

order by

背景

```
select city,name,age from t where city='杭州' order by name limit 1000 ;
```

全字段排序

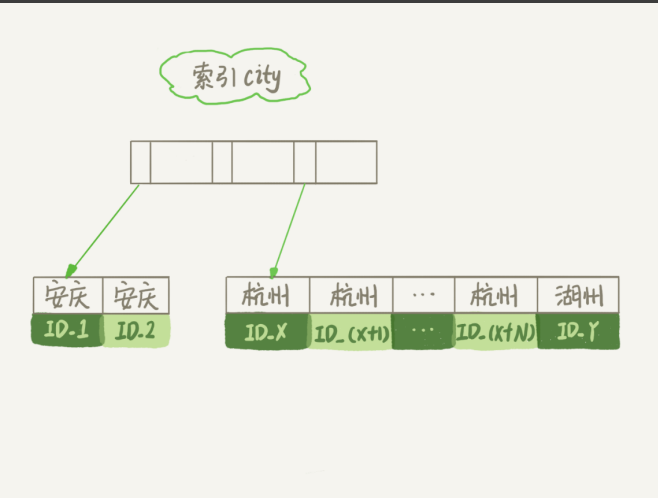
city字段上加上索引，执行explain

```
mysql> explain select city, name,age from T where city='杭州' order by name limit 1000;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	T	NULL	ref	city	city	51	const	4000	100.00	Using index condition; Using filesort

Extra字段中的Using filesort表示需要排序

MySQL会给每个线程分配一块内存用于排序，称为sort_buffer



city索引示意图

上面的语句大致执行流程

- 初始化sort_buffer，确定放入name、city、age这三个字段
- 从索引city找到第一个满足city='杭州'条件的主键id，也就是ID_X
- 到主键id索引取出整行，去name、city、age三字段的价值，存入sort_buffer中
- 从索引city取下一个记录的主键id
- 重复步骤3、4直到city值不满足查询条件为止，对应的主键id也就是图中的ID_Y
- 对sort_buffer中的数据按照字段name做快速排序
- 按照排序结果取前1000行返回给客户端

排序的方式

- sort_buffer_size的大小大于需要排序的数量，在内存中进行排序，
- sort_buffer_size的大小小于需要排序的数量，在外部排序，利用临时文件进行归并排序

问题

如果查询的返回字段很多，那么sort_buffer里面需要放的字段数太多，这样内存里面能够同时放下的行数很少，要分成很多个临时文件，排序的性能就会很差

rowid排序

max_length_for_sort_data，是MySQL中专门控制用于排序的行数据的长度的一个参数。它的意思是，如果单行的长度超过这个值，MySQL就认为单行太大，要换成rowid排序

city、name、age这三个字段的定义总长度是36，我把max_length_for_sort_data设置为16，那么使用rowid排序放入sort_buffer的字段，只有要排序的列（name）和主键id

大致执行流程

- 初始化sort_buffer，确定放入两个字段，name和id
- 从索引city中找到第一个满足city='杭州'的主键id，也就是ID_X
- 从id索引中取出整行，取name，id两个字段，存入sort_buffer中
- 从索引city中去下一个记录的主键id
- 重复步骤3、4直到不满足city='杭州'条件为止，也就是ID_Y
- 对sort_buffer中的数据按照name进行排序
- 遍历排序结果，取前1000行，并按照id的值回到原表中取出city、name、age三个字段返回给客户端

对比全字段排序，多了步骤7，也就是多访问了一次主键索引

全字段排序 VS rowid 排序

如果MySQL实在是担心排序内存太小，会影响排序效率，才会采用rowid排序算法，这样排序过程中一次可以排序更多行，但是需要再回到原表去取数据。

如果MySQL认为内存足够大，会优先选择全字段排序，把需要的字段都放到sort_buffer中，这样排序后就会直接从内存里面返回查询结果了，不用再回到原表去取数据。

体现了MySQL的一个设计思想：如果内存够，就要多利用内存，尽量减少磁盘访问

对于InnoDB表来说，rowid排序会要求回表多造成磁盘读，因此不会被优先选择。

并不是所有的order by语句都需要排序，对应天然有序的结构就不需要进行排序

并不是说要为了每个查询能用上覆盖索引，就要把语句中涉及的字段都建上联合索引，毕竟索引还是有维护代价的

建立city和name的联合索引

执行流程就大致变成了

- 从索引(city.name)找到一个满足city='杭州'条件的主键id
- 到主键id索引中取出整行，取name、city、age三个字段，作为结果集的一部分直接返回
- 从索引(city.name)去下一个记录的主键id
- 重复步骤2、3，直到查到第1000条记录，或者是不满足city='杭州'条件时循环结束

上述流程中有一次回表的操作，可以利用覆盖索引来优化，建立city、name和age的联合索引

执行流程就大致变成了

- 从索引(city.name.age)中找到一个满足city='杭州'条件的记录，取出其中的city、name、age这三个字段的值，作为结果集的一部分返回
- 从索引(city.name.age)取下一条记录，同样取这三个字段的值，作为结果集的一部分直接返回
- 重复执行步骤2，直到查到第1000条记录，或者是不满足city='杭州'条件时循环结束