<실습1>

1. 현재 접속한 DB USER가 소유하고 있는 테이블들을 조회

SQL> select \*

from tab;

2. 이름, 월급, 커미션, 월급 + 커미션 출력

SQL> select ename, sal, sal + comm

from emp;

3. 이름과 연봉(sal \* 12) 을 계산하고 컬럼별칭을 사용해서 한글로 이름, 연봉으로 출력

SQL> select ename 이름, sal \* 12 연봉

from emp;

4. 부서번호를 출력하는데 중복을 제거

SQL> select distinct deptno

from emp;

5. 아래와 같이 결과를 출력

KING 의 직업은 PRESIDENT 입니다.

SQL> select ename||' 의 직업은 '||job||' 입니다. '

from emp;

6. emp 테이블의 구조를 확인

SQL> describe emp 또는 desc emp

7. 1981 년 4월 2일에 입사한 사원의 이름과 입사일을 출력

SQL> select ename, hiredate

from emp

where hiredate ='1981/04/02';

8. 월급이 1000 에서 3000 사이가 아닌 사원들의 이름과 월급을 출력

SQL> select ename, sal

from emp

where sal not between 1000 and 3000;

9. 이름의 세번째 철자가 L 인 사원들의 이름을 출력

SQL> select ename

from emp

where ename like '\_\_L%';

10. 이름의 첫번째 철자가 A 로 시작하지 않는 사원들의 이름을 출력

SQL> select ename

from emp

where ename not like 'A%';

11. 이름의 두번째 철자와 세번째 철자가 % 인 사원의 이름을 출력

-->전체 자리수가 3자리라면

SQL> select ename

from emp

where ename like '\_\%\%' escape '\' ;

-->전체 자리수가 3자리이상일수도 있다면

SQL> select ename

from emp

where ename like '\_\%\%%' escape '\' ;

12. 커미션이 null 이 아닌 사원들의 이름과 커미션을 출력

SQL> select ename, comm

from emp

where comm is not null ;

13. 직업이 SALESMAN , ANALYST 가 아닌 사원들의 이름과 직업을 출력

SQL> select ename , job

from emp

where job not in ('SALESMAN','ANALYST');

14. 직업이 SALESMAN 이고 월급이 1000 이상인 사원들의 이름과 월급을 출력

SQL> select ename, sal

from emp

where job='SALESMAN' and sal >= 1000;

15. 이름과 연봉(sal\*12) 을 출력하는데 연봉이 36000 이상인 사원들의 이름과 연봉을 출력하되, 결과가 출력될때 컬럼명이 한글로 이름, 연봉으로 출력

SQL> select ename as 이름, sal \*12 as 연봉

from emp

where sal \* 12 >= 36000;

16. 직업이 SALESMAN 인 사원들의 이름과 직업과 월급을 출력하는데 월급이 높은 사원부터 출력

SQL> select ename, job, sal

from emp

where job='SALESMAN'

order by sal desc;

<실습2>

1. 부서번호가 10, 20번이 아닌 사원들의 이름과 입사일을 출력하는데 최근에 입사한 사원부터 출력

SQL> select ename, hiredate

from emp

where deptno not in ( 10, 20)

order by hiredate desc;

2. 이름의 철자의 갯수가 6개 이상인 사원들의 이름과 철자의 갯수를 출력

SQL> select ename, length(ename)

from emp

where length(ename) >= 6;

3. 이름에 철자중에 L 자를 포함하고 있는 사원들의 이름을 출력

SQL> select ename

from emp

where instr(ename,'L') > 0;

SQL> select ename

from emp

where ename like '%L%';

4. 이름과 월급을 출력하는데 월급의 숫자 0 을 \* 로 출력

SQL> select ename, replace(sal,0,'\*')

from emp;

5. 이름, 입사한 날짜부터 오늘까지 총 몇달 근무했는지 출력

select ename, trunc(months\_between(sysdate,hiredate))+1 "1달 초과",

trunc(months\_between(sysdate,hiredate)) "1달 미만"

from emp;

6. 오늘날짜에서 100 달뒤의 날짜를 출력

SQL> select add\_months(sysdate,100)

from dual;

7. 오늘날짜에서 앞으로 돌아올 금요일의 날짜를 출력

SQL> select next\_day( sysdate,'금요일')

from dual;

SQL> select next\_day( sysdate,6)

from dual;

8. 오늘부터 앞으로 200 달뒤에 돌아올 월요일의 날짜를 출력

SQL> select next\_day(add\_months(sysdate,200) ,'월요일')

from dual;

SQL> select next\_day(add\_months(sysdate,200) ,'MONDAY')

from dual;

SQL> select next\_day(add\_months(sysdate,200) ,2)

from dual;

9. 오늘부터 이번달 말일까지 총 몇일 남았는지 출력

SQL> select last\_day(sysdate) - sysdate

from dual;

10. 오늘이 무슨요일인 출력하시오 !

