MongoDB

的安装与使用

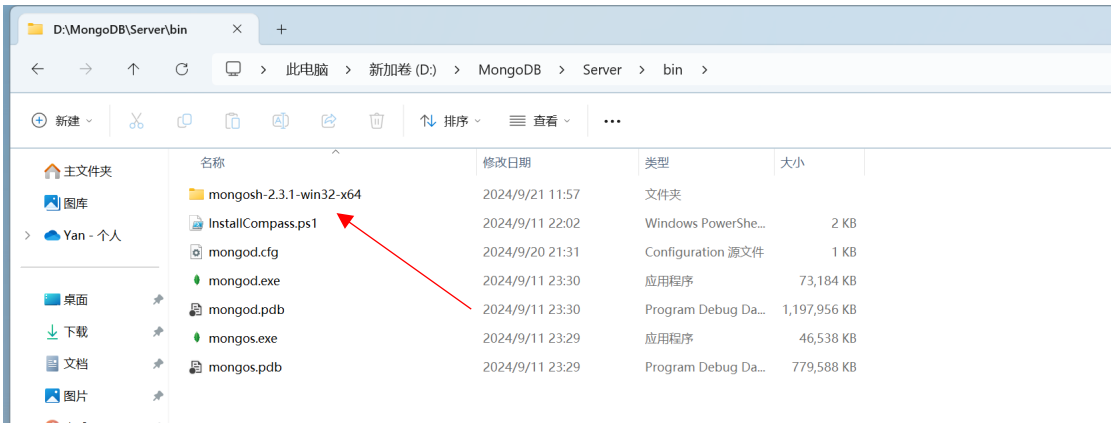
计算机科学与技术学院
2352018 刘彦
2024 年 9 月 21 日

一、 实验步骤

1. 下载安装与配置环境变量

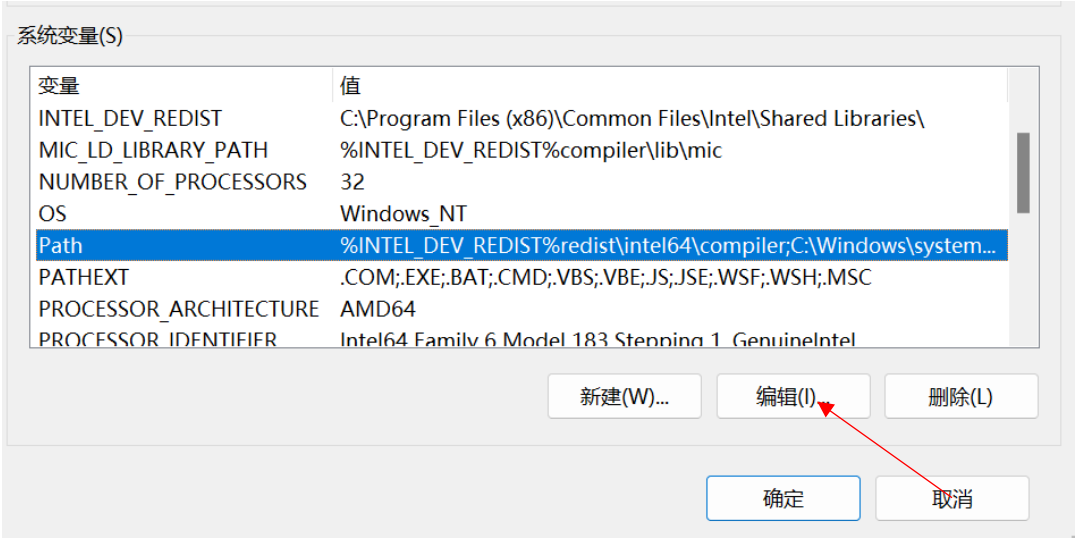
(1)下载后 MongoDB Server 后，在安装时点击安装 MongoDB Compass 可视化界面，桌面出现这样的快捷方式。

