


# MySQL 数据库索引原理与分类

 fqliutao (/u/c169fce5179b) [+ 关注](#)

2017.03.29 11:07\* 字数 1854 阅读 2529 评论 1 喜欢 20

(/u/c169fce5179b)

## 前言

数据库索引本质上是一种数据结构(存储结构+算法)，目的是为了加快目标数据检索的速度。

## 目录

- 1.索引的本质与原理？
- 2.索引的分类？

## 1.索引的本质与原理

我们先看一个问题：

假设现在有100000条从0到10000且从大到小排列的整型数据，1条数据的大小假设(真的只是假设)是1KB,操作系统的每次I/O数据块(页)大小是8KB。

如果现在我要查找其中 50001 这个数据值，有如下几个方式：

1.最蠢的方式，遍历，每次遍历到一个值，就用这个值跟目标值做对比，如果不等于那么查找下一个。这样的话那么每次I/O是8条数据，目标数据在50001/8 约6600多次I/O 才能找到目标数据。

2.二分查找，最好一次性将100000条数据全部读到内存，这样查找也是很快的。

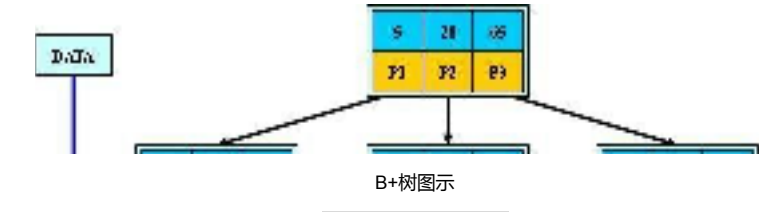
但是即使二分查找很快，但这些数据也不能单单通过一次I/O全部进入内存，进行运算。

那么怎样在I/O 块大小 的限制下快速利用二分查找找到目标值呢？我们得引入新的数据结构，B+树正好可以解决上述I/O块大小的限制，解决限制不是说增大了限制范围，而是我们在此限制上改变了数据的存储结构，即在同等限制条件下，快速检索到目标数据，如下是B+树的原理讲解：

**注意，我们主要讲解索引的原理，没有必要过于纠结B+树的各种操作，及代码实现**

### 1.1 B+ 树





根据上图所示，及其论文定义：

1. 图上蓝色的块为关键字，我们发现所有的关键字最终都会被包含在叶子节点当中。  
图上的黄色区块表示的是子树的指针域，比如根节点下的P2指向的就是28-65之间的索引。

2. 所有的叶子结点中包含了全部关键字的信息，及指向含有这些关键字记录的指针，且叶子结点本身依关键字的大小自小而大的顺序链接。（而B 树的叶子节点并没有包括全部需要查找的信息）

3. 所有的非终端结点可以看成是索引部分，结点中仅含有其子树根结点中最大（或最小）关键字。（而B 树的非终端节点也包含需要查找的有效信息）

现在来看下查找数据 60 的 查找过程，如下所示：

1. I/O第一次：读入5、28、65 数据块，在此同级别节点块上，60在28到65之间(其实是二分查找)，那走P2指针  
2. I/O第二次：读入28、35、56 数据块，在此同级别节点块上，60大于56，所以走P3指针指向的子树(上图中就是  
3. I/O第三次：读入叶子节点，在这个叶子节点中，使用二分查找算法找到目标值60。

