|  |  |
| --- | --- |
| From: | 京奇電腦 |
| To: |  |
| Date: | 2016-06-01 |
| Subject: | Oracle SQL command Start with connect by prior 階層式查詢用法 |

Reference: http://fecbob.pixnet.net/blog/post/43278427-oracle-start-with-connect-by-%E7%94%A8%E6%B3%95

Sample:

SELECT '1' FCLASS,

AG.AGENT\_CODE IBIGCOMP, --车商代码

AB.BRANCH\_CODE IBIG\_BRANCH, --车商营业所分支代码

AB.OCCUPY\_SP\_CODE IBIG\_OFFICE, --车商营业所据点代码

RPAD(NVL(O.REGISTRATION\_NAME, ' '), 40, ' ') NNAME,--全称

RPAD(NVL(O.name, ' '), 20, ' ') NNAME\_BRAN --简称

-- T.\*

FROM

(SELECT connect\_by\_root TO\_PTYR\_ID,

R.FROM\_PTYR\_ID,

R.RELATION\_TYPE

FROM T\_PTYR\_RELA R

START WITH R.to\_Ptyr\_Id IN

(SELECT RR.PTYR\_ID

FROM T\_AGT\_AGENT AG,

T\_PTYR RR

WHERE AG.PARTY\_ID = RR.PTY\_ID

)

CONNECT BY PRIOR R.FROM\_PTYR\_ID = R.TO\_PTYR\_ID

) T,

T\_CHT\_PTY\_AGENCY\_BRANCH AB,

T\_PTYR RA,

T\_AGT\_AGENT AG,

T\_PTY\_ORG O,

T\_PTYR BA

WHERE AB.PTYR\_ID = T.FROM\_PTYR\_ID

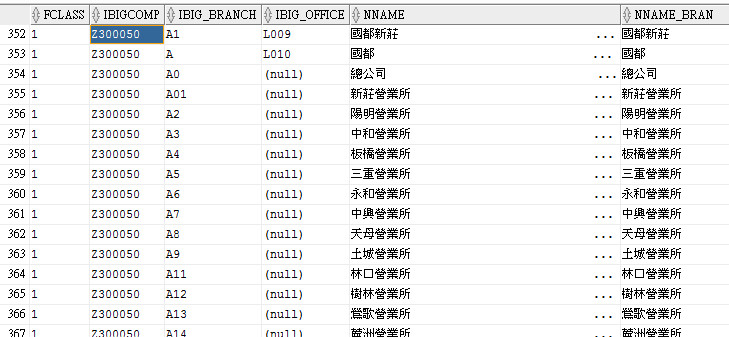
AND T.RELATION\_TYPE IN (109, 110)

AND AG.PARTY\_ID = RA.PTY\_ID

AND O.PTY\_ID = BA.PTY\_ID

AND BA.PTYR\_ID = AB.PTYR\_ID

AND RA.PTYR\_ID = CONNECT\_BY\_ROOTTO\_PTYR\_ID;



oracle 提供了start with connect by 語法結構可以實現遞迴查詢。

1. 一個簡單舉例:

SQL> select \* from test;

BILL\_MONTH DAY\_NUMBER MSISDN

-------------------- ---------- --------------------

200803 1 13800

200803 3 13800

200803 2 13800

200803 2 13801

200803 4 13804

200803 5 13804

200803 7 13804

200803 8 13804

200803 6 13802

200803 6 13801

200803 7 13801

200803 8 13801

12 rows selected

SQL>

SQL> select \* from test

2 start with day\_number=1

3 connect by prior day\_number=day\_number-1 and prior msisdn= msisdn

4 ;

BILL\_MONTH DAY\_NUMBER MSISDN

-------------------- ---------- --------------------

200803 1 13800

200803 2 13800

200803 3 13800

SQL>

上面的語句查找出了從1開始，並且day\_number 逐漸+1 遞增的，並且 msisdn 相同的哪些個資料.

