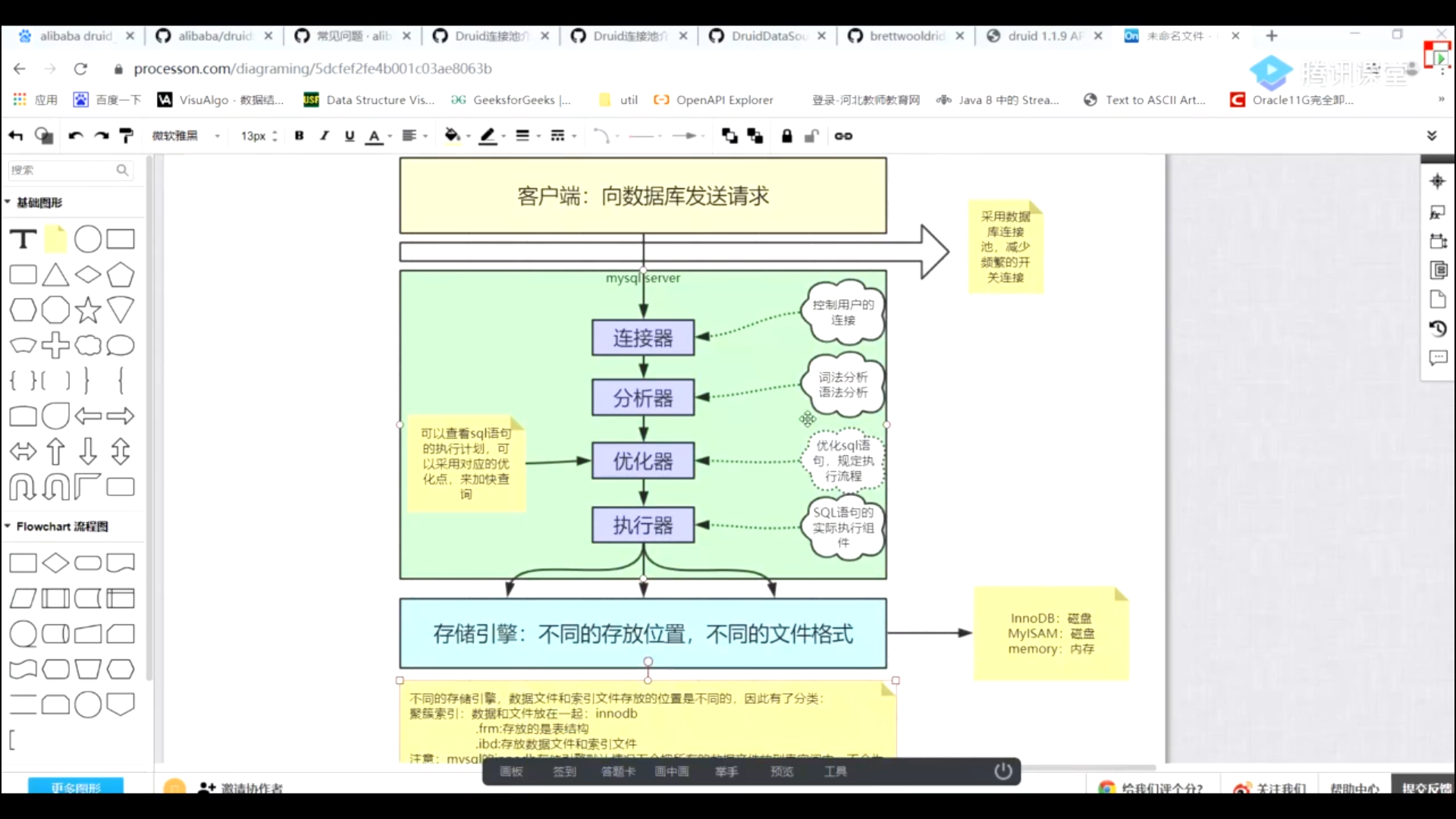
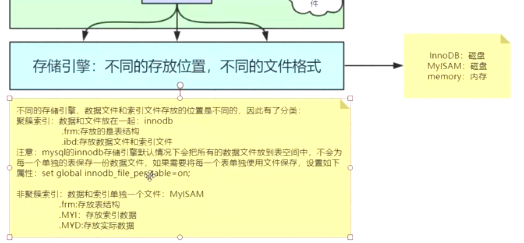
