



Redis快速入门

Redis的常见命令和客户端使用



目录

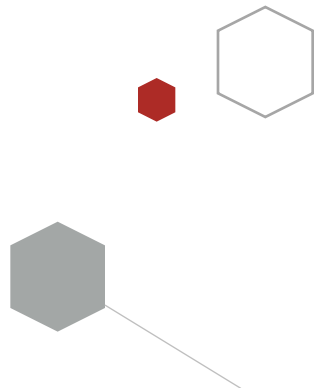
Contents

- 初识Redis
 - 认识NoSQL
 - 认识Redis
 - 安装Redis
- Redis常见命令
 - 5种常见数据结构
 - 通用命令
 - 不同数据结构的操作命令
- Redis的Java客户端
 - Jedis客户端
 - SpringDataRedis客户端

学习目标

Learning Objectives

1. 知道NoSQL与SQL的差别
2. 熟悉Redis的常用5种数据结构
3. 熟悉Redis的常用命令
4. 熟练使用Jedis或SpringDataRedis



初识Redis



目录

Contents

- ◆ 认识NoSQL
- ◆ 认识Redis
- ◆ 安装Redis

多一句没有，少一句不行，用最短时间，教会最实用的技术！

认识NoSQL



多一句没有，少一句不行，用最短时间，教会最实用的技术！

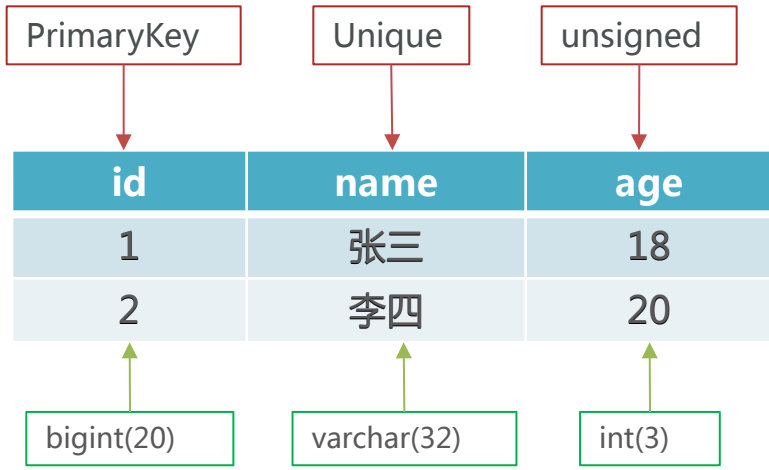
认识NoSQL

SQL



NoSQL

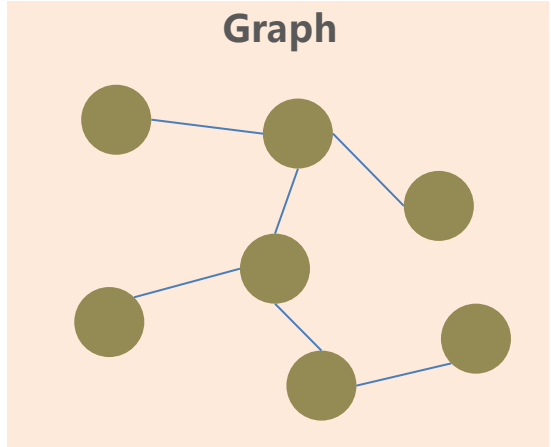
#1 结构化(Structured)



