

回表查询对性能的损害以及覆盖索引是什么？

通过之前的学习都知道，一般我们自己建的索引不管是单列索引还是联合索引，其实一个索引就对应着一颗独立的索引B+树，索引B+树的节点仅仅包含了索引里的几个字段的值以及主键值。

即使我们根据索引树按照条件找到了需要的数据，那也仅仅是索引里的几个字段的值和主键值，万一你搞了一个select *还需要很多其他的字段，那还得走一个回表操作，根据主键跑到主键的聚簇索引里去，聚簇索引的叶子节点是数据页，找到数据页里才能把一行数据的所有字段值提取出来。

所以其实大家可以思考一下，假设你是类似select * from table order by xx1,xx2,xx3的语句，可能你就是得从联合索引的索引树里按照顺序取出来所有数据，接着对每一条数据都走一个主键的聚簇索引的查找，其实性能也是不高的。

有的时候MySQL的执行引擎甚至可能会认为，你要是类似select * from table order by xx1,xx2,xx3的语句，相当于是得把联合索引和聚簇索引，两个索引的所有数据都扫描一遍了，那还不如就不走联合索引了，直接全表扫描得了，这样还就扫描一个索引而已。

但是你如果要是select * from table order by xx1,xx2,xx3 limit 10这样的语句，那执行引擎就知道了，你先扫描联合索引的索引树拿到10条数据，接着对10条数据在聚簇索引里查找10次就可以了，那么还是会走联合索引的。

所以说，上述原理大家首先得先知晓一下。

其次的话，就是给大家讲解一个覆盖索引的概念，其实覆盖索引不是一种索引，他就是一种基于索引查询的方式罢了。

他的意思就是针对类似select xx1,xx2,xx3 from table order by xx1,xx2,xx3这样的 语句，这种情况下，你仅仅需要联合索引里的几个字段的值，那么其实就只要扫描联合索引的索引树就可以了，不需要回表去聚簇索引里找其他字段了。

所以这个时候，需要的字段值直接在索引树里就能提取出来，不需要回表到聚簇索引，这种查询方式就是覆盖索引。

也正是这样，所以在写SQL语句的时候，一方面是你要注意一下也许你会用到联合索引，但是是否可能会导致大量的回表到聚簇索引，如果需要回表到聚簇索引的次数太多了，可能就直接给你做成全表扫描不走联合索引了；

一方面是尽可能还是在SQL里指定你仅仅需要的几个字段，不要搞一个select *把所有字段都拿出来，甚至最好是直接走覆盖索引的方式，不要去回表到聚簇索引。

即使真的要回表到聚簇索引，那你也尽可能用limit、where之类的语句限定一下回表到聚簇索引的次数，就从联合索引里筛选少数数据，然后再回表到聚簇索引里去，这样性能也会好一些。

好了，到这里为止，关于索引本身的工作原理以及SQL语句怎么写才能用上索引，就给大家都讲清楚了，下一讲我们给大家说说平时设计索引的时候，一些通用的原则，如何选择索引，如何设计索引。

End