【书评:Oracle 查询优化改写】第三章

BLOG 文档结构图

1.1 导读

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~O(∩_∩)O~：

① 隐含参数 _b_tree_bitmap_plans 介绍

② 11g 新特性 Native Full Outer Join

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

1.2 实验环境介绍

oracle：11.2.0.3 、8.1.7.0.0

OS: RHEL6.5

1.3 前言

前 2 章的链接参考相关连接：

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

【书评:Oracle 查询优化改写】第二章 http://blog.itpub.net/26736162/viewspace-1654252/

昨天晚上(5.14)看完了《Oracle 查询优化改写》的第三章，不得不说下这本书里边代码的排版有很大问题，格式老是不对齐，尤其是执行计划的格式，可能是印刷的时候出现的问题吧，不说这个了。这个第三章主要是讲多表的关联，包括各种连接的写法，如左联、右联，以及过滤条件错误地放在 WHERE 里会有什么影响；当数据有重复值时要直接关联还是分组汇总后再关联。

第 3 章 操作多个表

3.1 UNION ALL 与空字符串

3.2 UNION 与 OR

3.3 组合相关的行

3.4 IN、EXISTS 和 INNER JOIN

3.5 INNER JOIN、LEFT JOIN、RIGHT JOIN 和 FULL JOIN 解析

3.6 自关联

3.7 NOT IN、NOT EXISTS 和 LEFT JOIN

3.8 外连接中的条件不要乱放

3.9 检测两个表中的数据及对应数据的条数是否相同

3.10 聚集与内连接

3.11 聚集与外连接

3.12 从多个表中返回丢失的数据

3.13 多表查询时的空值处理

下边我就针对一些重点，或者说是我自己也不是很懂的部分做做研究吧。

1.4 隐含参数 _b_tree_bitmap_plans 实验

1.4.1 简介

该参数为隐含参数，是指是否将索引转换为 bitmap 索引然后执行，在 oracle9i 之前默认值为 false，之后的默认值为 true。可以这样认为，如有两个字段 A，B 都有 btree 索引，oracle 有可能将这两个索引转换成 bitmap 索引然后做 and 操作得出结果集。如果改为 false 就会选用其中的一个索引，走 btree 的索引，**我们可以将该参数在 session 或系统级别设置为 false，也可以加 hint /*+ opt_param('_b_tree_bitmap_plans', 'false') */ 来实现禁用该参数。**

- symptom: Execution plan operation shows bitmap conversion from rowids
- symptom: No bitmap indexes
- symptom: Execution plan shows BITMAP CONVERSION
- cause: In 7.3.4 and in 8.1.7 default value of _b_tree_bitmap_plans is FALSE

whereas as of 9.0.1 (and 9.2) the default value is TRUE When _b_tree_bitmap_plans set to true (advice not to change the default setting yourself) the optimizer is allowed to produce bitmap plans for normal b*tree indexes even if no bitmap indexes set.

相关的执行计划中可能转换为如下的形式：

（1）BITMAP CONVERSION FROM ROWIDS

将一批数据记录的 ROWID 映射为位图。

对于普通 B*树索引，Oracle 也可以将数据记录的 ROWID 映射成一个位图，然后进行位图操作。进行这样的转换需要将系统参数_b_tree_bitmap_plans 设置为 TRUE。