SQL> select to\_char(sysdate,'day') from dual;

SQL> select to\_char(sysdate,'DY') from dual;

SQL> select to\_char(sysdate,'D') from dual;

11. 이름, 입사일, 입사한 요일을 출력하시오

SQL> select ename, hiredate, to\_char(hiredate,'day')

from emp;

12. 이름, 입사한 년도를 출력하시오 !

SQL> select ename, to\_char(hiredate,'RR') from emp;

SQL> select ename, to\_char(hiredate,'YY') from emp;

SQL> select ename, to\_char(hiredate,'RRRR') from emp;

SQL> select ename, to\_char(hiredate,'YYYY') from emp;

SQL> select ename, extract(year from hiredate) from emp;

13. 직업이 SALESMAN 이 아닌 사원들의 이름과 연봉을 출력하는데 연봉이 높은 사원부터 출력하고 이름과 연봉을 출력할때에 아래와 같이 결과가 출력되게하시오 !

SCOTT 의 연봉은 36000 입니다.

:

:

SQL> SELECT ENAME||' 의 연봉은 '||SAL\*12||' 입니다'

FROM EMP

WHERE JOB <> 'SALESMAN'

ORDER BY SAL\*12 DESC;

14. 이름과 월급을 출력하는데 월급을 출력할때에 천단위를 부여하시오

(예: 3000 ---> 3,000 )

SQL> select ename, to\_char(sal,'999,999')

from emp;

15. 1981 년도에 입사한 사원들의 이름과 입사일을 출력하시오 !

SQL> select ename, hiredate

from emp

where hiredate between TO\_DATE('81/01/01','RR/MM/DD')

and to\_date('81/12/31','RR/MM/DD');

<test>

insert into emp (empno, ename, sal, hiredate)

values (9999,'test',3000,to\_date('1981/12/31 13:12:20','yyyy/mm/dd hh24:mi:ss'));

select \* from emp;

select ename, hiredate

from emp

where hiredate >= TO\_DATE('81/01/01','RR/MM/DD')

and hiredate < to\_date('81/12/31','RR/MM/DD')+1;

SQL> select ename, hiredate

from emp

where hiredate LIKE '81%';

SQL> select ename, hiredate

from emp

where EXTRACT(YEAR FROM hiredate) = 1981;

16. 이름, 월급, 커미션을 출력하는데 커미션이 null인 사원들은 0 으로 출력되게하시오 !

SQL> select ename, sal, nvl(comm,0) from emp;

17. 이름, 월급, 커미션을 출력하는데 커미션이 null인 사원들은 no comm이란 글씨로 출력

SQL> select ename, sal, nvl( to\_char(comm) , 'no comm')

from emp;

18. 이름, 월급, 직업, 보너스를 출력하는데 보너스가 만약에 직업이 SALESMAN 이면 자신의 월급의 20%를 보너스로 출력하고 만약에 직업이 ANALYST 면 자신의 월급의 40%를 보너스로 출력하고 만약에 직업이 CLERK 이면 자신의 월급의 10%를 보너스로 출력하고 나머지 직업인 사원들은 0 으로 출력되게하시오 !(decode 사용)

SQL> select ename, sal, job,

decode( job, 'SALESMAN', sal \* 0.2 ,

'ANALYST', sal \* 0.4,

'CLERK', sal \* 0.1, 0 ) as bonus

from emp;

19. 이름, 입사한 년도(4자리), 보너스를 출력하는데입사한 년도가 1981 년이면 보너스를 900 으로 출력 입사한 년도가 1982 년이면 보너스를 200 으로 출력 하고 나머지 년도는 0 으로 출력하시오 ! ( case 문으로 수행 )

SQL> select ename, to\_char(hiredate,'RRRR'),

case when to\_char(hiredate,'RRRR')='1981' then 900

when to\_char(hiredate,'RRRR')='1982' then 200

else 0 end as bonus

from emp;

SQL> select ename , to\_char(hiredate,'RRRR'),

case to\_char(hiredate,'RRRR') when '1981' then 900

when '1982' then 200

else 0 end as bonus

from emp;

<실습3>

1. 직업이 SALESMAN 인 사원들중에서의 최대월급을 출력하시오 !

SQL> select max(sal) from emp where job='SALESMAN';

2. 부서번호, 부서번호별 최소월급을 출력하는데 부서번호별 최소월급이 높은것부터 출력하시오 !

SQL> select deptno, min(sal)

from emp

group by deptno

order by min(sal) desc ;

3. 입사한 년도(4자리), 입사한 년도별 최대월급을 출력하는데 입사한 년도가 1981 년은 제외하고 출력하고 그리고 입사한 년도별 최대월급을 출력할때에 천단위를 부여하고 그리고 입사한 년도별 최대월급이 작은것부터 출력되게하고 그리고 컬럼별칭이 각각 한글로 입사한 년도, 최대월급 이렇게 출력되게하시오 !

