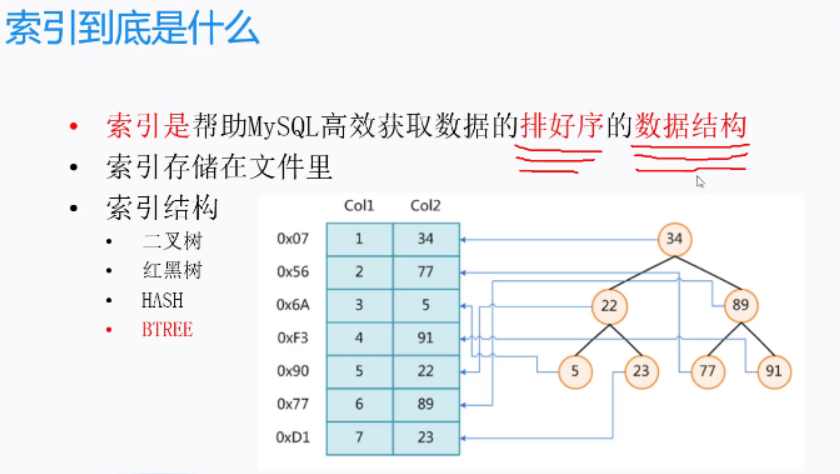
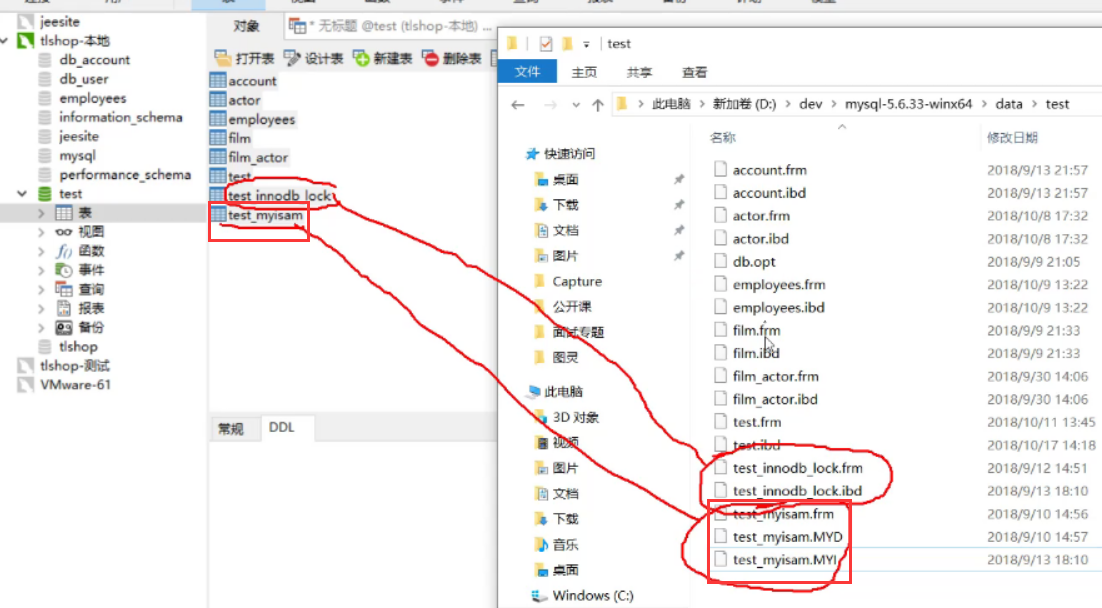
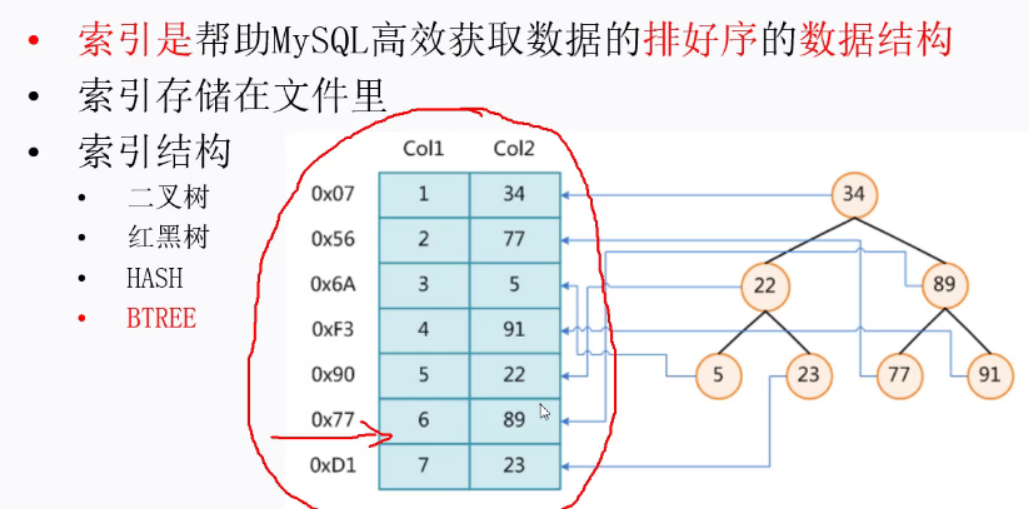
MySQL数据库索引