（2）BITMAP CONVERSION TO ROWIDS

将位图映射为 ROWID。在一个位图键值中，包含了一批数据记录的起始地址和结束地址，且这批记录是连续的，因此位图中的每一个位就按序对应了一条数据记录。

（3）BITMAP OR

对位图进行“或”（OR）操作。在查询的过滤条件中，如果位图索引字段直接的关系是“或”，可以通过 BITMAP OR 来判断位图所映射的一批数据记录是否满足条件。

eygle 大师的一个例子：

http://www.eygle.com/archives/2011/12/bitmap_conversion_cpu.html

1.4.2 11g 情况下

```
[root@rhel6_lhr ~]# su - oracle
[oracle@rhel6_lhr ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.3.0 Production on Fri May 10:16:10 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

10:16:10 SQL>

10:16:10 SQL> conn lhr/lhr
Connected.

10:16:10 SQL> create table emp_bk as select * from scott.emp;
Table created.

Elapsed: 00:00:03.43

10:16:15 SQL> create index idx_emp_empno on emp_bk(empno);

Index created.

Elapsed: 00:00:00.05
10:19:26 SQL> create index idx_emp_ename on emp_bk(ename);

Index created.

Elapsed: 00:00:00.04
```

```
10:20:48 SQL> explain plan for select empno,ename from emp_bk where empno=7788 or ename='SCOTT';

Explained.

Elapsed: 00:00:00.09
10:20:56 SQL> select * from table(dbms_xplan.display);


PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------
Plan hash value: 4193090541

-------------------------------------------------------------------------------------------
| Id  | Operation                           | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |                |     1 |    20 |     2   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID        | EMP_BK         |     1 |    20 |     2   (0)| 00:00:01 |
|   2 |   BITMAP CONVERSION TO ROWIDS       |                |       |       |            |          |
|   3 |    BITMAP OR                        |                |       |       |            |          |
|   4 |     BITMAP CONVERSION FROM ROWIDS   |                |       |       |            |          |
|*  5 |      INDEX RANGE SCAN               | IDX_EMP_EMPNO  |       |       |     1   (0)| 00:00:01 |
|   6 |     BITMAP CONVERSION FROM ROWIDS   |                |       |       |            |          |
|*  7 |      INDEX RANGE SCAN               | IDX_EMP_ENAME  |       |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

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

   5 - access("EMPNO"=7788)
   7 - access("ENAME"='SCOTT')

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

24 rows selected.

Elapsed: 00:00:00.52

10:24:06 SQL> conn / as sysdba
Connected.

Elapsed: 00:00:00.03
10:24:34 SQL> set pagesize 9999
10:24:41 SQL> set line 9999
10:24:41 SQL> col NAME format a30
10:24:41 SQL> col KSPPDESC format a50
10:24:41 SQL> col KSPPSTVL format a20
10:24:42 SQL> SELECT a.INDX,
10:24:42   2         a.KSPPINM NAME,
10:24:42   3         a.KSPPDESC,
10:24:42   4         b.KSPPSTVL
10:24:42   5  FROM   x$ksppi  a,
10:24:42   6         x$ksppcv b
10:24:42   7  WHERE  a.INDX = b.INDX
10:24:42   8  and lower(a.KSPPINM) like  lower('%&parameter%');
Enter value for parameter: _b_tree_bitmap_plans
old   8: and lower(a.KSPPINM) like  lower('%&parameter%')
new   8: and lower(a.KSPPINM) like  lower('%_b_tree_bitmap_plans%')

      INDX NAME                           KSPPDESC                                           KSPPSTVL
---------- ------------------------------ -------------------------------------------------- --------------------
      1910 _b_tree_bitmap_plans           enable the use of bitmap plans for tables w. only  TRUE
                                          B-tree indexes
```

```
Elapsed: 00:00:00.01

10:25:44 SQL> conn lhr/lhr
Connected.

10:26:56 SQL> alter session set "_b_tree_bitmap_plans" = false;

Session altered.

Elapsed: 00:00:00.00
10:27:01 SQL> show parameter _b_tree_bitmap_plans

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
_b_tree_bitmap_plans                 boolean     FALSE
10:27:05 SQL> explain plan for select empno,ename from emp_bk where empno=7788 or ename='SCOTT';

Explained.

Elapsed: 00:00:00.01
10:27:14 SQL> select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
Plan hash value: 370270337

-----------------------------------------------------------------------------
| Id  | Operation         | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |        |     1 |    20 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| EMP_BK |     1 |    20 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------

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

   1 - filter("EMPNO"=7788 OR "ENAME"='SCOTT')

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

17 rows selected.

Elapsed: 00:00:00.04
10:27:18 SQL> explain plan for select empno,ename from emp_bk where empno=7788
10:27:49   2  union
10:27:55   3  select empno,ename from emp_bk where ename='SCOTT';

Explained.

Elapsed: 00:00:00.00
10:28:07 SQL> select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------
Plan hash value: 3014579657

--------------------------------------------------------------------------------
| Id  | Operation                     | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |               |     2 |    40 |     6  (67)| 00:00:01 |
|   1 |  SORT UNIQUE                  |               |     2 |    40 |     6  (67)| 00:00:01 |
```

```
|   2 |   UNION-ALL                          |                |       |       |     2   (0)| 00:00:01 |
|   3 |    TABLE ACCESS BY INDEX ROWID       | EMP_BK         |     1 |    20 |     2   (0)| 00:00:01 |
|*  4 |     INDEX RANGE SCAN                 | IDX_EMP_EMPNO  |     1 |       |     1   (0)| 00:00:01 |
|   5 |    TABLE ACCESS BY INDEX ROWID       | EMP_BK         |     1 |    20 |     2   (0)| 00:00:01 |
|*  6 |     INDEX RANGE SCAN                 | IDX_EMP_ENAME  |     1 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------

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

   4 - access("EMPNO"=7788)
   6 - access("ENAME"='SCOTT')

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

23 rows selected.

Elapsed: 00:00:00.01
10:28:13 SQL> Select Name ,Value From v$parameter Where Name ='_b_tree_bitmap_plans' ;

NAME                           VALUE
------------------------------ ---------------------------------------------------------------
_b_tree_bitmap_plans           FALSE

Elapsed: 00:00:00.02
10:34:06 SQL> alter session set "_b_tree_bitmap_plans" = true;

Session altered.

Elapsed: 00:00:00.00

11:19:04 SQL> explain plan for select /*+ opt_param('_b_tree_bitmap_plans', 'false') */ empno,ename from emp_bk where empno=7788 or ename='SCOTT';

Explained.

Elapsed: 00:00:00.08
11:19:22 SQL> select * from table(dbms_xplan.display);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Plan hash value: 370270337

-----------------------------------------------------------------------------
| Id  | Operation          | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |        |     1 |    20 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL | EMP_BK |     1 |    20 |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------

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

   1 - filter("EMPNO"=7788 OR "ENAME"='SCOTT')

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

17 rows selected.

Elapsed: 00:00:00.24
```

