

## CRUD 2.1

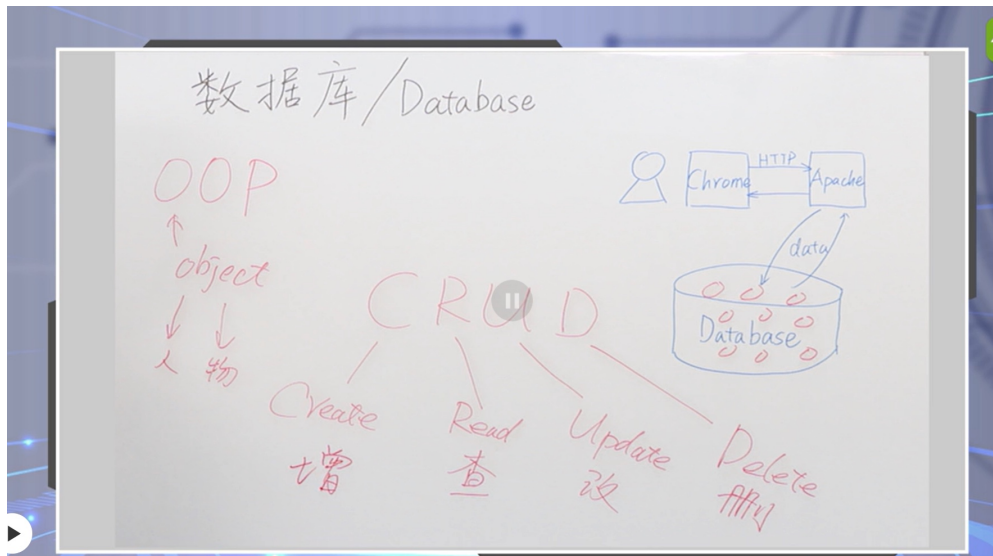
**Create** 把数据放入一个数据库系统里 增

**Read** 从数据库读取数据 查

**Update** 对数据库里的数据进行修改 改

**Delete** 删除不需要的数据 删

增删改查



描述对象的结构—数据模型（文档or表格）决定了数据是怎么被存放在数据库里的，数据模型不同，数据库的功能和使用方式就不同。

数据模型有 关系数据库系统 文档数据库系统 还有其他

使用不同的数据模型来描述数据的

## 文档数据库的数据模型or数据组织方式

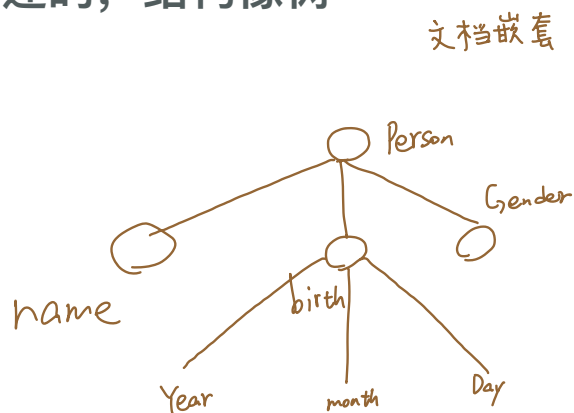
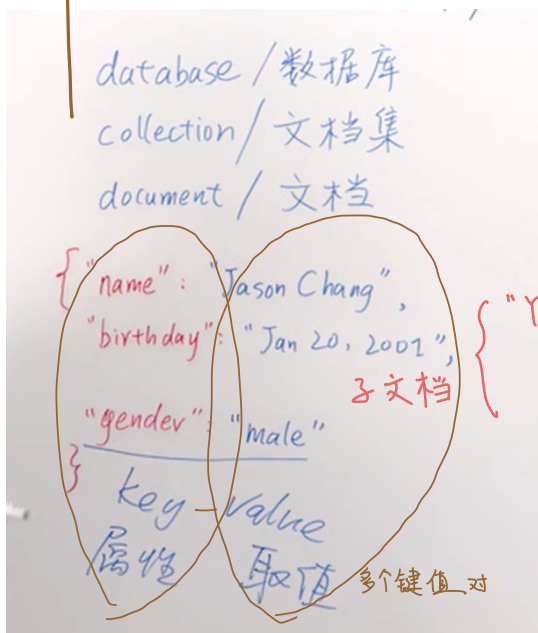
对于文档数据库而言，对象是以文档的形式被描述