Key	Value
id	
name	
age	

非结构化 #1

```
Document
{
  id:
  name: " ",
  age:
}
```



多一句没有，少一句不行，用最短时间，教会最实用的技术！

认识NoSQL

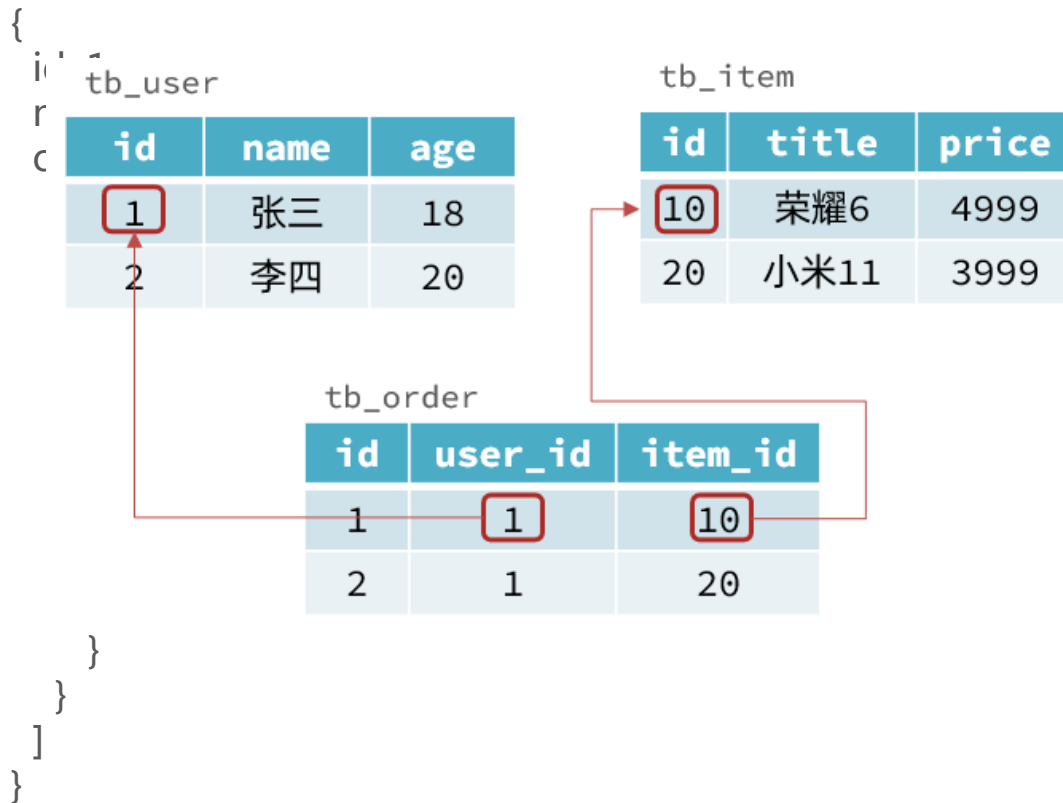
SQL



NoSQL

#1 结构化(Structured)

#2 关联的(Relational)



非结构化 #1

无关联的 #2

多一句没有，少一句不行，用最短时间，教会最实用的技术！

认识NoSQL

SQL



NoSQL

#1 结构化(Structured)

非结构化 #1

#2 关联的(Relational)

无关联的 #2

#3 SQL查询

非SQL #3

SQL `SELECT id, name age FROM tb_user WHERE id = 1`

Redis `get user:1`

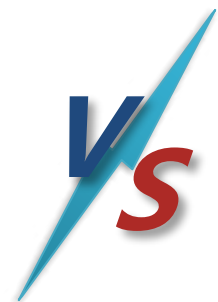
MongoDB `db.users.find({_id: 1})`

elasticsearch `GET http://localhost:9200/users/1`

多一句没有，少一句不行，用最短时间，教会最实用的技术！

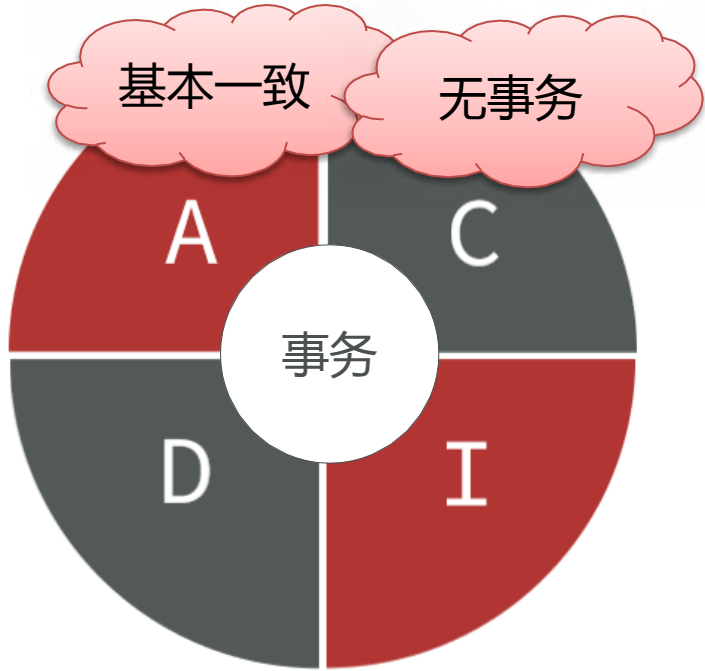
认识NoSQL

SQL



NoSQL

- #1 结构化(Structured)
- #2 关联的(Relational)
- #3 SQL查询
- #4 ACID



- 非结构化 #1
- 无关联的 #2
- 非SQL #3
- BASE #4

多一句没有，少一句不行，用最短时间，教会最实用的技术！

认识NoSQL

	SQL	NoSQL
数据结构	结构化(Structural)	
数据关联	关联的(Relational)	
查询方式	SQL查询	
事务特性	ACID	BASE

#1 键值类型 (Redis)

