**[문제1] PT2-109, REGEXP\_REPLACE() 함수를 사용하여 문자열(임의 값으로 조정가능) ‘aaa  bbb’를 대상으로 a{5 }, a{3, }, a{3, 5}, ab{2, 3} 패턴(임의 패턴으로 적용가능) 을 적용한 SQL를 실습하라.**

**a{5 },**

SQL> select 'kkiimmmmmgggyuyounng' as "name" , REGEXP\_REPLACE('kkiimmmmmgggyuyounng','(m){2}','\*') replace from dual;

name REPLACE

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

kkiimmmmmgggyuyounng kkii\*\*mgggyuyounng

**a{3, }**

SQL> select 'kkiimmmmmgggyuyounng' as "name" , REGEXP\_REPLACE('kkiimmmmmgggyuyounng','(m){3,}','\*') replace from dual;

name REPLACE

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

kkiimmmmmgggyuyounng kkii\*gggyuyounng

**a{1, 3}**

SQL> select 'kkiimmmmmgggyuyounng' as "name" , REGEXP\_REPLACE('kkiimmmmmgggyuyounng','(g){1,3}','\*') replace from dual;

name REPLACE

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

kkiimmmmmgggyuyounng kkiimmmmm\*yuyounn\*

**ab{2, 3}**

SQL> select 'kkiimmmmmgggyuyounng' as "name" , REGEXP\_REPLACE('kkiimmmmmgggyuyounng','ki{1,2}','\*') replace from dual;

name REPLACE

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

kkiimmmmmgggyuyounng k\*mmmmmgggyuyounng

**[실행결과]**

정규식에서 문자의 {a ,b } 등등이 헷갈렸었는데 이번에 어떤 의미인지 잘 알았고,

**a{3, } aaa, aaaa, aaaaaa, …**

**a{1,3} a, aa, aaa**

**a{2,3} aa, aaa**

**a{3} aaa**

정확히 숙지했습니다.

**[문제2] PT2-110, 예제 Test SQL 실습**

SQL> select REGEXP\_REPLACE('aaa bbb','( ){2,}','') "One",

2 REGEXP\_REPLACE('aaa bbb','( ){2,}','') "Two" from dual;

One Two

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

aaa bbb aaabbb

텍스트이(가) 표시된 사진

자동 생성된 설명

**[실행결과]**

Replace 함수를 적용하여 2번째 인자의 정규식에 해당하는 부분을 3번째 인자인 바꿀 문자열로 바뀌는 함수이다. 더 추가적인 인자로는 4번째 인자가 defalut 1 이고 이 1은 첫번째부터 스캔한다는 의미입니다. 또 마지막 인자로 ocuurence 하는 부분을 지정하여 몇번째로 해당하는 정규식인지를 표시하여 그 부분을 replace 합니다.

**[문제3] PT2-112, emp 테이블에서 사원명을 “SM  ITH”를 입력받아 위의 REGEXP\_REPLACE 함수를 적용하고 공백을 제거한후 EMPNO, ENAME과 REGEXP 함수 결과를 출력하는 SQL 실습**

SQL> select empno, ename from emp where ename = REGEXP\_REPLACE('&SMITH','( ){1,}','');

Enter value for smith: SM ITH

old 1: select empno, ename from emp where ename = REGEXP\_REPLACE('&SMITH','( ){1,}','')

new 1: select empno, ename from emp where ename = REGEXP\_REPLACE('SM ITH','( ){1,}','')

EMPNO ENAME

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

7369 SMITH

SQL> set verify off

SQL> select empno, ename from emp where ename = REGEXP\_REPLACE('&SMITH','( ){1,}','');

Enter value for smith: SM ITH

EMPNO ENAME

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

7369 SMITH

**[실행결과]**

전의 내용과 같은 내용으로 여기서는 새로 알게된 내용을 기술하겠습니다.

**SQL> set verify off**

구문으로

old 1: select empno, ename from emp where ename = REGEXP\_REPLACE('&SMITH','( ){1,}','')

new 1: select empno, ename from emp where ename = REGEXP\_REPLACE('SM ITH','( ){1,}','')

값을 다시 확인하는 내용을 없앴고, &는 사용자에게 입력받는 수식이라는 것을 알았습니다.

