

=====

-- ■ 1. emp 테이블에서 사원번호, 사원이름, 월급을 출력하시오. ■ scott 계정

```
select * from emp;
select empno, ename, sal from emp;
```

=====

-- ■ 2. emp 테이블에서 사원이름과 월급을 출력하는데 컬럼명은 이 름, 월 급으로 바꿔서 출력하시오. ■ scott 계정

```
select ename "이름", sal "월급" from emp;
```

=====

-- ■ 3. emp 테이블에서 사원번호, 사원이름, 월급, 연봉을 구하고 각각 컬럼명은 사원번호, 사원이름, 월급, 연봉으로 출력하시오, ■ scott 계정

```
select empno "사원번호", ename "사원이름", sal "월급", sal*12 "연봉" from emp;
```

=====

-- ■ 4. emp 테이블 의 업무(job)를 중복되지 않게 표시하시오. ■ scott 계정

```
select distinct job from emp;
```

=====

-- ■ 5. emp 테이블의 사원명과 업무로 연결(SMITH, CHECK)해서 표시하고 컬럼명은 Employee and job으로 표시하시오. ■ scott 계정

```
select ename || 'smith' || job || 'check' "employee and job" from emp;
```

=====

-- ■ 1. EMPLOYEES table의 모든 자료를 출력하라 ■ hr 계정

```
select * from employees;
```

=====

-- ■ 2. EMPLOYEES table의 컬럼들을 모두 출력하라. ■ hr 계정

```
DESC employees;
```

=====

-- ■ 3. EMPLOYEES에서 사원번호, 이름, 급여, 담당업무를 출력하라. ■ hr 계정

```
select employee_id, last_name||first_name, salary, job_id from employees;
```

=====

-- ■ 4. 모든 종업원의 급여를 \$300증가 시키기 위해서 덧셈 연산자를 사용하고 결과에 SALARY+300 디스플레이 합니다. ■ hr 계정

```
select last_name||first_name, salary, salary+300 from employees;
```

=====

-- ■ 5. EMP 테이블에서 사원번호, 이름, 급여, 보너스, 보너스 금액을 출력하라 (참고로 보너스는 월급 + (월급 * 커미션)) ■ hr 계정

```
select employee_id, last_name||first_name, salary, commission_pct * salary, salary+(commission_pct*salary)
from employees;
```

```
select employee_id, first_name, salary, nvl(salary * commission_pct, 0), nvl(salary + (salary *
commission_pct), 0) from employees;
```

=====

-- ■ 6. EMPLOYEES 테이블에서 LSAT_NAME을 이름으로 SALARY을 급여로 출력하여라. ■ hr 계정

```
select last_name "이름", salary "급여" from employees;
```

=====

-- ■ 7. EMPLOYEES 테이블에서 LASY_NAME을 Name으로 SALARY*12를 Annual Salary(연봉)로 출력하여라. ■ hr 계정

```
select last_name "name", salary * 12 "annual salary(연봉)" from employees;
```

=====

-- ■ 8. EMPLOYEES 테이블에서 이름과 업무를 연결하여 출력하여라. ■ hr 계정

```
select last_name|| first_name ||' '|| job_id "Employees" from employees;
```

=====

-- ■ 9. EMPLOYEES 테이블에서 이름과 업무를 "KING is a PRESIDENT" 형식으로 출력하여라. ■ hr 계정

```
select last_name || first_name ||' '|| job_id "KING is a PRESIDENT" from employees;
```

```
=====
-- ■ 10. EMPLOYEES 테이블에서 이름과 연봉을 "KING: 1 Year salary = 60000" 형식으로 출력하여라. ■ hr
계정
```

```
select last_name || first_name || 'King: 1Year salary * ' || salary * 12 from employees;
```

```
=====
-- ■ -- 11. EMPLOYEES 테이블에서 JOB을 모두 출력하라. ■ hr 계정
```

```
select job_id from employees;
```

```
=====
-- ■ 1. EMPLOYEES 테이블에서 급여가 3000이상인 사원의 정보를 사원번호, 이름, 담당업무, 급여를
출력하라. ■
```

```
select employee_id, last_name || ' ' || first_name, job_id, salary from employees
where salary >= 3000;
```

```
=====
-- ■ 2. EMPLOYEES 테이블에서 담당 업무가 ST_MAN 인 사원의 정보를 사원번호, 성명, 담당업무, 급여,
부서번호를 출력하라. ■
```

```
select employee_id, last_name || ' ' || first_name, job_id, salary, department_id from employees
where job_id = 'ST_MAN';
```

```
=====
-- ■ 3. EMPLOYEES 테이블에서 입사일자가 2006년 1월 1일 이후에 입사한 사원의 정보를 사원번호, 성명,
담당업무, 급여, 입사일자, 부서번호를 출력하라. ■
```

```
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, department_id from employees
where hire_date >= '06/01/01'; -- 06+/-01+/-01
-- where hire_date > to_date('060101');
-- where hire_date like '06%';
-- where hire_date > to_date('20060101', 'YYYYMMDD');
```

```
=====
-- ■ 4. EMPLOYEES 테이블에서 급여가 3000에서 5000사이의 정보를 성명, 담당업무, 급여, 부서번호를
출력하라. ■
```

```
select * from employees;
select last_name || ' ' || first_name, salary, department_id from employees
where salary BETWEEN 3000 AND 5000;
-- where salary >= 3000 and salary <= 5000;
```