由上述查找过程所示统共需要三次I/O就能查到目标值，性能大大提升。

## 2.索引的分类？

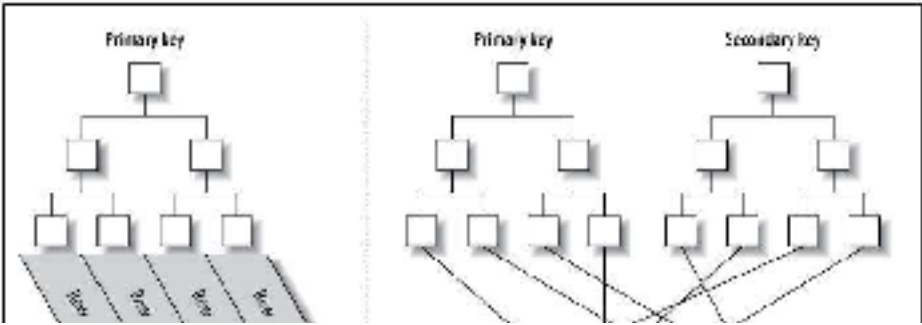
### 2.1 聚簇索引 & 非聚簇索引

InnoDB 主键使用的是聚簇索引，MyISAM 不管是主键索引，还是二级索引使用的都是非聚簇索引。

下图形象说明了聚簇索引表(InnoDB)和非聚簇索引(MyISAM)的区别：

聚簇索引：对正文内容按照一定规则排列的目录称为聚簇索引  
非聚簇索引：目录自己按照一定规则排列，正文自己按照另一种规则排列，目录主要是保存对正文的一个映射关系，这种称为非聚簇索引  
看完上边的定义，你可能对聚簇索引和非聚簇索引还是一头雾水，没关系，看看下边的例子你应该就明白了  
以我们的汉语字典为例，我们要查一个字“爱”，我们知道它的发音，我们会自然的翻开目录中的A目录下去找读音为“ai”的字，找到它并找到它具体对应到哪一页，这里A目录下所有字具体在哪一页都是连续的，假设A目录下“爱”对应18页，在A目录中“爱”的前一个字是“矮”，对应的是17页，也就是他们在目录中的相对顺序也是他们在具体页数中的相对顺序，这个就是聚簇索引。  
但是如果我们遇到一个字，并不知道它的读音，我们就会采用另一种查找方式，根据“偏旁部首”去查找，然后根据这个字后的页码直接翻到某页来找到您要找的字。但你结合“部首目录”和“检字表”而查到的字的排序并不是真正的正文的排序方法，比如你查“张”字，我们可以看到在查部首之后的检字表中“张”的页码是672页，检字表中“张”的上面是“驰”字，但页码却是63页，“张”的下面是“弩”字，页面是390页，这样目录中的排列方式并不是正文实际的排列方式，这就是非聚簇索引。





聚簇索引与非聚簇索引

1.对于非聚簇索引表来说（右图），表数据和索引是分成两部分存储的，主键索引和二级索引存储上没有任何区别。使用的是B+树作为索引的存储结构，所有的节点都是索引，叶子节点存储的是索引+索引对应的记录的地址。

2.对于聚簇索引表来说（左图），表数据是和主键一起存储的，主键索引的叶结点存储行数据(包含了主键值)，二级索引的叶结点存储行的主键值。使用的是B+树作为索引的存储结构，非叶子节点都是索引关键字，但非叶子节点中的关键字中不存储对应记录的具体内容或内容地址。叶子节点上的数据是主键与具体记录(数据内容)。

聚簇索引的优点

- 1.当你需要取出一定范围内的数据时，用聚簇索引 (<https://link.jianshu.com?t=http://baike.baidu.com/view/1028053.htm>)也比用非聚簇索引 (<https://link.jianshu.com?t=http://baike.baidu.com/view/1615249.htm>)好。
- 2.当通过聚簇索引查找目标数据时理论上比非聚簇索引要快，因为非聚簇索引定位到对应主键时还要多一次目标记录寻址,即多一次I/O。

聚簇索引的缺点

1.插入速度严重依赖于插入顺序，按照主键的顺序插入是最快的方式，否则将会出现页分裂，严重影响性能。因此，对于InnoDB表，我们一般都会定义一个自增的ID列为主键。

2.更新主键的代价很高，因为将会导致被更新的行移动。因此，对于InnoDB表，我们一般定义主键为不可更新。

3.二级索引访问需要两次索引查找，第一次找到主键值，第二次根据主键值找到行数据。

二级索引的叶节点存储的是主键值，而不是行指针（非聚簇索引存储的是指针或者说是地址），这是为了减少当出现行移动或数据页分裂时二级索引的维护工作，但会让二级索引占用更多的空间。

4.采用聚簇索引插入新值比采用非聚簇索引插入新值的速度要慢很多，因为插入要保证主键不能重复，判断主键不能重复，采用的方式在不同的索引下面会有很大的性能差距，聚簇索引遍历所有的叶子节点，非聚簇索引也判断所有的叶子节点，但是聚簇索引的叶子节点除了带有主键还有记录值，记录的大小往往比主键要大的多。这样就会导致聚簇索引在判定新记录携带的主键是否重复时进行昂贵的I/O代价。

唯一索引



主键就是唯一索引，但是唯一索引不一定是主键，唯一索引可以为空，但是空值只能有一个，主键不能为空。

普通唯一索引：单个字段上建立唯一索引，需要此字段所在的列上不能有重复的值，属于二级索引。

复合唯一索引：多个字段上联合建立唯一索引，属于二级索引。

覆盖索引

查找的目标数据，包含在索引中，如建立idx\_column1\_column2。

```
select column1 from table where column1 = ? and column2 > ?
```

通过查询索引就能确定最终的数据，不用再利用叶子节点中存储的主键值去查询对应的数据。

覆盖索引的性能是极高的。

索引原理篇讲述完，下一篇讲解索引的优化，以及 explain 工具的使用。

博客搬家：大坤的个人博客 (<https://link.jianshu.com?t=http://markfork.com/>)  
欢迎评论哦~

一切从简。

赞赏支持

数据库 (/nb/6034233)