由实验可以看出，_b_tree_bitmap_plans 设置为 false 后，emp_bk 走了全表扫描，并没有走位图索引转换。

1.4.3 8i 情况下

```
C:\Users\Administrator>sqlplus "lhr/lhr@orcl8i as sysdba"

SQL*Plus: Release 8.1.7.0.0 - Production on Mon May 18 10:44:28 2015

(c) Copyright 2000 Oracle Corporation.  All rights reserved.


Connected to:
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

SQL> set pagesize 9999
SQL> set line 9999
SQL> col NAME format a30
SQL> col KSPPDESC format a50
SQL> col KSPPSTVL format a20
SQL> SELECT a.INDX,
  2         a.KSPPINM NAME,
  3         a.KSPPDESC,
  4         b.KSPPSTVL
  5  FROM   x$ksppi  a,
  6         x$ksppcv b
  7  WHERE  a.INDX = b.INDX
  8  and lower(a.KSPPINM) like  lower('%&parameter%');
Enter value for parameter: _b_tree_bitmap_plans
old   8: and lower(a.KSPPINM) like  lower('%&parameter%')
new   8: and lower(a.KSPPINM) like  lower('%_b_tree_bitmap_plans%')

      INDX NAME                           KSPPDESC                                           KSPPSTVL
---------- ------------------------------ -------------------------------------------------- --------------------
       348 _b_tree_bitmap_plans           enable the use of bitmap plans for tables w. only  FALSE
                                          B-tree indexes


SQL>



SQL> create table lhr.emp_bk as select * from scott.emp;

Table created.

SQL> create index lhr.idx_emp_empno on lhr.emp_bk(empno);

Index created.

SQL> create index lhr.idx_emp_ename on lhr.emp_bk(ename);

Index created.

SQL> set line 9999 pagesize 9999
SQL> set autot on;
SQL> select empno,ename from lhr.emp_bk where empno=7788 or ename='SCOTT';

     EMPNO ENAME
```

```
---------- ----------
      7788 SCOTT


Execution Plan
----------------------------------------------------------
          0      SELECT STATEMENT Optimizer=CHOOSE
          1    0   CONCATENATION
          2    1     TABLE ACCESS (BY INDEX ROWID) OF 'EMP_BK'
          3    2       INDEX (RANGE SCAN) OF 'IDX_EMP_ENAME' (NON-UNIQUE)
          4    1     TABLE ACCESS (BY INDEX ROWID) OF 'EMP_BK'
          5    4       INDEX (RANGE SCAN) OF 'IDX_EMP_EMPNO' (NON-UNIQUE)



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

SQL>
```

8i 下默认为 false，执行计划也完全不同。

1.5 Native Full Outer Join

关于这个特性可以参考如下文章：

http://blog.itpub.net/26736162/viewspace-1660038/

我们在 10.2.0.4 下测试一下：

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

SQL*Plus: Release 10.2.0.4.0 - Production on Mon May 18 11:41:13 2015

Copyright (c) 1982, 2007, Oracle.  All Rights Reserved.

Connected to an idle instance.
```

```
SQL> startup
ORACLE instance started.

