

mysql SQL优化

1. 查询多的字段上建立索引
2. 默认情况下，MySQL对所有的GROUP BY字段进行排序，如果查询包括GROUP BY但是用户想要避免排序结果的消耗，则可以指定ORDER BY NULL禁止排序。
3. 使用like的查询，后面如果是常量并且只有%号不在第一个字符，索引才可能被使用
4. 用or分割的条件，如果or前的条件中的列有索引，二后面的列没有索引，那么涉及的索引都不会用到。
5. 复合索引有最左原则，如果不是索引列的第一部分，如下例子：可见虽然在money上面建有复合索引，但是由于money不是索引的第一列，那么在查询中，这个索引也不会被mysql采用。
6. 如果列类型是字符串，但在查询时把第一个数值型常量赋值给了一个字符类型的列名name，那么虽然在name列上有索引，但是也没有用到。（因为这样MySQL会将表中字符串类型转换为数字之后再比较，导致索引失效）
7. 在某些情况中，mysql可以使用一个索引来满足order by 子句，而不需要额外的排序。where条件和order by使用相同的索引，并且order by的顺序和索引顺序相同，并且order by的字段都是升序或者降序。
8. 使用not exists代替not in，not in不会使用索引

```
EXPLAIN select * from user where age not in (12);
```

结果 1	剖析	状态						
select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1 SIMPLE	user	ALL	age	(Null)	(Null)	(Null)	6	Using where

9. 多个单列索引在多条件查询时只会生效第一个索引！所以多条件联合查询时最好建联合索引！

```
EXPLAIN select * from user where name = "zs" and age = 12;
```

只会用name的索引，age的索引不会用到

10. 应尽量避免在 where 子句中使用!=或<>操作符，否则引擎将放弃使用索引而进行全表扫描。
11. 应尽量避免在 where 子句中对字段进行表达式操作，这将导致引擎放弃使用索引而进行全表扫描。如：

```
select id from t where num/2=100
```

应改为:

```
select id from t where num=100*2
```