**[문제1] PIVOT 기능을 사용하여 달력 만들기**

1. 테이블 만들기

create table cal (day varchar(3), weekno number);

1. 서브테이블 값 채우기

INSERT ALL into cal(day, weekno) values ('SUN',1)

into cal (day, weekno) values ('MON',1)

into cal (day, weekno) values ('TUE',1)

into cal (day, weekno) values ('WED',1)

into cal (day, weekno) values ('THU',1)

into cal (day, weekno) values ('FRI',1)

into cal (day, weekno) values ('SAT',1)

SELECT \* FROM DUAL ;

INSERT ALL into cal(day, weekno) values ('SUN',2)

into cal (day, weekno) values ('MON',2)

into cal (day, weekno) values ('TUE',2)

into cal (day, weekno) values ('WED',2)

into cal (day, weekno) values ('THU',2)

into cal (day, weekno) values ('FRI',2)

into cal (day, weekno) values ('SAT',2)

SELECT \* FROM DUAL ;

INSERT ALL into cal(day, weekno) values ('SUN',3)

into cal (day, weekno) values ('MON',3)

into cal (day, weekno) values ('TUE',3)

into cal (day, weekno) values ('WED',3)

into cal (day, weekno) values ('THU',3)

into cal (day, weekno) values ('FRI',3)

into cal (day, weekno) values ('SAT',3)

SELECT \* FROM DUAL ;

INSERT ALL into cal(day, weekno) values ('SUN',5)

into cal (day, weekno) values ('MON',5)

into cal (day, weekno) values ('TUE',5)

SELECT \* FROM DUAL ;

1. 서브테이블 확인하기 (ROWNUM 추가로 데이터 출력해보기)

create table calx as(select rownum dayno, day, weekno from cal);

DAYNO DAY WEEKNO

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

1 SUN 1

2 MON 1

3 TUE 1

4 WED 1

5 THU 1

6 FRI 1

7 SAT 1

8 SUN 2

1. 형식 지정후 달력 만들기

col week for 9999;

col sun for 9999;

col mon for 9999;

col tue for 9999;

col wed for 9999;

col thu for 9999;

col fri for 9999;

col sat for 9999;

select \* from (select weekno "WEEK", day,dayno from calx)

pivot(MAX(dayno) FOR day IN('SUN' AS "SUN",

'MON' AS "MON",

'TUE' AS "TUE",

'WED' AS "WED",

'THU' AS "THU",

'FRI' AS "FRI",

'SAT' AS "SAT"))

ORDER BY "WEEK";

WEEK SUN MON TUE WED THU FRI SAT

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

1 1 2 3 4 5 6 7

2 8 9 10 11 12 13 14

3 15 16 17 18 19 20 21

4 22 23 24 25 26 27 28

5 29 30 31

**[실행결과]**

처음 실행결과에서는 MAX(DECODE(DAY,’MON’,DAYNO)) “MAX …” 이런식으로 해당 주를 출력하면서 WEEK 순으로 정렬했는데 MAX에서 최종적인 인자인 DAYNO를 넣어놓고 FOR에는 그룹화할 컬럼의 이름을 넣어서 그룹화해주게 되면 위의 결과와 같이 달력으로 순서 매핑이 가능함.

두번째로 WEEK의 값이 없으면 어떻게 될지도 보았다.

select \* from (select day,dayno from calx)

pivot(MAX(dayno) FOR day IN('SUN' AS "SUN",

'MON' AS "MON",

'TUE' AS "TUE",

'WED' AS "WED",

'THU' AS "THU",

'FRI' AS "FRI",

'SAT' AS "SAT"))

SUN MON TUE WED THU FRI SAT

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

29 30 31 25 26 27 28

WEEK를 없애게 되면 단순하게 DAY로만 그룹화되어지고 WEEK에 상관없이 최댓값을 출력하여 나열합니다. 여기서 NUMBER 기준으로 최적화가 되어있는 것 같은데 문자로 하게되면 8이 20 보다 (예를 들어) 크게 되는 경우가 있습니다. ASCII함수로 인자를 할 때, 결과값으로 MAX 를 정합니다.

**[문제2] EMP 테이블에서 부서별로 각 직급별 인원이 몇 명인 지 계산해서 출력하세요**

select deptno,

COUNT(DECODE(job,'CLERK','0')) "CLERK",

COUNT(DECODE(job,'MANAGER','0')) "MANAGER",

COUNT(DECODE(job,'PRESIDENT','0')) "PRESIDENT",

COUNT(DECODE(job,'ANALYST','0')) "ANALYST",

COUNT(DECODE(job,'SALESMAN','0')) "SALESMAN"

FROM emp

GROUP BY deptno

ORDER BY deptno;

DEPTNO CLERK MANAGER PRESIDENT ANALYST SALESMAN

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

10 1 1 1 0 0

20 2 1 0 2 0

30 1 1 0 0 4

**[실행결과]**

COUNT를 이용하여 PIVOT말고 단순 GROUP BY 로 그룹화 했을 때의 결과인 점 입니다.