数据库 多个文档集是给同一个应用使用的，放在一起成为数据库

文档集 所有的文档整理起来，同一类型的文档放在一起成为文档集

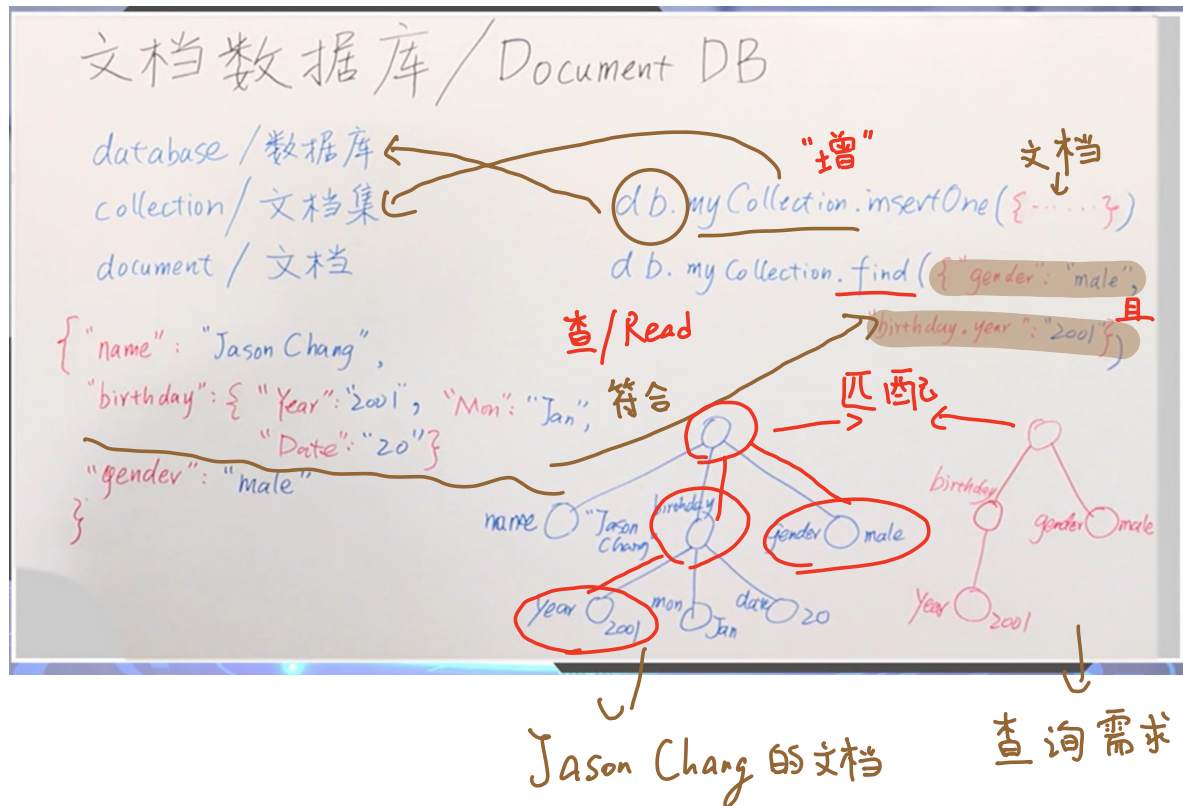
文档 每一个对象在文档数据库中被描述成一个文档

每一个文档是通过键值对来描述的，结构像树



{ "Year": "2001", "Mon": "Jan",  
"Day": "20" }

## 文档数据库的功能



右边的树是左边的树的一部份

通过文档匹配的方式，用户可以描述自己对某一个文档的需求  
(某个属性的取值是什么或者在什么范围内)

