

## MongoDB

```
brew services start mongodb-community@8.0
brew services stop mongodb-community@8.0

//進入數據庫
mongosh
//顯示所有數據庫
show dbs
//顯示目前在那個數據庫
db
//使用其他數據庫， 如 itcast
use itcast
//顯示當前數據庫的集合 collection
show collections
//顯示集合的具體數據， 例如users
db.users.find()

exit
```

通过Nodejs使用第三方包Mongoose操作MongoDB数据库；

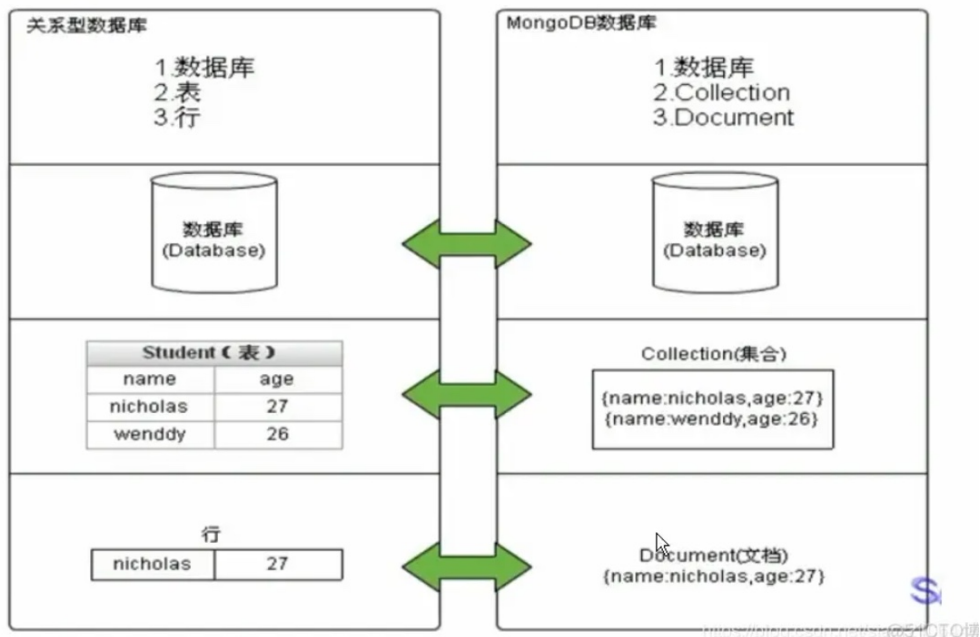
<https://mongoosejs.com/>

```
npm install mongoose --save
```

数据库结构：

最初都可以看做一个对象{}，对象里面有若干DB数据库{}，同样也是对象；数据库对象里面有若干collection集和[]集合是数组形式，集合里面有若干document文档{}，文档是对象形式；

MySQL和MongoDB对比



### 设计schema

```
//引用
import mongoose from 'mongoose';
//拿到mongoose的schema
const { Schema } = mongoose;

//连接数据库
mongoose.connect('mongodb://127.0.0.1:27017/myapp');

//设计schema 例如用户表
const userSchema = new Schema({
  username: {
    type: String,
    required: true
  },
  password: {
    type: String,
    required: true
  },
  email: {
    type: String,
    required: true
  }
});

// 将文档结构发布为模型：第一个参数是大写名词，表示数据库名称；第二个阐述是上面定义的schema名称；返回值
是 模型对象/模型构造函数；

const User = mongoose.model('User', userSchema);
```

### 增加数据

```
// 插入单个产品 用的是save () 方法
const insertSingleUser = async () => {
  const newUser = new User({
    username: 'admin',
    password: 1234,
    email: 'admin@admin.com'
  });

  try {
    const savedUser = await newUser.save();
  }
}
```

```
console.log('用户已保存:', savedUser);  
} catch (err) {  
  console.error('保存用户失败:', err);  
}  
};
```

## 查询/更改/删除

<https://mongoosejs.com/docs/queries.html>

查询中涉及比较查询运算符：

<https://www.mongodb.com/zh-cn/docs/manual/reference/operator/query-comparison/>

名称	说明
<code>\$eq</code>	匹配等于指定值的值。
<code>\$gt</code>	匹配大于指定值的值。
<code>\$gte</code>	匹配大于等于指定值的值。
<code>\$in</code>	匹配数组中指定的任何值。
<code>\$lt</code>	匹配小于指定值的值。
<code>\$lte</code>	匹配小于等于指定值的值。
<code>\$ne</code>	匹配所有不等于指定值的值。
<code>\$nin</code>	不匹配数组中指定的任何值。