-- ■ 5. EMPLOYEES 테이블에서 사원번호가 145, 152, 203 인 사원의 정보를 사원번호, 성명, 담당업무, 급여, 입사일자를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date from employees
where employee_id in(145, 152, 203);
```

-- ■ 6. EMPLOYEES 테이블에서 입사일자가 05년도에 입사한 사원의 정보를 사원번호, 성명, 담당업무, 급여, 입사일자, 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, department_id from employees
where hire_date like '05%';
```

-- ■ 7. EMPLOYEES 테이블에서 보너스가 NULL인 사원의 정보를 사원번호, 성명, 담당업무, 급여, 입사일자, 보너스, 부서번호를 출력하라. ■

```
select * from employees;
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, commission_pct * salary,
department_id from employees
-- select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, nvl(commission_pct * salary, 0),
department_id from employees
```

-- ■ 8. EMPLOYEES 테이블에서 급여가 1100이상이고 JOB이 ST_MAN 인 사원의 정보를 사원번호, 성명, 담당업무, 급여, 입사일자, 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, department_id from employees
where salary >= 1100 AND job_id = 'ST_MAN';
```

-- ■ 9. EMPLOYEES 테이블에서 급여가 10000이상이거나 JOB이 Manager 인 사원의 정보를 사원번호, 성명, 담당업무, 급여, 입사일자, 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, department_id from employees
where salary >= 10000 or job_id = 'ST_MAN';
```

-- ■ 10. EMPLOYEES 테이블에서 JOB이 ST_MAN, SA_MAN, SA_REP가 아닌 사원의 정보를 사원번호, 성명, 담당업무, 급여, 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, department_id from employees
where job_id not in('ST_MAN', 'SA_MAN', 'SA_REP');
```

-- ■ 11. 업무가 PRESIDENT(AD_PRES) 이고 급여가 12000이상이거나 업무가 SA_MAN인 사원의 정보를 사원번호, 이름, 업무, 급여를 출력하라. ■

```
select * from employees;
select employee_id, last_name || ' ' || first_name, job_id, salary from employees
where job_id = 'AD_PRES' and salary >= 12000 or job_id = 'SA_MAN';
```

-- ■ 12. 업무가 PRESIDENT(AD_PRES) 또는 SA_MAN 이고 급여가 12000이상이거나 사원의 정보를 사원번호, 이름, 업무, 급여를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary from employees
where job_id = 'AD_PRES' or salary >= 12000 and job_id = 'SA_MAN';
-- where job_id='AD_PRES' or job_id = 'SA_MAN' AND salary > 12000;
```

-- ■ 1. emp 테이블에서 사원번호가 7698 인 사원의 이름, 업무, 급여를 출력하시오. ■

```
select * from emp;
select ename, job, sal, mgr from emp
where mgr = '7698';
```

-- ■ 2. emp 테이블에서 사원이름이 SMITH인 사람의 이름과 월급, 부서번호를 구하시오. ■

```
select ename, sal, deptno from emp
where ename = 'SMITH';
```

-- ■ 3. 월급이 2500이상 3500미만인 사원의 이름, 입사일, 월급을 구하시오. ■

```
select ename, hiredate, sal from emp
where sal BETWEEN 2500 and 3500;
-- where sal >= 2500 and sal < 3500;
```

=====

-- ■ 4. 급여가 2000에서 3000사이에 포함되지 않는 사원의 이름, 업무, 급여를 출력하시오. ■

```
select * from emp;
select ename, job, sal from emp
where sal BETWEEN 2000 AND 3000;
```

=====

-- ■ 5. 81년05월01일과 81년12월03일 사이에 입사한 사원의 이름, 급여, 입사일을 출력하시오. ■

```
select ename, sal, hiredate from emp
where hiredate BETWEEN '81/05/01' and '81/12/03';
```

