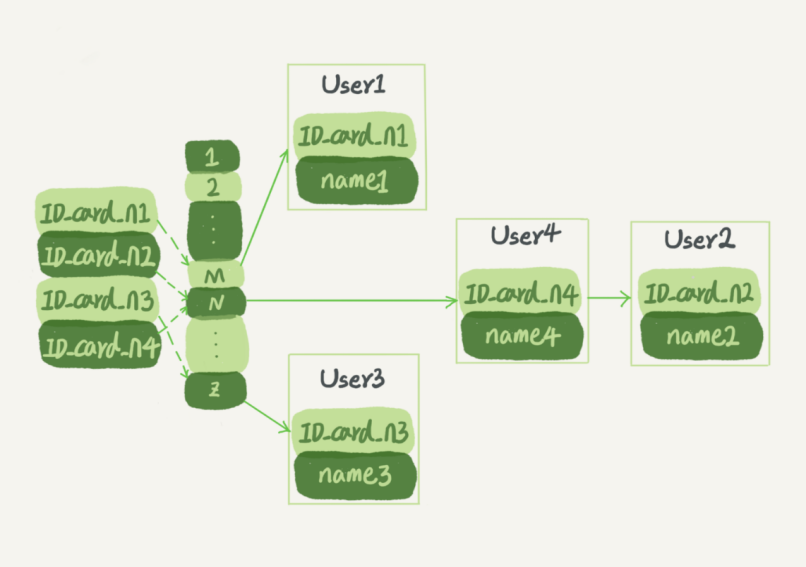
**MySQL索引**

**索引：为了提高数据查询的效率，**像书的目录一样。

常见实现：哈希表、有序数组、搜索树。

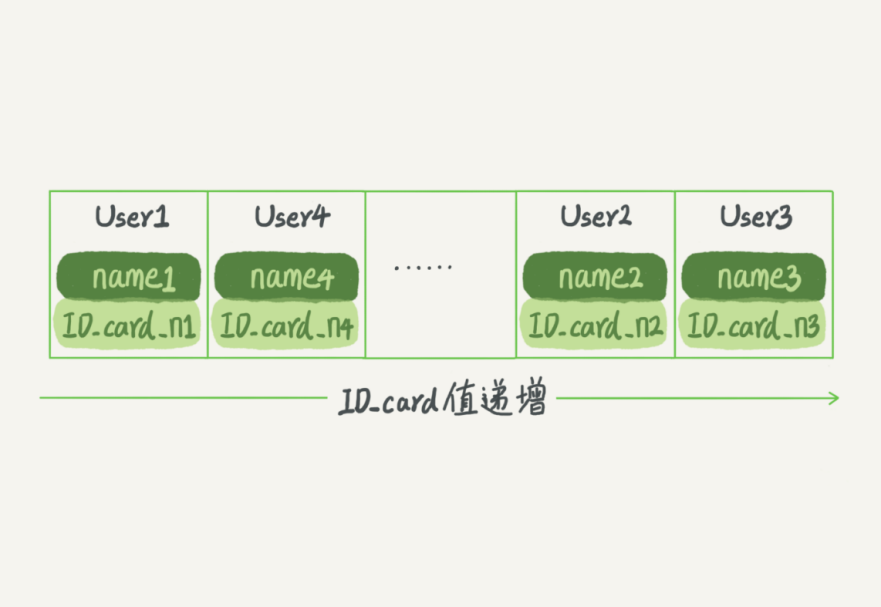
**1、哈希表：**



优点：等值查询快、插入快；只适合等值查询。

缺点：区间查询慢，相当于全部扫描。

**2、有序数组：**

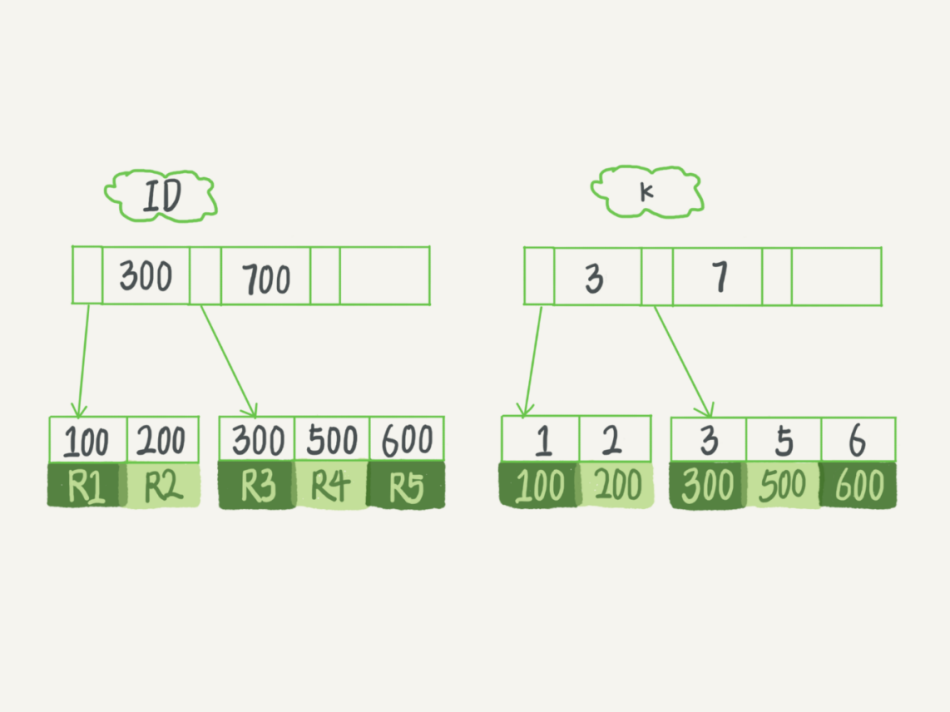


优点：等值查询和范围查询都快，可以用二分法 O(logN)

缺点：插入删除太麻烦，适合不会再修改的数据。

1. **N叉搜索树（5.6之后通过page大小间接控制N值）**

IO次数取决于数的高度；二叉树的树高仍然很高；为了让一个查询尽量少地读磁盘，就必须让查询过程访问尽量少的数据块。 一个数据块的空间远远大于一个二叉树节点空间，那么，我们就不应该使用二叉树，而是要使用“N 叉”树。 N 叉树由于在读写上的性能优点，以及适配磁盘的访问模式，已经被广泛应用在数据库引擎中了。B+树