库 / Document DB

增 / Create 文档  
 ↓  
`db.myCollection.insertOne({...})`

查 / Read  
`db.myCollection.find({ "gender": "male",  
 "birthday.year": "2001" })`

改 / update  
`db.myCollection.updateOne(  
 { "name": "Jason Chang" },  
 { $set: { "birthday.year": "2002" } })`  
 目标, 起查询作用  
 改动指令

删 / Delete  
`db.myCollection.deleteOne(  
 { "name": "Jason Chang" })`  
 删除与之匹配的一个或多个

怎样唯一的识别一个文档

文档数据库 / Document DB

database / 数据库  
 collection / 文档集  
 document / 文档

增 / Create 文档  
 ↓  
`db.myCollection.insertOne({...})`

查 / Read  
`db.myCollection.find({ "gender": "male",  
 "birthday.year": "2001" })`

改 / Update  
`db.myCollection.updateOne(  
 { "name": "Jason Chang" },  
 { $set: { "birthday.year": "2002" } })`  
 目标  
 改动指令

删 / Delete  
`db.myCollection.deleteOne(  
 { "name": "Jason Chang" })`

示例文档结构:

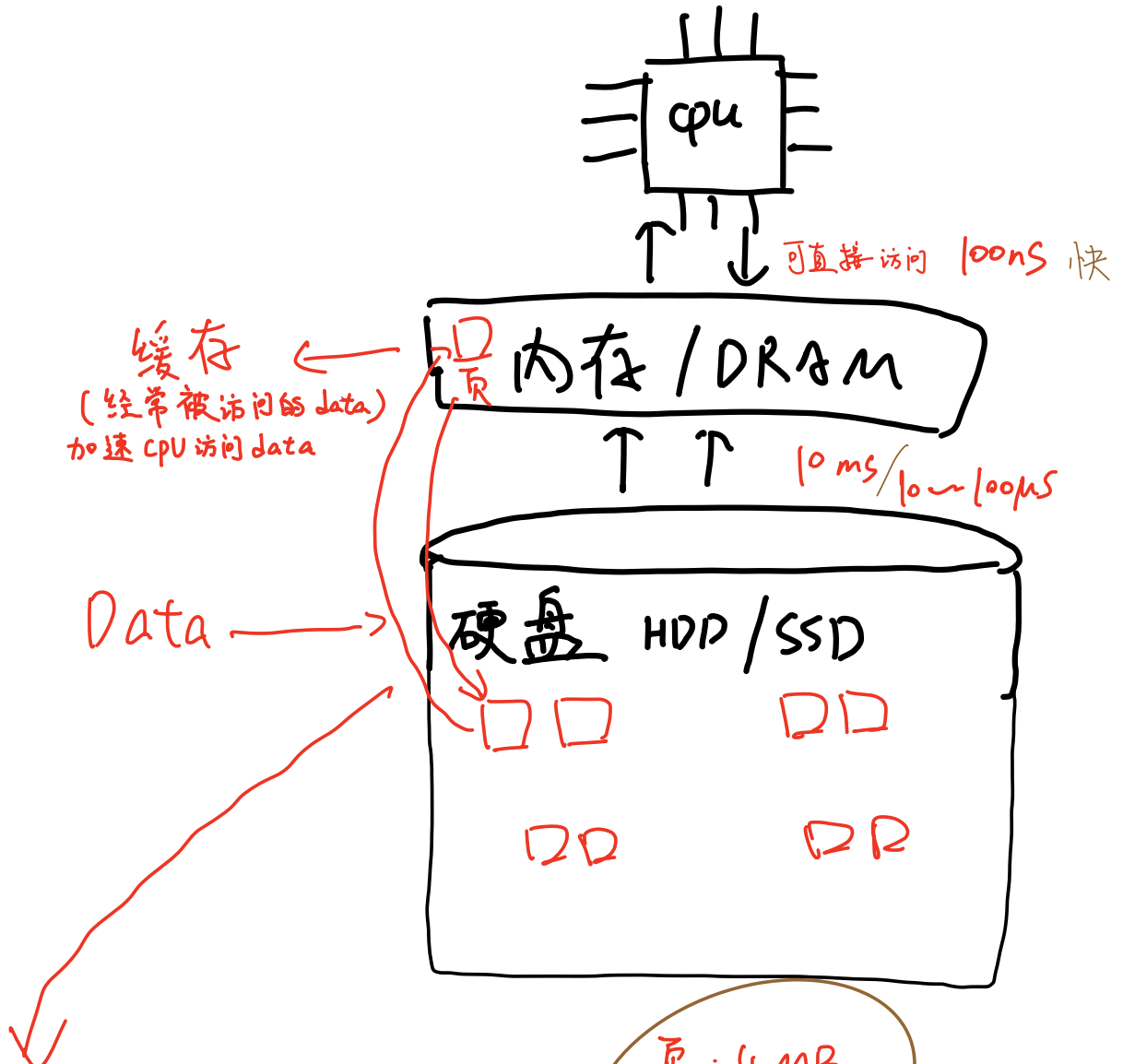
```
{
  "name": "Jason Chang",
  "birthday": {
    "Year": "2001",
    "Month": "Jan",
    "Date": "20"
  },
  "gender": "male",
  "Id": "ObjectID('9876543210')",
  ...
}
```