=====

-- ■ 6. emp테이블에서 사원번호가 7566,7782,7934인 사원을 제외한 사람들의 사원번호,이름, 월급을 출력하시오. ■

```
select empno, ename, sal from emp
where empno not in('7566','7782','7934');
```

=====

-- ■ 7. 부서번호 30(deptno)에서 근무하며 월2,000달러 이하를 받는 81년05월01일 이전에 입사한 사원의 이름, 급여, 부서번호, 입사일을 출력하시오. ■

```
select ename, sal, deptno, hiredate from emp
where deptno = 30 and sal <= 2000 and hiredate <= '81.05.01';
```

=====

-- ■ 8. emp테이블에서 급여가 2,000와 5,000 사이고 부서번호가 10 또는 30인 사원의 이름과 급여,부서번호를 나열하시오. ■

```
select ename, sal, deptno from emp
where sal BETWEEN 2000 AND 5000 or deptno in ('10','30');
```

=====

-- ■ 9. 업무가 SALESMAN 또는 MANAGER이면서 급여가 1,600, 2,975 또는 2,850이 아닌 모든 사원의 이름, 업무 및 급여를 표시하시오. ■

```
select * from emp;
select ename, job, sal from emp
where job = 'SALESMAN' OR job = 'MANAGER' AND sal not in('1600','2975','2850');
```

=====

=====

-- ■ 11. emp테이블에서 이름에 A와 E가 있는 모든 사원의 이름을 표시하시오. ■

```
select ename from emp
where ename like '%A%' OR ename like 'A%' OR ename like '%A' OR ename like '%E%' OR ename like 'E%'
OR ename like '%E';
```

=====

-- ■ 12. emp테이블에서 관리자가 없는 모든 사원의 이름과 업무를 표시하시오. ■

```
select ename, job from emp
where job != 'MANAGER';
```

=====

-- ■ 13. emp테이블에서 커미션 항목이 입력된 사원들의 이름과 급여, 커미션을 구하시오. ■

```
select ename, sal, comm from emp
where comm IS NOT NULL;
```

=====

-- ■ 1. EMPLOYEES 테이블에서 입사일자 순으로 정렬하여 사원번호, 이름, 업무, 급여, 입사일자, 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, department_id from employees
order by hire_date asc;
```

=====

-- ■ 2. EMPLOYEES 테이블에서 가장 최근에 입사한 순으로 사원번호, 이름, 업무, 급여, 입사일자, 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, hire_date, department_id from employees
order by hire_date desc;
```

=====

-- ■ 3. EMPLOYEES 테이블에서 부서번호로 정렬한 후 부서번호가 같을 경우 급여가 많은 순으로 정렬하여 사원번호, 성명, 업무, 부서번호, 급여를 출력하여라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, department_id, salary from employees
order by department_id, salary desc;
```

=====

-- ■ 4. EMPLOYEES 테이블에서 첫번째 정렬은 부서번호로 두번째 정렬은 업무로 세번째 정렬은 급여가 많은 순으로 정렬하여 사원번호, 성명, 입사일자, 부서번호, 업무, 급여를 출력하여라. ■

```
select employee_id, last_name || ' ' || first_name, hire_date, department_id, job_id, salary from employees
order by department_id, job_id, salary desc;
```

=====

-- ■ 5. EMPLOYEES 테이블에서 King의 정보를 소문자로 검색하고 사원번호, 성명, 담당업무(소문자로), 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, lower(job_id), department_id from employees
where lower(last_name) like lower('king');
```

=====

-- ■ 6. EMPLOYEES 테이블에서 King의 정보를 대문자로 검색하고 사원번호, 성명, 담당업무(대문자로), 부서번호를 출력하라. ■

```
select employee_id, last_name || ' ' || first_name, upper(job_id), department_id from employees
where upper(last_name) like upper('king');
```

=====

-- ■ 7. DEPARTMENTS 테이블에서 부서번호와 부서이름, 부서이름과 위치번호를 합하여 출력하도록 하라. ■

```
select department_id || department_name, department_name || location_id from departments;
```

=====

-- ■ 8. EMPLOYEES 테이블에서 이름의 첫 글자가 'K' 보다 크고 'Y' 보다 작은 사원의 정보를 사원번호, 이름, 업무, 급여, 부서번호를 출력하라. 단 이름순으로 정렬하여라. ■

```
select employee_id, last_name || ' ' || first_name, job_id, salary, department_id from employees
where last_name > 'K%' AND last_name < 'Y%'
order by first_name;
```