이전 문제 실행결과에 기술.

**[문제3] 부서별로 인원 수 하고 각 인원수 별로 급여 합계까지 함께 출력하는 화면**

COL C\_COUNT FOR 99999999

COL C\_SUM FOR 99999999

COL M\_COUNT FOR 99999999

COL M\_SUM FOR 99999999

COL P\_COUNT FOR 99999999

COL P\_SUM FOR 99999999

COL A\_COUNT FOR 99999999

COL A\_SUM FOR 99999999

COL S\_COUNT FOR 99999999

COL S\_SUM FOR 99999999

COL C\_AVG FOR 99999999

COL M\_AVG FOR 99999999

COL P\_AVG FOR 99999999

COL A\_AVG FOR 99999999

COL S\_AVG FOR 99999999

SELECT \* FROM (SELECT DEPTNO, JOB, EMPNO, SAL FROM EMP)

PIVOT(COUNT(EMPNO) AS COUNT,

SUM(NVL(SAL,0)) AS SUM,

AVG(NVL(SAL,0)) AS AVG FOR JOB IN ('CLERK' AS "C",

'MANAGER' AS "M",

'PRESIDENT' AS "P",

'ANALYST' AS "A",

'SALESMAN' AS "S"))

ORDER BY DEPTNO;

DEPTNO C\_COUNT C\_SUM C\_AVG M\_COUNT M\_SUM M\_AVG P\_COUNT P\_SUM P\_AVG A\_COUNT A\_SUM A\_AVG S\_COUNT S\_SUM S\_AVG

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

10 1 1300 1300 1 2450 2450 1 5000 5000 0 0

20 2 1900 950 1 2975 2975 0 2 6000 3000 0

30 1 950 950 1 2850 2850 0 0 4 5600 1400

**[실행결과]**

하나의 항목으로 그룹화를 시키면 PIVOT 사용법을 그대로 ALIAS 로 SUM.. 등 이름을 정해주고 JOB이 해당 값과 일치하면 M\_SUM 등으로 컬럼이름이 매핑이 되는 것을 확인할 수 있습니다.

PIVOT의 사용법은 이전 문제를 참조합니다.

**[문제4] emp 테이블에서 사원들의 empno, ename, sal, 급여순위를 출력하세요.**

SELECT EMPNO, ENAME, SAL,

RANK() OVER (ORDER BY SAL ASC) AS RANK\_ASC,

RANK() OVER (ORDER BY SAL DESC) AS RANK\_DESC,

RANK() OVER (ORDER BY SAL ) AS RANK

FROM EMP;

EMPNO ENAME SAL RANK\_ASC RANK\_DESC RANK

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

7839 KING 5000 16 1 16

1000 Tiger 3600 15 2 15

7902 FORD 3000 12 3 12

2000 Cat 3000 12 3 12

7788 SCOTT 3000 12 3 12

7566 JONES 2975 11 6 11

7698 BLAKE 2850 10 7 10

7782 CLARK 2450 9 8 9

7499 ALLEN 1600 8 9 8

7844 TURNER 1500 7 10 7

7934 MILLER 1300 6 11 6

7654 MARTIN 1250 4 12 4

7521 WARD 1250 4 12 4

7876 ADAMS 1100 3 14 3

7900 JAMES 950 2 15 2

7369 SMITH 800 1 16 1

**[실행결과]**

RANK() OVER을 사용하게 되면 순위별로 확인이 가능합니다. ORDER BY SAL 급여순으로 내림차순, 오름차순으로 순위를 출력한 결과를 나타냅니다.

**[문제5] emp 테이블을 조회하여 사번, 이름, 급여, 부서번호, 부서별 급여순위를 출력하세요**

SELECT EMPNO, ENAME, SAL, DEPTNO,

RANK() OVER (PARTITION BY DEPTNO ORDER BY SAL DESC) "RANK" FROM EMP;

EMPNO ENAME SAL DEPTNO RANK

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

7839 KING 5000 10 1

1000 Tiger 3600 10 2

7782 CLARK 2450 10 3

7934 MILLER 1300 10 4

7788 SCOTT 3000 20 1

7902 FORD 3000 20 1

7566 JONES 2975 20 3

7876 ADAMS 1100 20 4

7369 SMITH 800 20 5

2000 Cat 3000 30 1

7698 BLAKE 2850 30 2

7499 ALLEN 1600 30 3

7844 TURNER 1500 30 4

7654 MARTIN 1250 30 5

7521 WARD 1250 30 5

7900 JAMES 950 30 7

**[실행결과]**

전 문제를 가지고 PARTITION을 하여 부서별로도 RANK출력이 가능합니다.

