

Module 2 - Tuần 1 - Tổng hợp kiến thức Buổi học số 3

Time-Series Team

Ngày 3 tháng 7 năm 2025

Buổi học số 3 (Thứ 5, 03/07/2025) bao gồm ba nội dung chính:

- *Phần I: Giới thiệu về NoSQL*
- *Phần II: Giới thiệu về MongoDB*
- *Phần III: Ngôn ngữ truy vấn MongoDB*

Phần I: Giới thiệu về NoSQL

1 Khái niệm cơ bản về SQL

1.1 Database là gì?

Khái niệm cơ sở dữ liệu bắt đầu từ việc hiểu **dữ liệu (data)** là gì:

- **Dữ liệu** là các thông tin thô, có thể ở dạng số, chữ, hình ảnh, âm thanh hoặc video.
- **Cơ sở dữ liệu (Database)** là nơi tổ chức lưu trữ các dữ liệu đó một cách có hệ thống, giúp truy xuất – cập nhật – quản lý một cách hiệu quả.

Name: Thai

Age: 20

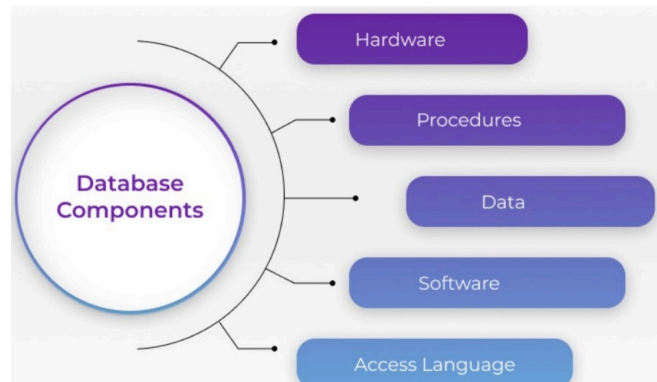
Image: [link](#)



Ở đây, mỗi phần tử là một trường dữ liệu, cấu thành nên một đơn vị dữ liệu hoàn chỉnh.

Ví dụ

1.2 Các thành phần chính trong hệ thống cơ sở dữ liệu



Một cơ sở dữ liệu hiện đại không thể hoạt động độc lập mà bao gồm nhiều thành phần hỗ trợ như:

- **Thiết bị phần cứng (Hardware):** máy chủ, bộ nhớ, ổ cứng...
- **Quy trình vận hành (Procedures):** hướng dẫn cách tương tác, sao lưu, bảo mật,...
- **Dữ liệu (Data):** Tất cả thông tin mà DBMS có thể thu thập, truy cập,...
- **Phần mềm quản lý (Software):** Chương trình kiểm soát, quản lý toàn bộ cơ sở dữ liệu.
- **Ngôn ngữ truy vấn:** Giúp viết các lệnh để làm việc với dữ liệu trong cơ sở dữ liệu

1.3 DBMS là gì?

DBMS (Database Management System) là phần mềm trung gian cho phép:

- Tạo, sửa, xoá cơ sở dữ liệu và các bảng dữ liệu.
- Đảm bảo dữ liệu toàn vẹn và không bị trùng lặp.
- Kiểm soát truy cập, phân quyền người dùng.
- Tối ưu truy vấn, cải thiện hiệu năng hệ thống.

Các hệ DBMS phổ biến:

- SQL: MySQL, PostgreSQL, Oracle, SQL Server.
- NoSQL: MongoDB, Cassandra, Firebase, Couchbase.

1.4 SQL Database

SQL (Structured Query Language) là hệ thống cơ sở dữ liệu quan hệ (RDBMS) lưu trữ dữ liệu trong các bảng (table) với các hàng (row) và cột (column).

Ví dụ:

studentID	studentName
1	Anh
2	Thai

studentID	subjectID	mark
1	S1	3.0
2	S2	3.0

subjectID	subjectName
S1	Machine Learning
S2	Deep Learning

2 NoSQL Database

NoSQL không có nghĩa là “không dùng SQL” mà là “**Not Only SQL**” – tức là:

- Không quan hệ.
- Không cần hệ quản trị RDBMS.
- Không sử dụng cấu trúc bảng.
- Có thể lưu dữ liệu linh hoạt dưới dạng cặp **key-value** hoặc tài liệu (document).