SQL> SELECT TO\_CHAR(HIREDATE,'YYYY') "입사년도",

TO\_CHAR(MAX(SAL),'999,999') "최대월급"

FROM EMP

WHERE TO\_CHAR(HIREDATE,'YYYY') <> '1981'

GROUP BY TO\_CHAR(HIREDATE,'YYYY')

ORDER BY 2;

4. 사원 테이블의 커미션의 평균값을 출력하시오 !

SQL> select avg(comm) from emp;

SQL> select avg( nvl(comm,0) ) from emp;

5. 직업, 직업별 토탈월급을 출력하는데 직업이 SALESMAN 은 제외하고 출력하고 그리고 직업별 토탈월급이 4000 이상인것만 출력하고 직업별 토탈월급이 높은것부터 출력하시오 !

<좋음>

SQL> select job, sum(sal)

from emp

where job !='SALESMAN'

group by job

having sum(sal) >= 4000

order by sum(sal) desc;

<별로>

SQL> select job, sum(sal)

from emp

group by job

having sum(sal) >= 4000 and job !='SALESMAN'

order by sum(sal) desc;

6. 사원 테이블에 전체 인원수를 출력하시오 !

<PK이용>

SQL> select count(empno)

from emp;

SQL> select count(\*)

from emp;

7. 직업, 직업별 인원수를 출력하시오 !

SQL> select job, count(\*)

from emp

group by job;

8. 직업별 인원수가 4명 이상인 직업만 출력하시오 !

SQL> select job, count(\*)

from emp

group by job

having count(\*) >= 4;

9. 사원테이블의 직업의 종류가 몇가지가 있는지 출력하시오 !

SQL> select count(distinct job)

from emp;

10. 입사한 년도(4자리) , 입사한 년도별 인원수를 출력하시오 !

SQL> select to\_char(hiredate,'RRRR') , count(\*)

from emp

group by to\_char(hiredate,'RRRR');

11. 직업별 토탈월급중 최대값을 출력하시오 !

SQL> select max( sum(sal) )

from emp

group by job;

12. 부서번호, 부서번호별 토탈월급을 출력하시오 !

( 결과를 가로로 출력되게하시오 !)

10 20 30 <-- 컬럼명

8750 10875 9400 <-- 결과 data

SQL> select sum( decode(deptno,10,sal) ) as "10",

sum( decode(deptno,20,sal) ) as "20",

sum( decode(deptno,30,sal) ) as "30"

from emp;

SQL> select \*

from ( select deptno, sal from emp )

pivot ( sum(sal) for deptno in (10,20,30) ) ;

13. 부서번호, 부서번호가 10번이면 월급이 출력되게하고 10번이 아니면 0이 출력되게하시오 ! (decode 를 이용해서)

SQL> select DEPTNO, decode(deptno,10,sal,0) as "10"

from emp;

14. 부서번호별 입사한 년도별 인원수를 출력하시오 !

SQL> SELECT DISTINCT TO\_CHAR(hiredate,'YYYY')

FROM EMP;-->1980, 1981,1982,1987년 있음

DEPTNO 1980 1981 1982 1987

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

30 0 6 0 0

20 1 2 0 0

10 0 2 1 0

/\*비효율\*/

SQL> select deptno,

sum(decode(to\_char(hiredate,'RRRR'),'1980',1,0)) "1980",

sum(decode(to\_char(hiredate,'RRRR'),'1981',1,0)) "1981",

sum(decode(to\_char(hiredate,'RRRR'),'1982',1,0)) "1982",

sum(decode(to\_char(hiredate,'RRRR'),'1987',1,0)) "1987"

from emp

group by deptno;

/\*효율\*/

SQL> select deptno,

NVL(sum(decode(to\_char(hiredate,'RRRR'),'1980',1)),0) "1980",

NVL(sum(decode(to\_char(hiredate,'RRRR'),'1981',1)),0) "1981",

NVL(sum(decode(to\_char(hiredate,'RRRR'),'1982',1)),0) "1982",

NVL(sum(decode(to\_char(hiredate,'RRRR'),'1987',1)),0) "1987"

from emp

group by deptno;

SQL> SELECT \*

FROM (SELECT DEPTNO, TO\_CHAR(HIREDATE,'YYYY') YEAR

FROM EMP)

PIVOT (COUNT(YEAR) FOR YEAR IN

('1980','1981','1982','1987') );

\*\*동적 PIVOT

SQL> SELECT \*

from ( select job, sal from emp )

pivot XML ( sum(sal) for JOB in

(select distinct job from emp ) );

15. 직업중에 인원수가 가장 많은 직업이 무엇인가 ?

SQL> select job, count(\*)

from emp

group by job

having count(\*) = (select max(count(\*))

from emp group by job);

<실습4>

1. 이름과 지역코드를 출력하시오 !