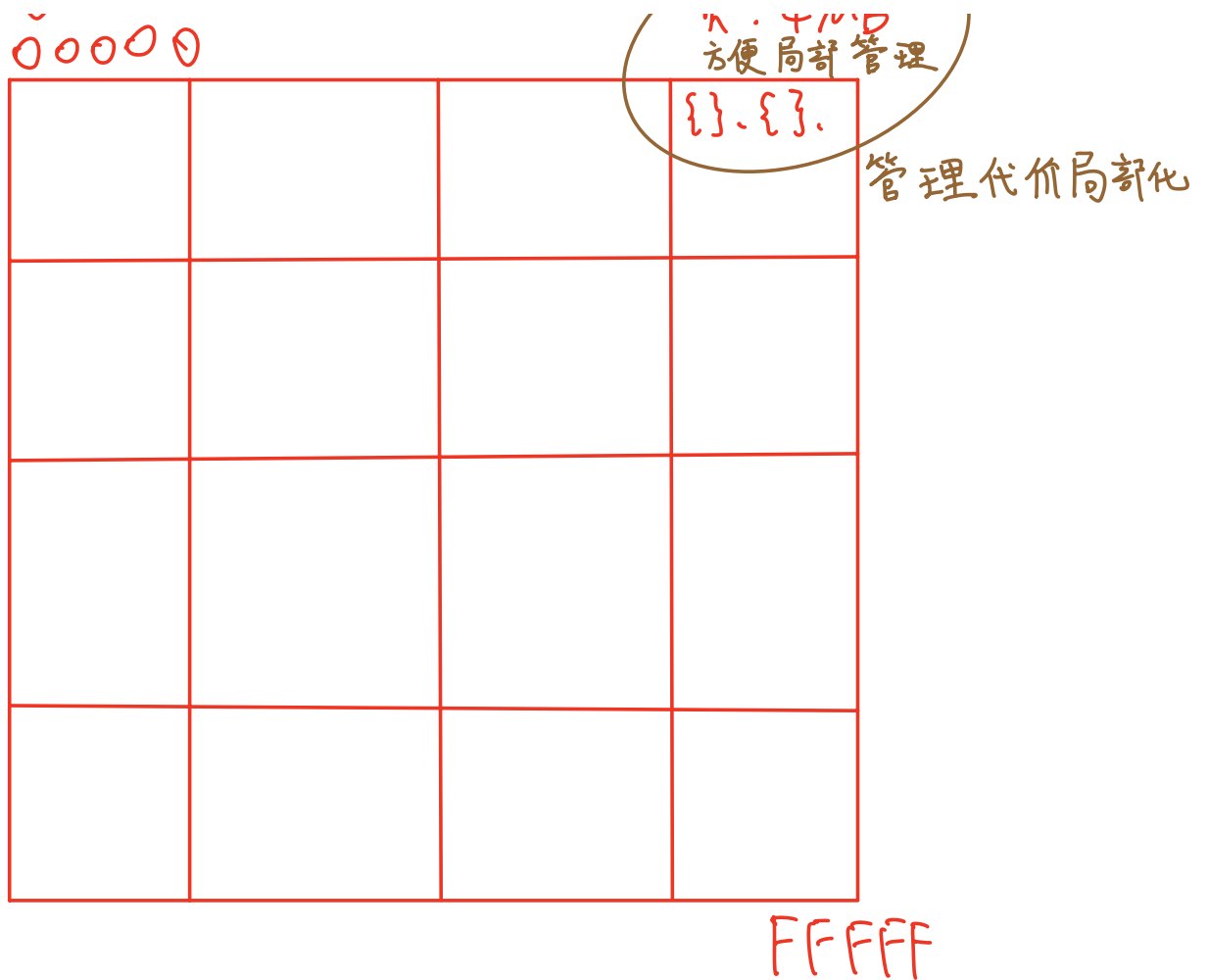
所有的文档的id属性取值在同一个数据库里都不同

## 文档是如何存储的

文档数据库里面把数据都放在一个个文档里面，每个文档描述一个对象，结构像树一样，每个文档里面可以嵌入更小的文档。一个个文档（对象）被包装在一个个的文档集里面，文档集被包装在数据库里。

