查看索引的使用情况

show status like ‘Handler\_read%’;

大家可以注意：

handler\_read\_key:这个值越高越好，越高表示使用索引查询到的次数

handler\_read\_rnd\_next:这个值越高，说明查询低效

但是如果是同样的sql如果在之前能够使用到索引，那么现在使用不到索引，以下几种主要情况:

1. 随着表的增长，where条件出来的数据太多，大于15%，使得索引失效（会导致CBO（查询优化器）计算走索引花费大于走全表）

2. 统计信息失效      需要重新搜集统计信息

3. 索引本身失效      需要重建索引

1. **如果条件中有or**(必须所有都加索引)，即使其中有条件带索引也不会使用(**这也是为什么尽量少用or的原因。要想使用or，又想让索引生效，只能将or条件中的每个列都加上索引**)
2. 对于多列索引，不是使用的第一部分，则不会使用索引（and的规则）
3. like查询是以%开头
4. 如果列类型是字符串，那一定要在条件中将数据使用引号引用起来,否则不使用索引
5. .如果mysql估计使用全表扫描要比使用索引快,则不使用索引

索引失效 （详细的一些索引失效的原因。）  
1) 没有查询条件，或者查询条件没有建立索引   
2) 在查询条件上没有使用引导列   
3) 查询的数量是大表的大部分，应该是30％以上。   
4) 索引本身失效   
5) 查询条件使用函数在索引列上（见12）   
6) 对小表查询   
7) 提示不使用索引   
8) 统计数据不真实   
9) CBO计算走索引花费过大的情况。其实也包含了上面的情况，这里指的是表占有的block要比索引小。   
10)隐式转换导致索引失效.这一点应当引起重视.也是开发中经常会犯的错误. 由于表的字段tu\_mdn定义为varchar2(20),   
但在查询时把该字段作为number类型以where条件传给Oracle,这样会导致索引失效.   
错误的例子：select \* from test where tu\_mdn=13333333333;   
正确的例子：select \* from test where tu\_mdn='13333333333';   
11)对索引列进行运算导致索引失效,我所指的对索引列进行运算包括(+，-，\*，/，! 等)   
错误的例子：select \* from test where id-1=9;   
正确的例子：select \* from test where id=10;   
12)**使用Oracle内部函数导致索引失效**.对于这样情况应当创建基于函数的索引.   
错误的例子：select \* from test where round(id)=10;   
说明，此时id的索引已经不起作用了 正确的例子：首先**建立函数索引，**  
create index test\_id\_fbi\_idx on test(round(id));   
然后 select \* from test where round(id)=10; 这时函数索引起作用了

1,<>

2,单独的>,<,(有时会用到，有时不会)   
3,like "%\_" 百分号在前.   
4,表没分析.   
5,单独引用复合索引里非第一位置的索引列.   
6,字符型字段为数字时在where条件里不添加引号.   
7,对索引列进行运算.需要建立函数索引.   
8,not in ,not exist.   
9,当变量采用的是times变量，而表的字段采用的是date变量时.或相反情况。   
10, 索引失效。   
11,基于cost成本分析(oracle因为走全表成本会更小)：查询小表,或者返回值大概在10%以上   
12,有时都考虑到了 但就是不走索引,drop了从建试试在   
13,B-tree索引 is null不会走,is not null会走,位图索引 is null,is not null 都会走   
14,联合索引 is not null 只要在建立的索引列（不分先后）都会走,   
in null时 必须要和建立索引第一列一起使用,当建立索引第一位置条件是is null 时,   
其他建立索引的列可以是is null（但必须在所有列 都满足is null的时候）,   
或者=一个值；当建立索引的第一位置是=一个值时,其他索引列可以是任何情况（包括is null =一个值）,   
以上两种情况索引都会走。其他情况不会走。