=====

```

-- ■ 9. EMPLOYEES 테이블에서 20번 부서 중 이름의 길이 및 급여의 자릿수를 사원번호, 이름, 이름의
자릿수, 급여 급여의 자릿수를 출력하라. ■
select employee_id, last_name || ' ' || first_name, length(first_name), salary, length(salary) from employees
where department_id = '20';
=====

-- ■ 10. EMPLOYEES 테이블에서 이름 중 'e' 자의 위치를 출력하라. ■
select last_name || ' ' || first_name, instr(first_name, 'e') from employees;
=====

-- ■ 11. 다음 쿼리를 실행하고 결과를 분석하라. ■
select round(4567.678), round(4567.678,0), round(4567.678,2), round(4567.678, -2) from dual;
-- 소수점을 잘른다. 뒤에 숫자는 해당 소수점 자리까지 잘르지 않도록 한다. ( - ) 인 경우 해당 정수 인자까지
잘른다.
=====

-- ■ 12. EMPLOYEES 테이블에서 부서번호가 80인 사람의 급여를 30으로 나눈 나머지를 구하여 출력하라. ■
SELECT * FROM EMPLOYEES;
select salary, round(salary/30,5), department_id from employees
where department_id = '80';
=====

-- ■ 13. EMPLOYEES 테이블에서 30번 부서 중 이름과 담당 업무를 연결하여 출력하여라. 단 담당 업무를 한
줄 아래로 출력하라. ■
select department_id ||chr(10)|| job_id from employees
where department_id = '30';
=====

-- ■ 14. EMPLOYEES 테이블에서 현재까지 근무일 수가 몇주 몇일 인가를 출력하여라. 단 근무 일수가 많은
사람 순으로 출력하여라. ■
select last_name || ' ' || first_name, hire_date, round(months_between(sysdate, hire_date),3) || '월' ,
round(months_between(sysdate, hire_date),3) * 7 || '일' from employees
order by extract(day from hire_date) desc;

```

-- ■ 15. EMPLOYEES 테이블에서 부서 50에서 급여 앞에 \$를 삽입하고 3자리마다 , 를 출력하라. ■

```
select department_id, to_char(salary, '$999,999,999') from employees
where department_id = '50';
```

-- ■ 16. EMPLOYEES 테이블에서 모든 SALESMAN(SA_MAN)에 대하여 급여의 평균,최고액,최저액,합계를 구하여 출력하여라. ■

```
select avg(salary), max(salary), min(salary), sum(salary) from employees
where job_id = 'SA_MAN';
```

-- ■ 17. EMPLOYEES 테이블에 등록되어 있는 인원수, 보너스가 NULL이 아닌 인원수, 보너스의 평균, 등록되어 있는 부서의 수를 구하여 출력하라. ■

```
select count(*), count(commission_pct), round(avg(commission_pct),3), count(distinct(department_id)) from employees;
```

-- ■ 18. EMPLOYEES 테이블에서 부서별로 인원수, 평균 급여, 최저급여, 최고급여, 급여의 합을 구하여 출력하라. ■

```
select department_id "부서명", count(department_id), round(avg(salary),3), min(salary), max(salary),
sum(salary) from employees
group by department_id
order by department_id desc;
```

-- ■ 19. EMPLOYEES 테이블에서 각 부서별로 인원수, 급여의 평균, 최저 급여, 최고 급여, 급여의 합을 구하여 급여의 합이 많은 순으로 출력하여라. ■

```
select department_id, count(department_id), round(avg(salary),3), min(salary), max(salary), sum(salary) from employees
group by department_id
order by sum(salary) desc;
```

-- ■ 20. EMPLOYEES 테이블에서 부서별, 업무별 그룹하여 결과를 부서번호, 업무, 인원수, 급여의 평균, 급여의 합을 구하여 출력하여라. ■

```
select department_id, job_id, count(department_id), avg(salary), sum(salary) from employees
group by department_id, job_id
order by sum(salary) desc;
```

-- ■ 21. EMPLOYEES 테이블에서 부서 인원이 4명보다 많은 부서의 부서번호, 인원수, 급여의 합을 구하여 출력하여라(GROUP BY, HAVING) ■

```
select count(*), sum(salary) from employees
group by department_id
having count(*) > 4;
```

-- ■ 22. EMPLOYEES 테이블에서 급여가 최대 10000이상인 부서에 대해서 부서번호, 평균급여, 급여의 합을 구하여 출력하여라. ■

```
select department_id, round(avg(salary), 3), sum(salary) from employees
group by department_id
having max(salary) > 10000;
```

