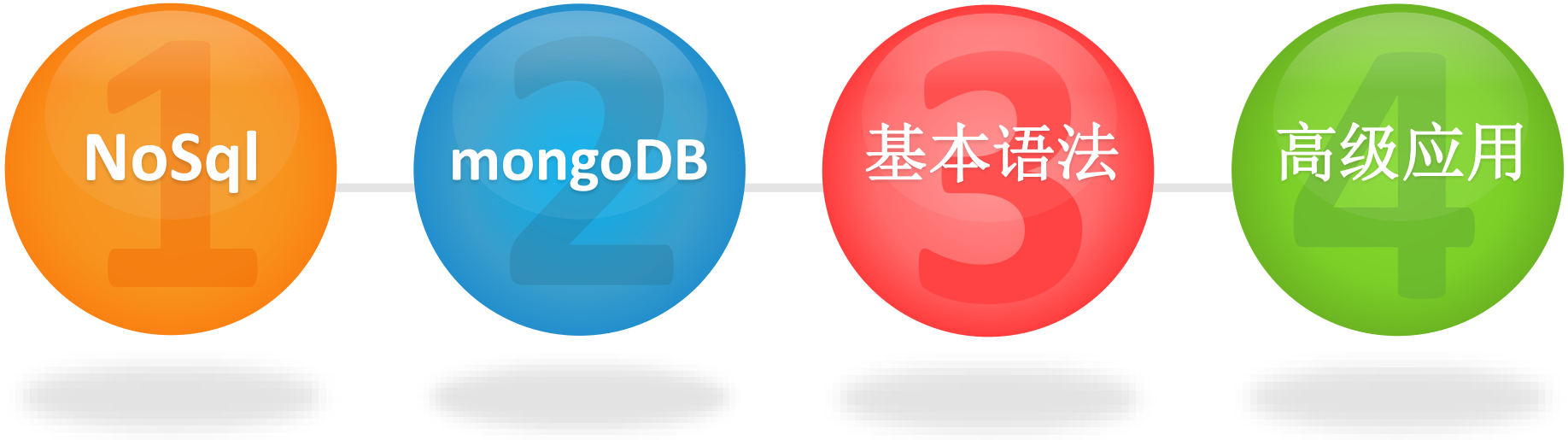
mongoDB 介绍


1
NoSql
2
mongoDB
3
基本语法
4
高级应用

什么是NoSQL？

什么是NoSQL

- NoSQL(NoSQL = Not Only SQL)是一种与关系型数据库管理系统截然不同的数据库管理系统，它的数据存储格式可以是松散的，通常不支持Join操作

为什么使用NoSQL

三高

- **High performance**
  对数据库高并发读写的需求
- **Huge Storage**
  对海量数据的高效率存储和访问的需求
- **High Scalability && High Availability**
  对数据库的高可扩展性和高可用性的需求

NoSQL解决的问题和存在的不足

解决的问题

- 对数据库的高并发读写
- 对海量数据的高效率存储和访问
- 数据库的可扩展性

存在的不足

- 绝大多数NoSQL只能提供简单的查询，无法进行多表联合查询等复杂的查询操作
- 功能相对贫乏，在一些要求事务一致性较高、业务逻辑比较复杂或者一些需要复杂分析查询的环境中，NoSQL还难以担当重任

NoSQL家族的数据库

mongoDB

mongoDB简介

- **mongoDB**是一个高性能，开源，无模式的文档型数据库，官方给自己的定义是Key-value存储(高性能和高扩展)和传统RDBMS(丰富的查询和功能)之间的一座桥梁。使用C++开发

- MongoDB is a scalable, high-performance, open source NoSQL database. Written in C++

mongoDB支持的语言

mongoDB数据类型

- null
- 布尔　　ture | false
- 整数　　123
- 浮点　　12.3
- 字符串　　“hello world”
- 对象ID　　用 new ObjectId（）来申明
- 日期　　用 new Date（）来申明
- 时间戳
- 数组　　[“apple” ,” blanan” ,” pear”]
- 内嵌文档　{ “username” : “jone” , “age” : 13, “contact” : { “home” :” 123” ,” moblie” :” 456” }}
- RegExp　　正则表达式 /[a-z]/