Ví dụ dạng document:

```
{ "studentID": 1, "studentName": "Anh" }
```

2.1 Các loại NoSQL Database

Các mô hình chính của NoSQL:

1. **Key-Value Store Database**
2. **Graph Database**
3. **Document Database**
4. (Khác): Column Store, Multi-model...

2.1.1 Key-Value Store Database

Cấu trúc dữ liệu dạng bảng băm (hashmap) hoặc dictionary:

- Mỗi bản ghi là một cặp **key - value**.
- Truy cập cực nhanh qua khóa chính.

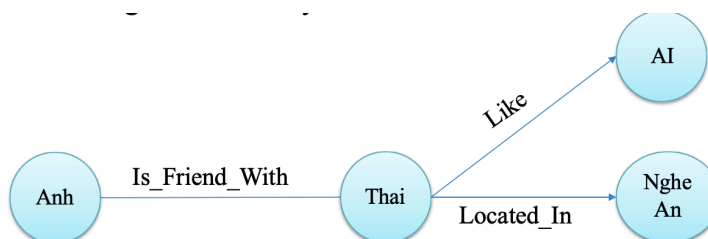
2.1.2 Graph Database

Dành cho hệ thống có mối quan hệ phức tạp giữa các thực thể.

Dữ liệu được biểu diễn bằng:

- **Node** (nút) (thực thể)
- **Edge** (liên kết giữa các nút) (quan hệ)
- **Attribute** (thuộc tính)

Ví dụ:



Hữu ích với các hệ thống như: mạng xã hội, đề xuất bạn bè, hệ thống phân tích quan hệ.

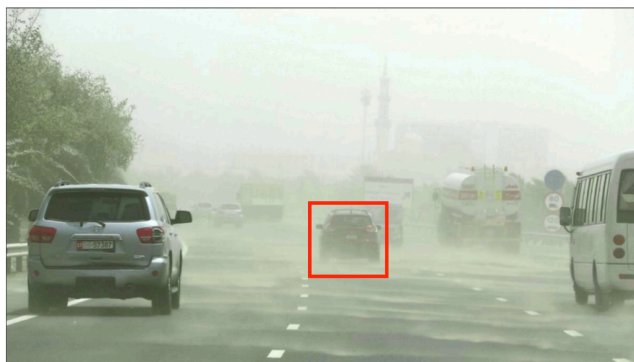
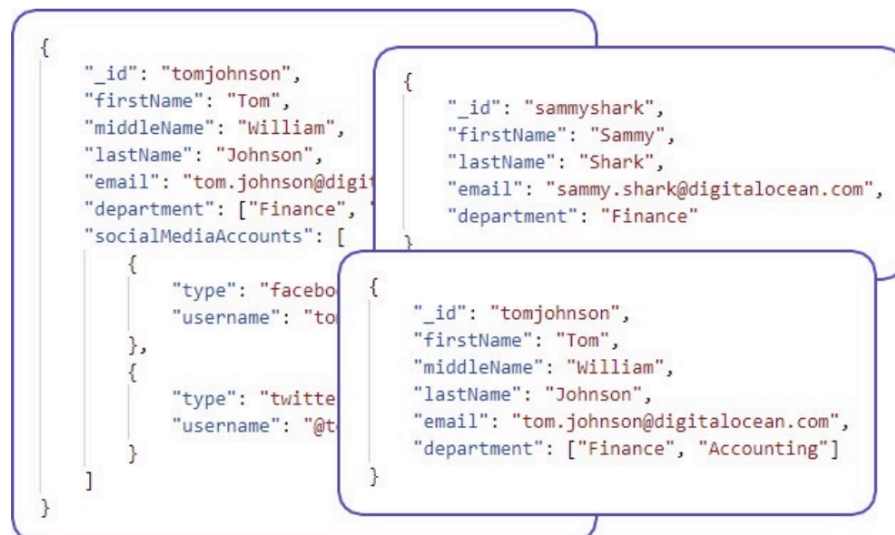
2.1.3 Document Database

Lưu trữ dữ liệu dưới dạng tài liệu (document), thường là:

- JSON
- BSON (Binary JSON)
- XML

Mỗi document có:

- `_id` – định danh duy nhất
- Các cặp key-value
- Cấu trúc linh hoạt (schema-less)