物理上来看，整个数据管理的方式是以页为单位的，页在内存和磁盘中调进调出。访问数据的时候，因其局部性，cpu对同一页中的多个文档会一起访问，提高存储系统的效率。

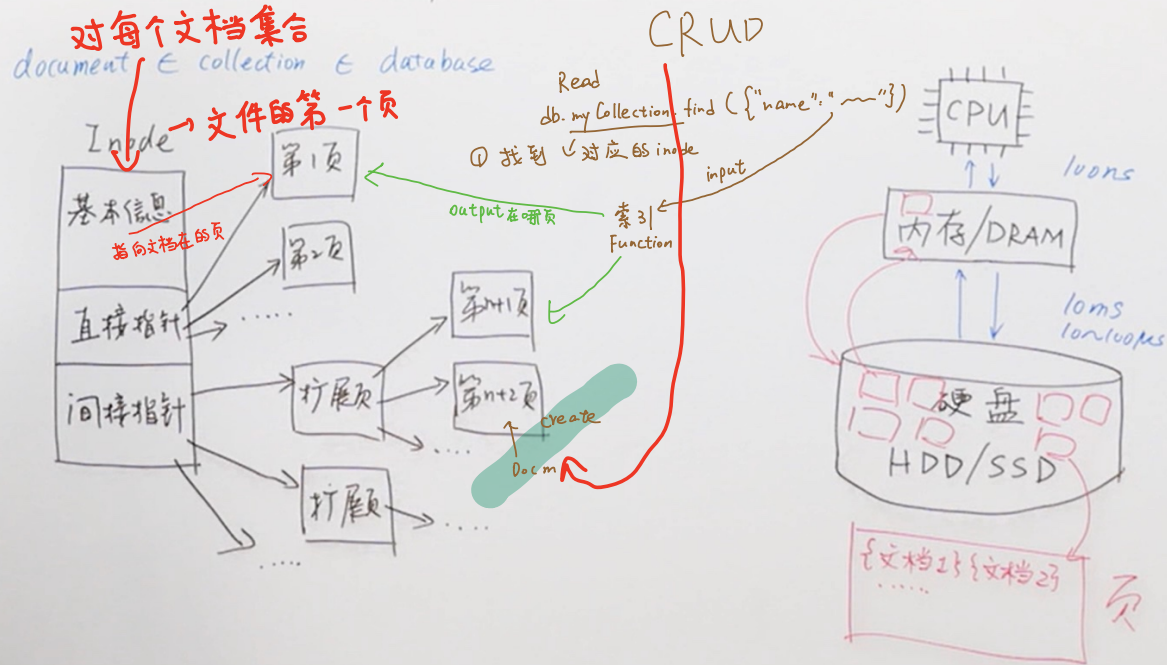




数据管理的方式以页为单位  
页在内存和磁盘之间调入调出



