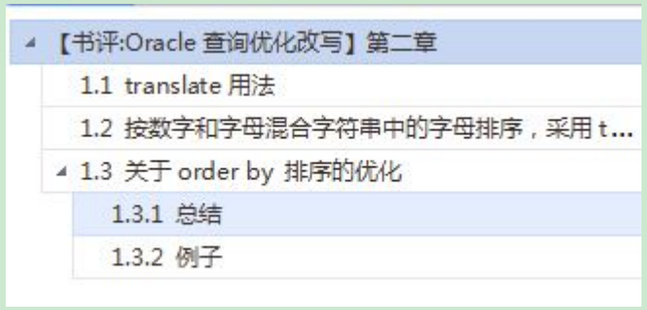
【书评:Oracle 查询优化改写】第二章

BLOG 文档结构图



在上一篇中 **http://blog.itpub.net/26736162/viewspace-1652985/**，我们主要分析了一些**单表查询**的时候需要注意的内容，今天第二章也很简单，主要是关于排序方面的内容，以下贴出第二章的内容：

排序基本上没有什么可以讲的，不过书中着重介绍了下 translate 的用法。

1. 1 translate 用法

语法：TRANSLATE(char, from, to)

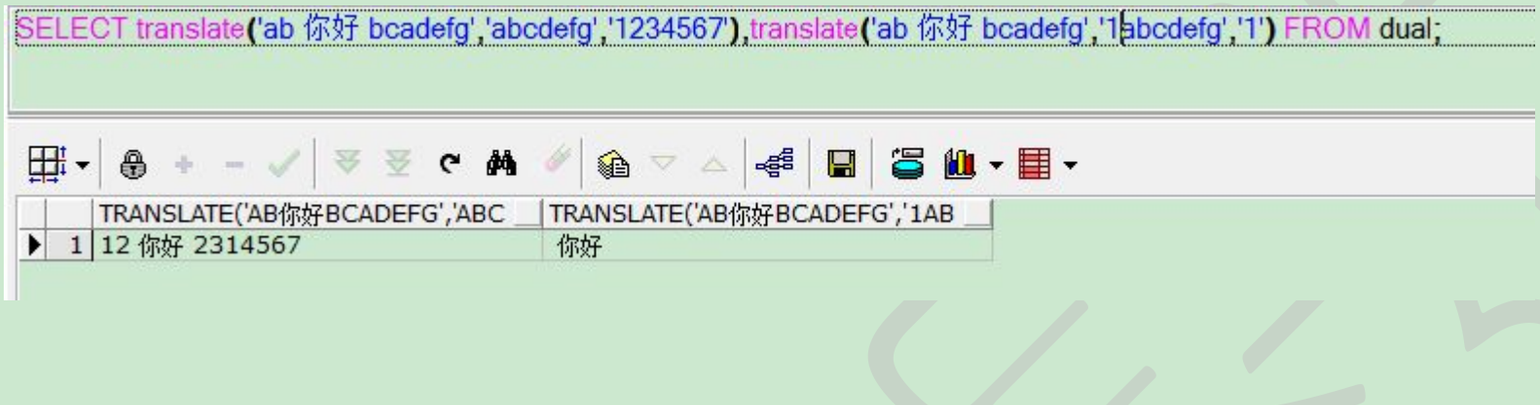
用法：

1. 返回将出现在 from 中的每个字符替换为 to 中的相应字符以后的字符串。
2. 若 from 比 to 字符串长，那么在 from 中比 to 中多出的字符将会被删除，或者认为 from 中多出的字符在 to 中与空对应

3. 三个参数中有一个是空，返回值也将是空值。

```
09:43:50 SQL>  select translate('abcdefga','abc','wo')  from dual;

TRANSLA
-------
wodefgw

Elapsed: 00:00:00.14
09:43:57 SQL>  select translate('abcdefga','abc','')  from dual;

T
-


Elapsed: 00:00:00.00
```

```
SELECT translate('ab 你好 bcadefg','abcdefg','1234567'),translate('ab 你好 bcadefg','1abcdefg','1') FROM dual;
```