Image

```

<object>
  <name>LP</name>
  <pose>Unspecified</pose>
  <truncated>Unspecified</truncated>
  <difficult>Unspecified</difficult>
  - <bndbox>
    <xmin>1195</xmin>
    <ymin>2091</ymin>
    <xmax>1585</xmax>
    <ymax>2203</ymax>
  </bndbox>
</object>

```

XML label

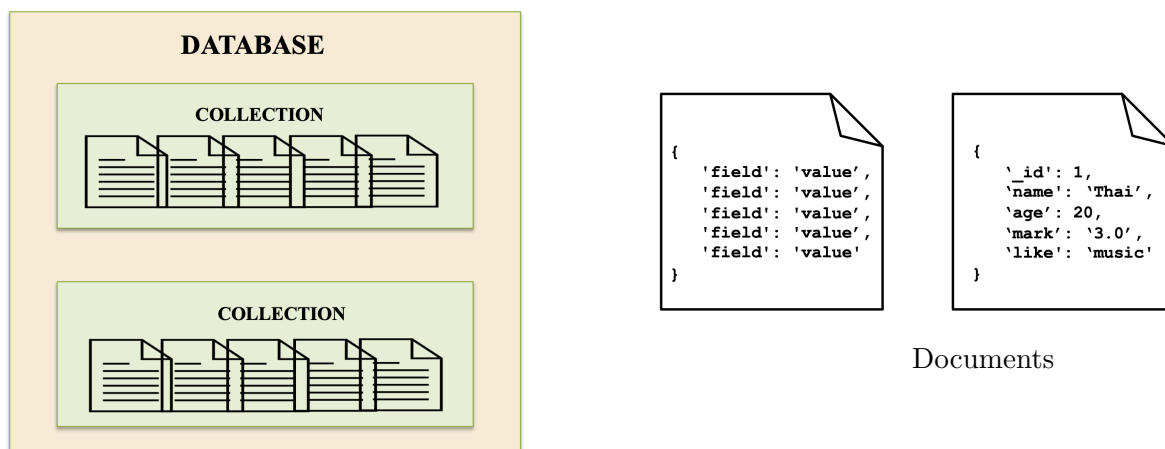
Phần II: Giới thiệu về MongoDB

1 Mongo DB là gì?

MongoDB là hệ quản trị cơ sở dữ liệu NoSQL dạng tài liệu, lưu trữ dữ liệu theo mô hình:

- **Database:** hệ thống tổng thể chứa nhiều collections.
- **Collection:** tương tự như “table” trong SQL, chứa nhiều documents.
- **Document:** đơn vị lưu trữ chính, dạng JSON (hoặc BSON).

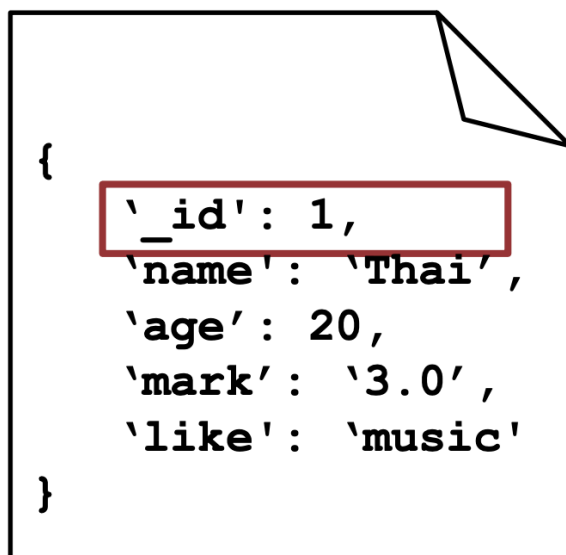
Mỗi document là một cấu trúc JSON với các cặp key-value, có thể chứa mảng hoặc document lồng nhau.



1.1 Tài liệu (Documents) trong MongoDB

MongoDB sử dụng định dạng **JSON (Java Script Object Notation)** cho dữ liệu – một định dạng rất phổ biến trong các ứng dụng web và API.

Có thể lưu dữ liệu có cấu trúc linh hoạt, không cần schema cố định.



Trường `_id` là một định danh duy nhất, MongoDB sẽ tự động sinh nếu không cung cấp.