2. start with connect by 語法結構

如上面說看到的 例子， 其語法結構為 start with condition connect by condition （含 prior 關鍵字)

start with conditon 給出的seed 資料的範圍, connect by 後面給出了遞迴查詢的條件,prior 關鍵字表示父資料，prior 條件表示子資料需要滿足父資料的什麼條件。

在下面的這個start with connect by 結構中，就表示 查找出了從1開始，並且day\_number 逐漸+1 遞增的，並且 msisdn 相同的那些個資料.

start with day\_number=1

connect by prior day\_number=day\_number-1 and prior msisdn= msisdn

3. 執行計畫

對於這個特殊的語法結構，我們來看看它的執行計畫。

通過下面的執行計畫，我們可以看出，對於簡單的訪問一個物件的遞迴查詢，實際上oracle 要三次訪問要查詢的物件。因此，這一個告訴我們，在使用遞迴查詢時，一定要謹慎，因為即使原表資料不多，但是三倍的訪問喜愛來，代價也會很大。

SQL> explain plan for

2

2 select \* from test

3 --where bill\_month='200803'

4 start with day\_number=1

5 connect by prior day\_number=day\_number-1 and prior msisdn= msisdn

6 ;

Explained

SQL> select \* from table( dbms\_xplan.display);

PLAN\_TABLE\_OUTPUT

--------------------------------------------------------------------------------

-------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost |

-------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | | | |

|\* 1 | CONNECT BY WITH FILTERING| | | | |

|\* 2 | FILTER | | | | |

| 3 | TABLE ACCESS FULL | TEST | | | |

| 4 | NESTED LOOPS | | | | |

| 5 | BUFFER SORT | | | | |

| 6 | CONNECT BY PUMP | | | | |

|\* 7 | TABLE ACCESS FULL | TEST | | | |

| 8 | TABLE ACCESS FULL | TEST | | | |

-------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

1 - filter("TEST"."DAY\_NUMBER"=1)

2 - filter("TEST"."DAY\_NUMBER"=1)

PLAN\_TABLE\_OUTPUT

--------------------------------------------------------------------------------

7 - filter("TEST"."MSISDN"=Null AND "TEST"."DAY\_NUMBER"-1=Null)

Note: rule based optimization

23 rows selected

SQL>

另外，發現了在含有其他條件的遞迴中，是先處理所有的遞迴查詢，最後才用加入的條件過濾.

請看下面的例子。

和上面的執行計畫對比下我們可以知道，加入條件 where bill\_month='200803' 後，實際上卻是在遞迴完成後，最後才執行的 1 - filter("TEST"."BILL\_MONTH"='200803') 。

所以，為了確保語句的性能，不要直接加入條件在start with connect by 結構中，而是要想辦法將原表的資料控制住。這個可以採用子查詢的辦法，或者使用臨時表等（最好採用臨時表，將資料量從本源上控制住；因為從子查詢的執行計畫我們可以看到，它每次也都是訪問全表，再用條件過濾，要重複三次，不是一次過濾就夠了).

--直接加入條件後的執行計畫

SQL> explain plan for

2

2 select \* from test

3 where bill\_month='200803'

4 start with day\_number=1

5 connect by prior day\_number=day\_number-1 and prior msisdn= msisdn

6 ;

Explained

SQL> select \* from table( dbms\_xplan.display);

PLAN\_TABLE\_OUTPUT

--------------------------------------------------------------------------------

--------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost |

--------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | | | |

|\* 1 | FILTER | | | | |

|\* 2 | CONNECT BY WITH FILTERING| | | | |

|\* 3 | FILTER | | | | |

| 4 | TABLE ACCESS FULL | TEST | | | |

| 5 | NESTED LOOPS | | | | |

| 6 | BUFFER SORT | | | | |

| 7 | CONNECT BY PUMP | | | | |

|\* 8 | TABLE ACCESS FULL | TEST | | | |

| 9 | TABLE ACCESS FULL | TEST | | | |

--------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

1 - filter("TEST"."BILL\_MONTH"='200803')

PLAN\_TABLE\_OUTPUT

--------------------------------------------------------------------------------

2 - filter("TEST"."DAY\_NUMBER"=1)

3 - filter("TEST"."DAY\_NUMBER"=1)

8 - filter("TEST"."MSISDN"=Null AND "TEST"."DAY\_NUMBER"-1=Null)

Note: rule based optimization

25 rows selected

SQL>

--使用子查詢，將過濾條件嵌在子查詢中

SQL> explain plan for

2

2 select \* from (select \* from test

3 where bill\_month='200803')

4 start with day\_number=1

5 connect by prior day\_number=day\_number-1 and prior msisdn= msisdn

6 ;

Explained

SQL> select \* from table(dbms\_xplan.display);