Total System Global Area  448790528 bytes
Fixed Size                  2084616 bytes
Variable Size             121635064 bytes
Database Buffers          318767104 bytes
Redo Buffers                6303744 bytes
Database mounted.
Database opened.

SQL> create table lhr.emp_bk as select * from scott.emp;

Table created.

SQL> create table lhr.emp_bk as select * from scott.emp;

Table created.

SQL> set autot on;
SQL> set line 9999 pagesize 9999
SQL> select * from lhr.emp_bk a full outer join lhr.emp_bk2 b on a.empno=b.empno;

     EMPNO ENAME      JOB              MGR HIREDATE                 SAL       COMM     DEPTNO      EMPNO ENAME      JOB              MGR HIREDATE
SAL       COMM     DEPTNO
---------- ---------- --------- ---------- ------------------- ---------- ---------- ---------- ---------- ---------- --------- ---------- -------------------
---------- ---------- ----------
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00        800                    20       7369 SMITH      CLERK           7902 1980-12-17 00:00:00
800                    20
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00       1600        300         30       7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00
1600        300         30
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00       1250        500         30       7521 WARD       SALESMAN        7698 1981-02-22 00:00:00
1250        500         30
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00       2975                    20       7566 JONES      MANAGER         7839 1981-04-02 00:00:00
2975                   20
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00       1250       1400         30       7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00
1250       1400         30
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00       2850                    30       7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00
2850                   30
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00       2450                    10       7782 CLARK      MANAGER         7839 1981-06-09 00:00:00
2450                   10
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00       3000                    20       7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00
3000                   20
      7839 KING       PRESIDENT            1981-11-17 00:00:00       5000                    10       7839 KING       PRESIDENT            1981-11-17
00:00:00       5000                   10
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00       1500          0         30       7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00
1500          0         30
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00       1100                    20       7876 ADAMS      CLERK           7788 1987-05-23 00:00:00
1100                   20
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00        950                    30       7900 JAMES      CLERK           7698 1981-12-03 00:00:00
950                    30
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00       3000                    20       7902 FORD       ANALYST         7566 1981-12-03 00:00:00
3000                   20
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00       1300                    10       7934 MILLER     CLERK           7782 1982-01-23 00:00:00
1300                   10