1.2 Các kiểu dữ liệu trong MongoDB

MongoDB hỗ trợ rất nhiều kiểu dữ liệu:

Kiểu dữ liệu	Mô tả
Number	Số nguyên, số thực
String	Chuỗi ký tự
Date	Ngày giờ
Array	Mảng các giá trị
Object	Document lồng nhau

```
{
  '_id': 1,
  'name': 'Thai',
  'phone': ['0123', '1235'],
  'dateOfBirth': 10/07/1997,
}
```

1.3 Tính linh hoạt trong cấu trúc tài liệu (Document Structure)

MongoDB không yêu cầu các document trong cùng một collection phải có cấu trúc giống nhau (tính polymorphic).

Có thể:

- Thêm bớt trường tùy ý
- Sử dụng document lồng nhau (nested document)
- Sử dụng mảng (arrays)

```
{
  '_id': 1,
  'name': 'Thai',
  'age': 20,
  'address': {
    'city': 'Ha Noi',
    'postcode': '100'
  }
}
```

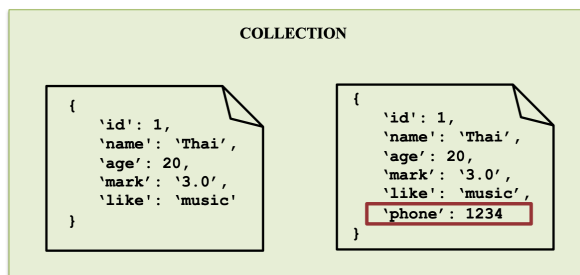
Nested Document

```

{
  '_id': 1,
  'name': 'Thai',
  'phone': ['0123', '1235'],
  'age': 20,
  'address': {
    'city': 'Ha Noi',
    'postcode': '100'
  }
}

```

Array



- Không cần định nghĩa schema trước như SQL. Điều này rất hữu ích khi dữ liệu thay đổi thường xuyên hoặc không thể đoán trước cấu trúc.
- Có thể dễ dàng thay đổi cấu trúc tài liệu riêng lẻ, ví dụ: có thể thêm cặp giá trị mới vào trường.

Có thể có các document trong cùng một collection

1.4 Hệ sinh thái MongoDB

MongoDB không chỉ là một hệ quản trị CSDL, mà còn là một hệ sinh thái bao gồm nhiều công cụ và nền tảng hỗ trợ:

1.4.1 MongoDB Distribution:

- **Community Edition:** mã nguồn mở, miễn phí.
- **Enterprise Edition:** dành cho doanh nghiệp, có tính năng bảo mật và giám sát cao cấp.
- **MongoDB Atlas:** dịch vụ MongoDB trên nền tảng đám mây (Cloud), do chính Mongo vận hành.

1.4.2 Công cụ hỗ trợ:

- **Mongo Shell:** công cụ dòng lệnh tương tác với MongoDB.
- **MongoDB Compass:** giao diện đồ họa (GUI) giúp quản lý cơ sở dữ liệu dễ dàng.
- **Drivers:** hỗ trợ nhiều ngôn ngữ như Java, Python, Node.js...
- **BI Connector:** tích hợp với các công cụ Business Intelligence.

2 Cài đặt MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL phổ biến. VSCode là trình soạn thảo hỗ trợ tốt việc kết nối MongoDB qua các extension. Trước tiên, cần có tài khoản và database trên MongoDB Atlas.