-- ■ 23. EMPLOYEES 테이블에서 업무별 급여의 평균이 10000 이상인 업무에 대해서 업무명, 평균 급여, 급여의 합을 구하여 출력하여라. ■

```
select job_id, AVG(salary), sum(salary) from employees
group by job_id
having avg(salary) >= 10000;
```

-- ■ 24. EMPLOYEES 테이블에서 전체 월급이 10000을 초과하는 각 업무에 대해서 업무와 월급여 합계를 출력하라. 단 판매원은 제외하고 월 급여 합계로 정렬(내림차순) 하라. (SA_) ■

```
select job_id, sum(salary) from employees
where job_id not like 'SA_%'
group by job_id
having sum(salary) > 10000
order by sum(salary) desc;
```

=====

-- ■ 25. 테이블에서 사원번호, 사원이름, 입사일을 출력하는데 입사일이 빠른 사람순으로 정렬하시오. ■

```
select * from employees;
select employee_id, last_name || ' ' || first_name, hire_date from employees
order by hire_date asc;
```

-- 26. emp 테이블에서 사원이름, 급여, 연봉을 구하고 연봉이 많은 순으로 정렬하시오.

```
select last_name || ' ' || first_name, salary, salary * 12 from employees
order by salary * 12 desc;
```

=====

-- ■ 27. 10번 부서와 20번부서에서 근무하고 있는 사원의 이름과 부서번호를 출력하는데 이름을 영문자순으로 표시하시오. ■

```
select lower(last_name || ' ' || first_name), department_id from employees
where department_id between 10 and 20
order by lower(last_name || ' ' || first_name) asc;
```

=====

-- ■ 28 .커미션을 받는 모든 사원의 이름,급여 및 커미션을 커미션을 기준으로 내림차순으로 정렬하여 표시하십시오. ■

```
select last_name || ' ' || first_name, salary, nvl(commission_pct, 0) from employees
order by commission_pct desc;
```

=====

-- ■ 1. emp테이블의 업무(job)을 첫글자는 대문자 나머지는 소문자로 출력하시오. ■

```
select initcap(job) from emp;
```

=====

-- ■ 2. emp테이블에서 사원이름 중 A가 포함된 사원이름을 구하고 그 이름 중 앞에서 3자만 추출하여 출력 ■

```
select substr(ename, 1,3) from emp where ename like '%A%';
```

=====

-- ■ 3. 이름의 세번째 문자가 A인 모든 사원의 이름을 표시하시오. ■

```
select ename from emp where substr(ename, 3,1) ='A';
```

=====

-- ■ 4. 이름이 J,A 또는 M으로 시작하는 모든 사원의 이름(첫 글자는 대문자로, 나머지 글자는 소문자로 표시) 및 이름의 길이를 표시하시오.(열 레이블은 name과 length로 표시) ■

```
select initcap(ename), length(ename) from emp
where ename like '%J%' or ename like '%A%' or ename like '%M%';
```

=====

-- ■ 5. 이름의 글자수가 6자 이상인 사원의 이름을 소문자로 이름만 출력하시오 ■

```
select lower(ename), length(ename) from emp
where length(ename) >= 6;
```

-- ■ 6. 이름의 글자수가 6자 이상인 사람의 이름을 앞에서 3자만 구하여 소문자로 출력하시오. ■

```
select substr(lower(ename),1,3), length(ename) from emp
where length(ename) >=6;
```

-- ■ 7. 모든 사원의 이름과 급여를 표시하시오. 급여는 15자 길이로 왼쪽에 \$기호가 채워진 형식으로 표기하고 열레이블을 SALARY로 지정하시오. ■

```
select * from emp;
select ename, '$' || rpad(sal, 15, '0') as salary from emp;
```

-- ■ 1. Steven King의 부서명을 출력하라. ■

```
select e.last_name, e.first_name, e.department_id, d.department_name
from employees e, departments d
where e.department_id = d.department_id and e.first_name = 'Steven' and e.last_name = 'King';
/*
```

```
select e.employee_id, e.first_name, e.last_name, e.department_id, d.department_id, department_name
from employees e, departments d
where e.department_id = d.department_id and e.first_name = 'Steven' and lower(e.last_name) = 'king';
*/
```

-- ■ 2. IT 부서에서 근무하고 있는 사람들을 출력하라. ■

```
select e.last_name, e.first_name, e.department_id, d.department_name
from employees e, departments d
where e.department_id = d.department_id and d.department_id = 90;
/*
```

```
select d.department_name, e.employee_id, e.first_name, e.last_name
from departments d, employees e
where d.department_id = e.department_id
and d.department_name = 'IT';
*/
```

