

## 2011년 4월 19일 화요일

### Oracle 문자열 구분자로 나누기

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | SQL> SELECT  REGEXP\_SUBSTR(full\_txt, '[^\_]+', 1, LEVEL) txt    2  FROM    (SELECT '첫번째\_두번째\_세번째' full\_txt FROM DUAL)    3  CONNECT BY LEVEL <= LENGTH(REGEXP\_REPLACE(full\_txt, '[^\_]+',''))+1    4  ;    TXT  -----------  첫번째  두번째  세번째    SQL> |

구분자가 연속으로 나타날 경우엔 아래와 같은 결과가 나타난다.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | SQL> SELECT  REGEXP\_SUBSTR(full\_txt, '[^\_]+', 1, LEVEL) txt    2  FROM    (SELECT '첫번째\_\_세번째' full\_txt FROM DUAL)    3  CONNECT BY LEVEL <= LENGTH(REGEXP\_REPLACE(full\_txt, '[^\_]+',''))+1    4  ;    TXT  --------  첫번째  세번째      SQL> |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | SQL> SELECT  SUBSTR(full\_txt, INSTR(full\_txt, '\_', 1, LEVEL)+1,    2          INSTR(full\_txt, '\_', 1, LEVEL+1) -    3          INSTR(full\_txt, '\_', 1, LEVEL)-1) txt    4  FROM    (SELECT '\_'||'첫번째\_두번째\_세번째'||'\_' full\_txt FROM DUAL)    5  CONNECT BY LEVEL <= LENGTH(REGEXP\_REPLACE(full\_txt, '[^\_]+',''))-1    6  ;    TXT  -------------  첫번째  두번째  세번째    SQL> |

구분자가 연속으로 나타날 경우엔 아래와 같은 결과가 나타난다.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | SQL> SELECT  SUBSTR(full\_txt, INSTR(full\_txt, '\_', 1, LEVEL)+1,    2          INSTR(full\_txt, '\_', 1, LEVEL+1) -    3          INSTR(full\_txt, '\_', 1, LEVEL)-1) txt    4  FROM    (SELECT '\_'||'첫번째\_\_세번째'||'\_' full\_txt FROM DUAL)    5  CONNECT BY LEVEL <= LENGTH(REGEXP\_REPLACE(full\_txt, '[^\_]+',''))-1    6  ;    TXT  ----------  첫번째    세번째    SQL> |

[**네이버**](http://blog.naver.com/PostPrint.nhn?blogId=outthree&logNo=60061859450)

[**블로그**](http://blog.naver.com/PostPrint.nhn?blogId=outthree&logNo=60061859450)

[인쇄하기](javascript:print();)

**참고 SQL 쿼리(구분자로 된 문자열을 행으로 변환 등)**

개발참고

2009/02/03 08:35

[http://blog.naver.com/outthree/60061859450](http://blog.naver.com/PostPrint.nhn?blogId=outthree&logNo=60061859450)

-- 구분자로 된 문자열 행으로 반환  
SELECT  
       ROLE ,  
       SUBSTR(ROLE,val1+1,val2-val1-1)  
  FROM  
       (SELECT  
              ROLE,  
              DECODE(rnum,1,0,INSTR(ROLE, '^',1,RNUM-1)) val1,  
              DECODE (INSTR(ROLE, '^',1,RNUM),0,LENGTH(ROLE)+1, INSTR(ROLE, '^',1,RNUM)) val2,  
              LENGTH(ROLE)+1 len,  
              RNUM  
         FROM  
              (SELECT  
                     ROWNUM RNUM ,  
                     --'그룹12341^그룹212^그룹23^그룹64^그룹5^그룹6^그룹7^그룹8^그룹9^그룹6^그룹7^그룹8^그룹9' ROLE  
                     '5001^5002^7001^7004^' ROLE  
                FROM  
                     DUAL CONNECT BY LEVEL <= (LENGTH('5001^5002^7001^7004^') - LENGTH(REPLACE('5001^5002^7001^7004^', '^'))) / LENGTH('^') + 1

              )  
       )  
       WHERE ROLE <> SUBSTR(ROLE,val1+1,val2-val1-1)  
;

--여러행을 구분자로 붙여서 반환  
SELECT LTRIM(SYS\_CONNECT\_BY\_PATH(CODE\_NAME,', '),', ') CODE\_NAME  
  FROM (SELECT USE\_YN  
              ,CODE\_NAME  
              ,ROW\_NUMBER() OVER(PARTITION BY USE\_YN ORDER BY USE\_YN) RN  
              ,COUNT(\*) OVER(PARTITION BY USE\_YN) CNT  
          FROM SCV\_TASK)  
WHERE LEVEL = CNT  
START WITH RN = 1  
CONNECT BY PRIOR USE\_YN = USE\_YN AND PRIOR RN = RN-1;

|  |
| --- |
|  |
|  |
|  |
| [출력하기](javascript:print();)[취소](javascript:self.close();) |