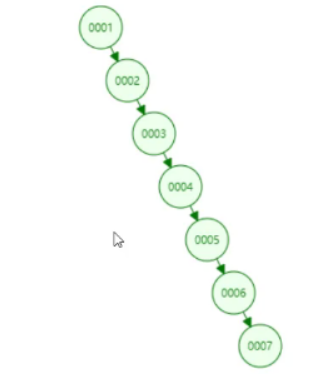






## 二叉树

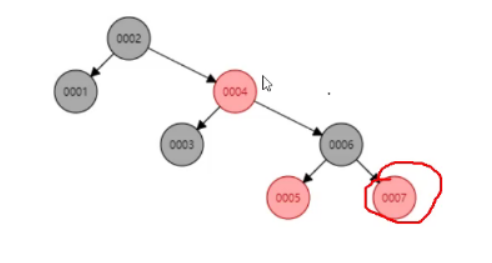
使用二叉树作为索引的数据结构：会导致二叉树严重倾斜，单边增长，带来的查询效率并不高。



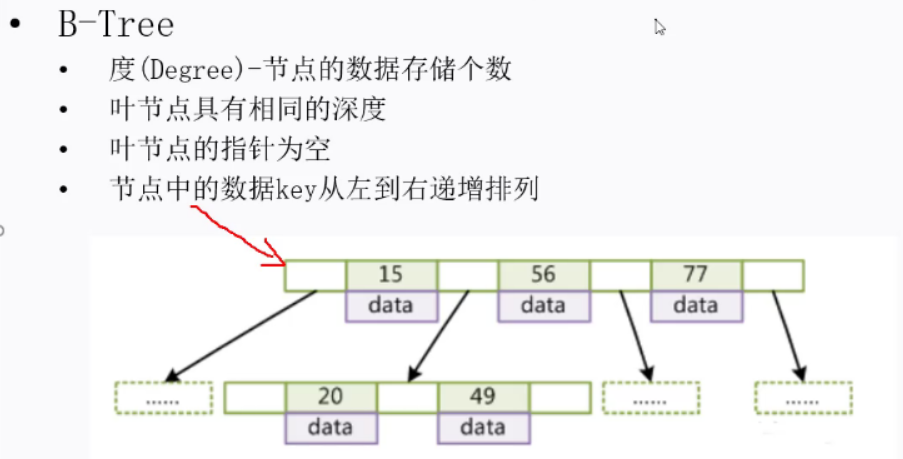
红黑树：Red-Black Tree -----**二叉平衡树**

为什么不用红黑树作为索引的数据结构？

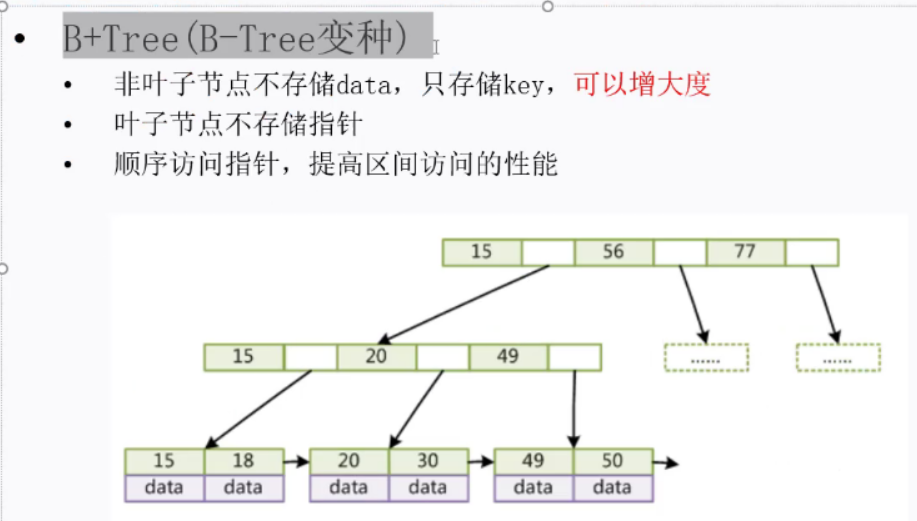
因为红黑树的高度不稳定，高度较大，查询次数会比较大。



