## 索引设计原则

### 代码先行，索引后上

不知大家一般是怎么给数据表建立索引的，是建完表马上就建立索引吗？

这其实是不对的，一般应该等到主体业务功能开发完毕，把涉及到该表相关sql都要拿出来分析之后再建立索引。

### **联合索引尽量覆盖条件**

比如可以设计一个或者两三个联合索引(尽量少建单值索引)，让每一个联合索引都尽量去包含sql语句里的where、order by、group by的字段，还要确保这些联合索引的字段顺序尽量满足sql查询的最左前缀原则。

### **不要在小基数字段上建立索引**

索引基数是指这个字段在表里总共有多少个不同的值，比如一张表总共100万行记录，其中有个性别字段，其值不是男就是女，那么该字段的基数就是2。

如果对这种小基数字段建立索引的话，还不如全表扫描了，因为你的索引树里就包含男和女两种值，根本没法进行快速的二分查找，那用索引就没有太大的意义了。

一般建立索引，尽量使用那些基数比较大的字段，就是值比较多的字段，那么才能发挥出B+树快速二分查找的优势来。

### **长字符串我们可以采用前缀索引**

尽量对字段类型较小的列设计索引，比如说什么tinyint之类的，因为字段类型较小的话，占用磁盘空间也会比较小，此时你在搜索的时候性能也会比较好一点。当然，这个所谓的字段类型小一点的列，也不是绝对的，很多时候你就是要针对varchar(255)这种字段建立索引，哪怕多占用一些磁盘空间也是有必要的。

对于这种varchar(255)的大字段可能会比较占用磁盘空间，可以稍微优化下，比如针对这个字段的前20个字符建立索引，就是说，对这个字段里的每个值的前20个字符放在索引树里，类似于 KEYindex(name(20),age,position)。

此时你在where条件里搜索的时候，如果是根据name字段来搜索，那么此时就会先到索引树里根据name字段的前20个字符去搜索，定位到之后前20个字符的前缀匹配的部分数据之后，再回到聚簇索引提取出来完整的name字段值进行比对。

但是假如你要是order by name，那么此时你的name因为在索引树里仅仅包含了前20个字符，所以这个排序是没法用上索引的， group by也是同理。所以这里大家要对前缀索引有一个了解。

### **where与order by冲突时优先where**

在where和order by出现索引设计冲突时，到底是针对where去设计索引，还是针对order by设计索引？到底是让where去用上索引，还是让order by用上索引?

一般这种时候往往都是让where条件去使用索引来快速筛选出来一部分指定的数据，接着再进行排序。

因为大多数情况基于索引进行where筛选往往可以最快速度筛选出你要的少部分数据，然后做排序的成本可能会小很多。

### **基于慢sql查询做优化**

可以根据监控后台的一些慢sql，针对这些慢sql查询做特定的索引优化。

关于慢sql查询不清楚的可以参考这篇文章：<https://blog.csdn.net/qq_40884473/article/details/89455740>

## Sql优化总结

1、MySQL支持两种方式的排序filesort和index，Using index是指MySQL扫描索引本身完成排序。**index效率高，filesort效率低。**

2、order by满足两种情况会使用Using index。

1) order by语句使用索引最左前列。

2) 使用where子句与order by子句条件列组合满足索引最左前列。

3、尽量在索引列上完成排序，遵循索引建立（索引创建的顺序）时的最左前缀法则。

4、如果order by的条件不在索引列上，就会产生Using filesort。

5、能用覆盖索引尽量用覆盖索引

6、group by与order by很类似，其实质是先排序后分组，遵照索引创建顺序的最左前缀法则。对于group by的优化如果不需要排序的可以加上order by null禁止排序。注意，where高于having，能写在where中的限定条件就不要去having限定了。