mongoDB简介

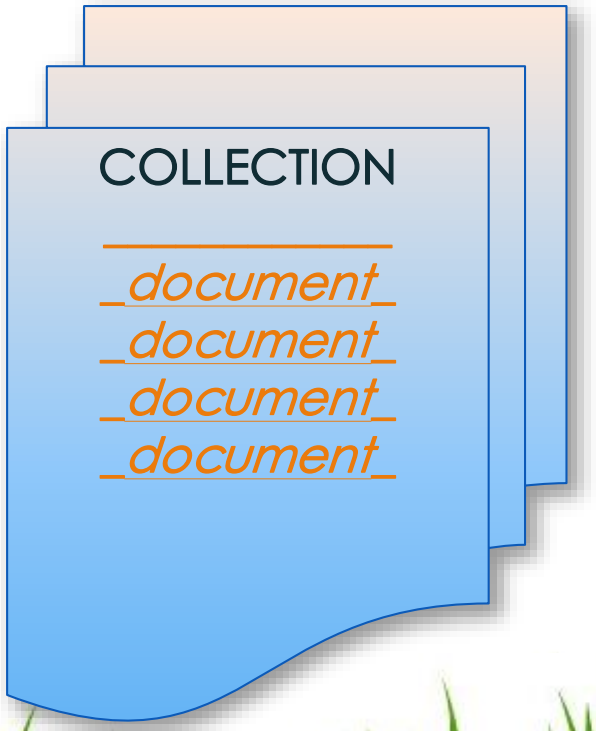
- mongoDB的最小存储单位就是文档对象，对应于关系型数据库的行，数据在mongo中以BSON（Binary-JSON）文档的格式存储在磁盘上。每一个文档对象， mongo都会为它分配一个唯一的id号，名为“_id”。

```
{
  _id      :    ObjectId( “xxxxxxxx" ),
  name     :    "zzm",
  gender   :    "M",
  address  :    "Beijing"
}
```

_id：507fbbb2a7b49259f1bf274c

mongoDB存储结构

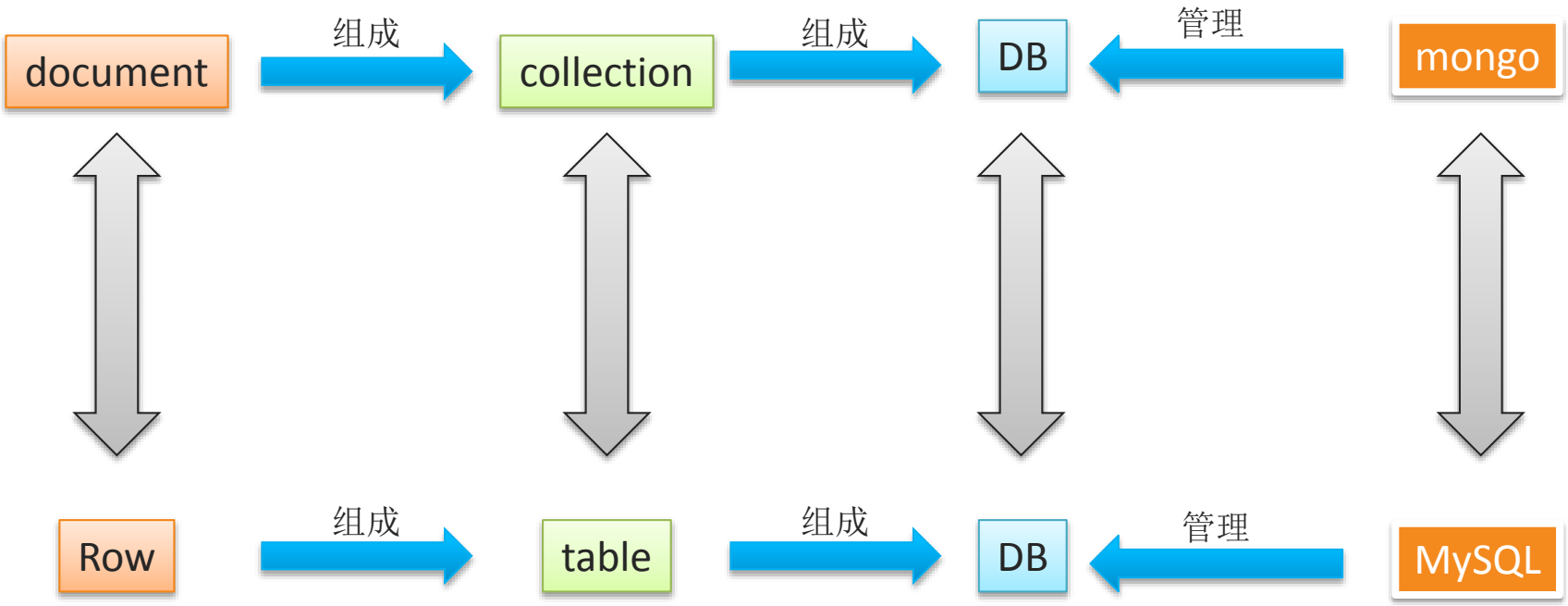
- Collection是documents的集合，可以理解为关系型数据库中的表，也可以看成一个文件夹，用来专门储存同一类文档。