(2)下载 MongoDB Shell，将文件夹解压到 bin 目录下。

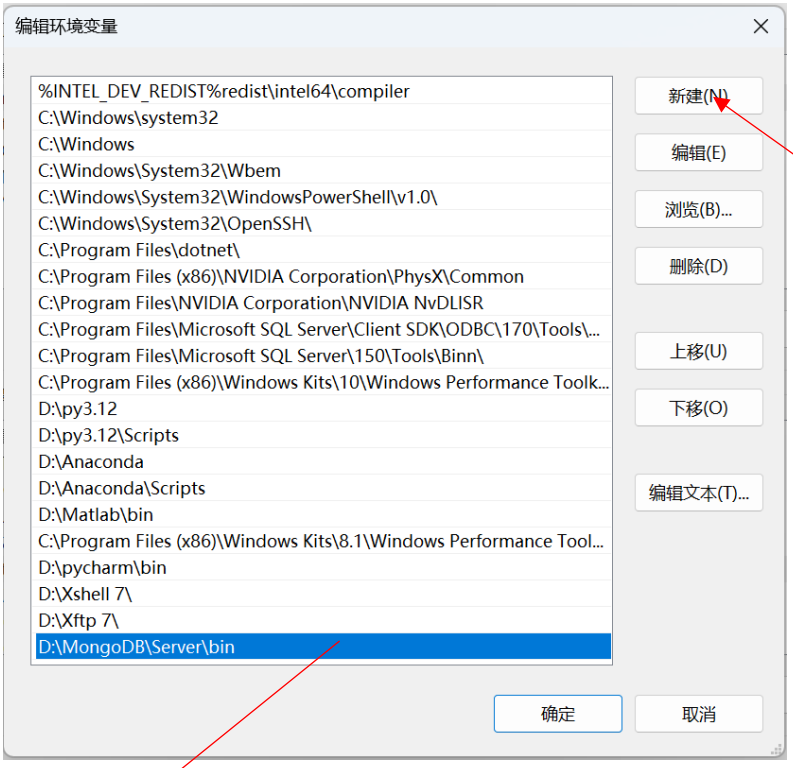


(3)配置环境变量

在安装完成后，在系统设置中配置环境变量。



新建两条 MongoDB Server 和 MongoDB Shell 的环境变量：



D:\MongoDB\Server\bin

D:\MongoDB\Server\bin\mongosh-2.3.1-win32-x64\bin

(4)安装后设置数据目录，运行 MongoDB 服务器。

启动 MongoDB 服务器： 使用 mongod 命令并指定 dbpath：

```
C:\Users\PC>mongod --dbpath d:\mongodb_data\db
```

输入该指令后，mongod 服务器启动，数据储存在 d:\mongodb_data\db 目录中。

- "mongod startup complete"：这是确认启动完成的关键日志。
- "Listening on" 和 "Waiting for connections"：这表明服务正在监听指定的地址和端口（通常是 127.0.0.1:27017），并准备接受连接。

需要先启动 mongod 服务器，然后才能在另一个终端中运行 mongosh 来连接它。可以确认 MongoDB 服务是否正在运行以及它是否在监听预期的端口。出现如下情况说明服务器已经启动。

```
C:\Users\PC>tasklist | findstr mongod
mongod.exe                   6060 Services                   0     54,648 K

C:\Users\PC>netstat -ano | findstr :27017
  TCP    127.0.0.1:27017        0.0.0.0:0              LISTENING       6060
```

2. 启动与数据库

(1)MongoDB Shell 的启动

配置好环境变量且 mongod 服务器启动后，在命令行输入 mongosh 启动 MongoDB Shell，出现如下界面即为启动成功。会显示当前 mongosh LOG ID，和 mongodb、mongosh 的版本。

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [版本 10.0.22631.4169]
(c) Microsoft Corporation。保留所有权利。

C:\Users\PC>mongosh
Current Mongosh Log ID: 66ed8217d0eede2673c73bf7
Connecting to:          mongodb://127.0.0.1:27017/?directConn
ection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3
.1
Using MongoDB:          8.0.0
Using Mongosh:          2.3.1

For mongosh info see: https://www.mongodb.com/docs/mongodb-sh
ell/

------
   The server generated these startup warnings when booting
   2024-09-20T21:31:44.163+08:00: Access control is not enabl
ed for the database. Read and write access to data and config
uration is unrestricted
------

test>
```

(2)创建或选择数据库

use …

出现如下界面即为创建或选择成功，并自动跳转到这个数据库。

```
test> use education
switched to db education
education>
```

(3)显示数据库

show dbs

显示所有数据库，其中，admin、config 和 local 为自带数据库。