SQL> select ename, loc\_code

from emp, dept

where emp.deptno = dept.deptno;

☞ 여러개의 테이블을 조인한 결과가 제대로 나오려면

where 절에 유효한 조인조건을 주면된다.

2. 이름, 지역코드, 월급, 부서번호를 출력하시오 !

SQL> select emp.ename, dept.loc\_code, emp.sal, emp.deptno

from emp , dept

where emp.deptno = dept.deptno;

☞ 조인시에 컬럼명 앞에 테이블명 또는 테이블 별칭을 써줘야 더 성능이 좋다.

♣ Table alias 를 사용

SQL> select e.ename, d.loc\_code, e.sal, e.deptno

from emp e, dept d

where e.deptno = d.deptno;

3. 직업이 SALESMAN 인 사원들의 이름과 직업과 월급과 지역코드를 출력하시오!

SQL> select e.ename, e.job, e.sal, d.loc\_code

from emp e, dept d

where e.deptno = d.deptno -- 조인 조건

and e.job ='SALESMAN'; -- 검색 조건

4. 월급이 1000 에서 3000 사이인 사원들의 이름과 월급과 직업과 지역코드를 출력하시오

SQL> select e.ename , e.sal, e.job, d.loc\_code

from emp e, dept d

where e.deptno = d.deptno

and e.sal between 1000 and 3000;

5. DALLAS 에서 근무하는 사원들의 이름과 지역코드를 출력하시오 !

SQL> SELECT e.ename, d.loc\_code

from emp e ,dept d, locations l

where e.deptno = d.deptno

and d.loc\_code = l.loc\_code

and l.city ='DALLAS';

6. 도시, 도시별 근무사원들의 급여의 합계를 출력하시오 !

SQL> select l.city, sum(e.sal)

from emp e, dept d, locations l

where e.deptno = d.deptno

and d.loc\_code = l.loc\_code

group by l.city;

7. 근무도시, 도시별 인원수를 출력하는데 도시별 인원수가 3명 이상인것만 출력하고 도시별 인원수가 높은것부터 출력하시오 !

SQL> select l.city, count(e.sal)

from emp e, dept d, locations l

where e.deptno = d.deptno

and d.loc\_code = l.loc\_code

group by l.city

having count(\*) >= 3

order by 2 desc;

**8. 이름, 월급, grade(급여등급) 을 출력하시오**

**- non equi join : 조인하려는 테이블 사이의 연결고리**

**즉 조인조건이 = 이 아닌 경우**

**<test데이터>**

**SQL> UPDATE emp**

**SET sal = sal\*100**

**WHERE ename = ‘SMITH’; 🡪등급표에 없는 급여값으로 변경(80000)**

**SQL> select e.ename, e.sal, s.grade**

**from emp e, salgrade s**

**where e.sal between s.losal and s.hisal; 🡪 SMITH누락 13명만 조회됨**

**SQL> select e.ename, e.sal, nvl(s.grade,0)**

**from emp e, salgrade s**

**where e.sal between s.losal(+) and s.hisal(+);🡪SMITH포함 14명모두 나옴**

**SQL> COMMIT;**

9. 지역코드별 근무사원을 출력하시오 !

- outer join : equi join 으로는 조인관계에서 누락된 특정테이블의 데이터까지 보고자할때 사용하는 조인방법

SQL> select d.loc\_code, e.ename

from emp e, dept d

where e.deptno(+) = d.deptno;

10. 지역코드, 지역코드별 토탈월급을 출력하는데 부서테이블에는 존재하는데 결과로 출력안되는 지역코드정보도 출력하시오

SQL> select d.loc\_code, nvl(sum(e.sal),0)

from emp e, dept d

where e.deptno (+) = d.deptno

group by d.loc\_code ;

11. 이름과 지역코드를 출력하는데 outer join sign 을 양쪽에 써서 출력하고자 한다면 어떻게 해야하는가 ?

SQL> select e.ename, d.loc\_code

from emp e, dept d

where e.deptno (+) = d.deptno (+);

-->에러(아우터조인 연산자는 한쪽에만 지정가능), ansi join으로는 가능

SQL> select e.ename, d.loc\_code

from emp e, dept d

where e.deptno (+) = d.deptno

union

select e.ename, d.loc\_code

from emp e, dept d

where e.deptno = d.deptno (+);

SQL> select e.ename, d.loc\_code

from emp e full outer join dept d

on (e.deptno = d.deptno);