mongoDB存储结构

- mongoDB的最外层结构，和关系型数据库一样，是存放多个Collections的容器。
- 可以把mongoDB看成一个文件储藏柜，每个document就如同一页纸；成千上万张纸被存放在文件夹里，这些文件夹就可以看做是Collection；多个文件夹存放在一个储藏柜里，也就是Database

mongoDB存储结构



基本语法

mongoDB基本语法

- List database
  - Show dbs || show databases
- List collections
  - Show collections || show tables
- Create database
  - Use test
- Create collection
  - Db.user_list.insert({name: “fish”})
- Remove database
  - Db.dropDatabase()
- Remove collection
  - Db.user_list.drop()

Db.help() **db.collections.help()**

mongoDB基本语法

SQL INSERT Statements	MongoDB insert() Statements

`INSERT INTO users(user_id,`
`age,`
`status)`
`VALUES ("bcd001",`
`45,`
`"A")`
	`db.users.insert(`
	`{ user_id: "bcd001", age: 45, status: "A" }`
	`)`

mongoDB基本语法

SQL SELECT Statements	MongoDB find() Statements
`SELECT *` `FROM users`	`db.users.find()`

`SELECT id,`
`user_id,`
`status`
`FROM users`
	`db.users.find(`
	`{ },`
	`{ user_id: 1, status: 1 }`
	`)`

`SELECT user_id, status`
`FROM users`
	`db.users.find(`
	`{ },`
	`{ user_id: 1, status: 1, _id: 0 }`
	`)`

`SELECT *`
`FROM users`
`WHERE status = "A"`
	`db.users.find(`
	`{ status: "A" }`
	`)`

mongoDB基本语法

```
SELECT *
FROM users
WHERE status != "A"
```

```
db.users.find(
    { status: { $ne: "A" } }
)
```

```
SELECT *
FROM users
WHERE status = "A"
AND age = 50
```

```
db.users.find(
    { status: "A",
      age: 50 }
)
```

```
SELECT *
FROM users
WHERE status = "A"
OR age = 50
```

```
db.users.find(
    { $or: [ { status: "A" } ,
             { age: 50 } ] }
)
```

```
SELECT *
FROM users
WHERE age > 25
```

```
db.users.find(
    { age: { $gt: 25 } }
)
```

```
SELECT *
FROM users
WHERE age < 25
```

```
db.users.find(
   { age: { $lt: 25 } }
)
```

mongoDB基本语法

SQL Update Statements	MongoDB update() Statements

`UPDATE users`
`SET status = "C"`
`WHERE age > 25`
	`db.users.update(`
	`{ age: { $gt: 25 } },`
	`{ $set: { status: "C" } },`
	`{ multi: true }`
	`)`

`UPDATE users`
`SET age = age + 3`
`WHERE status = "A"`
	`db.users.update(`
	`{ status: "A" } ,`
	`{ $inc: { age: 3 } },`
	`{ multi: true }`
	`)`

mongoDB基本语法

SQL Delete Statements	MongoDB remove() Statements
`DELETE FROM users` `WHERE status = "D"`	`db.users.remove( { status: "D" } )`
`DELETE FROM users`	`db.users.remove({})`

mongoDB简介

- db.users.insert({"name":"zzm","pass":"123456"})
  - insert into users(`name`,`pass`) values (‘zzm’,’123456’);
- db.users.find();
  - Select * from users;
- db.users.find().skip(20).limit(10);
  - Select * from users limit 20,10;
- db.users.find({name:{$nin: [“a”,”b”]}});
  - Select * from users where name is not in (“a”,”b”);
- db.users.find({name:"zzm",age:{$gte:15}})
  - Select * from users where name=“zzm” and age>=15;

mongoDB简介

- db.users.find().sort({age:1})
  - select * from users order by age asc;
- db.users.remove()
  - delete from users;
- db.users.remove({“name”:”zzm”});
  - delete from users where `name`=“zzm”;
- db.users.update({name:"zzm"},{$set:{pass:12345}})
  - update users set `pass`=“12345” where `name`=“zzm”;

高级应用

高级应用

- 索引
- 同步复制
- 分布式
- 备份/还原

Thank You!