**[문제6] PT3-57, 부서별 급여 순위를 오름차순으로 구분하고 사원번호, 이름, 급여, 소속부서명, 순위를 출력하는 SQL 문을 실습하라.**

SQL> select empno, ename, sal, deptno, rank() over(partition by deptno order by sal asc) "rank" from emp;

EMPNO ENAME SAL DEPTNO rank

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

7934 MILLER 1300 10 1

7782 CLARK 2450 10 2

1000 Tiger 3600 10 3

7839 KING 5000 10 4

7369 SMITH 800 20 1

7876 ADAMS 1100 20 2

7566 JONES 2975 20 3

7902 FORD 3000 20 4

7788 SCOTT 3000 20 4

7900 JAMES 950 30 1

7521 WARD 1250 30 2

7654 MARTIN 1250 30 2

7844 TURNER 1500 30 4

7499 ALLEN 1600 30 5

7698 BLAKE 2850 30 6

2000 Cat 3000 30 7

**[실행결과]**

순위를 구하기 위해서는 rank로 써주는데 over절은 이 다음 문제에서 이해하는 과정을 서술했습니다. (이상 없음)

**[문제7] PT3-61, emp 테이블에서 10번 부서에 소속된 사원들의 사원번호, 사원명, 월급, 월급누계를 출력하는 SQL 문을 실습하라.**

SQL> select empno, ename, sal, deptno ,sum(sal) over(partition by deptno order by sal) "TOTAL" from emp where deptno=10;

EMPNO ENAME SAL DEPTNO TOTAL

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

7934 MILLER 1300 10 1300

7782 CLARK 2450 10 3750

1000 Tiger 3600 10 7350

7839 KING 5000 10 12350

**[실행결과]**

월급의 누계 합계를 구할 땐 sum() over()문으로 구현 하였습니다. Sum()에 sal 을 넣어 어떤 컬럼의 누계합계를 구할지 적고 over절에 order by와 partition by를 쓸 수 있었습니다. 그래서 deptno로 부서에 소속된 사원들 별로 추출하며 order를 sal을 기본 오름차순으로 정렬하는 order by를 씀과 동시에 where 절에서는 deptno = 10; 이라는 조건을 넣어줘서 그 절만 출력했습니다.

**[문제8] PT3-62, emp 테이블에서 부서별로 소속된 사원들의 사원번호, 사원명, 월급, 월급누계를 출력하는 SQL 문을 실습하라.**

SQL> select empno, ename, sal, deptno ,sum(sal) over(partition by deptno order by sal) "TOTAL" from emp;

EMPNO ENAME SAL DEPTNO TOTAL

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

7934 MILLER 1300 10 1300

7782 CLARK 2450 10 3750

1000 Tiger 3600 10 7350

7839 KING 5000 10 12350

7369 SMITH 800 20 800

7876 ADAMS 1100 20 1900

7566 JONES 2975 20 4875

7902 FORD 3000 20 10875

7788 SCOTT 3000 20 10875

7900 JAMES 950 30 950

7521 WARD 1250 30 3450

7654 MARTIN 1250 30 3450

7844 TURNER 1500 30 4950

7499 ALLEN 1600 30 6550

7698 BLAKE 2850 30 9400

2000 Cat 3000 30 12400

**[실행결과]**

위 문제에서 where 절을 제거하여 모든 부서별로 나타내어 출력했다. (이상 없음)

처음 이해가 어려웠던 것은 테이블에서 GROUP BY로 매핑하는 것과 PIVOT을 써서 테이블로 매핑해주는 것이 정말 헷갈렸습니다. 아무리 많이 쿼리를 쳐도 이해가 되지 않아서

select \* from (select weekno "WEEK", day,dayno from calx)

pivot(MAX(dayno) FOR day IN('SUN' AS "SUN",

'MON' AS "MON",

'TUE' AS "TUE",

'WED' AS "WED",

'THU' AS "THU",

'FRI' AS "FRI",

'SAT' AS "SAT"))

WEEK SUN MON TUE WED THU FRI SAT

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

1 1 2 3 4 5 6 7

2 8 9 10 11 12 13 14

4 22 23 24 25 26 27 28

5 29 30 31

3 15 16 17 18 19 20 21

이런식으로 정렬도 하지 않아보고 위의 문제에서 WEEK 컬럼을 빼주기도 하면서 이해를 했습니다.