**[문제4] PT2-116, select LTRIM(REGEXP\_SUBSTR(‘http://www.ut.ac.kr', ‘/([[:alnum:]]+ \.?) {3,4}?'), ‘/’)  from dual;  SQL 실습**

SQL> select LTRIM(REGEXP\_SUBSTR('http://www.ut.ac.kr','/([[:alnum:]]+\.?){3,4}?'),'/') from dual;

LTRIM(REGE

----------

www.ut.ac.

SQL> select LTRIM(REGEXP\_SUBSTR('http://www.ut.ac.kr','/([[:alnum:]]+\.?){3,4}'),'/') from dual;

LTRIM(REGEXP

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

[www.ut.ac.kr](http://www.ut.ac.kr)

**[실행결과]**

문제의 있는 정규식인 REGEXP\_SUBSTR('http://www.ut.ac.kr'**,'/([[:alnum:]]+\.?){3,4}?'**)

를쓰게 되면 /뒤의 모든 알파벳들에 겹치거나 추가로 오는 문자열 (.) 에 대해서 .이 0개나 1개 이상인게 3,4개가 오고 그게 없거나 1이상이면 출력하라 입니다.

자세히 생각해보니 앞의 정규식 **/([[:alnum:]]+\.?)** 이부분에서 마지막 kr이 뒤에 점이 없는 것을 확인했습니다. 그리고 ?를 지워준 후 정상 url 경로를 호출했습니다.

**[문제5] PT2-117, select LTRIM(REGEXP\_SUBSTR('kim@abc.net', '@([[:alnum:]]+ \.?) {3,4 }'), '@') from dual; SQL 실습**

SQL> select LTRIM(REGEXP\_SUBSTR('kim@abc.net','@([[:alnum:]]+\.?){3,4}'),'@') from dual;

LTRIM(R

-------

abc.net

SQL> select LTRIM(REGEXP\_SUBSTR('kim@ut.ac.kr','@([[:alnum:]]+\.?){2,3}'),'@') from dual;

LTRIM(RE

--------

ut.ac.kr

**[실행결과]**

실행결과 같은 문제임 이상없음.

**[문제6] PT2-117, [^/:]+에서 : 기호를 생략되는 경우의 SQL 실습 결과 확인**

SQL> col result for a15

SQL> select REGEXP\_SUBSTR('sys/oracle@racdb:1521:rac','[^/:]+',1,2) result from dual;

RESULT

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

oracle@racdb

SQL> select REGEXP\_SUBSTR('sys/oracle@racdb:1521:rac','[^/]+',1,2) result from dual;

RESULT

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

oracle@racdb:15

21:rac

**[실행결과]**

사용법에 대해서는 하단 문제에 대해 기술했습니다.

**[문제7] PT2-117, [^/:]+의 1, 1에서 1,3을 적용하였을 경우의 SQL 실습 결과 확인**

SQL> select REGEXP\_SUBSTR('sys/oracle@racdb:1521:rac','[^/:]+',1,3) result from dual;

RESULT

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

1521

**[실행결과]**

이번 문제에서는 [^문자] 형태에 대한 사용법을 알았습니다. 이는 [^:] 이것을 예로 들면 :을 기준으로 문자열을 슬라이싱하여 번호를 매기고 뒤의 인자 1,3은 첫번째부터 스캔하고 슬라이싱된 3번째 문자열인 rac를 출력하자는 의미입니다.

이게 [^:/] 이 되면서 문자의 슬라이싱은 4개로 이루어집니다. 기준은 /과 : 이기 때문입니다. 그래서 이기준 1번째부터 스캔해서 3번째것을 보면 1521이 정상출력됩니다.

**[문제8] PT2-124, 예제 Test SQL 실습**

SQL> select ename, REGEXP\_COUNT(ename,'f',1,'i') RESULT1, REGEXP\_COUNT(ename,'t{2}',1,'i') RESULT2, REGEXP\_COUNT(ename,'(l)(l)',1,'i') RESULT3 from emp;

ENAME RESULT1 RESULT2 RESULT3

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

KING 0 0 0

BLAKE 0 0 0

CLARK 0 0 0

JONES 0 0 0

SCOTT 0 1 0

FORD 1 0 0

SMITH 0 0 0

ALLEN 0 0 1

WARD 0 0 0

MARTIN 0 0 0

TURNER 0 0 0

ADAMS 0 0 0

JAMES 0 0 0

MILLER 0 0 1

Tiger 0 0 0

Cat 0 0 0

16 rows selected.