**12. 사원번호, 이름, 관리자사원번호(MGR), 관리자이름을 출력하시오 !**

**☞ self join ~> 자기 자신의 테이블과 조인**

**SQL> select e.empno "사번", e.ename "사원명",**

**m.empno "관리자사번", m.ename "관리자명"**

**from emp e, emp m**

**where e.mgr = m.empno; 🡪13명조회,최고관리자인 king이 mgr이 null이므로 누락됨**

**SQL> SELECT EMPNO, ENAME,**

**(SELECT ENAME FROM EMP WHERE EMPNO = E.MGR)**

**FROM EMP E;🡪self join에서 누락되었던 최고관리자인 King도 나옴(outer join결과가 만들어짐)**

**SQL> SELECT EMPNO, ENAME,**

**(SELECT DNAME FROM DEPT WHERE DEPTNO = E.DEPTNO)**

**FROM EMP E;**

**SQL> select e.empno "사번", e.ename "사원명",**

**m.empno "관리자사번", m.ename "관리자명"**

**from emp e, emp m**

**where e.mgr = m.empno(+) ;**

13. 지역코드, 지역코드별 인원수를 출력하는데 지역코드가 A1은 제외하고 지역코드별 인원수가 3명 이상인 데이터만 조회하고 지역코드별 인원수가 높은것부터 출력하시오 !

SQL> select d.loc\_code, count(\*)

from emp e, dept d

where e.deptno = d.deptno

and d.loc\_code <> 'A1'

group by d.loc\_code

having count(\*) >= 3

order by count(\*) desc;

14. 지역코드, 지역코드별 토탈월급, 지역코드별 최대월급, 지역코드별 최소월급, 지역코드별 인원수를 출력하시오 !

SQL> select d.loc\_code, sum(e.sal), max(e.sal), min(e.sal), count(\*)

from emp e, dept d

where e.deptno = d.deptno

group by d.loc\_code;

15. 1981 년도에 입사한 사원들의 이름과 월급과 입사일과 지역코드를 출력하는데 월급이 높은사원부터 출력하시오!

SQL> select e.ename, e.sal, e.hiredate, d.loc\_code

from emp e, dept d

where e.deptno = d.deptno

and e.hiredate between to\_date('1981/01/01','RRRR/MM/DD')

and to\_date('1981/12/31','RRRR/MM/DD')

order by e.sal desc;

16. 이름과 월급과 지역코드를 출력하시오 (ANSI JOIN ON절 사용)

SQL> select e.ename , e.sal, d.loc\_code

from emp e join dept d

on (e.deptno = d.deptno);

17. 16번의 조건을 월급이 3000 이상인 사원들만 출력하도록 수정하시오 !

SQL> select e.ename , e.sal, d.loc\_code

from emp e join dept d

on (e.deptno = d.deptno)

where e.sal >= 3000 ;

18. 이름, 월급, grade (급여등급)을 출력하시오!(ON절 사용)

SQL> select e.ename, e.sal, s.grade

from emp e left outer join salgrade s

on (e.sal between s.losal and s.hisal);

19. 이름과 월급과 지역코드를 출력하시오 ! (ANSI JOIN USING 사용)

SQL> select e.ename, e.sal, d.loc\_code

from emp e join dept d

using (deptno);

20. 19번을 natural join 으로 수행하시오 !

SQL> select e.ename, e.sal, d.loc\_code

from emp e natural join dept d;

21. 이름과 지역코드를 출력하는데 아래와 같이 결과를 출력하시오 !(T-SQL과 ANSI JOIN 두가지로 모두 작성)

SQL> select e.ename, d.loc\_code

from emp e, dept d

where e.deptno (+) = d.deptno ;

SQL> select e.ename, d.loc\_code

from emp e right outer join dept d

on (e.deptno = d.deptno);

22. EMP와 DEPT의 카테시안 곱을 발생시키시오.

SQL> select e.ename, d.loc\_code

from emp e, dept d

SQL> select e.ename, d.loc\_code

from emp e cross join dept d;

23. 사원이름, 월급, 지역코드, grade(급여등급) 을 출력하시오 !

SQL> select e.ename, e.sal, d.loc\_code, s.grade

from emp e, dept d, salgrade s

where e.deptno = d.deptno

and e.sal between s.losal and s.hisal ;

SQL> select e.ename, e.sal, d.loc\_code, s.grade

from emp e JOIN dept d ON (e.deptno = d.deptno)

JOIN salgrade s ON (e.sal between s.losal and s.hisal) ;

24. 부서번호, 부서번호별 토탈월급을 출력하시오

SQL> select deptno, sum(sal)

from emp

group by deptno;

25. 사원 테이블 전체의 토탈월급을 출력하시오 !

SQL> select sum(sal)

from emp;

26. 1번의 결과와 2번의 결과를 아래와 같이 하나의 결과로 출력하시오

DEPTNO SUM(SAL)

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

30 9400

20 10875

10 8750

29025

SQL> select deptno, sum(sal)

from emp

group by deptno

union all

select to\_number(null) as deptno, sum(sal)

from emp ;

SQL> SELECT DEPTNO, SUM(SAL)

FROM EMP

GROUP BY ROLLUP(DEPTNO);

27. 직업, 직업별 토탈월급을 출력하는데 맨 아래쪽에 전체 토탈월급을 출력하시오 !

