B+ 树索引

没有索引时进行查找

在一个页中

数据比较少，在一个页中的话，

- 主键，二分，找对应的槽，然后在页内逐条找，这种情况下，每页最多 8 条。
- 其他列， 从 Infimum 记录开始依次遍历单向链表中的每条记录，依次对比。

很多页中

- 定位到该记录所在的页
- 从所在的页内查找相应的记录

  如果没有索引，无论是根据主键还是非主键查找，都是从第一页的 Infimum 记录开始，逐条找起，找完了沿着双向链表去下一页中找。效率更低了。

索引

为每个数据页 设置一个目录项，包括两部分：

- 页的 用户记录中最小的主键值，用 key 来表示
- 页号，用 page_no 表示

这些目录项 也可以用存 记录的方式来存储，叫做目录项记录。

目录就是索引

用 记录头信息中的 `record_type` 属性来区分记录

`record_type` 取值为

- 0 普通用户记录
- 1 目录项记录
- 2 Infimum 记录
- 3 Supremum 记录

目录项记录与普通的用户记录的区别：

- `record_type` 的值不一样

- 目录项中只有两个列，主键值和页的编号，但是普通用户记录的列，包括用户自定义的列，还有 InnoDB 的隐藏列。
- 记录头信息中的 min_rec_flag 属性， 目录项的 min_rec_flag 属性的值 可能为 1，普通用户记录的 min_rec_flag 属性的值 可能为 0

除此之外，目录项记录和普通的用户记录，是一样的，二者用的是一样的数据页来存储，页面类型为 0x45BF， 页也是由 7个组成部分组成的，都会为主键生成 页目录。

根据某个主键值去查找记录

1. 先存储 目录项记录 的页

   每一个目录项记录都有主键值，那么对应的存储目录项记录的页，它记录的主键值的范围可以确定：[本页最小目录项记录的主键值，下一页最小目录项记录的主键值） 注意是左闭右开。那么就可以根据这个来确定目录项记录在哪一个页里面

2. 确定存储 目录项记录

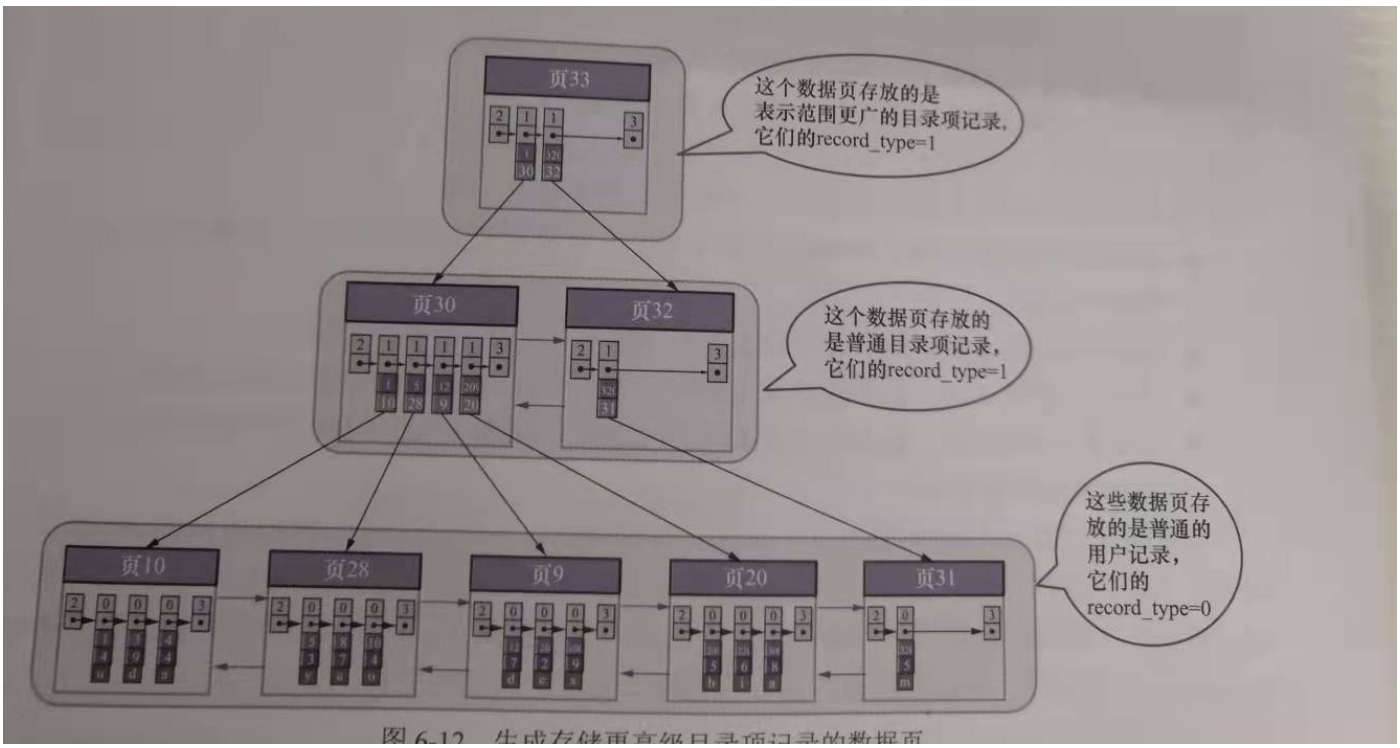
   在存储目录项记录的页中，通过 2 分法找到对应的目录项记录。还是通过目录项中的主键值。来确定一条记录的目录项记录 。

3. 找具体的 用户记录

   目录项记录确定，那么里面的页号可以确定，可以定位到具体的页，

   定位到具体的页后，就是 在一个数据页中，根据主键来查找对应的记录 InnoDB 数据页结构

我们还可以为这些存储目录项记录的页再生成一个更高级的目录，



这样就形成了一种树形结构，要做 b+树。

B+ 树：

真正的用户记录其实都存放在页子节点上，

最上面的节点叫做根节点。

数据页作为节点时，其层级用一个名叫 `PAGE_LEVEL` 的属性来表示。

一般情况下，用到的 b+树不会超过 4层