```
test> show dbs
admin       40.00 KiB
config      12.00 KiB
education   40.00 KiB
local       72.00 KiB
test>
```

(4)删除数据库

db.dropDatabase()

可以删除当前数据库，会显示是否删除成功和删除数据库名称。

```
education> db.dropDatabase();
{ ok: 1, dropped: 'education' }
education>
```

3. 集合

(1)创建集合

db.createCollection(name,……)

```
education> db.createCollection("student", { capped: true, autoIndexId: true, size: 6142800, max: 10000 })
{ ok: 1 }
education>
```

(2)显示数据库中的集合

show collections

```
education> show collections
student
education>
```

(3)删除集合

db.collection_name.drop()

```
education> db.student.drop();
true
education>
```

4. 文档

(1)创建文档

db.collection_name.insertOne({……})

```
education> db.student.insertOne({ name: 'San Zhang', age: 18, dept: 'Computer Science' })
{
  acknowledged: true,
  insertedId: ObjectId('66ed8688d0b7439fcec73bf9')
}
education>
```

输入执行后还会显示其 ObjectID。

还可以同时插入多条：

db.collection_name.insertMany({……})

```
education> db.student.insertMany([
...     { name: 'Wu Wang', age: 18, dept: 'Data Science' },
...     { name: 'Li Si', age: 19, dept: 'Mathematics' }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66ed86d4d0b7439fcec73bfa'),
    '1': ObjectId('66ed86d4d0b7439fcec73bfb')
  }
}
education>
```

(2)显示文档

db.collection_name.find()

```
education> db.student.find()
[
  {
    _id: ObjectId('66ed8688d0b7439fcec73bf9'),
    name: 'San Zhang',
    age: 18,
    dept: 'Computer Science'
  },
  {
    _id: ObjectId('66ed86d4d0b7439fcec73bfa'),
    name: 'Wu Wang',
    age: 18,
    dept: 'Data Science'
  },
  {
    _id: ObjectId('66ed86d4d0b7439fcec73bfb'),
    name: 'Alice',
    age: 19,
    dept: 'Mathematics'
  },
  {
    _id: ObjectId('66ed8cedd0b7439fcec73bfc'),
    name: 'San Zhang',
    age: 18,
    dept: 'Computer Science'
  }
]
education>
```

(3)替换文档

db.collection_name.replaceOne({……}, {……})

此时会替换第一个满足条件的文档：

```
education> db.student.replaceOne(
...     { name: "Wu Wang" },
...     { name: "Liu Zhao", age: 20, dept: 'Art' }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
education>
```

如果要替换所有满足条件的文档，则使用

db.collection_name.replaceMany({……}, {……})

(4)删除文档

db.collection_name.deleteOne({……})

此时会删除第一个满足条件的文档：

```
education> db.student.deleteOne({ name: 'San Zhang', age: 18 })
{ acknowledged: true, deletedCount: 1 }
education>
```

如果要删除所有满足条件的文档，则使用

db.collection_name.deleteMany({……})

(5)查询文档

db.collection_name.find({…: {…}})

比较操作符：

- $eq: 等于
- $ne: 不等于
- $gt: 大于
- $gte: 大于或等于
- $lt: 小于
- $lte: 小于或等于

例如：