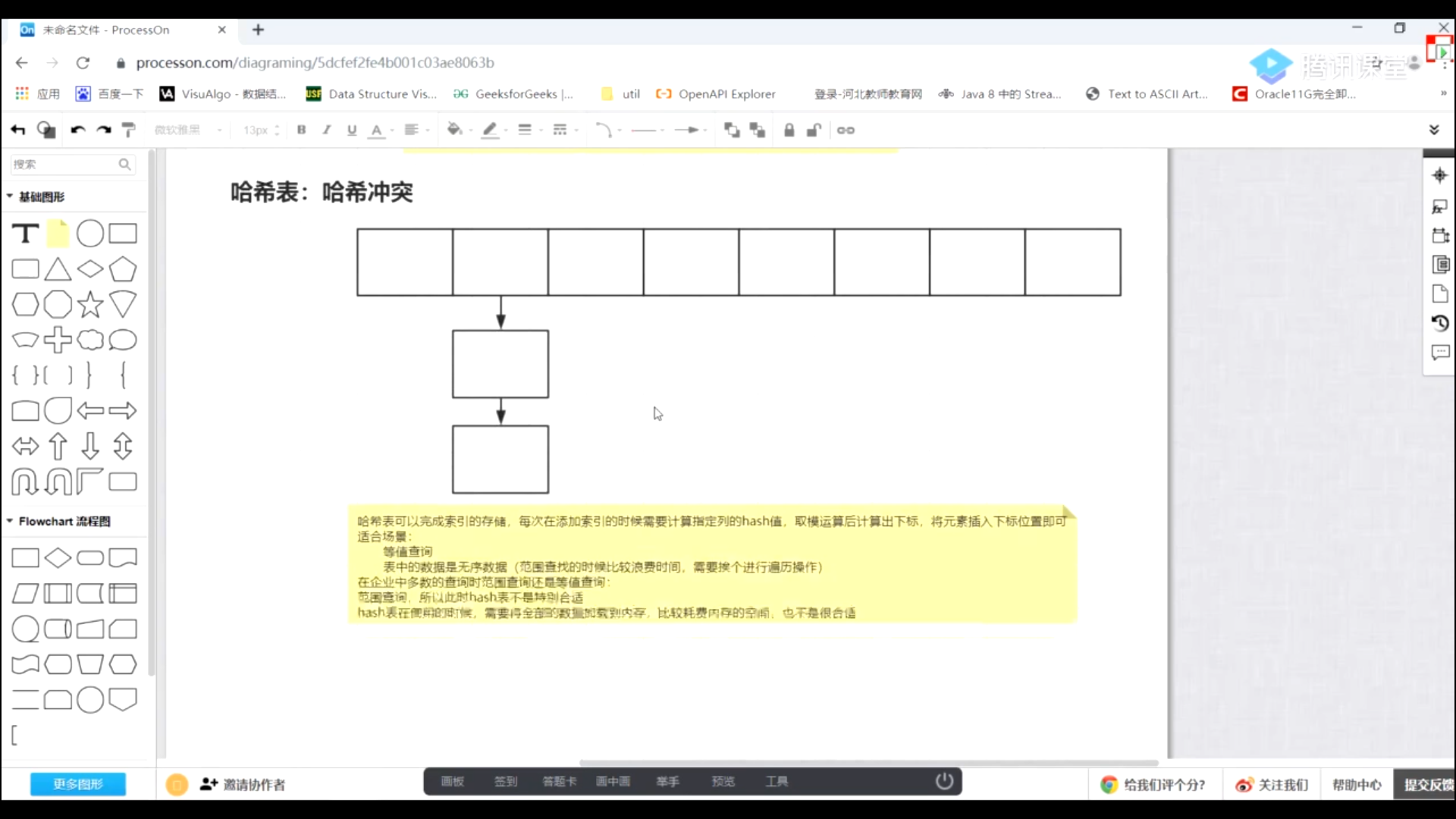
**（树的结构永远都是根大于左子树，小于右子树）**