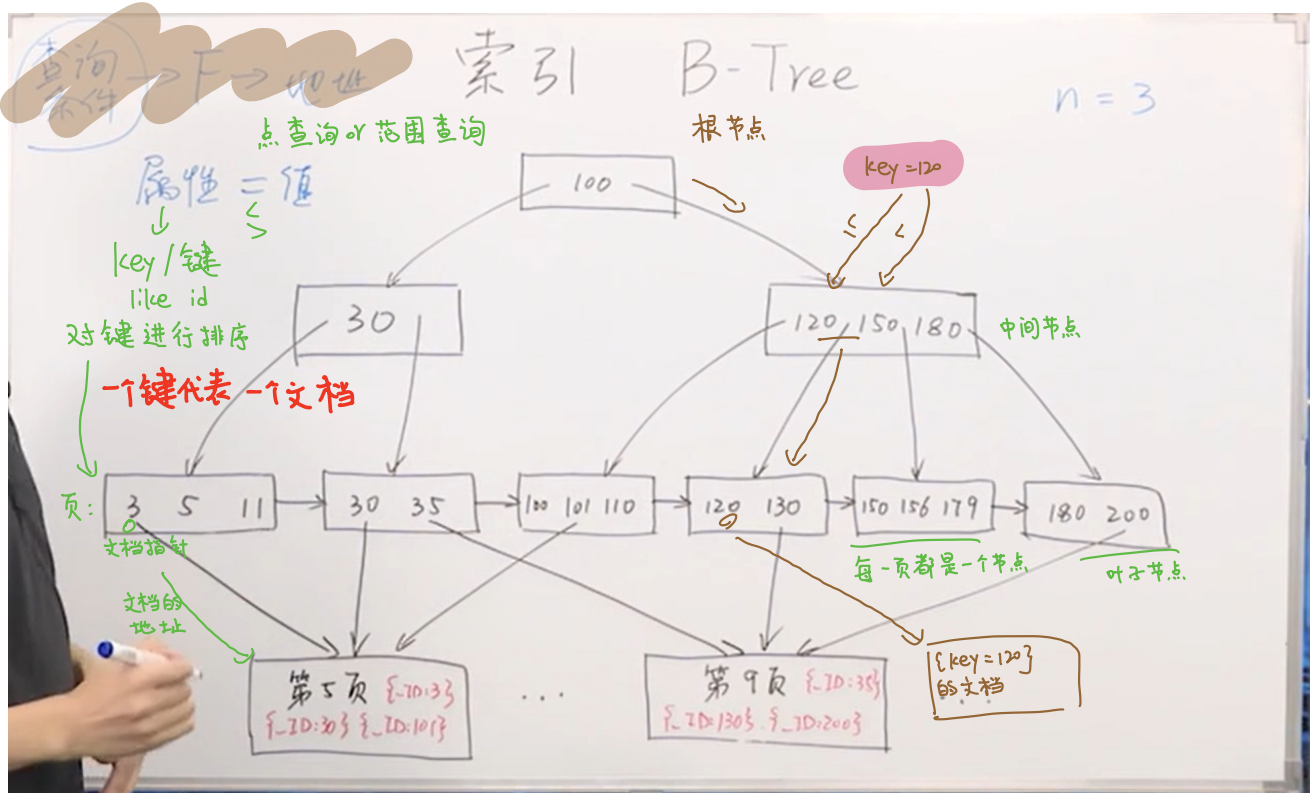
# 文档数据库—数据存储



## 文档集的物理知识

用文件的方式来组织文档，变成文档的集合，再用目录的方式来组织文档集合变成数据库，需要把数据页打包在一起放在同一个地方

## B树索引



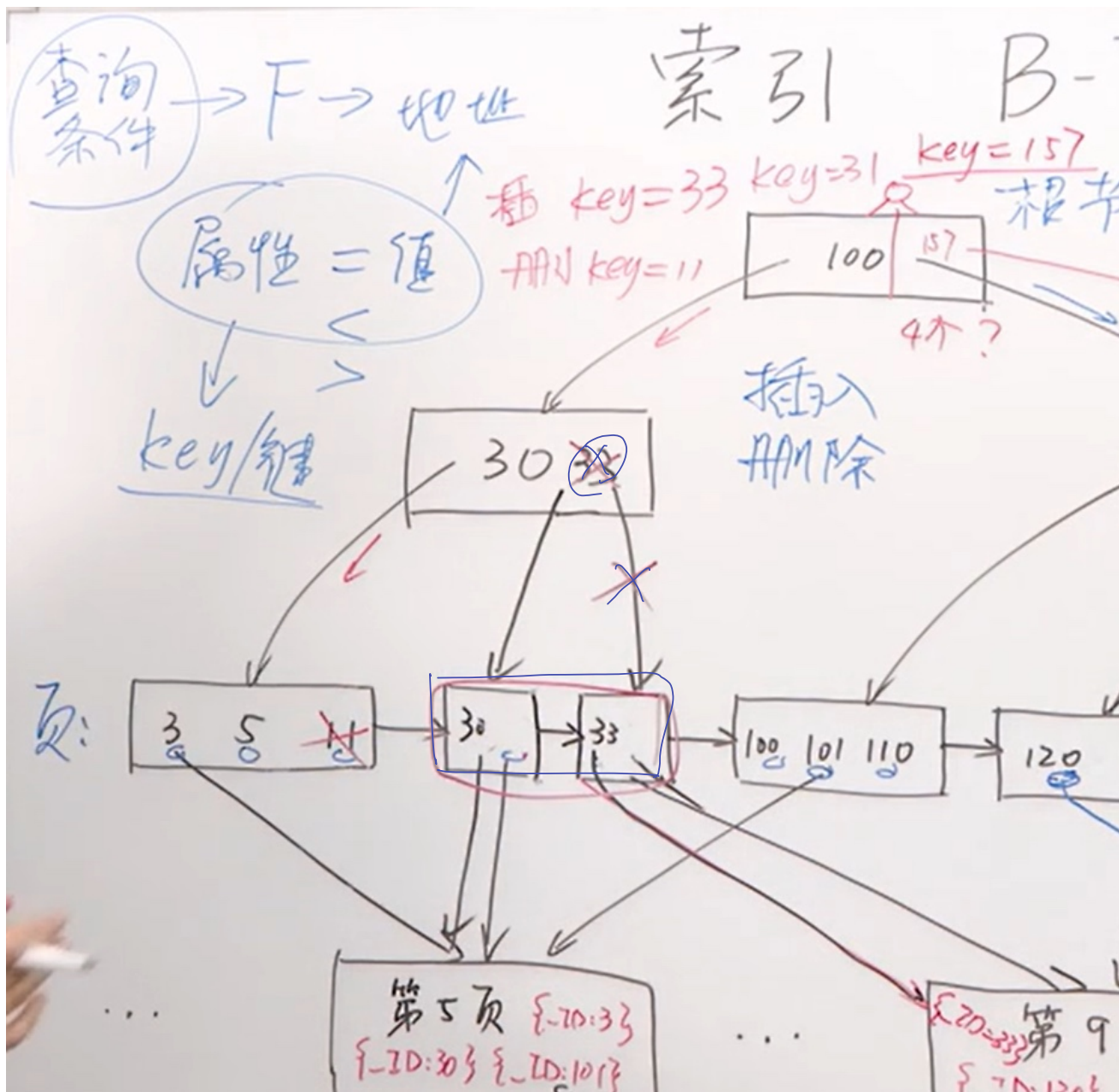