14 rows selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 914601651

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

```
|   0 | SELECT STATEMENT    |         |    15 |  2610 |    13   (8)| 00:00:01 |
|   1 |  VIEW               |         |    15 |  2610 |    13   (8)| 00:00:01 |
|   2 |   UNION-ALL         |         |       |       |            |          |
| * 3 |    HASH JOIN OUTER  |         |    14 |  2436 |     7  (15)| 00:00:01 |
|   4 |     TABLE ACCESS FULL| EMP_BK  |    14 |  1218 |     3   (0)| 00:00:01 |
|   5 |     TABLE ACCESS FULL| EMP_BK2 |    14 |  1218 |     3   (0)| 00:00:01 |
| * 6 |    HASH JOIN ANTI   |         |     1 |   100 |     7  (15)| 00:00:01 |
|   7 |     TABLE ACCESS FULL| EMP_BK2 |    14 |  1218 |     3   (0)| 00:00:01 |
|   8 |     TABLE ACCESS FULL| EMP_BK  |    14 |   182 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------

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

   3 - access("A"."EMPNO"="B"."EMPNO"(+))
   6 - access("A"."EMPNO"="B"."EMPNO")

Note
-----
   - dynamic sampling used for this statement


Statistics
----------------------------------------------------------
        338  recursive calls
          0  db block gets
         61  consistent gets
          6  physical reads
          0  redo size
       2521  bytes sent via SQL*Net to client
        492  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          4  sorts (memory)
          0  sorts (disk)
         14  rows processed

SQL> select /*+ NATIVE_FULL_OUTER_JOIN */ * from lhr.emp_bk a full outer join lhr.emp_bk2 b on a.empno=b.empno;

     EMPNO ENAME      JOB              MGR HIREDATE                 SAL       COMM     DEPTNO      EMPNO ENAME      JOB              MGR HIREDATE
SAL       COMM     DEPTNO
---------- ---------- --------- ---------- ------------------ ---------- ---------- ---------- ---------- ---------- --------- ---------- ------------------
---------- ---------- ----------
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00        800                    20       7369 SMITH      CLERK           7902 1980-12-17 00:00:00
800                    20
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00       1600        300         30       7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00
1600        300         30
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00       1250        500         30       7521 WARD       SALESMAN        7698 1981-02-22 00:00:00
1250        500         30
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00       2975                    20       7566 JONES      MANAGER         7839 1981-04-02 00:00:00
2975                   20
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00       1250       1400         30       7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00
1250       1400         30
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00       2850                    30       7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00
2850                   30
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00       2450                    10       7782 CLARK      MANAGER         7839 1981-06-09 00:00:00
2450                   10
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00       3000                    20       7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00
3000                   20
      7839 KING       PRESIDENT            1981-11-17 00:00:00       5000                    10       7839 KING       PRESIDENT            1981-11-17
00:00:00       5000                   10
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00       1500          0         30       7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00
1500          0         30
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00       1100                    20       7876 ADAMS      CLERK           7788 1987-05-23 00:00:00
1100                   20
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00        950                    30       7900 JAMES      CLERK           7698 1981-12-03 00:00:00
950                    30
```

```
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00       3000                 20      7902 FORD       ANALYST         7566 1981-12-03 00:00:00
3000                   20
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00       1300                 10      7934 MILLER     CLERK           7782 1982-01-23 00:00:00
1300                   10

14 rows selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 2812081866

-------------------------------------------------------------------------------------
| Id  | Operation             | Name     | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |          |    14 |  2436 |     7  (15)| 00:00:01 |
|   1 |  VIEW                 | VW_FOJ_0 |    14 |  2436 |     7  (15)| 00:00:01 |
|*  2 |   HASH JOIN FULL OUTER|          |    14 |  2436 |     7  (15)| 00:00:01 |
|   3 |    TABLE ACCESS FULL  | EMP_BK   |    14 |  1218 |     3   (0)| 00:00:01 |
|   4 |    TABLE ACCESS FULL  | EMP_BK2  |    14 |  1218 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   2 - access("A"."EMPNO"="B"."EMPNO")

Note
-----
   - dynamic sampling used for this statement


Statistics
----------------------------------------------------------
          7  recursive calls
          0  db block gets
         15  consistent gets
          0  physical reads
          0  redo size
       2521  bytes sent via SQL*Net to client
        492  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          2  sorts (memory)
          0  sorts (disk)
         14  rows processed

SQL> set pagesize 9999
SQL> set line 9999
SQL> col NAME format a40
SQL> col KSPPDESC format a50
SQL> col KSPPSTVL format a20
SQL> SELECT a.INDX,
  2         a.KSPPINM NAME,
  3         a.KSPPDESC,
  4         b.KSPPSTVL
  5  FROM   x$ksppi  a,
  6         x$ksppcv b
  7  WHERE  a.INDX = b.INDX
  8  and lower(a.KSPPINM) like  lower('%&parameter%');
Enter value for parameter: optimizer_native_full_outer_join
old   8: and lower(a.KSPPINM) like  lower('%&parameter%')
new   8: and lower(a.KSPPINM) like  lower('%optimizer_native_full_outer_join%')

      INDX NAME                           KSPPDESC                                           KSPPSTVL
---------- ------------------------------ -------------------------------------------------- --------------------
      1318 _optimizer_native_full_outer_j execute full outer join using native implementaion off
           oin
```

```
SQL>
```

1.6 多表查询时候的 null 值处理

我们在第一篇（http://blog.itpub.net/26736162/viewspace-1652985/）中总结了一下 null 值特征，今天我们再来看一下多表查询的时候 null 值得处理。

1.6.1 情形一：

若子查询中的结果中包含 null 值，那么 not in（null、xx、bb、cc）返回为空。

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

SQL*Plus: Release 11.2.0.3.0 Production on Mon May 18 13:38:09 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

13:38:09 SQL> drop table lhr.emp_bk;

Table dropped.

Elapsed: 00:00:04.16
13:38:15 SQL> create table lhr.emp_bk as select * from scott.emp;

Table created.

Elapsed: 00:00:00.77
13:41:01 SQL> create table lhr.dept_bk as select * from scott.dept;

Table created.

Elapsed: 00:00:00.13
13:41:43 SQL>  insert into lhr.dept_bk  values(50,'lhr','China');

1 row created.