SQL> select job, sum(sal)

from emp

group by job

union all

select to\_char(null) as job, sum(sal)

from emp;

SQL> select job, sum(sal)

from emp

group by job

union

select to\_char(null) as job, sum(sal)

from emp;

SQL> SELECT JOB, SUM(SAL)

FROM EMP

GROUP BY ROLLUP(JOB);

28. 입사한 년도(4자리), 입사한 년도별 평균월급을 출력하고 맨 아래쪽에 사원 테이블 전체의 평균월급을 출력하시오 !

SQL> select to\_char(hiredate,'RRRR') as h\_year,

avg(sal)

from emp

group by to\_char(hiredate,'RRRR')

union all

select to\_char(null) as h\_year, avg(sal)

from emp

order by h\_year desc nulls last ;

29. 다음문장을 수행하여 emp\_dev테이블을 만들고 emp 테이블과 emp\_dev 테이블에 중복으로 존재하는 data 를 찾으시오 ! 사원번호, 이름와 직업을 출력하시오 !

SQL> create table emp\_dev

as

select \*

from emp

where job = 'SALESMAN';

SQL> select \* from emp\_dev;

SQL> SELECT \*

FROM EMP

INTERSECT

SELECT \*

FROM EMP\_DEV;

30. emp 테이블에는 존재하는데 emp\_dev 테이블에는 존재하지 않는 사원의 사원번호, 이름과 직업을 출력하시오 !

SQL> select empno, ename, job

from emp

minus

select empno, ename, job

from emp\_dev;

31. 아래의 결과를 출력하시오 !

JOB 10 20 30

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

MANAGER 2450 2975 2850

ANALYST 0 6000 0

PRESIDENT 5000 0 0

SALESMAN 0 0 5600

CLERK 1300 1900 950

토탈값: 8750 10875 9400

SQL> select job,

NVL(sum( decode(deptno,10,sal) ),0) as "10",

NVL(sum( decode(deptno,20,sal) ),0) as "20",

NVL(sum( decode(deptno,30,sal) ),0) as "30"

from emp

group by job

union all

select '토탈값:' as job,

NVL(sum( decode(deptno,10,sal) ),0) as "10",

NVL(sum( decode(deptno,20,sal) ),0) as "20",

NVL(sum( decode(deptno,30,sal) ),0) as "30"

from emp;

SQL> select nvl(job,'토탈값:'),

nvl(sum( decode(deptno,10,sal)),0) as "10",

nvl(sum( decode(deptno,20,sal)),0) as "20",

nvl(sum( decode(deptno,30,sal)),0) as "30"

from emp

group by rollup(job);

<실습5>

1. 최대월급을 받는 사원의 이름과 그 월급을 출력하시오 !

SQL> select max(sal)

from emp; --> 6000

SQL> select ename, SAL

from emp

WHERE SAL = 6000;

☞ 위의 문제를 해결하려면 subquery 를 사용해야한다.

SQL> select ename, SAL

from emp

WHERE SAL = (select max(sal)

from emp) ;

2. SCOTT 보다 더 많은 월급을 받는 사원의 이름과 월급을 출력하시오 !

SQL> select ename,sal

from emp

where sal > ( select sal

from emp

where ename='SCOTT' ) ;

3. SCOTT와 같은 월급을 받는 사원의 이름과 월급을 출력하시오 !

SQL> select ename, sal

from emp

where sal = ( select sal

from emp

where ename='SCOTT' )

and ename != 'SCOTT';

4. ALLEN 보다 늦게 입사한 사원들의 이름과 입사일을 출력하시오 !

SQL> select ename, hiredate

from emp

where hiredate > ( select hiredate

from emp

where ename='ALLEN' );

5. 직업이 SALESMAN 인 사원들 중에서의 최대월급보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오 !

SQL> select ename, sal

from emp

where sal > ( select max(sal)

from emp

where job='SALESMAN' );

6. 최대월급을 받는 사원의 이름과 월급을 출력하시오 !

SQL> select ename, sal

from emp

where sal = (select max(sal)

from emp );

7. DALLAS에 있는 부서에 근무하는 사원들의 이름과 월급을 출력하시오 !

SQL> select ename, sal

from emp

where deptno = ( select deptno

from dept

where loc\_code = 'DALLAS' );

8. 직업이 SALESMAN 인 사원들과 같은 월급을 받는 사원들의 이름과 월급을 출력하시오

SQL> select ename, sal,JOB

from emp

where sal in ( select sal

from emp

where job = 'SALESMAN' );

9. 부서번호가 30번인 사원들과 같은 커미션을 받는 사원들의 이름과 월급과 직업을 출력하시오

-->comm을 받지 않는 사람이 null로 있고 0이 없을때

SQL> select ename, sal, job

from emp

where comm in ( select comm

from emp

where deptno = 30 and COMM is not null);

-->comm을 받지 않는 사람이 null이나 0으로 되어 있을때

SQL> select ename, sal, job

from emp

where comm in ( select comm

from emp

where deptno = 30 and nvl(COMM,0) <> 0);

10. 1982 년도 입사한 사원들과 직업이 같지않는 사원들의 이름과 직업을 출력하시오 !