PLAN\_TABLE\_OUTPUT

--------------------------------------------------------------------------------

-------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost |

-------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | | | |

|\* 1 | CONNECT BY WITH FILTERING| | | | |

|\* 2 | FILTER | | | | |

|\* 3 | TABLE ACCESS FULL | TEST | | | |

| 4 | NESTED LOOPS | | | | |

| 5 | BUFFER SORT | | | | |

| 6 | CONNECT BY PUMP | | | | |

|\* 7 | TABLE ACCESS FULL | TEST | | | |

|\* 8 | TABLE ACCESS FULL | TEST | | | |

-------------------------------------------------------------------------

Predicate Information (identified by operation id):

---------------------------------------------------

1 - filter("TEST"."DAY\_NUMBER"=1)

2 - filter("TEST"."DAY\_NUMBER"=1)

PLAN\_TABLE\_OUTPUT

--------------------------------------------------------------------------------

3 - filter("TEST"."BILL\_MONTH"='200803')

7 - filter("TEST"."BILL\_MONTH"='200803' AND "TEST"."MSISDN"=Null AND

"TEST"."DAY\_NUMBER"-1=Null)

8 - filter("TEST"."BILL\_MONTH"='200803')

Note: rule based optimization

26 rows selected

SQL>

4. 實際中 遞迴查詢的使用。

問題：

資料庫裡有欄位day\_number，msisdn。如何寫月度連續3天有記錄的手機號？表結構如下：

id bill\_month day\_number msisdn

1 200803 1 13800000000

2 200803 1 130137.....

3 200803 2 13800000000

4 200803 3 13800000000

..............................

表中3月份連續3天有記錄的紀錄就是1380000000。請問如何寫這樣的sql？

解決方案：

SQL> create table test ( bill\_month Varchar2(20),day\_number number ,msisdn Varchar2(20));

Table created

SQL> insert into test values ( '200803',1,'13800');

1 row inserted

SQL> insert into test values ( '200803',3,'13800');

1 row inserted

SQL> insert into test values ( '200803',2,'13800');

1 row inserted

SQL> insert into test values ( '200803',2,'13801');

1 row inserted

SQL> insert into test values ( '200803',4,'13804');

1 row inserted

SQL> insert into test values ( '200803',5,'13804');

1 row inserted

SQL> commit;

Commit complete

SQL> select \* from test;

BILL\_MONTH DAY\_NUMBER MSISDN

-------------------- ---------- --------------------

200803 1 13800

200803 3 13800

200803 2 13800

200803 2 13801

200803 4 13804

200803 5 13804

6 rows selected

SQL>

SQL> select distinct msisdn from test a

2 where bill\_month='200803'

3 and exists

4 ( select msisdn from test

5 where bill\_month='200803' and msisdn=a.msisdn

6 start with day\_number=a.day\_number

7 connect by prior day\_number=day\_number-1 and prior msisdn= msisdn

8 group by msisdn

9 having count(\*)>=3

10 );

MSISDN

--------------------

13800

SQL> select \* from test;

BILL\_MONTH DAY\_NUMBER MSISDN

-------------------- ---------- --------------------

200803 1 13800

200803 3 13800

200803 2 13800

200803 2 13801

200803 4 13804

200803 5 13804

6 rows selected

SQL> insert into test values ( '200803',7,'13804');

1 row inserted

SQL> insert into test values ( '200803',8,'13804');

1 row inserted

SQL> insert into test values ( '200803',6,'13802');

1 row inserted

SQL> insert into test values ( '200803',6,'13801');

1 row inserted

SQL> insert into test values ( '200803',7,'13801');

1 row inserted

SQL> insert into test values ( '200803',8,'13801');

1 row inserted

SQL> select \* from test;

BILL\_MONTH DAY\_NUMBER MSISDN

-------------------- ---------- --------------------

200803 1 13800

200803 3 13800

200803 2 13800

200803 2 13801

200803 4 13804

200803 5 13804

200803 7 13804

200803 8 13804

200803 6 13802

200803 6 13801

200803 7 13801

200803 8 13801

12 rows selected

SQL> commit;

Commit complete

SQL>

SQL> select distinct msisdn from test a

2 where bill\_month='200803'

3 and exists

4 ( select msisdn from test

5 where bill\_month='200803' and msisdn=a.msisdn

6 start with day\_number=a.day\_number

7 connect by prior day\_number=day\_number-1 and prior msisdn= msisdn

8 group by msisdn

9 having count(\*)>=3

10 );

MSISDN

--------------------

13800

13801

SQL>