## B树:Btree-------多叉平衡树。



## B+树



由于每一行的存储容量有限，如果把data移动到叶子节点，则会增大当前行的存储的节点个数，从而可以**大大降低B+树的深度**。

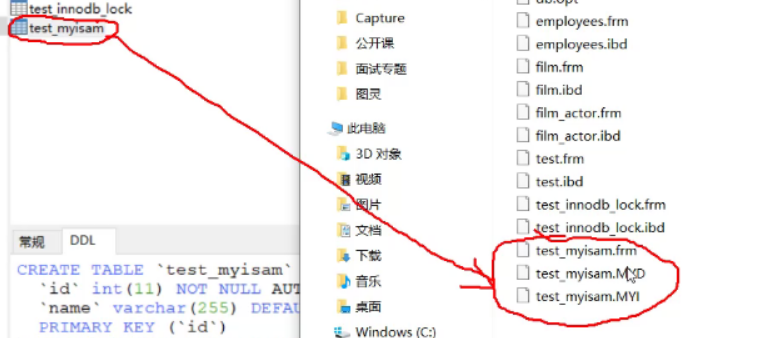
B+树相对于B树的两大改进：

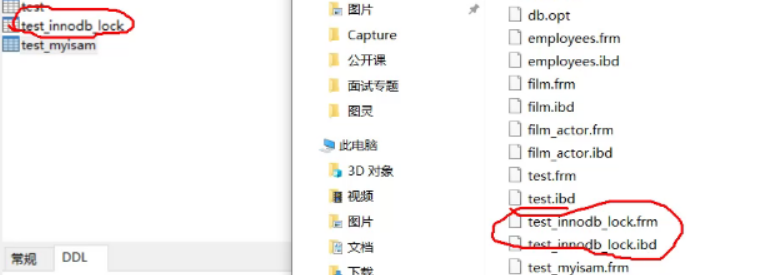
### 非叶子节点不存储data，只存储key，叶子节点不存储key只存储data；

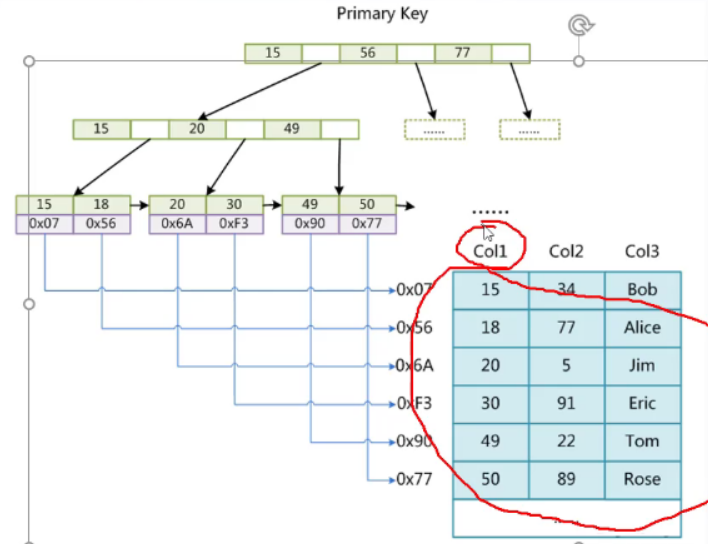
### 叶子节点之间增加顺序访问指针，提高区间(范围)访问的性能；==>范围查找