-- ■ 3. EMPLOYEES 테이블과 DEPARTMENTS 테이블을 Cartesian Product(모든 가능한 행들의 Join)하여 ■

```
select e.employee_id, e.last_name, e.first_name, e.job_id, e.department_id, d.department_name, d.location_id
from employees e cross join departments d;
```

```
/*
select * from employees e, departments d;
*/
```

-- ■ 4. EMPLOYEES 테이블에서 사원번호, 이름, 업무, EMPLOYEES 테이블의 부서번호, DEPARTMENTS 테이블의 부서번호, 부서명, 근무지를 출력하여라. ■

```
select e.employee_id, e.last_name, e.first_name, e.job_id, e.department_id, d.department_id,
d.department_name, d.location_id
from employees e cross join departments d;
```

```
/*
select e.employee_id, e.first_name, e.job_id, e.department_id, d.department_id, d.department_name,
d.location_id
from employees e, departments d
where e.department_id = d.department_id;
*/
```

```
=====
-- ■ 5. EMPLOYEES 테이블과 DEPARTMENTS 테이블의 부서번호를 조인하고 SA_MAN 직원만의 직원번호,
이름, 급여, 부서명, 근무지를 출력하라. (Alias를 사용) ■
```

```
select e.employee_id, e.last_name, e.first_name, e.salary, d.department_name, d.location_id
from employees e join departments d
on e.department_id = d.department_id;
```

```
/*
```

```
select e.employee_id, e.first_name, e.salary, e.department_id, d.department_id, d.department_name,
d.location_id
from employees e, departments d
where e.department_id = d.department_id
and e.job_id = 'SA_MAN';
```

```
*/
```

```
=====
-- ■ 6. EMPLOYEES 테이블과 DEPARTMENTS 테이블에서 DEPARTMENTS 테이블에 있는 모든 자료를
직원번호, 이름, 업무, EMPLOYEES 테이블의 부서번호, DEPARTMENTS 테이블의 부서번호, 부서명,
근무지를 출력하여라. ■
```

```
select e.employee_id, e.first_name, e.last_name, e.job_id, e.department_id, d.department_name, d.location_id
from employees e cross join departments d;
```

```
/*
```

```
select e.employee_id, e.first_name, e.job_id, e.department_id, d.department_id, d.department_name,
d.location_id;
from employees e, departments d
where d.department_id = e.department_id(+);
```

```
*/
```

```
=====
-- ■ 7. EMPLOYEES 테이블에서 Self join하여 관리자(매니저)를 출력하여라. ■
```

```
select a.employee_id, a.last_name, a.manager_id, a.job_id, b.employee_id, b.first_name, b.job_id
from employees a, employees b
where a.manager_id = b.employee_id;
```

```
/*
```

```
SELECT a.first_name || '의 매니저는 ' || b.first_name || '입니다'
FROM employees a, employees b
WHERE a.manager_id = b.employee_id;
```

```
*/
```

```
=====
-- ■ 8. EMPLOYEES 테이블에서 left join하여 관리자(매니저)를 출력하고 매니저 아이디가 없는 사람은
배제하고 하향식으로 하며, 급여는 역순으로 출력하여라. ■
```

```
select a.employee_id, a.last_name, a.manager_id, a.job_id, a.salary, b.employee_id, b.first_name, b.job_id ,
b.salary
from employees a join employees b
on a.manager_id(+) = b.employee_id
where a.manager_id is not null;
order by salary desc;
```

```
/*
```

```
select e.employee_id, a.first_name, a.manager_id, b.employee_id, b.first_name
from employees a, employees b
where a.manager_id = b.employee_id(+)
start with a.manager_id is not null
connect by a.manager_id = prior a.employee_id
order by a.salary desc;
```

```
*/
```


=====

-- ■ 9. EMPLOYEES 테이블에서 right join 하여 관리자(매니저)가 108번 상향식으로 급여는 역순으로 출력하라. ■

```
SELECT a.employee_id, a.first_name, a.manager_id,
       b.employee_id, b.first_name
FROM employees a, employees b
WHERE a.manager_id(+) = b.employee_id
      AND a.manager_id = 108
      CONNECT BY PRIOR a.manager_id = a.employee_id
ORDER BY a.salary DESC;
```

=====

-- ■ 10. 모든 사원의 이름, 부서번호, 부서이름을 표시하시오. (emp, dept) ■

```
select e.ename, e.deptno, d.deptno, d.dname
from emp e, dept d
where e.deptno = d.deptno;
```

=====

-- ■ 11. 업무가 MANAGER인 사원의 정볼르 이름, 업무, 부서명, 근무지 순으로 출력하시오. (emp, dept) ■

```
SELECT e.ename, e.job, d.dname, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.job = 'MANAGER';
```