SQL> select ename, job

from emp

where job not in ( select job

from emp

where hiredate between

to\_date('1982/01/01','RRRR/MM/DD') AND

to\_date('1982/12/31','RRRR/MM/DD') );

11. KING 에게 보고하는 사원들의 이름을 출력하시오! (KING 의 직속부하사원들)

SQL> SELECT EMPNO, ENAME, MGR

FROM EMP

WHERE MGR = (SELECT EMPNO FROM EMP WHERE ENAME = 'KING');

12. 직업, 직업별 토탈월급을 출력하는데 직업별 토탈월급들의 평균값보다 더 큰것을 출력하시오 !

SQL> select job, sum(sal)

from emp

group by job

having sum(sal) > ( select avg(sum(sal))

from emp

group by job );

13. 부서번호, 부서번호별 인원수를 출력하는데 부서번호별 평균인원수보다 더 큰것만 출력하시오 !

SQL> select deptno, count(\*)

from emp

group by deptno

having count(\*) > ( select avg(count(\*))

from emp

group by deptno );

14. 관리자인 사원들의 이름을 출력하시오 ! ( 자신의 사원번호가 부하직원들의 mgr 번호 인 사원들 )

SQL> select ename

from emp

where empno in ( select mgr

from emp );

select ename

from emp a

where exists ( select 1

from emp

where a.empno = mgr);

15. 관리자가 아닌 사원들의 이름을 출력하시오!

#not in에서는 서브쿼리의 결과중에 null인게 없도록 where mgr is not null을 반드시 써야함

SQL> select ename

from emp

where empno not in ( select mgr

from emp

where mgr is not null );

select ename

from emp a

where not exists ( select 1

from emp

where a.empno = mgr);

16. 직업이 SALESMAN 인 사원들중에서의 최대월급 보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오 !(단일행 연산자와 복수행 연산자 두가지형태의 문장을 모두 작성하시오)

SQL> select ename, sal

from emp

where sal > ( select max(sal)

from emp

where job = 'SALESMAN' );

SQL> select ename, sal

from emp

where sal >all ( select sal

from emp

where job = 'SALESMAN' );

17. 직업이 SALESMAN 인 사원들의 월급중에서 가장 작은월급보다 더 많은 월급을 받는 사원들의 이름과 월급을 출력하시오 ! (단일행 연산자와 복수행 연산자 두가지형태의 문장을 모두 작성하시오)

SQL> select ename, sal

from emp

where sal > ( select min(sal)

from emp

where job = 'SALESMAN' );

SQL> select ename, sal

from emp

where sal >any ( select sal

from emp

where job='SALESMAN' );

18. 부서 테이블에 있는 부서번호중에서 사원 테이블에 있는 부서번호만 출력하시오 !(소속사원이 있는 부서만 조회하시오)

SQL> select deptno, dname

from DEPT

where deptno in ( select deptno

from emp );

SQL> select deptno

from DEPT D

where exists ( select 1

from EMP E

where E.deptno = D.deptno );

19. 부서 테이블에는 존재하는 부서번호인데 사원 테이블에는 없는 부서번호를 출력하시오!(소속사원이 없는 부서만 조회하시오)

SQL> select deptno, dname

from DEPT

where deptno not in ( select deptno

from emp );

SQL> select deptno, dname

from DEPT D

where NOT exists ( select 1

from EMP E

where E.deptno = D.deptno );

20. 사원 테이블의 직업중에 인원수가 가장 많은 직업이 무엇인지 출력하시오 !

SQL> select job, count(\*)

from emp

group by job

having count(\*) = ( select max( count(\*) )

from emp

group by job );

<실습6>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

dml문장처리후에는 데이터 확인후 바로 rollback해주세요.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. 사원테이블의 아래의 data 를 입력하시오 !

사원번호 9392

사원이름 JACK

월급 4500

부서번호 20

SQL> insert into emp( empno, ename, sal, deptno )

values( 9392,'JACK', 4500, 20 );

SQL> select \* from emp;

SQL> rollback;

2. 사원 테이블의 아래의 DATA 를 입력하시오 !

사원번호 : 9394

사원이름 : ADAMS

월급 : 4600

입사일 : 오늘날짜

SQL> insert into emp( empno, ename, sal, hiredate)

values( 9394,'ADAMS', 4600, sysdate );

SQL> select \* from emp where empno = 9394;

SQL> rollback;

SQL> insert into emp( empno, ename, sal, hiredate )

values( 9394,'ADAMS', 4600,

to\_date('2015/06/18','RRRR/MM/DD') );

SQL> select \* from emp where empno = 9394;

SQL> rollback;

3. SCOTT 의 월급을 7000 으로 수정하시오 !

SQL> update emp

set sal = 7000

where ename='SCOTT';

SQL> select \* from emp where ename='SCOTT';

SQL> rollback;

4. ALLEN 의 직업을 SALESMAN 으로 변경하시오 !

SQL> update emp

set job='SALESMAN'

where ename='ALLEN';

SQL> select \* from emp where ename='ALLEN';

SQL> rollback;

5. 사원 테이블의 모든 사원들의 월급을 0 으로변경하시오 !