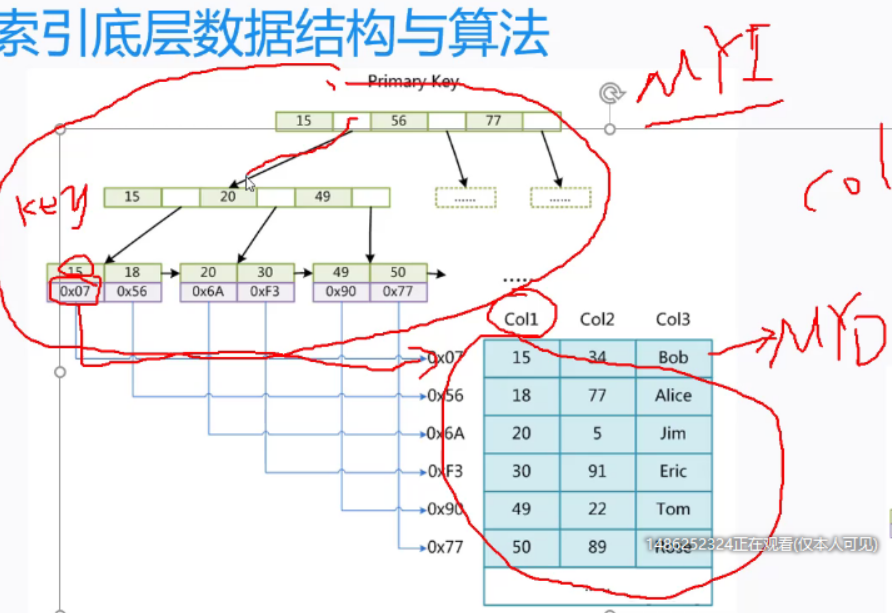
## hash索引

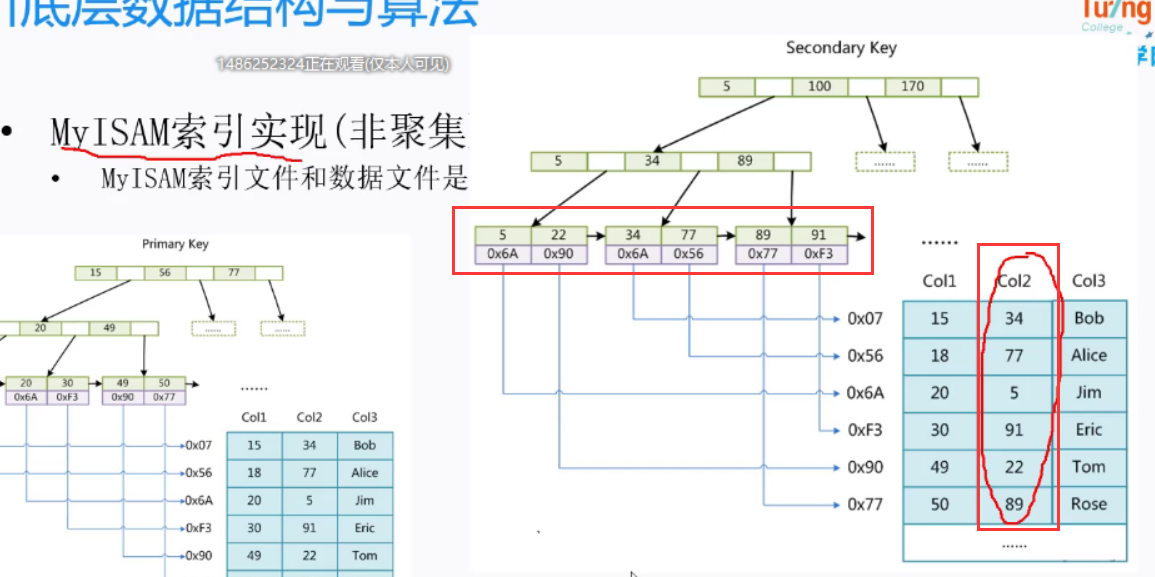
hasn索引不适用于范围查找。

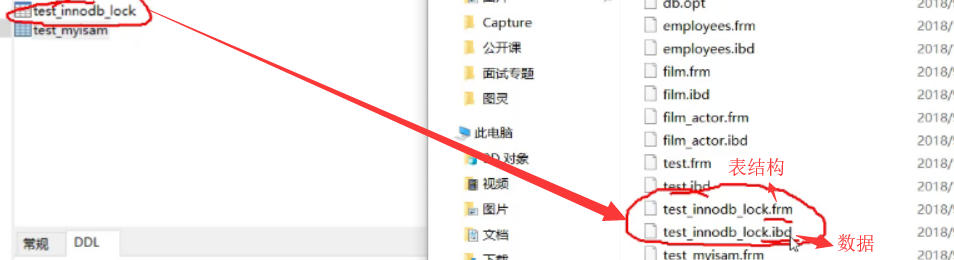




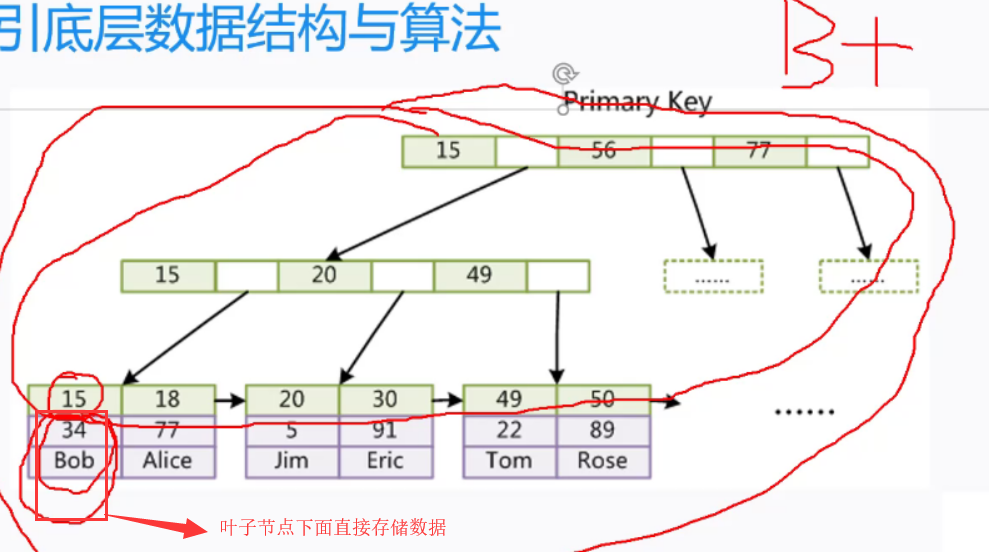




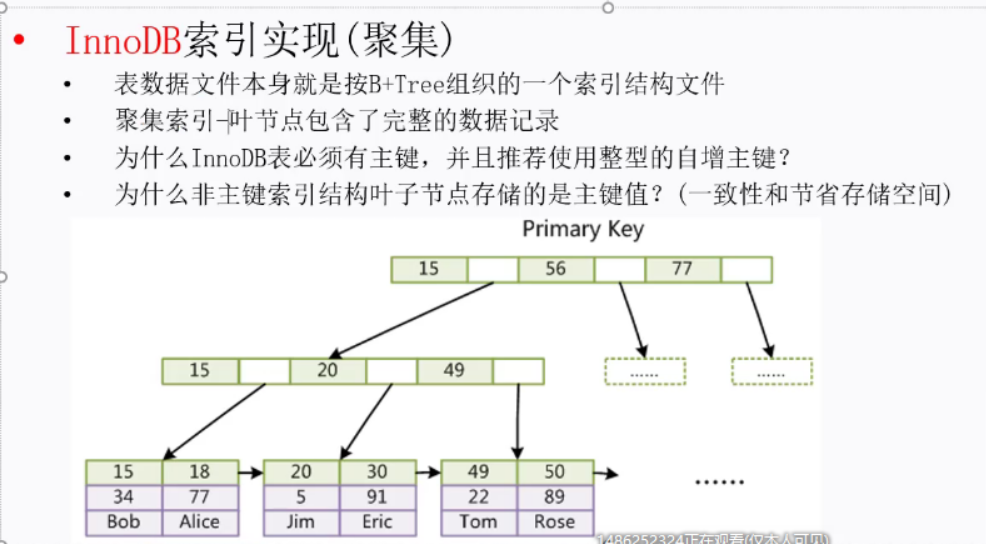