=====

-- ■ 12. 커미션을 받고 급여가 1,600이상인 사원의 사원이름, 부서명, 근무지를 출력하시오. ■

```
SELECT e.ename, d.dname, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.comm IS NOT NULL AND e.comm != 0
      AND e.sal >= 1600;
```

=====

-- ■ 13. 근무지가 CHICAGO인 모든 사원의 이름, 업무, 부서번호 및 부서이름을 표시하시오. ■

```
SELECT e.ename, e.job, e.deptno, d.deptno, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND d.loc = 'CHICAGO';
```

=====

-- ■ 14. 근무지별로 근무하는 사원의 수가 5명 이하인 경우, 인원이 적은 도시순으로 정렬하시오. (근무 인원이 0 명인 곳도 표시) ■

```
SELECT d.loc, COUNT(e.empno)
FROM emp e, dept d
WHERE e.deptno(+) = d.deptno
GROUP BY d.loc
HAVING COUNT(e.empno) <= 5
ORDER BY COUNT(e.empno);
```

=====

-- ■ 15. 사원의 이름 및 사원 번호를 관리자의 이름과 관리자 번호와 함께 표시하고 각각의 열레이블은 employee, emp#, manager, mag#로 지정하시오. ■

```
SELECT a.ename "사원 이름", a.empno "사원 번호",
       b.ename "관리자 이름", b.empno "관리자 번호"
FROM emp a, emp b
WHERE a.mgr = b.empno;
```

=====

-- ■ 16. 관리자보다 먼저 입사한 모든 사원의 이름 및 입사일을 관리자의 이름 및 입사일과 함께 표시하고 열레이블을 각각 employee, emp hired, manger, mgr hired로 지정 ■

```
SELECT a.ename "사원 이름", a.hiredate "사원의 입사일",
```

```

    b.ename "매니저 이름", b.hiredate "매니저 입사일"
FROM emp a, emp b
WHERE a.mgr = b.empno
    AND a.hiredate < b.hiredate;
=====

```

-- ■ 17. 사원의 이름 및 사원번호를 관리자의 이름과 관리자 번호와 함께 표시하고 각각의 열 레이블은 employee, emp#, manager, mgr#로 지정하는데 King을 포함하여 관리자가 없는 모든 사원을 표시하도록 하고 결과를 사원번호를 기준으로 정렬 ■

```

SELECT a.ename "사원이름", a.empno "사원번호",
    b.ename "매니저이름", b.empno "매니저번호"
FROM emp a, emp b
WHERE a.mgr = b.empno(+)
ORDER BY a.empno ASC;
=====

```

-- ■ 18. 지정한 부서번호, 사원이름 및 지정한 사원과 동일한 부서에서 근무하는 모든 사원을 표시하도록 질의를 작성하고 부서번호는 department, 사원이름은 employee, 동일한 부서에서 근무하는 사원은 colleague로 표시하시오.(부서번호, 사원이름,동료 순으로 오름차순 정렬) ■

```

SELECT a.ename, a.deptno, b.deptno, b.ename "colleague"
FROM emp a, emp b
WHERE a.deptno = b.deptno
    AND a.empno <> b.empno
    -- AND a.ename = 'SMITH'
ORDER BY a.deptno, a.ename, b.ename;
=====

```

-- ■ 19. 10번부서에서 근무하는 직원들의 부서번호, 부서이름, 사원이름,월급,급여등급을 출력하시오. (emp,dept,salgrade) ■

```

SELECT e.deptno, d.deptno, d.dname, e.ename, e.sal, s.grade
FROM emp e, dept d, salgrade s
WHERE e.deptno = d.deptno
    AND e.sal BETWEEN s.losal AND s.hisal
    AND e.deptno = 10;
=====

```

-- ■ 1. BLAKE와 같은 부서에 있는 직원들의 이름과 입사일을 구하는데 BLAKE는 제외하고 출력하시오.(BLAKE가 여러명일 수 있음) ■

```

select ename,hiredate, deptno
from emp
where deptno = (select deptno from emp where ename= 'BLAKE') AND ENAME != 'BLAKE';
-- select deptno from emp where ename= 'BLAKE' 는 30이라는 값을 가지고 있음
-- 30을 가지고 있으면서 blake 가 아닌 사람을 가지고 온다.
=====

```

-- ■ 2. (평균급여보다 많은 급여)를 받는 직원들의 사원번호, 이름, 월급을 출력하는데 (월급이 높은 사람순으로 출력 하시오.) ■

```

select empno, ename, sal from emp
where sal > (select floor(avg(sal)) from emp)
order by sal desc;
-- select floor(avg(sal)) from emp 는 2077 이라는 값을 가지고 있음 그래서 평균이라는것보다 높은것을 where
조건으로 보여짐
-- order by 로 desc 해서 높은사람순으로 출력하게됨.
=====

```

-- ■ 3. (10번부서에서 급여를 가장 적게 받는 직원)과 동일한 급여를 받는 직원의 이름을 출력하시오. ■

```

select * from emp;
select ename, sal from emp where sal = (select min(sal) from emp where deptno = '10');
-- select min(sal) from emp where deptno = '10'; 는 1300을 가져옴.

