

此图中，最上部分是数据页的存储结构。

下半部分是表中有索引，这里就出现了B-Tree结构，索引的根级会引用索引的下一级，直到索引的最后一级，这一级引用的对象是RID(当表中不存在聚集索引时，RID会指向每一行数据存储位置，RID的字段长度为16字节)或聚集索引列(当表存在索引时，聚集索引指向每一行数据存储位置)。

如果一个表每行存储200个字符，那么一个8Kb页面最多存储8060/200=40行数据。如果索引的字段是20个字节，表中聚集索引字段为16个字节或者没有聚集索引，那么索引行的长度为20+16=36个字节，每一个8Kb索引页存储索引的行数为8060/20=223行，也就是说：当数据小于223行时，索引只需要1个8Kb页面，而数据则需要223/40=6个8Kb页面。当数据大于223行时，索引页会分页，分成两个索引页，此时，会出现更高一级的索引节点，该节点引用它的下一级节点。此时，索引有2级，根级和子页级。根级保存的是对子页级的引用，此时，保存的数据最多为223\*223行，如果数据继续增加，那么根级会再次分页并变成第一子页级，同时生成新的根级，也就是上面图的结果，这个时候索引有3级，存储的最大数据为223\*223\*223。

当需要查询一行数据所有的列时，查询分析器只需要扫描4次就可以找数据。

注意：

1. 当表存在聚集索引时，并且只需要查询聚集索引键值，而查询条件使用到索引时，那么查询一次只需要扫描3次则可以找到数据。因为不需要扫描数据页级。如果查询的是整行，则需要扫描4次，多出来的一次是扫描数据页。
2. 当表不存在聚集索引时，查询某列(非索引列)或整行数据时，查询一次需要扫描4次才能找到数据，因为需要扫描数据页。