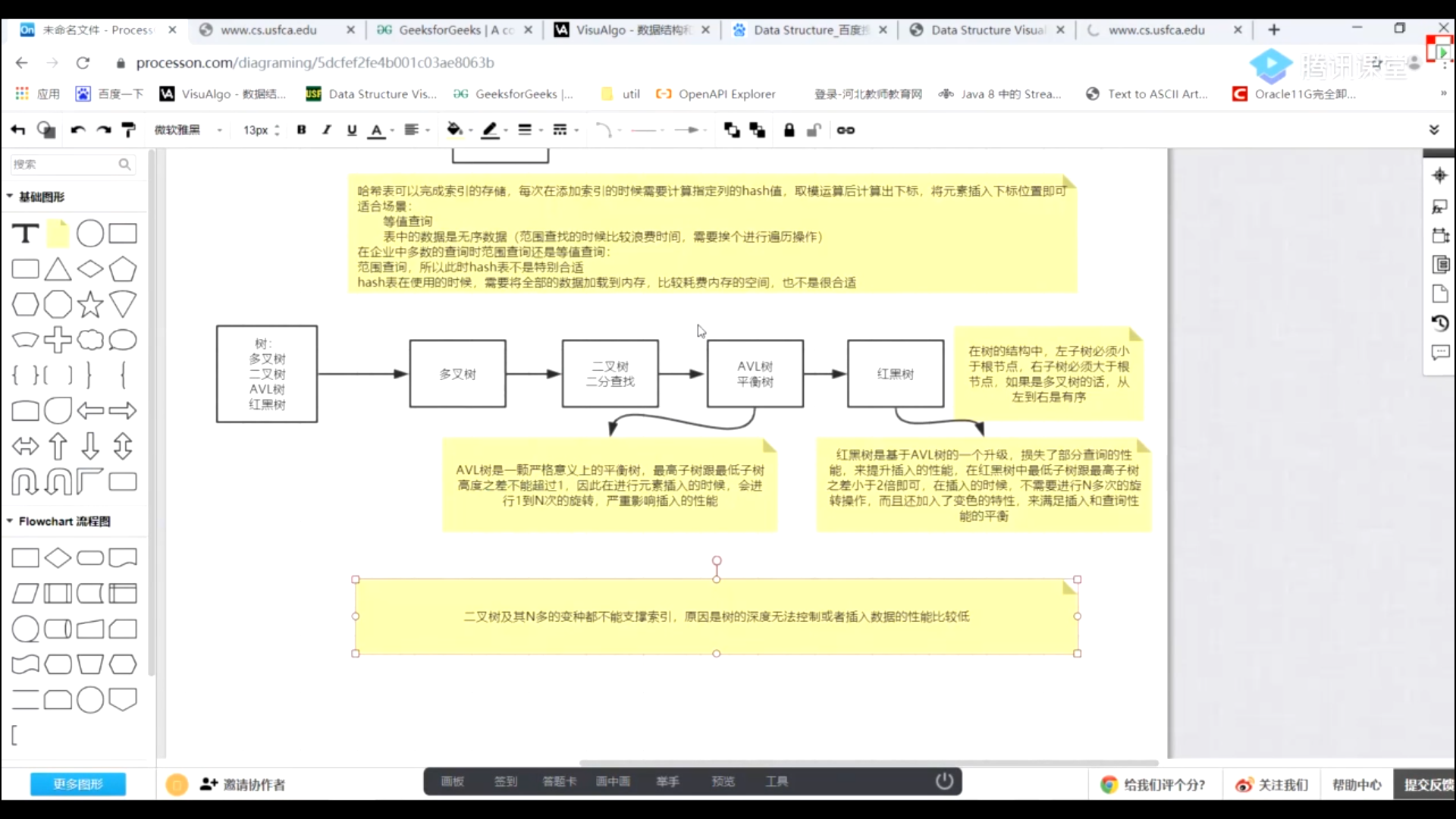
**数据库的索引是存放在磁盘中的**





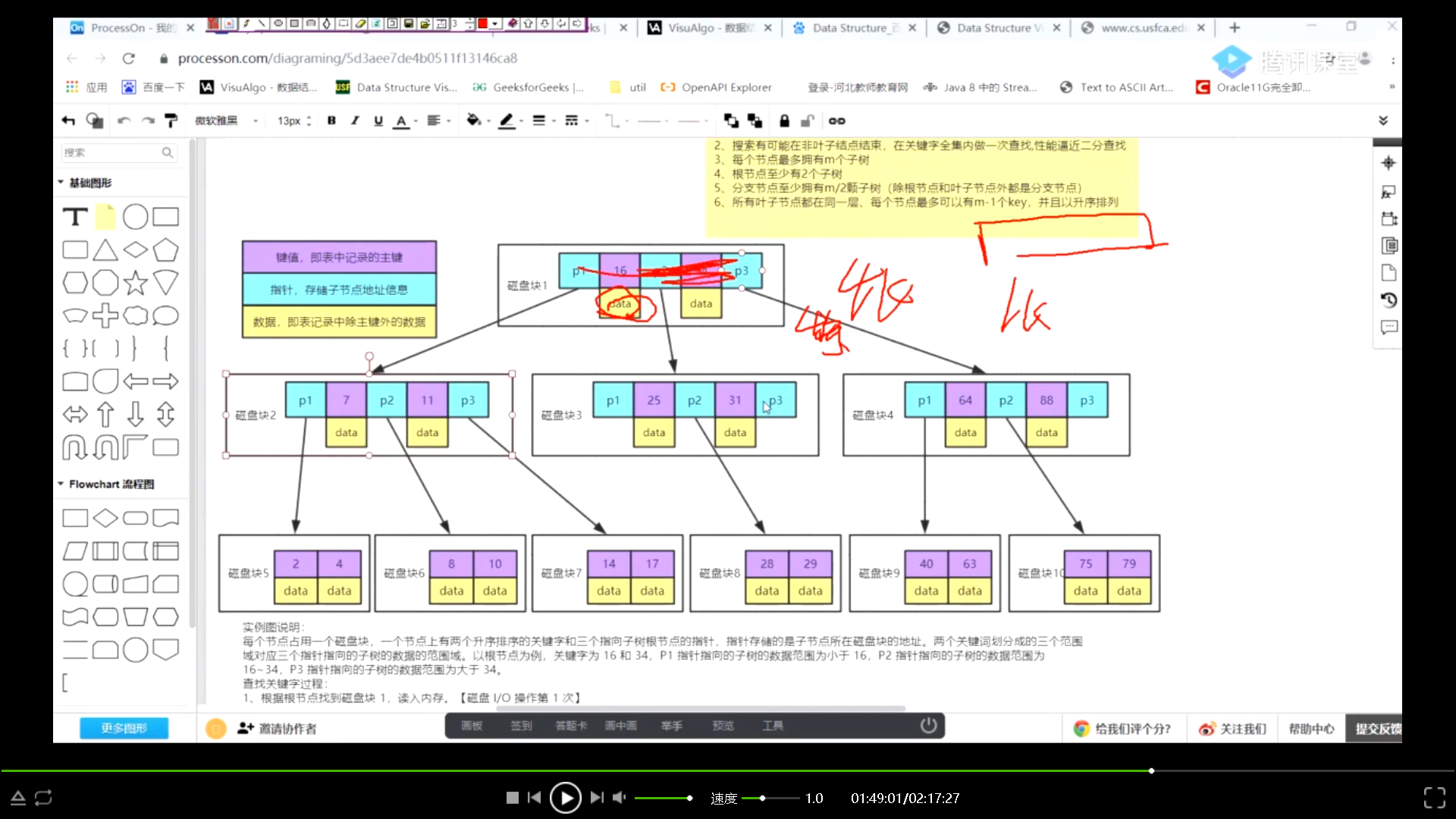


**为什么二叉树极其N多的树形变种都不足以支撑数据库的索引？**



**B树及B+树概念：**

**B树又叫B-树（B杠树）：（自我理解）**

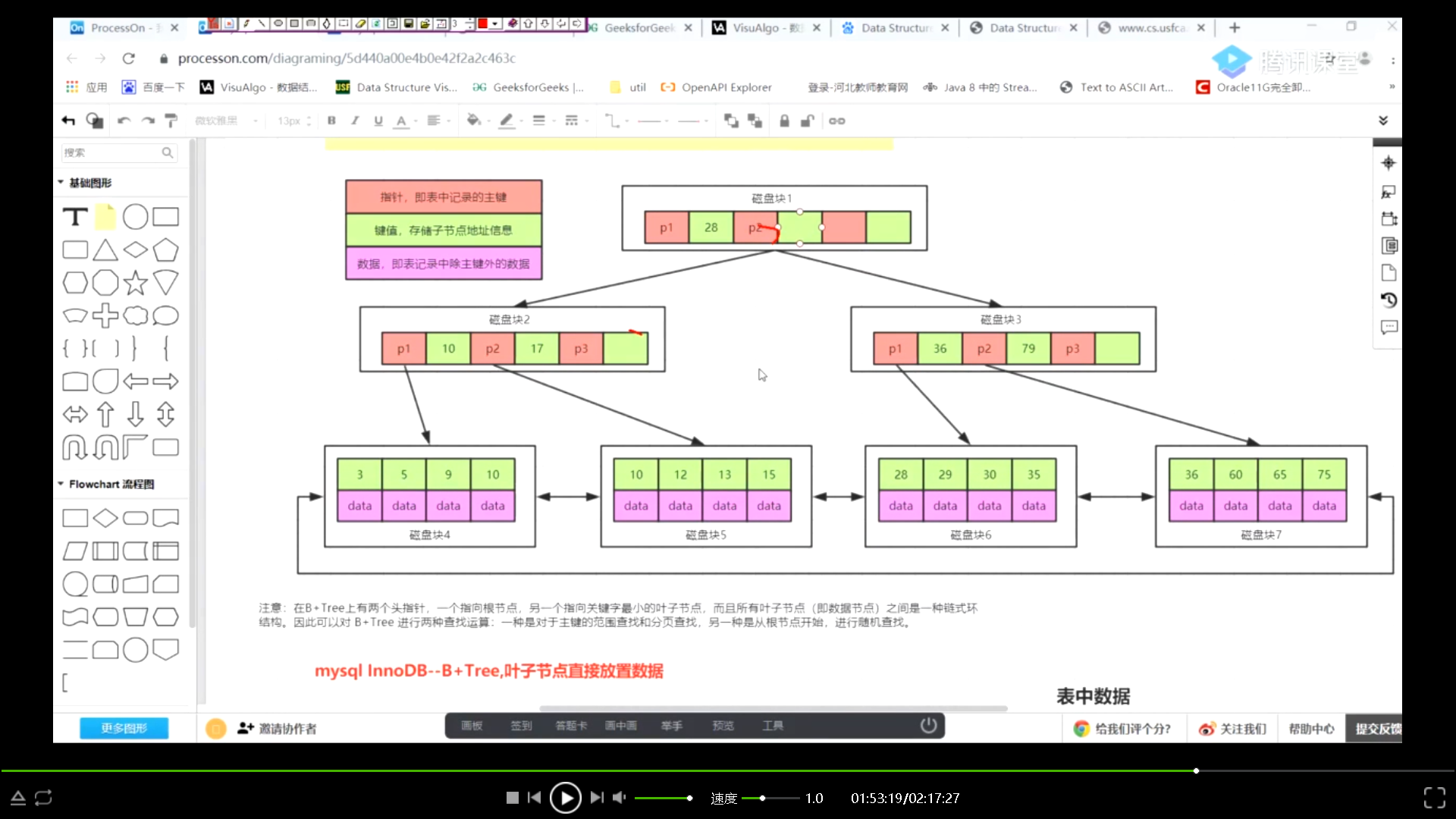