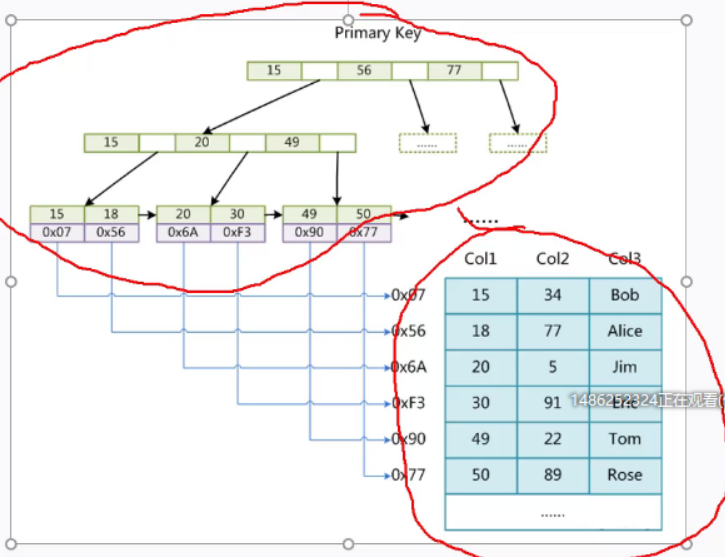
主键 自带索引。主键索引。



InnoDB的主键索引就是聚集索引：



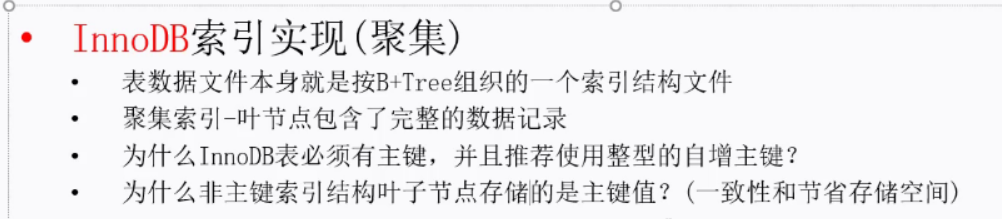
**MyIsm的主键是非聚簇索引**，数据文件与索引文件是分开的。



面试题目：为什么InnoDB的表需要建立一个整型的自增的主键？

原因之一：为什么整型？整型占据空间小；uuid为字符串型，占用空间大。

原因之二：为什么自增？连续存储。