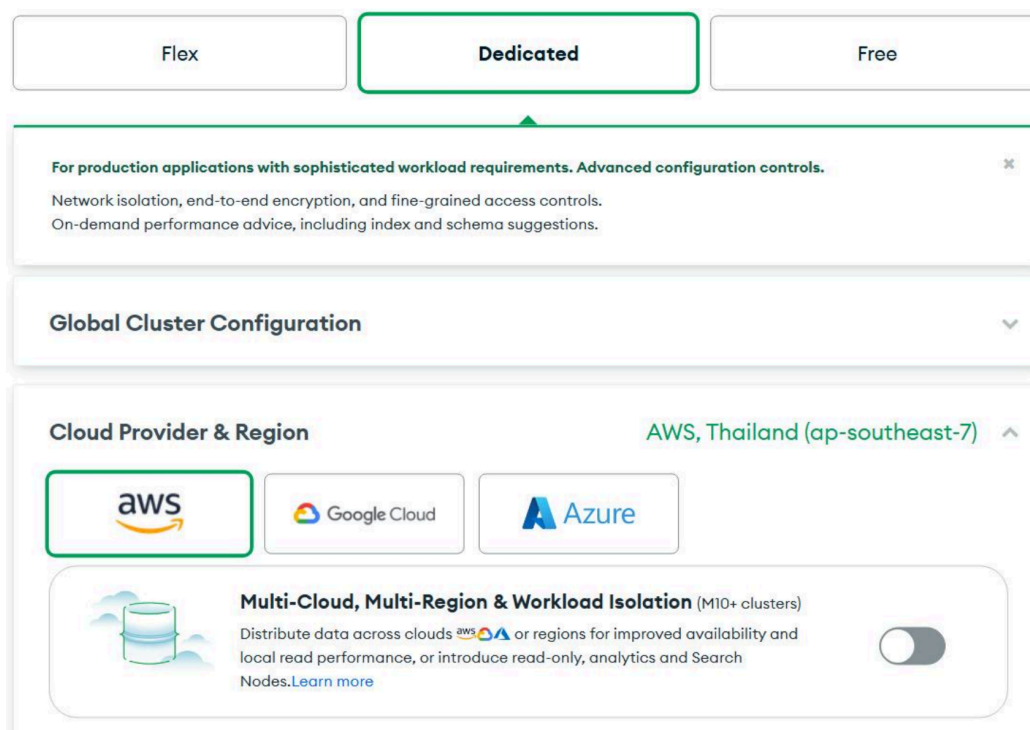
2.1 Cài đặt và cấu hình MongoDB Atlas

2.1.1 Tạo tài khoản MongoDB Atlas

1. Truy cập: <https://www.mongodb.com/atlas/database>
2. Nhấn **Sign Up**, dùng email hoặc Google để đăng ký.

2.1.2 Tạo Cluster miễn phí

1. Chọn **Build a Database**
2. Chọn gói **Shared - Free**
3. Chọn vùng gần nhất (VD: AWS Singapore)
4. Nhấn **Create Cluster**



Create Free Cluster

2.1.3 Thêm người dùng Database

1. Vào mục **Database Access**
2. Chọn **Add New Database User**
3. Đặt username, password, quyền truy cập
4. Nhấn **Add User**

Edit User: haopros123er@admin

Update a database user to grant an application or user access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding [Access Manager](#)

Authentication Method

Password Certificate AWS IAM Federated Auth (MongoDB 7.0 and up)

MongoDB uses [SCRAM](#) as its default authentication method.

Password Authentication

haopros123er

Edit Password

User Description

Add an optional description to your user.

Database User Privileges

Add Database User

2.1.4 Cấu hình IP Access

1. Vào **Network Access**
2. Nhấn **Add IP Address**
3. Nhập IP: 0.0.0.0/0 hoặc IP máy
4. Nhấn **Confirm**

2.2 Kết nối VSCode với MongoDB

2.2.1 Cài đặt Extension

1. Mở VSCode
2. Nhấn Ctrl + Shift + X
3. Tìm: MongoDB for VSCode

2.2.2 Lấy connection string

1. Vào Cluster → Connect → MongoDB for VSCode
2. Tìm mục **Connect to your MongoDB deployment**

2.2.3 Tạo kết nối

1. Mở sidebar MongoDB trong VSCode
2. Nhấn **Connect**, nhập connection string:

```
1 mongodb+srv://<username>:<password>@<cluster-url>/test
```

2.2.4 Kiểm tra kết nối và thao tác cơ bản

```
1 db.students.insertMany([
2   { name: "Nguyen Van A", age: 20, major: "CNTT" },
3   { name: "Tran Thi B", age: 22, major: "Kinh ết" },
4   { name: "Le Van C", age: 21, major: "CNTT" }
5 ]);
```

```
1 db.students.find({ age: { $gte: 21 } });
```

2.3 Lỗi thường gặp

- Không kết nối: Kiểm tra connection string, username, password.
- Lỗi IP: Thêm IP vào whitelist trong MongoDB Atlas.
- Thiếu extension: Cài lại và khởi động lại VSCode.