```
education> db.student.find({ age: { $gt: 18 } })
[
  {
    _id: ObjectId('66ed86d4d0b7439fcec73bfb'),
    name: 'Li Si',
    age: 19,
    dept: 'Mathematics'
  }
]
education>
```

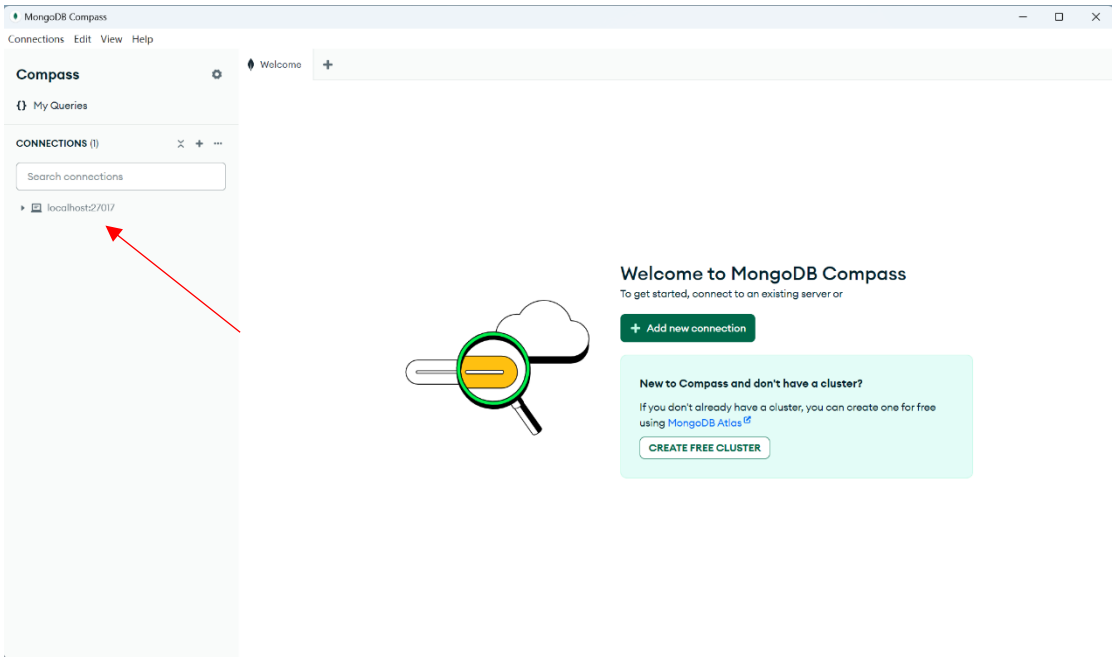
还可以附加多个条件：