#2 文档类型
(MongoDB)

#3 列类型 (HBase)

#4 Graph类型 (Neo4j)



目录

Contents

- ◆ 认识NoSQL
- ◆ 认识Redis
- ◆ 安装Redis

多一句没有，少一句不行，用最短时间，教会最实用的技术！

认识Redis

Redis诞生于2009年全称是**Remote Dictionary Server**，远程词典服务器，是一个基于内存的键值型NoSQL数据库。

特征：

- 键值（key-value）型，value支持多种不同数据结构，功能丰富
- 单线程，每个命令具备原子性
- 低延迟，速度快（基于内存、IO多路复用、良好的编码）。
- 支持数据持久化
- 支持主从集群、分片集群
- 支持多语言客户端



SALVATORE SANFILIPPO

AKA **ANTIREZ**



目录

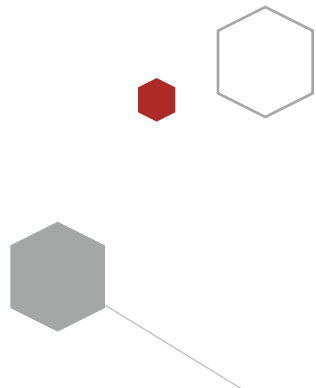
Contents

- ◆ 认识NoSQL
- ◆ 认识Redis
- ◆ 安装Redis

多一句没有，少一句不行，用最短时间，教会最实用的技术！

安装Redis

参考课前资料《Redis安装说明》



Redis常见命令



目录

Contents

- ◆ Redis数据结构介绍
- ◆ Redis通用命令
- ◆ String类型
- ◆ Hash类型
- ◆ List类型
- ◆ Set类型
- ◆ SortedSet类型

多一句没有，少一句不行，用最短时间，教会最实用的技术！

Redis数据结构介绍

Redis是一个key-value的数据库，key一般是String类型，不过value的类型多种多样：

String	hello world	基本类型
Hash	{name: "Jack", age: 21}	
List	[A -> B -> C -> C]	
Set	{A, B, C}	
SortedSet	{A: 1, B: 2, C: 3}	
GEO	{A: (120.3 , 30.5) }	特殊类型
BitMap	0110110101110101011	
HyperLog	0110110101110101011	

多一句没有，少一句不行，用最短时间，教会最实用的技术！

Redis数据结构介绍

Redis为了方便我们学习，将操作不同数据类型的命令也做了分组，在官网（<https://redis.io/commands>）可以看到不同的命令：

```
127.0.0.1:6379> help
redis-cli 6.2.6
To get help about Redis commands type:
"help @<group>" to get a list of commands in <group>
"help <command>" for help on <command>
"help <tab>" to get a list of possible help topics
"quit" to exit

To set redis-cli preferences:
":set hints" enable online hints
":set nohints" disable online hints
Set your preferences in ~/.rediscliirc
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> help @generic
```

查看通用的命令



目录

Contents

- ◆ Redis数据结构介绍
- ◆ Redis通用命令
- ◆ String类型
- ◆ Hash类型
- ◆ List类型
- ◆ Set类型
- ◆ SortedSet类型

Redis通用命令

通用指令是部分数据类型的，都可以使用的指令，常见的有：

- KEYS：查看符合模板的所有key，**不建议在生产环境设备上使用**
- DEL：删除一个指定的key
- EXISTS：判断key是否存在
- EXPIRE：给一个key设置有效期，有效期到期时该key会被自动删除
- TTL：查看一个KEY的剩余有效期

通过help [command] 可以查看一个命令的具体用法，例如：

```
127.0.0.1:6379> help keys

KEYS pattern
summary: Find all keys matching the given pattern
since: 1.0.0
group: generic
```



