQL

Усі логічні оператори запитів

Ім'я	опис
<code>\$and</code>	Об'єднує речення запиту з логічним <code>AND</code> поверненням усіх документів, які відповідають умовам обох речень.
<code>\$not</code>	Інвертує дію виразу запиту та повертає документи, які не відповідають виразу запиту.
<code>\$nor</code>	Об'єднує речення запиту з логічним <code>NOR</code> повертає всі документи, які не відповідають обом реченням.
<code>\$or</code>	Об'єднує пропозиції запиту з логічним <code>OR</code> поверненням усіх документів, які відповідають умовам будь-якої пропозиції.

Aggregation framework

Бази даних та SQL

Aggregation framework

Aggregation framework надає функціонал для створення та виконання агрегатних запитів (угруповання документів за ключами, обчислення загального або середнього значення тощо) до MongoDB.

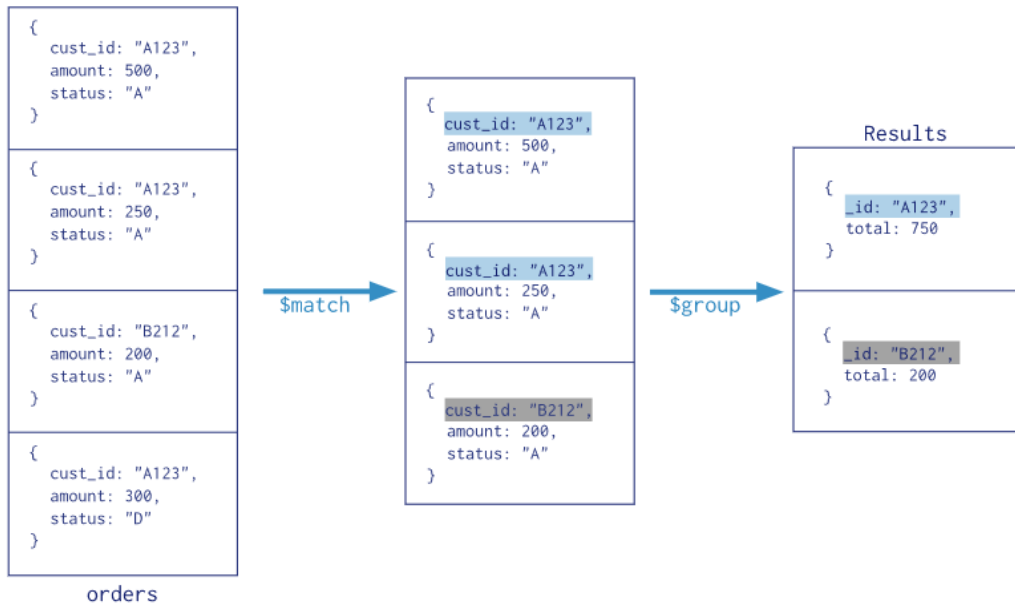


Бази даних та SQL

Порівняння з Map-Reduce

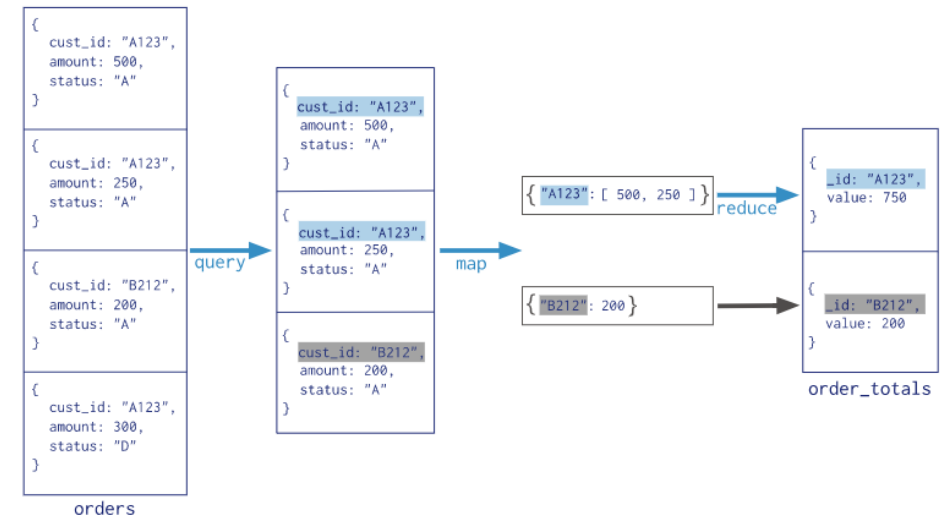
Aggregation framework

```
Collection
↓
db.orders.aggregate( [
  $match stage → { $match: { status: "A" } },
  $group stage → { $group: { _id: "$cust_id", total: { $sum: "$amount" } } }
] )
```



Map-Reduce

```
Collection
↓
db.orders.mapReduce(
  map → function() { emit( this.cust_id, this.amount ); },
  reduce → function(key, values) { return Array.sum( values ); },
  query → { query: { status: "A" },
  output → out: "order_totals"
} )
```



Бази даних та SQL

Оператори aggregation pipeline

SQL

WHERE



MongoDB

\$match

GROUP BY



\$group

HAVING



\$match

SELECT



\$project

ORDER BY



\$sort

Бази даних та SQL

Ресурси

Aggregation framework документація:

<https://docs.mongodb.com/manual/core/aggregation-pipeline/>

Оператори та вирази

<https://docs.mongodb.com/manual/meta/aggregation-quick-reference/>

Бази даних та SQL

Що ми сьогодні вивчили

Бази даних та SQL

План уроку

- Введення в MongoDB
- Як розпочати роботу з MongoDB
- Базові запити
- Агрегуючі запити

Бази даних та SQL

Підсумки

Що одне, найголовніше, ви дізналися сьогодні?

Бази даних та SQL

Дякую за увагу! До нових зустрічей!

Інформаційний відеосервіс для розробників програмного забезпечення