**[실행결과]**

이 문제에서는 REGEXP\_COUNT 함수에 대해서 배웠습니다. ➊번째 인자에 대해서 ➋정규식을 이용해 정규식에 해당하는 문자를 인식한 후 ➌그 정규식에 해당하는 문자열이 몇 개인지 카운트합니다. 디폴트로 첫번째부터 스캔하며 ➍ ‘i’ 와 같은 대표적인 기호로 구분합니다. (i의 뜻, 대 소문자를 서로 거르지말고 a이면 A도 같이 스캔해라. 라는 뜻입니다.

**[문제9] PT3-31, 8가지 유형의 Group By 함수 SQL 실습**



SQL> select job, null deptno, null empno, round(avg(sal),-1), max(hiredate) from emp group by job union all

2 select null job, deptno, null empno, round(avg(sal),-1), max(hiredate) from emp group by deptno union all

3 select null job, deptno, empno, round(avg(sal),-1), max(hiredate) from emp group by deptno,empno;

JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

3300

CLERK 1040 23-MAY-87

SALESMAN 1400 28-SEP-81

PRESIDENT 5000 17-NOV-96

MANAGER 2760 04-FEB-01

ANALYST 3000 17-JUN-03

30 1770 05-JAN-91

20 2180 23-MAY-87

10 3090 23-JAN-03

30 7654 1250 28-SEP-81

10 1000 3600

10 7839 5000 17-NOV-96

30 7900 950 12-MAR-81

30 7698 2850 05-JAN-91

10 7782 2450 06-SEP-99

20 7566 2980 04-FEB-01

30 7844 1500 09-AUG-81

20 7902 3000 12-MAR-81

30 2000 3000

20 7788 3000 17-JUN-03

20 7369 800 01-DEC-07

10 7934 1300 23-JAN-03

30 7521 1250 22-FEB-81

20 7876 1100 23-MAY-87

30 7499 1600 20-FEB-81

25 rows selected.

SQL> select job, deptno, empno, round(avg(sal),-1), max(hiredate) from emp group by grouping sets(job,deptno,(deptno,empno));

JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

10 1000 3600

10 7782 2450 06-SEP-99

10 7839 5000 17-NOV-96

10 7934 1300 23-JAN-03

10 3090 23-JAN-03

20 7369 800 01-DEC-07

20 7566 2980 04-FEB-01

20 7788 3000 17-JUN-03

20 7876 1100 23-MAY-87

20 7902 3000 12-MAR-81

20 2180 23-MAY-87

30 2000 3000

30 7900 950 12-MAR-81

30 7499 1600 20-FEB-81

30 7521 1250 22-FEB-81

30 7654 1250 28-SEP-81

30 7698 2850 05-JAN-91

30 7844 1500 09-AUG-81

30 1770 05-JAN-91

3300

CLERK 1040 23-MAY-87

SALESMAN 1400 28-SEP-81

PRESIDENT 5000 17-NOV-96

MANAGER 2760 04-FEB-01

ANALYST 3000 17-JUN-03

25 rows selected.



SQL> select job, deptno, empno, round(avg(sal),-1), max(hiredate) from emp group by grouping sets((job,deptno,empno));

JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

SALESMAN 30 7521 1250 22-FEB-81

CLERK 20 7876 1100 23-MAY-87

10 1000 3600

CLERK 10 7934 1300 23-JAN-03

PRESIDENT 10 7839 5000 17-NOV-96

MANAGER 10 7782 2450 06-SEP-99

ANALYST 20 7902 3000 12-MAR-81

SALESMAN 30 7844 1500 09-AUG-81

CLERK 30 7900 950 12-MAR-81

SALESMAN 30 7654 1250 28-SEP-81

ANALYST 20 7788 3000 17-JUN-03

CLERK 20 7369 800 01-DEC-07

30 2000 3000

MANAGER 30 7698 2850 05-JAN-91

MANAGER 20 7566 2980 04-FEB-01

SALESMAN 30 7499 1600 20-FEB-81

16 rows selected.