```
education> db.student.find({ name: 'San Zhang', age: { $gte: 18 } })
[
  {
    _id: ObjectId('66ed8688d0b7439fcec73bf9'),
    name: 'San Zhang',
    age: 18,
    dept: 'Computer Science'
  },
  {
    _id: ObjectId('66ed8cedd0b7439fcec73bfc'),
    name: 'San Zhang',
    age: 18,
    dept: 'Computer Science'
  }
]
education>
```

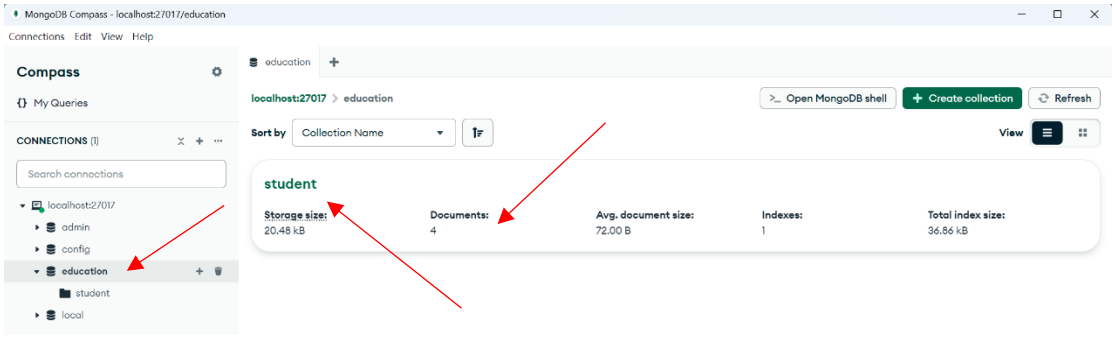
5. MongoDB Compass 可视化界面的使用

(1)开启和连接

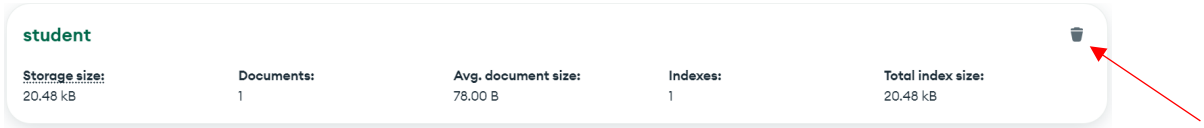
打开后在 connections 中连接：



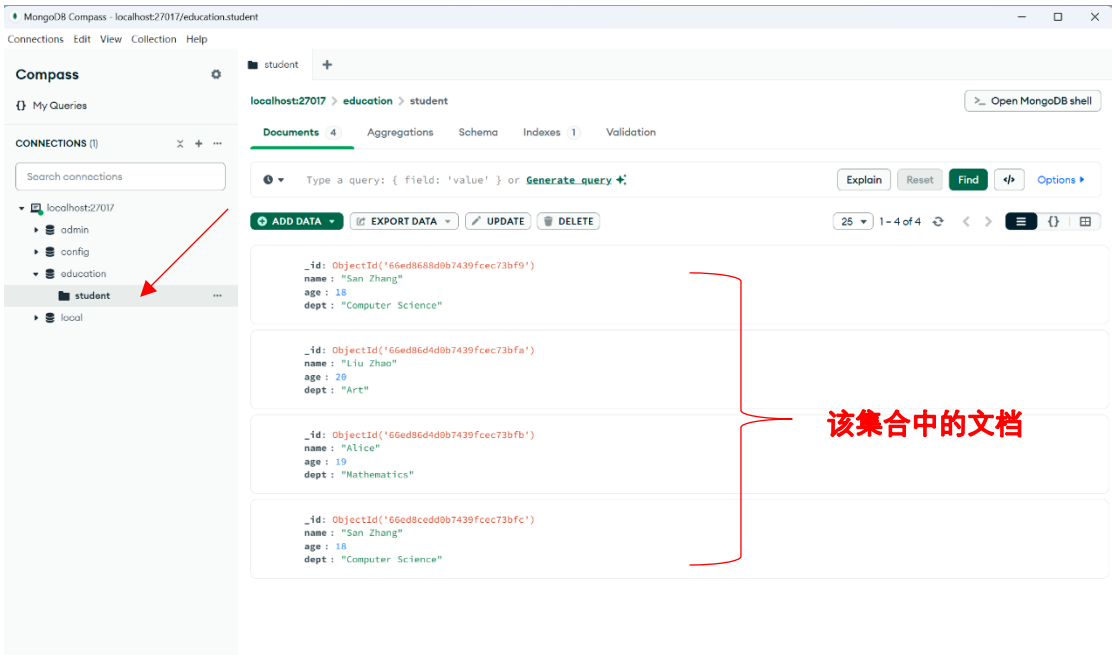
点击本地连接，即可看到现有数据库和集合及其相关参数：



也可以快速删除集合：



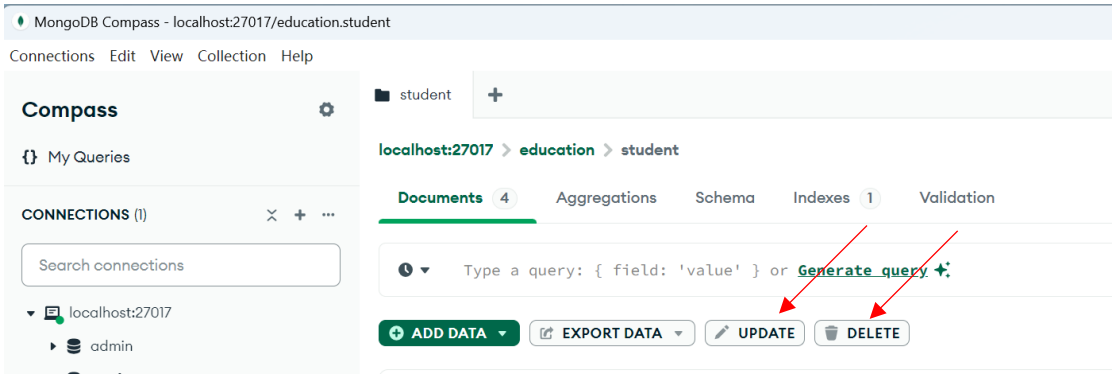
进入相应集合后，可以看到里面的文档：

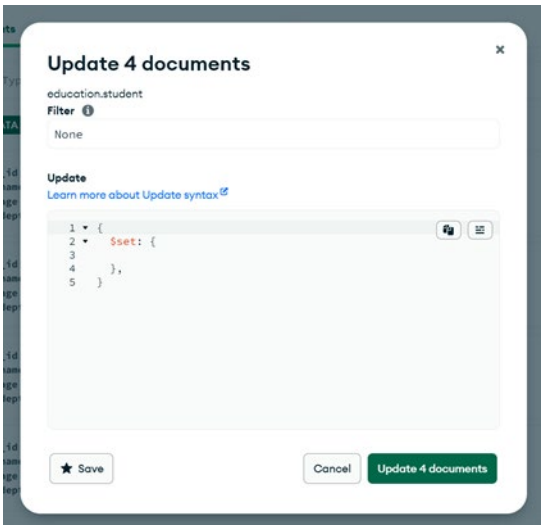


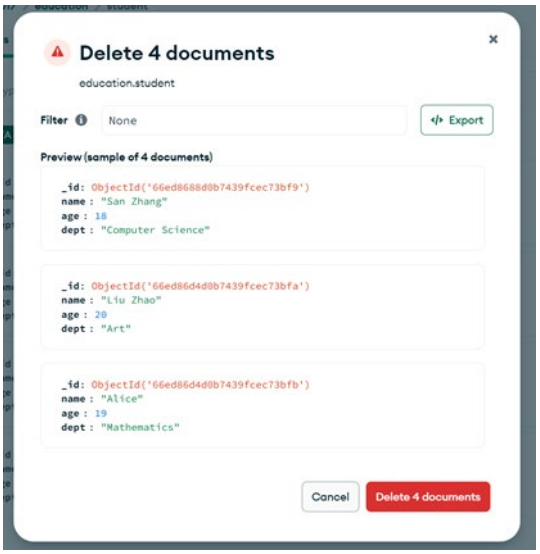
(2) 数据的修改

① 整体修改

点击上方 UPDATE/DELETE 可以整体更换/删除数据：