SQL> update emp

set sal = 0 ;

SQL> select \* from emp;

SQL> rollback;

6. 사원번호가 7788 번인 사원을 삭제하시오

SQL> delete from emp

where empno = 7788;

SQL> select \* from emp where empno = 7788;

SQL> rollback;

7. 월급이 3000 이상인 사원들을 삭제하시오 !

SQL> delete from emp

where sal >= 3000;

SQL> select \* from emp where sal >= 3000;

SQL> rollback;

8. SCOTT 보다 많은 월급을 받는 사원들을 삭제하시오

SQL> select \*

from emp

where sal > ( select sal

from emp

where ename='SCOTT') ;

SQL> delete from emp

where sal > ( select sal

from emp

where ename='SCOTT' );

SQL> select \* from emp

where sal > ( select sal

from emp

where ename='SCOTT') ;

SQL> rollback;

9. ALLEN 보다 늦게 입사한 사원들의 월급을 8000 으로 변경하시오 !

SQL> select \*

from emp

where hiredate > ( select hiredate

from emp

where ename='ALLEN' );

SQL> update emp

set sal = 8000

where hiredate > ( select hiredate

from emp

where ename='ALLEN' );

SQL> rollback;

10. ALLEN 보다 늦게 입사한 사원들의 월급을 KING의 월급으로 변경하시오 !

SQL> select \*

from emp

where hiredate > ( select hiredate

from emp

where ename='ALLEN' );

SQL> update emp

set sal = ( select sal

from emp

where ename='KING' )

where hiredate > ( select hiredate

from emp

where ename='ALLEN' );

SQL> rollback;

11. emp 테이블과 똑같은 구조를 갖고있는 테이블을 emp\_backup 이란 이름으로 생성하시오 !

SQL> drop table emp\_backup ;

SQL> create table emp\_backup

as

select \*

from emp

where 1 = 2;

SQL> select \* from emp\_backup;

12. emp\_backup 테이블에 emp 테이블의 직업이 SALESMAN 인 사원들의 사원번호, 이름, 직업과 월급을 입력하시오 !

SQL> insert into emp\_backup(empno, ename, job, sal)

select empno, ename, job, sal

from emp

where job='SALESMAN';

SQL> select \* from emp\_backup;

SQL> rollback;

13. emp 테이블의 모든 data 를 emp\_backup 에 입력하시오 !

SQL> insert into emp\_backup

select \*

from emp;

SQL> select \* from emp\_backup;

SQL> rollback;

14. 부서 테이블에는 존재하는데 사원 테이블에는 존재하지않는 부서번호를 부서테이블에서 지우시오 !

SQL> select \* from dept

where deptno not in (select deptno from emp);

SQL> delete from dept

where deptno not in (select deptno from emp);

SQL> select \* from dept;

SQL> rollback;

15. 직업이 SALESMAN 인 사원을 삭제하시오 !

SQL> delete from emp

where job='SALESMAN';

SQL> selet \* from emp;

SQL> rollback;

16. ALLEN 과 같은 부서번호에서 근무하는 사원들을 삭제하시오 !

SQL> select deptno

from emp

where ename='ALLEN'; -->30번

SQL> delete from emp

where deptno = ( select deptno

from emp

where ename='ALLEN' );

SQL> select \* from emp;-->30번부서에 속한사람 없음

SQL> rollback;

17. 아래의 문장을 수행하여 컬럼을 추가하고 update 문을 이용해서 emp 테이블의 loc\_code 컬럼을 해당하는 지역코드로 값을 갱신하시오 !

- 실습을 위해 컬럼 추가

SQL> alter table emp add loc\_code varchar2(10);

SQL> alter table dept add sumsal number(10);

SQL> update emp e

set loc\_code = ( select loc\_code

from dept d

where d.deptno = e.deptno );

SQL> select \* from emp;

SQL> rollback;

18. 1) merge와 2)update문장으로 각각 dept 테이블에 sumsal 컬럼에 해당 부서번호의 토탈월급으로 값을 갱신하시오 !

SQL> merge into dept d

using ( select deptno, sum(sal) 토탈월급

from emp

group by deptno ) e

on ( d.deptno = e.deptno )

when matched then

update set d.sumsal = e.토탈월급;

SQL> select \* from emp;

SQL> rollback;

SQL> update dept d

set sumsal = (select sum(sal)

from emp e

where e.deptno = d.deptno );

SQL> select \* from emp;