1.2 按数字和字母混合字符串中的字母排序，采用 translate 函数来实现

```
09:52:01 SQL> create or replace view v as select empno || ' '||ename as data from scott.emp;

View created.

Elapsed: 00:00:00.54
09:52:07 SQL> select * from V
09:52:15   2  ;

DATA
--------------------------------------------------
9000 lastwiner
9001 lastwiner
7369 SMITH
7499 ALLEN
7521 WARD
7566 JONES
7654 MARTIN
7698 BLAKE
7782 CLARK
7788 SCOTT
7839 KING
7844 TURNER
7876 ADAMS
7900 JAMES
```

```
7902 FORD
7934 MILLER

16 rows selected.

Elapsed: 00:00:00.20

09:55:07 SQL> select data,translate(data,'- 0123456789','-') from V order by 2;

DATA                                               TRANSLATE(DATA,'-0123456789','-')
-------------------------------------------------- --------------------------------------------------------------------------------
7876 ADAMS                                         ADAMS
7499 ALLEN                                         ALLEN
7698 BLAKE                                         BLAKE
7782 CLARK                                         CLARK
7902 FORD                                          FORD
7900 JAMES                                         JAMES
7566 JONES                                         JONES
7839 KING                                          KING
7654 MARTIN                                        MARTIN
7934 MILLER                                        MILLER
7788 SCOTT                                         SCOTT
7369 SMITH                                         SMITH
7844 TURNER                                        TURNER
7521 WARD                                          WARD
9001 lastwiner                                     lastwiner
9000 lastwiner                                     lastwiner

16 rows selected.

Elapsed: 00:00:00.10
09:55:33 SQL>
```

1.3 关于 order by 排序的优化

关于 SQL 优化中有一个原则叫：避免使用耗费资源的操作(DISTINCT、UNION、MINUS、INTERSECT、ORDER BY、group by、SMJ、created index)

带有 DISTINCT,UNION,MINUS,INTERSECT,ORDER BY 的 SQL 语句会启动 SQL 引擎执行耗费资源的排序(SORT)功能. DISTINCT 需要一次排序操作，而其他的至少需要执行两次排序.

例如,一个 UNION 查询,其中每个查询都带有 GROUP BY 子句, GROUP BY 会触发嵌入排序(NESTED SORT)；这样，每个查询需要执行一次排序，然后在执行 UNION 时，又一个唯一排序(SORT UNIQUE)操作被执行而且它只能在前面的嵌入排序结束后才能开始执行. 嵌入的排序的深度会大大影响查询的效率.

通常，带有 UNION, MINUS , INTERSECT 的 SQL 语句都可以用其他方式重写.

ORDER BY 语句决定了 Oracle 如何将返回的查询结果排序。Order by 语句对要排序的列没有什么特别的限制，也可以将函数加入列中（象联接或者附加等）。任何在 Order by 语句的非索引项或者有计算表达式都将降低查询速度。

仔细检查 order by 语句以找出非索引项或者表达式，它们会降低性能。解决这个问题的办法就是重写 order by 语句以使用索引，也可以为所使用的列建立另外一个索引，同时应绝对避免在 order by 子句中使用表达式。

●在频繁进行排序或分组（即进行 group by 或 order by 操作）的列上建立索引。

●如果待排序的列有多个，可以在这些列上建立复合索引（compound index）。

磁盘排序的开销是很大的，有几个方面的原因。首先，和内存排序相比较，它们特别慢；而且磁盘排序会消耗临时表空间中的资源。Oracle 还必须分配缓冲池块来保持临时表空间中的块。无论什么时候，内存排序都比磁盘排序好，磁盘排序将会令任务变慢，并且会影响 Oracle 实例的当前任务的执行。还有，过多的磁盘排序将会令 free buffer waits 的值变高，从而令其它任务的数据块由缓冲中移走。