SQL> select job, deptno, empno, round(avg(sal),-1), max(hiredate) from emp group by job,deptno,empno;

JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

SALESMAN 30 7521 1250 22-FEB-81

CLERK 20 7876 1100 23-MAY-87

10 1000 3600

CLERK 10 7934 1300 23-JAN-03

PRESIDENT 10 7839 5000 17-NOV-96

MANAGER 10 7782 2450 06-SEP-99

ANALYST 20 7902 3000 12-MAR-81

SALESMAN 30 7844 1500 09-AUG-81

CLERK 30 7900 950 12-MAR-81

SALESMAN 30 7654 1250 28-SEP-81

ANALYST 20 7788 3000 17-JUN-03

CLERK 20 7369 800 01-DEC-07

30 2000 3000

MANAGER 30 7698 2850 05-JAN-91

MANAGER 20 7566 2980 04-FEB-01

SALESMAN 30 7499 1600 20-FEB-81

16 rows selected.



JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

3300

CLERK 1040 23-MAY-87

SALESMAN 1400 28-SEP-81

PRESIDENT 5000 17-NOV-96

MANAGER 2760 04-FEB-01

ANALYST 3000 17-JUN-03

10 1000 3600

10 7782 2450 06-SEP-99

10 7839 5000 17-NOV-96

10 7934 1300 23-JAN-03

10 3090 23-JAN-03

20 7369 800 01-DEC-07

20 7566 2980 04-FEB-01

20 7788 3000 17-JUN-03

20 7876 1100 23-MAY-87

20 7902 3000 12-MAR-81

20 2180 23-MAY-87

30 2000 3000

30 7900 950 12-MAR-81

30 7499 1600 20-FEB-81

30 7521 1250 22-FEB-81

30 7654 1250 28-SEP-81

30 7698 2850 05-JAN-91

30 7844 1500 09-AUG-81

30 1770 05-JAN-91

2230 23-MAY-87

SQL> select job, deptno, empno, round(avg(sal),-1), max(hiredate) from emp group by grouping sets(job,rollup(deptno,empno));

JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

10 1000 3600

10 7782 2450 06-SEP-99

10 7839 5000 17-NOV-96

10 7934 1300 23-JAN-03

10 3090 23-JAN-03

20 7369 800 01-DEC-07

20 7566 2980 04-FEB-01

20 7788 3000 17-JUN-03

20 7876 1100 23-MAY-87

20 7902 3000 12-MAR-81

20 2180 23-MAY-87

30 2000 3000

30 7900 950 12-MAR-81

30 7499 1600 20-FEB-81

30 7521 1250 22-FEB-81

30 7654 1250 28-SEP-81

30 7698 2850 05-JAN-91

30 7844 1500 09-AUG-81

30 1770 05-JAN-91

2230 23-MAY-87

3300

CLERK 1040 23-MAY-87

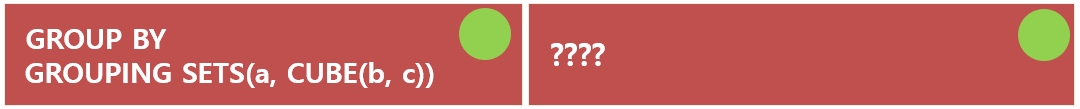
SALESMAN 1400 28-SEP-81

PRESIDENT 5000 17-NOV-96

MANAGER 2760 04-FEB-01

ANALYST 3000 17-JUN-03

26 rows selected.



SQL> select job, deptno, empno, round(avg(sal),-1), max(hiredate) from emp group by grouping sets(job,cube(deptno, empno));

JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

10 1000 3600

30 2000 3000

30 7900 950 12-MAR-81

20 7369 800 01-DEC-07

30 7499 1600 20-FEB-81

30 7521 1250 22-FEB-81

20 7566 2980 04-FEB-01

30 7654 1250 28-SEP-81

30 7698 2850 05-JAN-91

10 7782 2450 06-SEP-99

20 7788 3000 17-JUN-03

10 7839 5000 17-NOV-96

30 7844 1500 09-AUG-81

20 7876 1100 23-MAY-87

20 7902 3000 12-MAR-81

10 7934 1300 23-JAN-03

1000 3600

2000 3000

7900 950 12-MAR-81

7369 800 01-DEC-07

7499 1600 20-FEB-81

7521 1250 22-FEB-81

7566 2980 04-FEB-01

7654 1250 28-SEP-81

7698 2850 05-JAN-91

7782 2450 06-SEP-99

7788 3000 17-JUN-03

7839 5000 17-NOV-96

7844 1500 09-AUG-81

7876 1100 23-MAY-87

7902 3000 12-MAR-81

7934 1300 23-JAN-03

10 3090 23-JAN-03

20 2180 23-MAY-87

30 1770 05-JAN-91

2230 23-MAY-87

3300

CLERK 1040 23-MAY-87

SALESMAN 1400 28-SEP-81

PRESIDENT 5000 17-NOV-96

MANAGER 2760 04-FEB-01

ANALYST 3000 17-JUN-03

42 rows selected.

SQL> select job, null deptno, null empno, round(avg(sal),-1), max(hiredate) from emp group by job union all

2 select null job, deptno, empno, round(avg(sal),-1), max(hiredate) from emp group by cube(deptno, empno);

JOB DEPTNO EMPNO ROUND(AVG(SAL),-1) MAX(HIRED

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

3300

CLERK 1040 23-MAY-87

SALESMAN 1400 28-SEP-81

PRESIDENT 5000 17-NOV-96

MANAGER 2760 04-FEB-01

ANALYST 3000 17-JUN-03

2230 23-MAY-87

1000 3600

2000 3000

7900 950 12-MAR-81

7369 800 01-DEC-07

7499 1600 20-FEB-81

7521 1250 22-FEB-81

7566 2980 04-FEB-01

7654 1250 28-SEP-81

7698 2850 05-JAN-91

7782 2450 06-SEP-99

7788 3000 17-JUN-03

7839 5000 17-NOV-96

7844 1500 09-AUG-81

7876 1100 23-MAY-87

7902 3000 12-MAR-81

7934 1300 23-JAN-03

10 3090 23-JAN-03

10 1000 3600

10 7782 2450 06-SEP-99

10 7839 5000 17-NOV-96

10 7934 1300 23-JAN-03

20 2180 23-MAY-87

20 7369 800 01-DEC-07

20 7566 2980 04-FEB-01

20 7788 3000 17-JUN-03

20 7876 1100 23-MAY-87

20 7902 3000 12-MAR-81

30 1770 05-JAN-91

30 2000 3000

30 7900 950 12-MAR-81

30 7499 1600 20-FEB-81

30 7521 1250 22-FEB-81

30 7654 1250 28-SEP-81

30 7698 2850 05-JAN-91

30 7844 1500 09-AUG-81

42 rows selected.

**[실행결과]**

여기서 중요한 포인트는 ROLLUP 함수와 CUBE 함수 입니다.

ROLLUP은 M+1의 쿼리를 날리며, 예를 들어 ROLLUP(COL,COL2) 로 GROUPBY를 사용한다면

COL로 한번 그룹핑 실행하고, COL, COL2 두개로 그룹핑을 실행합니다.

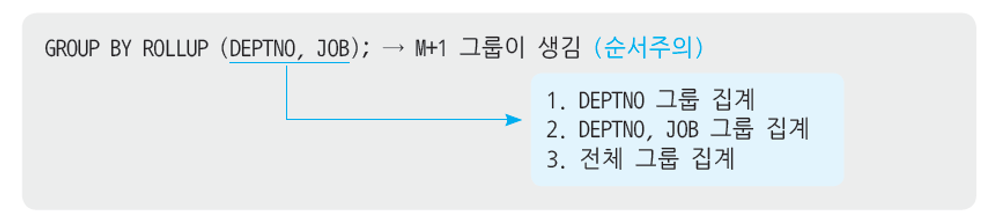
마지막으로 그룹핑 없이 전체 계수를 출력합니다.

CUBE는 똑같이 인자를 COL COL2로 한다면 COL로 한번 그룹핑하고 COL2를 한번 그룹핑하고, COL, COL2로 그룹핑하고

마지막으로 전체 소계를 출력합니다.

그리고 각각의 계산 테이블에 UNION ALL을 하는 쿼리를 날립니다.

**ROLLUP**



**CUBE**

텍스트이(가) 표시된 사진

자동 생성된 설명