目录

Contents

- ◆ Redis数据结构介绍
- ◆ Redis通用命令
- ◆ **String类型**
- ◆ Hash类型
- ◆ List类型
- ◆ Set类型
- ◆ SortedSet类型

多一句没有，少一句不行，用最短时间，教会最实用的技术！

String类型

String类型，也就是字符串类型，是Redis中最简单的存储类型。

其value是字符串，不过根据字符串的格式不同，又可以分为3类：

- string：普通字符串
- int：整数类型，可以做自增、自减操作
- float：浮点类型，可以做自增、自减操作

不管是哪种格式，底层都是字节数组形式存储，只不过是编码方式不同。字符串类型的最大空间不能超过512m.

KEY	VALUE

String类型的常见命令

String的常见命令有：

- SET：添加或者修改已经存在的一个String类型的键值对
- GET：根据key获取String类型的value
- MSET：批量添加多个String类型的键值对
- MGET：根据多个key获取多个String类型的value
- INCR：让一个整型的key自增1
- INCRBY:让一个整型的key自增并指定步长，例如：`incrby num 2` 让num值自增2
- INCRBYFLOAT：让一个浮点类型的数字自增并指定步长
- SETNX：添加一个String类型的键值对，前提是这个key不存在，否则不执行
- SETEX：添加一个String类型的键值对，并且指定有效期

多一句没有，少一句不行，用最短时间，教会最实用的技术！



思考

Redis没有类似MySQL中的Table的概念，我们该如何区分不同类型的key呢？

- 例如，需要存储用户、商品信息到redis，有一个用户id是1，有一个商品id恰好也是1

多一句没有，少一句不行，用最短时间，教会最实用的技术！

key的结构

- Redis的key允许有多个单词形成层级结构，多个单词之间用':'隔开，格式如下：
项目名:业务名:类型:id

- 这个格式并非固定，也可以根据自己的需求来删除或添加词条。

KEY	VALUE
heima:user:1	{"id":1, "name": "Jack", "age": 21}
heima:product:1	{"id":1, "name": "小米11", "price": 4999}

有user和
我们可以

product内容不同产生的数据，我们可以

这样它就以key

多一句没有，少一句不行，用最短时间，教会最实用的技术！



总结

String类型的三种格式：

- 字符串
- int
- float

Redis的key的格式：

- [项目名]:[业务名]:[类型]:[id]



目录

Contents

- ◆ Redis数据结构介绍
- ◆ Redis通用命令
- ◆ String类型
- ◆ **Hash类型**
- ◆ List类型
- ◆ Set类型
- ◆ SortedSet类型

多一句没有，少一句不行，用最短时间，教会最实用的技术！

Hash类型

Hash类型，也叫散列，其value是一个无序字典，类似于Java中的HashMap结构。

String结构是将对象序列化为JSON字符串后存储，当需要修改对象某个字段时很不方便：

KEY	VALUE
heima:user:1	{name:"Jack", age:21}
heima:user:2	{name:"Rose", age:18}

Hash结构可以将对象中的每个字段独立存储，可以针对单个字段做CRUD：

KEY	VALUE	
	field	value
heima:user:1	name	Jack
	age	21
heima:user:2	name	Rose
	age	18

Hash类型的常见命令

Hash的常见命令有：

- HSET key field value：添加或者修改hash类型key的field的值
- HGET key field：获取一个hash类型key的field的值
- HMSET：批量添加多个hash类型key的field的值
- HMGET：批量获取多个hash类型key的field的值
- HGETALL：获取一个hash类型的key中的所有的field和value
- HKEYS：获取一个hash类型的key中的所有的field
- HVALS：获取一个hash类型的key中的所有的value
- HINCRBY:让一个hash类型key的字段值自增并指定步长
- HSETNX：添加一个hash类型的key的field值，前提是这个field不存在，否则不执行



目录

Contents

- ◆ Redis数据结构介绍
- ◆ Redis通用命令
- ◆ String类型
- ◆ Hash类型
- ◆ List类型
- ◆ Set类型
- ◆ SortedSet类型

List类型

Redis中的List类型与Java中的LinkedList类似，可以看做是一个双向链表结构。既可以支持正向检索和也可以支持反向检索。

特征也与LinkedList类似：

- 有序
- 元素可以重复
- 插入和删除快
- 查询速度一般

常用来存储一个有序数据，例如：朋友圈点赞列表，评论列表等。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/737025001012006104>