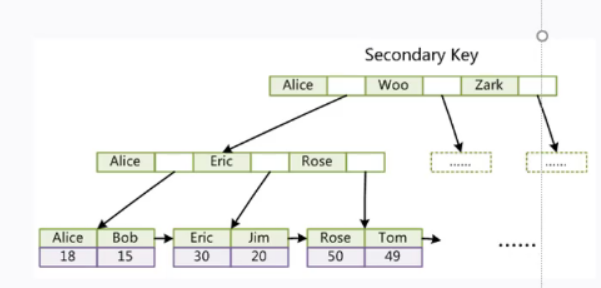
# 联合索引



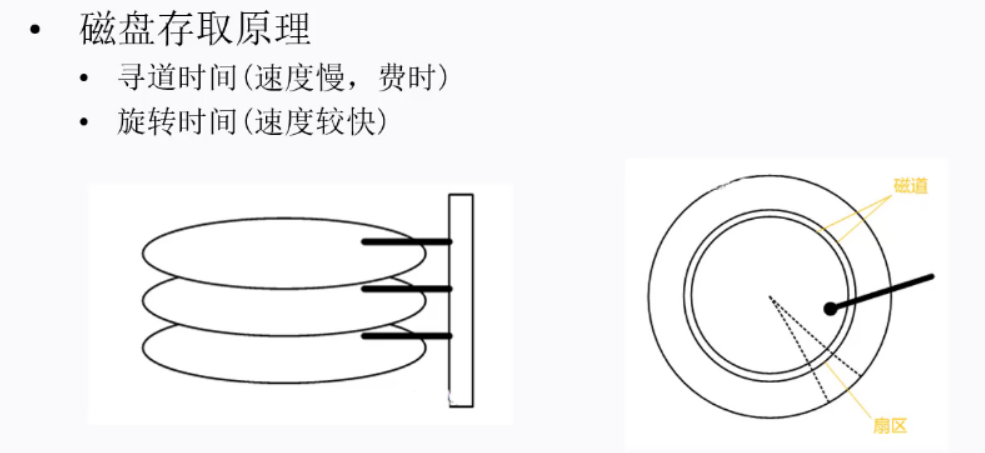


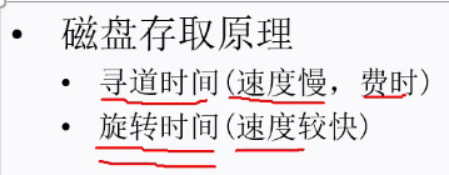
聚簇索引与非聚簇索引的区别：

InnoDB的二级索引不是一个聚簇索引。









多少转说的就是磁盘的旋转速度；

磁头只能左右移动，这叫寻道，速度比较慢。



<https://www.cs.usfca.edu/~galles/visualization/>