Phần III: Ngôn ngữ truy vấn MongoDB

1. Tạo/Xoá Cơ sở dữ liệu và Collection

[Use MongoDB Atlas](#)[Use MongoDB Compass](#)[Use MongoDB Shell](#)

```
1 show dbs // show databases
2 use myDatabase // create/switch to a database
3 db.dropDatabase() // delete the database
4 show collection // show collections in the database
5 db.createCollection("students") // create a collection
6 db.students.drop() // delete a collection
```

2. Thêm Dữ Liệu (Insert)

Thêm một document:

```
1 db.students.insert({ "name": "Thai" });
```

Thêm nhiều document:

```
1 db.students.insert([
2   { "name": "Thai" },
3   { "age": 20 }
4 ]);
```

Chú ý lỗi trùng `_id`:

```
1 db.users.insertOne({ _id: 1, name: "Alice" });
2 db.users.insertOne({ _id: 1, name: "Bob" }); // lỗi
```

3. Import / Export dữ liệu

```
1 mongoexport --uri="mongodb+srv://..." --collection=students --out=students.json
2 mongoimport --uri="mongodb+srv://..." --collection=companies --file=companies.json
```

4. Truy vấn Dữ liệu

Tìm 1 document đầu tiên:

```
1 db.students.findOne();
```

Tìm tất cả document có điều kiện:

```
1 db.students.find({ "name": "Anh" });
2 db.students.find({ "name": "Anh", "age": 20 });
```

5. Toán Tử So Sánh

- `$eq` – bằng
- `$lt` / `$lte` – nhỏ hơn (hoặc bằng)
- `$gt` / `$gte` – lớn hơn (hoặc bằng)
- `$in` – thuộc tập hợp
- `$nin` – không thuộc tập hợp

```
1 db.students.find({ "salary": { $eq: 50000 } });
2 db.students.find({ "salary": { $lt: 30000 } });
3 db.students.find({ "salary": { $in: [1000, 2000] } });
```

6. Toán Tử Logic

```
1 db.students.find({
2   $and: [
3     { "age": 20 },
4     { "mark": { $gt: 8.0 } }
5   ]
6 });
```

Tương đương:

```
1 db.students.find({
2   "age": 20,
3   "mark": { $gt: 8.0 }
4 });
```

7. `$expr` – So sánh giữa các trường

```
1 db.routes.find({
2   $expr: { $eq: ["$src_airport", "$dst_airport"] }
3 });
```

8. `$exists` và `$type`

Kiểm tra trường tồn tại:

```
1 db.companies.find({ "ipo": { $exists: true } });
```

Kiểm tra kiểu dữ liệu:

```
1 db.companies.find({ "homepage_url": { $type: 2 } }); // string
```

9. Cursor Methods

```
1 db.students.find().count();
2 db.students.find().limit(10);
3 db.students.find().skip(5);
4 db.students.find().sort({ "mark": -1 });
```

10. Projection – Lọc trường kết quả

```
1 db.students.find({}, { "name": 1, "mark": 1, "_id": 0 });
```

11. Truy vấn document lồng nhau

```
1 db.inspections.find({ "address.zip": 11385 });
```

12. Truy vấn mảng (Array)

\$all – chứa đủ phần tử:

```
1 db.students.find({ "skills": { $all: ["math", "english"] } });
```

\$size – kiểm tra độ dài:

```
1 db.students.find({ "hobbies": { $size: 2 } });
```

\$elemMatch – phần tử thỏa nhiều điều kiện:

```
1 db.grades.find({
2   "scores": {
3     $elemMatch: { "type": "exam", "score": { $gt: 80 } }
4   }
5 });
```

13. Xoá Dữ liệu

```
1 db.students.deleteOne({ "name": "Anh" });
2 db.students.deleteMany({ "name": "Anh" });
```

14. Cập nhật Dữ liệu

\$set, **\$unset**, **\$rename**, **\$inc**, **\$push**:

```
1 db.students.updateOne(
2   { "name": "Thai" },
3   { $set: { "name": "Anh" } }
4 );
5
6 db.students.updateMany(
7   {},
8   { $set: { "address": "Ha Noi" } }
9 );
10
11 db.students.updateMany(
12   {},
13   {
14     $unset: { "address": "" },
15     $rename: { "hobbies_list": "hobbies" }
16   }
17 );
18
19 db.students.updateMany(
20   { "age": 20 },
21   { $inc: { "age": 1 } }
22 );
```