1.3.1 总结

（1）采用索引避免排序：排序数据较多时
（2）去掉不必要的 distinct，很多 distinct 是由于程序员对数据的了解不自信而多加的。

总而言之，排序是非常耗费 CPU 资源的，能不排序就不要排序，如果非得排序，可以考虑在排序列上建立合适的索引。

记得之前有个 SQL，不加排序的话，秒级可以出结果，即响应速度很快，但是加上排序后得 5 或 6 分钟才可以，看了下是结果集很大，又得排序造成的。

这里简单举个例子吧：

1.3.2 例子

```
[oracle@rhel6_lhr ~]$ sqlplus / as sysdba
```

```
SQL*Plus: Release 11.2.0.3.0 Production on Thu May 14 10:55:26 2015

Copyright (c) 1982, 2011, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining
and Real Application Testing options

10:55:26 SQL> conn lhr/lhr
Connected.
12:08:08 SQL> create table test_index_lhr as select * from dba_objects;

Table created.

Elapsed: 00:00:03.70
12:08:27 SQL> insert into test_index_lhr select * from test_index_lhr;

77241 rows created.

12:08:39 SQL> commit;

Commit complete.

Elapsed: 00:00:00.00

12:08:41 SQL> set autot traceonly explain stat
12:08:41 SQL> select  object_name from test_index_lhr where object_name is not null order by object_name;

154482 rows selected.

Elapsed: 00:00:01.18

Execution Plan
----------------------------------------------------------
Plan hash value: 1466335622

-------------------------------------------------------------------------------------------------
| Id  | Operation          | Name           | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |                |   155K|     9M|       |  3078   (1)| 00:00:37 |
|   1 |  SORT ORDER BY     |                |   155K|     9M|    10M|  3078   (1)| 00:00:37 |
|*  2 |   TABLE ACCESS FULL| TEST_INDEX_LHR |   155K|     9M|       |   623   (1)| 00:00:08 |
-------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - filter("OBJECT_NAME" IS NOT NULL)

Note
-----
   - dynamic sampling used for this statement (level=2)


Statistics
----------------------------------------------------------
         10  recursive calls
          6  db block gets
       2808  consistent gets
        614  physical reads
      34996  redo size
    3787521  bytes sent via SQL*Net to client
     113801  bytes received via SQL*Net from client
      10300  SQL*Net roundtrips to/from client
          1  sorts (memory)
```

```
          0  sorts (disk)
     154482  rows processed

12:08:48 SQL> create index ind_test_inde on test_index_lhr(object_name) ;

Index created.

Elapsed: 00:00:02.45
12:08:58 SQL> EXEC dbms_stats.gather_table_stats(ownname => 'LHR', tabname=> 'test_index_lhr',   cascade => TRUE );

PL/SQL procedure successfully completed.

Elapsed: 00:00:02.62
12:09:04 SQL> select  object_name from test_index_lhr where object_name is not null order by object_name;

154482 rows selected.

Elapsed: 00:00:01.35

Execution Plan
----------------------------------------------------------
Plan hash value: 712275200

-------------------------------------------------------------------------------
| Id  | Operation        | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT |               |   154K|  3771K|   766   (1)| 00:00:10 |
|*  1 |  INDEX FULL SCAN | IND_TEST_INDE |   154K|  3771K|   766   (1)| 00:00:10 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("OBJECT_NAME" IS NOT NULL)

Note
-----
   - SQL plan baseline "SQL_PLAN_8kcy12j8f3s5n2a6f2b1f" used for this statement


Statistics
----------------------------------------------------------
        704  recursive calls
         64  db block gets
      11715  consistent gets
         37  physical reads
      33236  redo size
    3787521  bytes sent via SQL*Net to client
     113801  bytes received via SQL*Net from client
      10300  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
     154482  rows processed

12:09:09 SQL> select  owner, object_name from test_index_lhr where object_name is not null order by object_name;

154482 rows selected.

Elapsed: 00:00:01.28

Execution Plan
----------------------------------------------------------
Plan hash value: 1466335622

-----------------------------------------------------------------------------------------------
| Id  | Operation          | Name          | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
```

```
|   0 | SELECT STATEMENT   |                | 154K|  4676K|       | 1947   (1)| 00:00:24 |
|   1 |  SORT ORDER BY     |                | 154K|  4676K|  6072K| 1947   (1)| 00:00:24 |
|*  2 |   TABLE ACCESS FULL| TEST_INDEX_LHR | 154K|  4676K|       |  623   (1)| 00:00:08 |
---------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - filter("OBJECT_NAME" IS NOT NULL)


Statistics
----------------------------------------------------------
          6  recursive calls
          6  db block gets
       2241  consistent gets
        895  physical reads
        680  redo size
    4232382  bytes sent via SQL*Net to client
     113801  bytes received via SQL*Net from client
      10300  SQL*Net roundtrips to/from client
          1  sorts (memory)
          0  sorts (disk)
     154482  rows processed

12:09:22 SQL> select /*+index(a,IND_TEST_INDE)*/ owner,  object_name from test_index_lhr a where object_name is not null order by object_name;

154482 rows selected.

Elapsed: 00:00:09.59

Execution Plan
----------------------------------------------------------
Plan hash value: 880046030

-----------------------------------------------------------------------------------------------
| Id  | Operation                   | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |                |   154K|  4676K|   109K  (1)| 00:21:57 |
|   1 |  TABLE ACCESS BY INDEX ROWID| TEST_INDEX_LHR |   154K|  4676K|   109K  (1)| 00:21:57 |
|*  2 |   INDEX FULL SCAN           | IND_TEST_INDE  |   154K|       |   766   (1)| 00:00:10 |
-----------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - filter("OBJECT_NAME" IS NOT NULL)


Statistics
----------------------------------------------------------
          6  recursive calls
          4  db block gets
     122955  consistent gets
       2198  physical reads
        724  redo size
    4392715  bytes sent via SQL*Net to client
     113801  bytes received via SQL*Net from client
      10300  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
     154482  rows processed

12:14:23 SQL>
```

相关连接：

【书评:Oracle 查询优化改写】第一章 http://blog.itpub.net/26736162/viewspace-1652985/

.....

本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用

ITPUB BLOG：http://blog.itpub.net/26736162

本文地址：http://blog.itpub.net/26736162/viewspace-1654252/

本文pdf版：http://yunpan.cn/QCwUAI9bn7g7w 提取码：af2d

QQ：642808185 注明：ITPUB 的文章标题

.....