一、以下的方法会引起索引失效

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。

二、索引失效解决方法

1. 选用适合的 Oracle 优化器

Oracle 的优化器共有 3 种:

a. RULE (基于规则) b. COST (基于成本) c. CHOOSE (选择性)。

设置缺省的优化器,可以通过对 init.ora 文件中 OPTIMIZER_MODE 参数的各种声明,如 RULE，COST，CHOOSE，ALL_ROWS，FIRST_ROWS 。你当然也在 SQL 句级或是会话(session)级对其进行覆盖。

为了使用基于成本的优化器(CBO, Cost-Based Optimizer) ，你必须经常运行 analyze 命令,以增加数据库中的对象统计信息(object statistics)的准确性。

如果数据库的优化器模式设置为选择性(CHOOSE),那么实际的优化器模式将和是否运行过 analyze 命令有关。如果 table 已经被 analyze 过，优化器模式将自动成为 CBO ，反之,数据库将采用 RULE 形式的优化器。

（分析 table

analyze table PROD_PARTS compute statistics;

ANALYZE TABLE PROD_PARTS COMPUTE STATISTICS FOR ALL INDEXED COLUMNS;

analyze table PROD_PARTS compute statistics for table for all indexes for all indexed columns;

）【有一次索引失效之后，请教 DBA 后，发现是数据统计的问题，具体的解决办法是执行以上语句】

在缺省情况下,Oracle 采用 CHOOSE 优化器, 为了避免那些不必要的全表扫描（full table scan），你必须尽量避免使用 CHOOSE 优化器,而直接采用基于规则或者基于成本的优化器。

2、重建索引

alter index 索引名 rebuild 【online】

3、强制索引

给该语句加上 hint 后，强制其使用'RECORD_ENTITYID' 这个索引

sql 语句变成这样

引用

select /*+ index(record,record_entityid) */ *

from RECORD

where entityId='24' and entityType='blog';

/*+ index(record,record_entityid) */ 中，index 表示强制使用 index，record 是表名，record_entityid 是索引名。其执行计划跟测试数据库上一致，都是使用用 'RECORD_ENTITYID' 这个索引，逻辑读写同样为 4。

后来经过测试，在不加 hint 的情况下，对该表和两个索引执行 analyze 后，同样也能使用 'RECORD_ENTITYID' 这个索引。但是因为该表更新颇为频繁，不知道要多久就要再分析一次

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:索引失效的原因

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 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

下面是一些不会使用到索引的原因

索引失效

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