主键索引（聚簇索引）、非主键索引（二级索引）

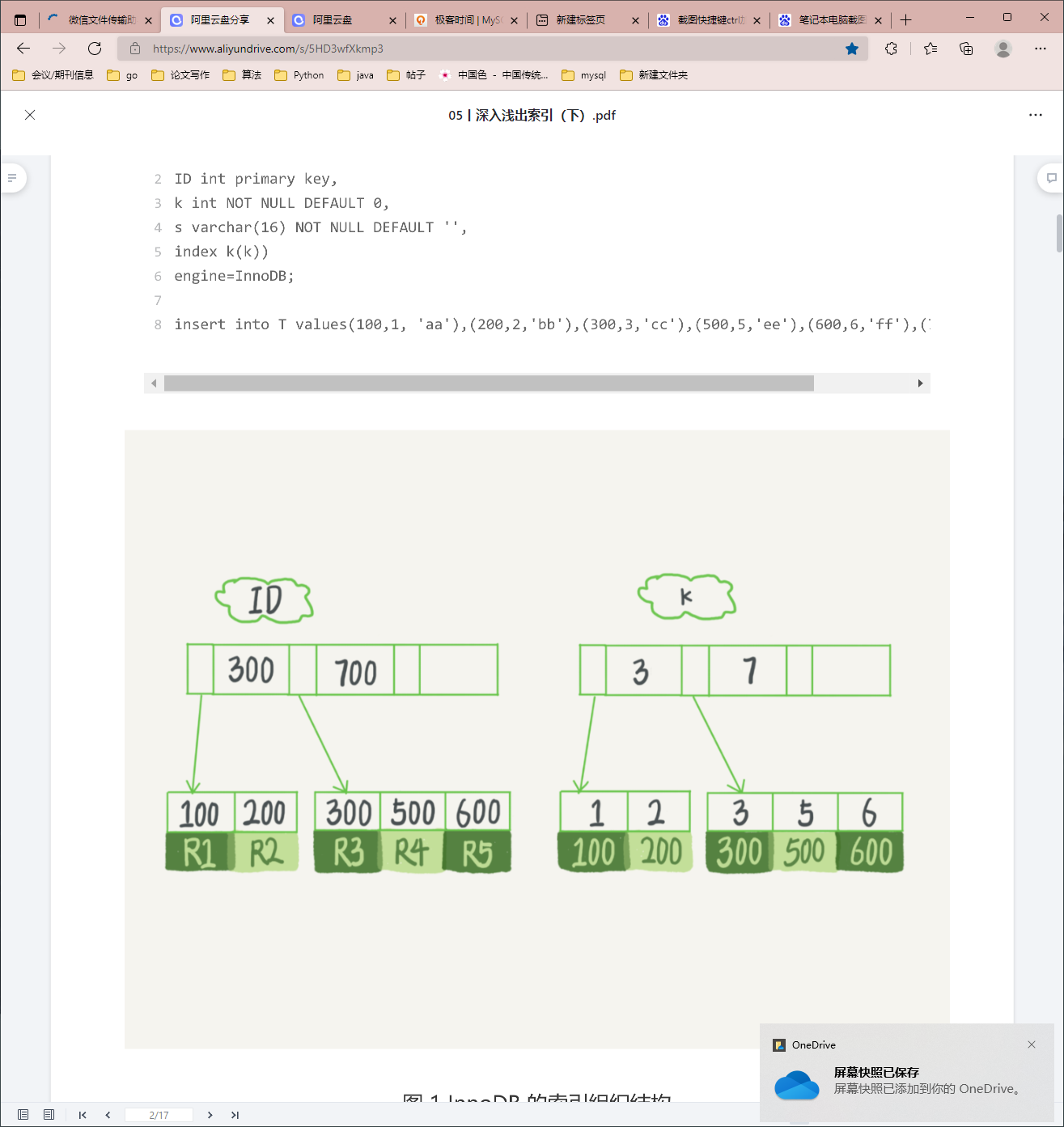
1. when用自增主键、when用业务逻辑字段做主键？

左图+700 ok +400会页分裂；

主键长度越小，普通索引的叶子节点就越小，普通索引占用的空间也就越小；

2）“尽量使用主键查询”原则

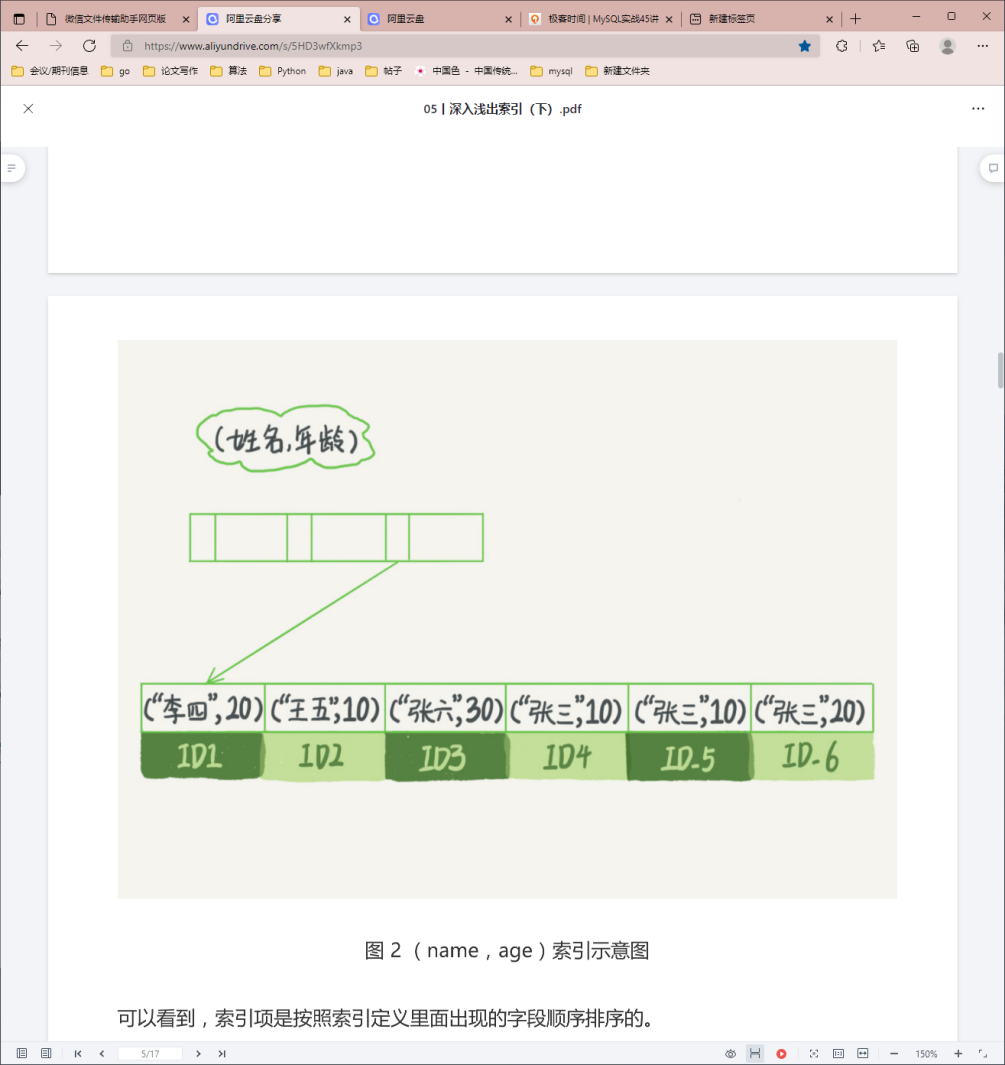
select \* from T where k between 3 and 5；（回表两次）



**回表：**回到主键索引树搜索的过程。

**索引优化：**

1. **覆盖索引：**select ID from T where k between 3 and 5，ID 的值已经在 k 索引树上了，因此可以直接提供查询结果，不需要回表。索引 k 已经“覆盖了”我们的查询需求，称为覆盖索引。
2. 最左前缀原则，**联合索引合理安排字段顺序**。索引项是按照索引定义里面出现的字段顺序排序的。
3. **索引下推：** 可以在索引遍历过程中，对索引中包含的字段先做判断，直接过滤掉不满足条件的记录，减少回表次数。

最左前缀原则

索引下推