

霍格沃兹测试学院 - 测试开发工程师的黄埔军校

MongoDB 数据库

MrDong



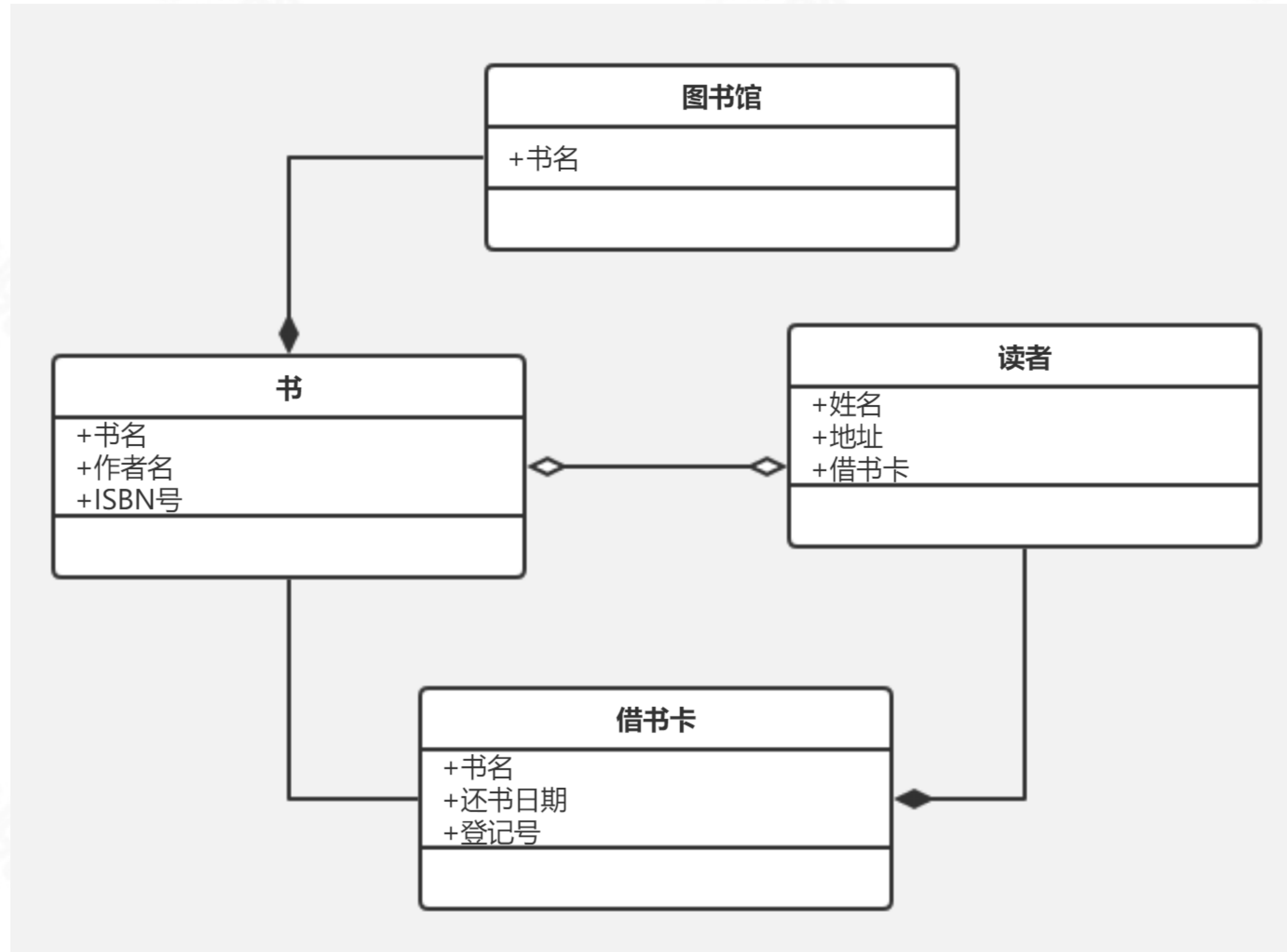
目录

- ❖ 关系与非关系
- ❖ MongoDB 简介
- ❖ 常用命令



关系型数据库





关系

- ❖ 书与读者 多对多
- ❖ 书与作者 一对多
- ❖ 书对出版日期 一对一



学生表 (Student)

S_No	S_Name	S_Sex	S_Age
201201	Edward	M	32
201202	Catty	F	23

课程表 (Course)

C_No	C_Name	C_Credit
1001	DataBase	3
1002	DataStructure	2

选课表 (SC)

SC_Sno	SC_Cno	SC_Score
201201	1001	98
201201	1002	95
201202	1001	95
201202	1002	86