举报文章 © 著作权归作者所有



**fxliutao (/u/c169fce5179b) ♂**  
写了 51916 字，被 237 人关注，获得了 332 个喜欢  
(/u/c169fce5179b)


+ 关注

简单的人


喜欢 | 20



(<http://cwb.assets.jianshu.io/notes/images/1071186>)



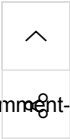
下载简书 App ▶  
随时随地发现和创作内容



(/apps/download?utm\_source=nbc)




登录 (/sign-in?utm\_source=desktop&utm\_medium=not-signed-in-comment-form)



1条评论 只看作者

按喜欢排序 按时间正序 按时间倒序



kuff (/u/9b3e39d69ca6)

2楼 · 2017.07.28 18:19

(/u/9b3e39d69ca6)





文章写得很赞 能否讲解下 数据库里 表空间 页 这些概念 一直很模糊

赞

回复

(/a  
utn

被以下专题收入，发现更多相似内容

- 程序员 (/c/NEt52a?utm\_source=desktop&utm\_medium=notes-included-collection)
- IT部落 (/c/309ff0f2e062?utm\_source=desktop&utm\_medium=notes-included-collection)
- 架构&设计&技术 (/c/a5a1c271979b?utm\_source=desktop&utm\_medium=notes-included-collection)
- 数据库 (/c/09da0deadd7?utm\_source=desktop&utm\_medium=notes-included-collection)

推荐阅读

更多精彩内容 > (/)

Spring loc 源码分析(四)--parseBeanDefinitions()与BeanDefinitionParse...

上一篇博客说到，BeanDefinition的解析,已经走到了DefaultBeanDefinitionDocumentReader里，这时候配置文件已经被加载，并解析成w3c的Document对象。这篇博客就接着介绍，...

fxliutao (/u/c169fce5179b?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

IntelliJ Idea--Debug使用 ( Mac ) (/p/adf995f3c6ae?u...

(/p/adf995f3c6ae?utm\_campaign=maleskine&utm\_content=note&utm...

前言：本文的核心是记录debug的使用，与具体的操作系统无关。1.什么是Debug Debug是程序的一种运行模式。用来发掘程序的走向，以及程序在运...

fxliutao (/u/c169fce5179b?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

突然不想努力了怎么办？认真看完帮你找回好状态 (/p/...

(/p/c83f8a1144c3?utm\_campaign=maleskine&utm\_content=note&utm...

01. 我们每个人，都有那么几个瞬间：知道任务的deadline只有几天了，就是不想动手去做；本来计划周末认真读完1本书，结果就是看不进去；被闹钟叫醒...

杨枫先生 (/u/61c389460cd6?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

写给未来男朋友的一封信 (/p/d46393205485?utm\_ca...

(/p/d46393205485?utm\_campaign=maleskine&utm\_content=note&utm...

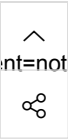
亲爱的某某：你好啊，见字如面。今天是2018年3月13日，现在是晚上，我在成都，还是一名大三学生。我想快点遇到你，给我一个温暖可靠的拥抱。我...

静若517 (/u/0f63d758cf5a?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

简宝玉分享会|我为什么说越是二三本大学越要考研？ (/...

(/p/80c4eb636baf?utm\_campaign=maleskine&utm\_content=note&utm...

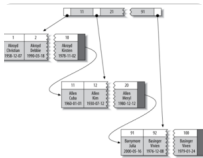
大家好哇，我是简书作者考研侠。我曾经一路从三本逆袭至专业第一的财经名校，热衷帮助出身不好的孩子走向逆袭，文末会有关于考研择校择专业课的相...



P学长 (/u/497fe8c65a53?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

(/p/54c6d5db4fe6?)



(/a  
utm

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

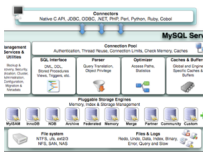
《高性能MySQL》读后感——聚簇索引 (/p/54c6d5db4fe6?utm\_campaig...

聚簇索引并不是一种单独的索引类型，而是一种数据存储方式。比如，InnoDB的聚簇索引使用B+Tree的数据结构存储索引和数据。当表有聚簇索引时，它的数据行实际上存放在索引的叶子页(leaf page)中。因为无...

大头8086 (/u/a4719179fc52?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/bd8675e5c7b2?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

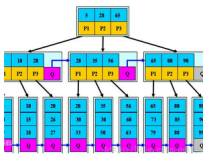
《高性能MySQL》&《MySQL技术内幕 InnoDB存储引擎》笔记 (/p/bd867...