如图：B-树其实就是多叉树，但是它的非叶子节点也存储了数据，存储的数据两边是指向其他数据的指针。可以作为数据库的索引，但是在数据存储量非常大的情况下会增大查询的代价，所以被淘汰了。

总结：为什么会被淘汰，因为存储的数据量大时，会增加树的高度，问题就接踵而至。

**B+树概念：（改良，降低树高）**

存储数据的时候非叶子结点不存储数据，数据全部放在叶子结点中存储，降低了树高。也就代表着可以存放更多的数据，在非叶子结点中存储的索引也就更多、分的更加详细，在后期查询数据的时候速度也就越快。



注意：在B+tree上有两个头指针，一个指向根节点，一个指向关键字最小的叶子结点，而且所有的叶子结点（即数据结点）是一种链式环结构，因此可以对B+tree上进行两种查找运算：一种是对于主键的范围查找和分页查找，另一种是从根节点开始，进行随机查找。

B-树图中的非叶子结点使用了键值对的一个存储方式，即有key，但是同时存储了对应的value值。

B+树中的非叶子结点有指针也有赋值，但是没有存储对应的value值，所有的值都存放在叶子结点中，且叶子结点相互也有指针，满足范围查找，

回表：

查询数据所经历的两颗B+tree的这个过程就叫做**回表**