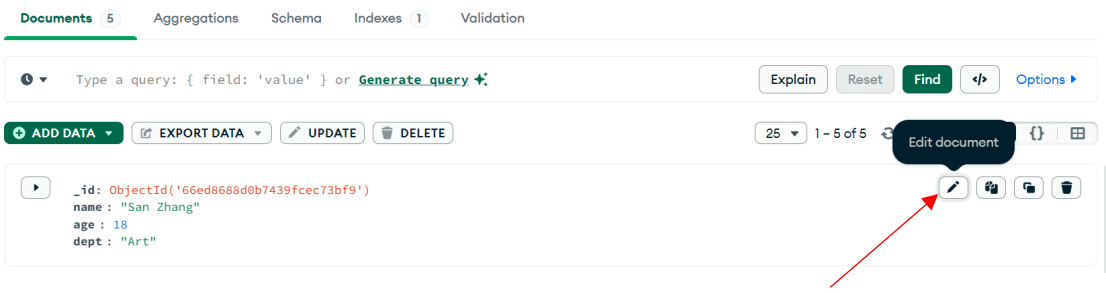




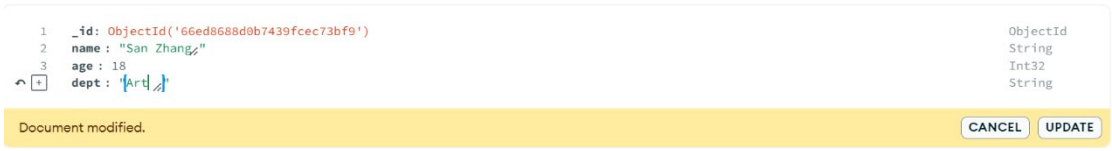


② 修改具体文件

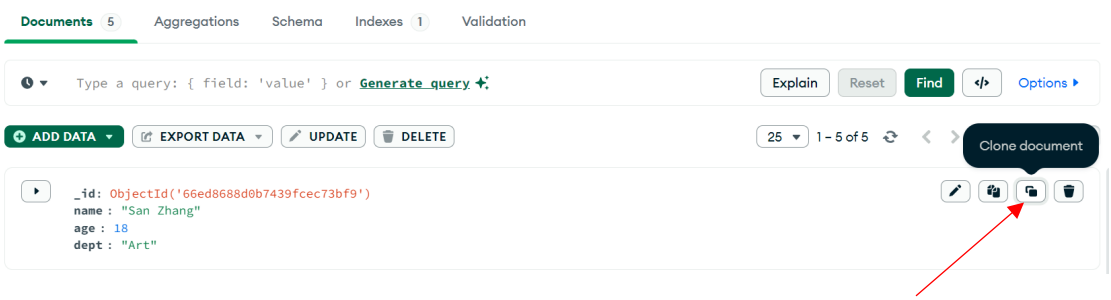
a. 修改文件



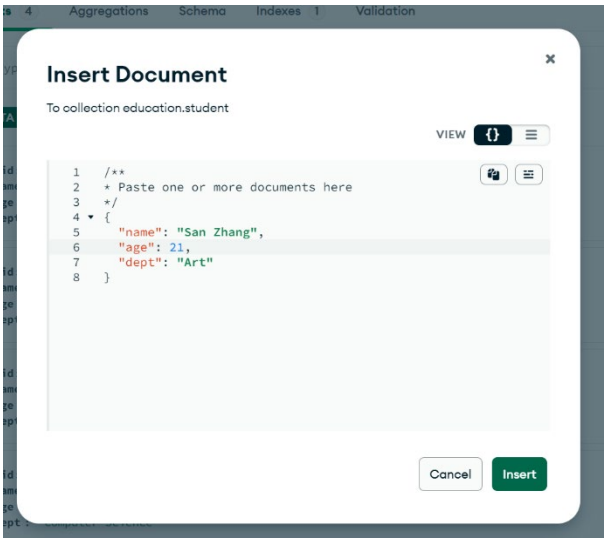
直接修改想要修改的数据即可。



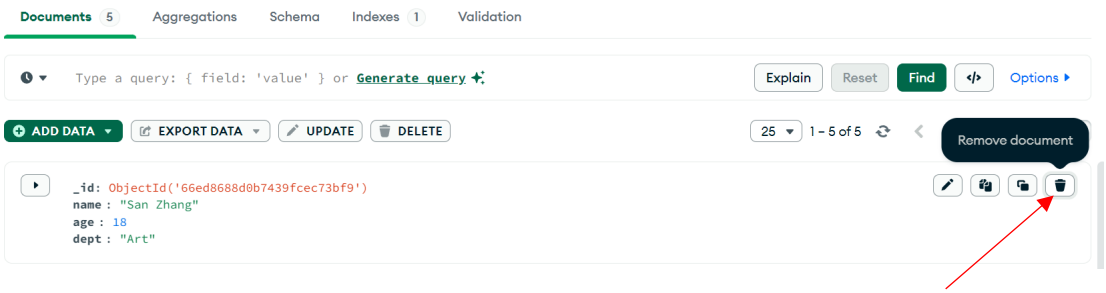
b. 克隆/插入文件



点击后可以复制此文件，也可以修改数据后相当于插入新文件。



c. 删除文件



点击 DELETE 即可。

二、 总结

1. 个人想法

(1)MongoDB 是一种文档型数据库，数据以文档的形式存储，每个文档可以包含不同的字段和值，这种数据模型比关系型数据库更加灵活。

(2)MongoDB 的查询语言丰富，还有官方可视化工具，帮助用户更直观地管理

和查询数据库。无需编写复杂的代码，适合不太熟悉 MongoDB 查询语言的用户。而且可以在 Compass 中查看和管理索引，了解索引的使用情况和性能，提高查询效率。

2. 遇到问题

(1)运行 mongosh 无法连接服务器

```
C:\Users\PC>mongosh
Current Mongosh Log ID: 66eeae8d30e99116c2c73bf7
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.1
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017
```

经检查原因是 mongod 服务器并未启动。在输入 mongod --dbpath d:\mongodb_data\db 启动 mongod 服务器后，就可以正常连接了。

(2)插入文档时遇到警告

```
education> db.student.insert({ name: 'San Zhang', age: 18, dept: 'Computer Science' })
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('66ed8615d0b7439fcec73bf8') }
}
education>
```

经查阅资料，insert()方法已经被弃用，需要加上 One/Many 来标出插入数量。

(3)再替换和删除文档时报错

```
education> db.students.replace(
...     { name: 'San Zhang' },
...     { name: 'Alice', age: 21, dept: 'Software Engineering' }
... );
TypeError: db.students.replace is not a function
education>
```

```
education> db.student.delete({ name: 'San Zhang' });
TypeError: db.student.delete is not a function
education>
```

在 MongoDB 中，replace()和 delete()方法并不存在，都需要加上 One/Many。