关系型数据库

- ❖ MySql, Oracle, SqlServer, Access...
- ❖ SQL (结构化查询语言)



存在哪些问题

- ❖ 存储在硬盘上 读写慢
- ❖ 保持 ACID 难扩展



非关系型数据库



非关系型数据优点

- ❖ NoSql(Not Only SQL)
- ❖ key : value
- ❖ CAP - > BASE
 - ❖ 一致性 (Consistency) (所有节点在同一时间具有相同的数据)
 - ❖ 可用性 (Availability) (保证每个请求不管成功或者失败都有响应)
 - ❖ 分隔容忍 (Partition tolerance) (系统中任意信息的丢失或失败不会影响系统的继续运作)



非关系型数据缺点

- ❖ 没有标准化
- ❖ 有限的查询功能（到目前为止）



```
“_id”:1, “S_No” : 201201, “S_Name” : “Edward”, “S_Sex”: ‘M’,  
“S_Age”: 32
```

```
“_id”:2, “S_Name”: “Catty”, “S_No”: 201202, “S_Sex”: ‘F’, “S_Age”:  
23
```

“_id”在集合内唯一标识一行

```
“_id”:1, “C_No”:1001, “C_Name”:“DataBase”, “C_Credit”:3
```

```
“_id”:2, “C_Name”:“DataStructure”, “C_No”:1002, “C_Credit”:2
```

```
“_id”:1, “SC_Sno”:201201, “SC_Cno”:1001, “SC_Score”:98
```

```
“_id”:2, “SC_Sno”:201201, “SC_Cno”:1002, “SC_Score”:95
```

```
“_id”:3, “SC_Sno”:201202, “SC_Cno”:1001, “SC_Score”:95
```

```
“_id”:4, “SC_Sno”:201202, “SC_Cno”:1002, “SC_Score”:86
```



id	user_name	email	age	city
1	Mark Hanks	mark@abc.com	25	Los Angeles
2	Richard Peter	richard@abc.com	31	Dallas



```
{
  "_id": ObjectId("5146bb52d8524270060001f3"),
  "age": 25,
  "city": "Los Angeles",
  "email": "mark@abc.com",
  "user_name": "Mark Hanks"
}
{
  "_id": ObjectId("5146bb52d8524270060001f2"),
  "age": 31,
  "city": "Dallas",
  "email": "richard@abc.com",
  "user_name": "Richard Peter"
}
```



MongoDB

- ❖ 非关系型数据库 nosql
- ❖ 文档存储：类 json 格式
- ❖ 有机会对字段建立索引，实现关系数据库



下载与安装

- ❖ <https://www.mongodb.com/download-center/community>



Select the server you would like to run:

MongoDB Community Server

FEATURE RICH. DEVELOPER READY.

Version

4.2.3 (current release)

Package

TGZ

Download

https://fastdl.mongodb.org/osx/mongodb-macos-x86_64-4.2.3.tgz

Select OS
Ubuntu 16.04 Linux x64
Windows x64
RHEL 7.2 Linux IBM Z Series
macOS x64
RHEL 7.0 Linux x64
RHEL 6.2 Linux x64
Amazon Linux x64
Amazon Linux 2 x64
Debian 9 Linux x64
SUSE 12 Linux x64
Ubuntu 18.04 Linux IBM Z Series
SUSE 12 Linux IBM Z Series
RHEL 6.7 Linux IBM Z Series
Ubuntu 18.04 Linux x64
SUSE 15 Linux x64
Ubuntu 18.04 Linux ARM 64
Debian 10 Linux x64
RHEL 8.0 Linux x64

MongoDB Enterprise Server

ADVANCED FEATURES. PERFORMANCE GRADE.

- Release notes
- Changelog
- All version binaries
- Installation instructions
- Download source (tgz)
- Download source (zip)

MongoDB Atlas is a fully-managed and fully-automated global cloud database service available on AWS, Azure, and Google Cloud.



MongoDB 操作



配置数据库目录

- ❖ 运行 MongoDB 服务器
 - ❖ `mongod -dbpath xxx`
- ❖ 连接 MongoDB
 - ❖ `mongo`



删除数据库

- ❖ 查看当前数据库
 - ❖ show dbs
- ❖ 切换数据库（如果没有会自动创建）
 - ❖ use demo1
- ❖ 往数据库中插入一条数据
 - ❖ `db.demo1.insert({"name": "小红"})`



删除数据库

- ❖ 切换数据库
 - ❖ use demo1
- ❖ 删除数据库
 - ❖ db.dropDatabase()