属性等于某个值到地址的映射





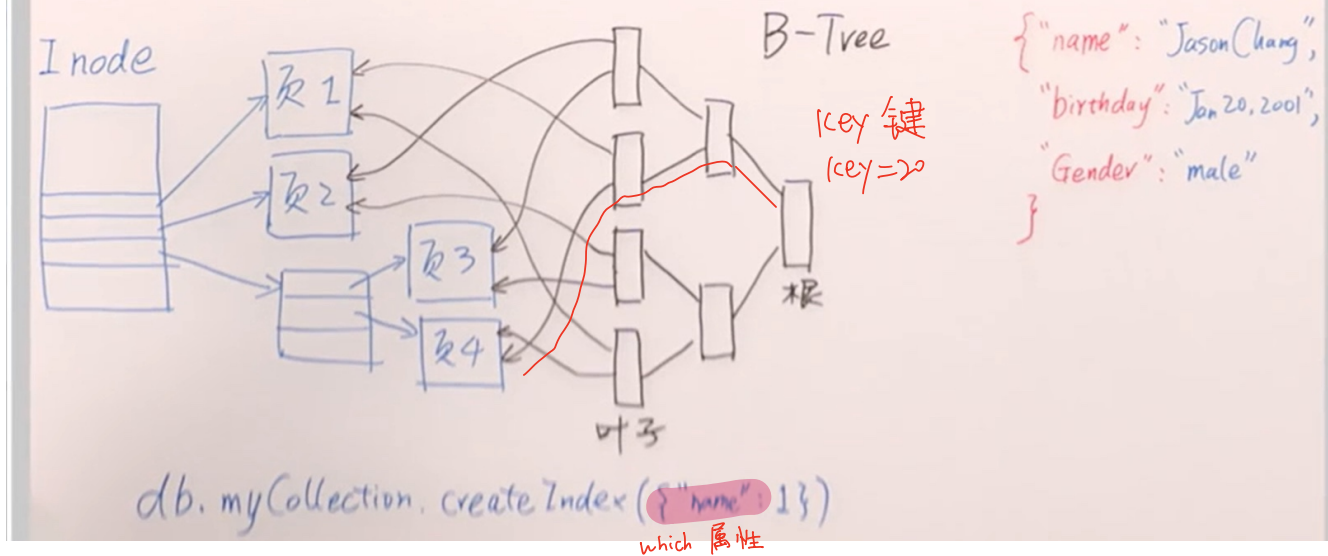


## B树索引的插入与删除



删除

## 索引的使用



### 索引的创建与使用

可以找到键值key=20的文档所在页面的位置，直接读取文档，不用遍历整个文档集合所在页面

### 适合用什么样的属性创造索引呢？

1. 常用的属性 (like 人名)
2. 属性不常被修改，即很稳定 (人名基本不变)  
主索引
3. 索引的有效性

如果是用id 作为key，则不需要inode结构。一般来说，对于文档数据库而言，不去创建索引，而是使用id的属性去访问数据，一般都是比较高效的。