《高性能MySQL》&《MySQL技术内幕 InnoDB存储引擎》笔记 第一章 MySQL架构与历史 MySQL的架构从上图可以看出，MySQL数据库区别于其他数据库的最重要的一个特点就是其插件式的表存储引擎。需要...

xiaogmail (/u/59cac9fff87c?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/6b080a787b61?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

MySQL优化笔记（三）--索引的使用、原理和设计优化 (/p/6b080a787b61?...)

之前的文章一直在规避索引的建立去优化数据库，不是不想讲，而是这个太重要，必须抽出来讲。今天我们就来研究下数据库索引的设计与优化（MySQL为例）。文章结构：（1）索引的概述和使用；（2）索引...

JackFrost\_fuzhu (/u/8dad2b82bce7?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

(/p/bf0c2ba00b03?)

| Index Page 满 | 操作   |
|--------------|--|
| No           | 直接将记录插入到叶...   |
| No           | 1) 新分 Leaf Page<br>2) 将中间节点插入<br>3) 小于中间节点的记<br>4) 大于或等于中间节...   |
| Yes          | 1) 新分 Leaf Page<br>2) 小于中间节点的记<br>3) 大于或等于中间节<br>4) 新分 Index Page<br>5) 小于中间节点的记<br>6) 大于中间节点的记<br>7) 中间节点插入上... |

utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

MySQL索引 (/p/bf0c2ba00b03?utm\_campaign=maleskine&utm\_conten...

索引的本质 MySQL官方对索引的定义为：索引（Index）是帮助MySQL高效获取数据的数据结构。在关系型数据库中设计索引其实并不是复杂的事情，很多开发者都觉得设计索引能够提升数据库的性能，相关的知...

Rick617 (/u/fee7a6fc9e87?)

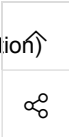
utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/486a514b0ded?)

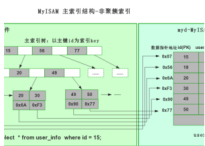
utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

MYSQL-B+TREE索引原理 (/p/486a514b0ded?utm\_campaign=maleskin...

1.什么是索引？索引：加速查询的数据结构。2.索引常见数据结构：#1.顺序查找: 最基本的查询算法-复杂程度O（n）,大数据量此算法效率糟糕。#2.二叉树查找（binary tree search）：O(log2n) 左边是数据表，一...



 kimze1107 (/u/3742a0a65ed4?)



(/a  
utn

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/4aea14c1d28a?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

**【星涵每日绘】20170509 (/p/4aea14c1d28a?utm\_campaign=maleskine...**


大家是不是有同感：每天感觉时间不够用啊~ 看到深度思考训练营里丽影同学分享自己的心得，深有同感啊！（此处加一个握爪的表情）。想想自己也是这样的，每天早五点多起床，做早饭，送她到小区门口，...

 那里面 (/u/567ed9b531ae?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

**送别（3） (/p/4c616b82fd0c?utm\_campaign=maleskine&utm\_content=...**

她回头，不断回头 她皱着眉头 让我不敢回头 她拖起箱子，她来回奔走 她伸起脖子，活像一头鹿 隔离带外站着的那个女人 是唯一一个可以让我哭的

 丁千 (/u/b2ac7f05581f?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/ce2c5c92d65a?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

**一味省钱，省钱，是一个很大的误区 (/p/ce2c5c92d65a?utm\_campaign=...**

文章约2500字，建议阅读时间10分钟 图片来源：百度 一、没钱的日子让我们害怕 钱不是万能的，但没有钱是万万不能的。不得不说，从读书的时候到现在，这句话一直深得我心。钱不是万能的，有些东西我们确...

 成长的功夫熊猫 (/u/606013ad3026?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)


(/p/ff7eb564409f?)



utm\_campaign=maleskine&utm\_content=note&utm\_medium=seo\_notes&utm\_source=recommendation)

**愿你历尽千帆，归来仍是少年 (/p/ff7eb564409f?utm\_campaign=maleskin...**


收到此书，是在今天上午，有些出乎意料，这个以干旱少雨著称的西北城市罕见地下起了雨，下雨天是个读书的好天气，因为空气清新湿润，静谧安静，读一本好书来消磨时光实在是美事一桩。很喜欢这本书的书...

 笑薇 (/u/b62c188dc635?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)

**写在2016年的最后一个凌晨 (/p/9dff8c8ef171?utm\_campaign=maleskine...**

现在时间:2016年12月31日2:29。我刚刚从嘈杂的酒吧返回家中-----这是我人生中第三次去酒吧。多多少少感觉到了一些不一样，无论是身边人的态度，还是自己的感觉。首先让我惊奇的是，晚上十二点以后，爸...

 乐呵的卉子 (/u/cfd59b79d63a?)

utm\_campaign=maleskine&utm\_content=user&utm\_medium=seo\_notes&utm\_source=recommendation)