集合



什么是集合

- ❖ 集合相当于一张表
- ❖ 集合可以有多个文档



创建 / 查看 / 删除集合

- ❖ 创建集合
 - ❖ `db.createCollection('collection1')`
- ❖ 查看已经创建的集合
 - ❖ `show collections`
- ❖ 删除集合
 - ❖ `db.collection1.drop()`



插入和查看文档



插入文档

- ❖ 文档是 mongodb 中的基本单元
- ❖ 相当于表中的一条记录
- ❖ 以 json 格式显示



插入 / 查看文档

- ❖ 插入文档时，MongoDB 自动创建集合
 - ❖ `db.collectionDemo.insert({"name": "hello"})`
- ❖ 查看插入的内容
 - ❖ `db.collectionDemo.find().pretty()`



插入复杂文档

- ❖ 声明变量，插入文档
- ❖ `document=({`
- ❖ `title: 'MongoDB',`
- ❖ `by: 'RuotongYu',`
- ❖ `tags: ['mongodb', 'database', 'NoSQL'],`
- ❖ `});`
- ❖ `db.collectionDemo.insert(document)`



操作	格式	范例	RDBMS 中的类似语句
等于	{<key>:<value>}	db.col.find({"by":"2"}).pretty()	where by = '2'
小于	{<key>: {\$lt:<value>}}	db.col.find({"likes": {\$lt:50}}).pretty()	where likes < 50
小于或等于	{<key>: {\$lte:<value>}}	db.col.find({"likes": {\$lte:50}}).pretty()	where likes <= 50
大于	{<key>: {\$gt:<value>}}	db.col.find({"likes": {\$gt:50}}).pretty()	where likes > 50
大于或等于	{<key>: {\$gte:<value>}}	db.col.find({"likes": {\$gte:50}}).pretty()	where likes >= 50
不等于	{<key>: {\$ne:<value>}}	db.col.find({"likes": {\$ne:50}}).pretty()	where likes != 50



修改和删除文档



格式

- ❖ db.collection.update(
 - ❖ <query>,
 - ❖ <update>,
 - ❖ {
 - ❖ upsert: <boolean>,
 - ❖ multi: <boolean>,
 - ❖ writeConcern: <document>
 - ❖ }
- ❖)



可选选项

- ❖ `upsert` : 如果不存在 `update` 的记录, 是否插入, 默认 `false`
- ❖ `multi` : 把按条件查出来多条记录全部更新, 默认 `false`
- ❖ `writeConcern` : 抛出异常的级别。



注意

- ❖ update 是替换，没有修改的字段会消失！



修改文档

- ❖ 修改文档
 - ❖ `db.col.update({'title':'MongoDB'},`
 - ❖ `{'title':'MongoDB123'})`
- ❖ 修改多个文档
 - ❖ `db.col.update({'name':'hello'},`
 - ❖ `{$set: {'name':'hello123'}},`
 - ❖ `{multi:true})`



删除多个文档

- ❖ 删除多个文档
 - ❖ `db.col.remove({'name':'hello123'})`
- ❖ 删除一个文档
 - ❖ `db.col.remove({'name':'hello'},1)`



修改操作符



\$inc

- ❖ 用法：`{$inc:{field:value}}`
- ❖ 作用：对一个数字字段的某个 field 增加 value
- ❖ 示例：将 name 为 xiaohong 的学生的 age 增加 5
 - ❖ `db.students.update({name:"xiaohong"},{$inc:{age:5}})`



\$set

- ❖ 用法：`{$set:{field:value}}`
- ❖ 作用：把文档中某个字段 `field` 的值设为 `value`
- ❖ 示例：把 `xiaohong` 的年龄设为 10 岁
 - ❖ `db.students.update({name:"xiaohong"},{$set:{age:10}})`



\$unset

- ❖ 用法：`{$unset:{field:1}}`
- ❖ 作用：删除某个字段 field
- ❖ 示例：将 chenzhou 的年龄字段删除
 - ❖ `db.students.update({name:"chenzhou"},{$unset:{age:1}})`



\$push

- ❖ 用法：`{ $push: { field: value } }`
- ❖ 作用：把 value 追加到 field 里。注：field 只能是数组类型，如果 field 不存在，会自动插入一个数组类型
- ❖ 示例：给 chenzhou 添加别名 "michael"
 - ❖ `db.students.update({name:"chenzhou"},{$push:{"ailas":"Michael"}})`



\$rename

- ❖ 用法：`{ $rename: { old_field_name: new_field_name } }`
- ❖ 作用：对字段进行重命名
- ❖ 示例：把 xiaohong 记录的 name 字段重命名为 hello
 - ❖ `db.students.update({ name: "xiaohong" }, { $rename: { "name": "name2" } })`