Elapsed: 00:00:00.03
13:41:57 SQL> select * from lhr.dept_bk ;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 lhr            China
```

```
Elapsed: 00:00:00.01
13:42:48 SQL> select * from  lhr.emp_bk b;

     EMPNO ENAME      JOB              MGR HIREDATE                 SAL       COMM     DEPTNO
---------- ---------- --------- ---------- ------------------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00      20800                    20
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00      31600        300         30
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00      31250        500         30
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00      22975                    20
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00      31250       1400         30
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00      32850                    30
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00      12450                    10
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00      23000                    20
      7839 KING       PRESIDENT            1981-11-17 00:00:00      15000                    10
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00      31500          0         30
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00      21100                    20
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00      30950                    30
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00      23000                    20
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00      11300                    10

16 rows selected.

Elapsed: 00:00:00.02
13:44:00 SQL> select * from lhr.dept_bk a where  a.deptno not in(select b.deptno from lhr.emp_bk b);

    DEPTNO DNAME          LOC
---------- -------------- -------------
        50 lhr            China
        40 OPERATIONS     BOSTON

Elapsed: 00:00:00.93
13:44:07 SQL> update lhr.emp_bk b set b.deptno=null where empno=7788;

1 row updated.

Elapsed: 00:00:00.04
13:45:17 SQL> select * from  lhr.emp_bk b;

     EMPNO ENAME      JOB              MGR HIREDATE                 SAL       COMM     DEPTNO
---------- ---------- --------- ---------- ------------------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 1980-12-17 00:00:00      20800                    20
      7499 ALLEN      SALESMAN        7698 1981-02-20 00:00:00      31600        300         30
      7521 WARD       SALESMAN        7698 1981-02-22 00:00:00      31250        500         30
      7566 JONES      MANAGER         7839 1981-04-02 00:00:00      22975                    20
      7654 MARTIN     SALESMAN        7698 1981-09-28 00:00:00      31250       1400         30
      7698 BLAKE      MANAGER         7839 1981-05-01 00:00:00      32850                    30
      7782 CLARK      MANAGER         7839 1981-06-09 00:00:00      12450                    10
      7788 SCOTT      ANALYST         7566 1987-04-19 00:00:00      23000
      7839 KING       PRESIDENT            1981-11-17 00:00:00      15000                    10
      7844 TURNER     SALESMAN        7698 1981-09-08 00:00:00      31500          0         30
      7876 ADAMS      CLERK           7788 1987-05-23 00:00:00      21100                    20
      7900 JAMES      CLERK           7698 1981-12-03 00:00:00      30950                    30
      7902 FORD       ANALYST         7566 1981-12-03 00:00:00      23000                    20
      7934 MILLER     CLERK           7782 1982-01-23 00:00:00      11300                    10

14 rows selected.

Elapsed: 00:00:00.14
13:45:23 SQL> select * from lhr.dept_bk a where  a.deptno not in(select b.deptno from lhr.emp_bk b);

no rows selected

Elapsed: 00:00:00.00
13:45:39 SQL> select * from lhr.dept_bk a where  a.deptno not in(select b.deptno from lhr.emp_bk b where b.deptno is not null);

    DEPTNO DNAME          LOC
---------- -------------- -------------
```

```
        50 lhr            China
        40 OPERATIONS     BOSTON

Elapsed: 00:00:00.04
13:46:01 SQL>
```

1.6.2 情形二：

要求返回所有比“ALLEN”提成低的员工：

```
14:01:07 SQL> select a.ename,a.comm from scott.emp a;

ENAME            COMM
---------- ----------
SMITH
ALLEN             300
WARD              500
JONES
MARTIN           1400
BLAKE
CLARK
SCOTT
KING
TURNER              0
ADAMS
JAMES
FORD
MILLER

14 rows selected.

Elapsed: 00:00:00.23
14:01:17 SQL> select a.ename,a.comm from scott.emp a where a.comm < ( select b.comm from scott.emp b where b.ename='ALLEN');

ENAME            COMM
---------- ----------
TURNER              0

Elapsed: 00:00:00.11
14:01:28 SQL> select a.ename,a.comm from scott.emp a where coalesce(a.comm,0) < ( select b.comm from scott.emp b where b.ename='ALLEN');

ENAME            COMM
---------- ----------
SMITH
JONES
BLAKE
CLARK
SCOTT
KING
TURNER              0
ADAMS
JAMES
FORD
MILLER

11 rows selected.

Elapsed: 00:00:00.02
14:01:55 SQL>
```

1.7 总结

到此个人觉得本章的一些难点或需要补充的地方就这些了，希望大家看完有所收获。

1.8 about me

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-1660422/

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

QQ：642808185 若加 QQ 请注明你所正在读的文章标题

创作时间地点：2015-05-15 10:00~ 2015-05-18 15:00 于外汇交易中心