```

-- sal이 1300이 인 사람을 보여지게 됨.

=====

-- ■ 4. 사원수가 3명이 넘는 부서의 부서명과 사원수를 출력하시오. ■

```
select d.dname,count(*) from emp e, dept d
where e.deptno = d.deptno group by d.dname having count(*) > 3;
```

=====

-- ■ 5. 사원번호가 7844인 사원보다 빨리 입사한 사원의 이름과 입사일을 출력하시오. ■

```
select ename, hiredate from emp where hiredate < (select hiredate from emp where empno = '7844');
```

=====

-- ■ 6. 직속상사(MGR)가 KING인 모든 사원의 이름과 급여를 출력하시오. ■

```
select ename, sal from emp where mgr= (select empno from emp where ename = 'KING');
```

=====

-- ■ 7. 20번 부서에서 가장 급여를 많이 받는 사원과 동일한 급여를 받는 사원의 이름과 부서명, 급여, 급여등급을 출력하시오. (emp, dept, salgrade) ■

```
select e.ename, d.dname, e.sal, s.grade from emp e, dept d, salgrade s
where e.deptno = d.deptno and sal between s.losal and s.hisal
and sal in(select max(sal) from emp where deptno = '20');
```

=====

-- ■ 8. 총급여(sal+comm)가 평균 급여보다 많은 급여를 받는 사람의 부서번호, 이름, 총급여, 커미션을 출력하시오. ■

```
select deptno, ename, sal+comm, nvl2(comm, '유','무') "comm유무"
from emp
where sal+comm > (select floor(avg(sal)) from emp);
```

=====

-- ■ 9. CHICAGO 지역에서 근무하는 사원의 평균 급여보다 높은 급여를 받는 사원의 이름과 급여, 지역명을 출력하시오. ■

```
select e.ename, e.sal, d.loc from emp e, dept d where e.deptno = d.deptno
and e.sal > (select floor(avg(e.sal)) from emp e, dept d
where e.deptno = d.deptno
group by d.loc
having d.loc = 'CHICAGO');
```

=====

-- ■ 10. 업무가 SALESMAN인 직원이 2명 이상인 부서의 이름, 근무하는 사원의 이름, 업무를 출력하시오.(컬럼명은 부서명, 사원명, 업무로 출력) ■

```
select d.dname 부서명, e.ename 사원명, e.job 업무
from emp e, dept d
where e.deptno = d.deptno
and d.dname in( select d.dname from emp e, dept d where e.deptno = d.deptno
and job = 'SALESMAN' group by d.dname having count(*) > 2);
```

=====

-- ■ 11. 커미션이 없는 사원들 중 월급이 가장 높은 사원의 이름과 급여등급을 출력하시오. ■

```
select ename, grade from emp, salgrade where sal between losal and hisal
and sal = (select max(sal) from emp where comm is null);
```

=====

-- ■ 12. SMITH의 관리자(MGR)의 이름과 부서명, 근무지역을 출력하시오. ■

```
select ename, dname, loc from emp e, dept d
where e.deptno = d.deptno and empno = (select mgr from emp where ename = 'SMITH');
```

=====

-- ■ 1. EMPLOYEES 테이블에서 Kochhar 의 급여보다 많은 사원의 정보를 사원번호, 이름, 담당업무, 급여를 출력하라. ■

```
select employee_id, first_name, job_id, salary from employees
where salary > (select salary from employees where last_name = 'Kochhar');
```

=====

-- ■ 2. EMPLOYEES 테이블에서 급여의 평균보다 적은 사원의 정보를 사원번호, 이름, 담당업무, 급여, 부서번호를 출력하여라 ■

```
select employee_id, first_name, job_id, salary, department_id
from employees
where salary > (select floor(avg(salary)) from employees) order by salary desc;
```

```

=====
-- ■ 3. EMPLOYEES 테이블에서 100번 부서의 최소 급여보다 최소 급여가 많은 다른 모든 부서를 출력하라.
■
select department_id, min(salary) from employees
group by department_id having min(salary) > (select min(salary) from employees where department_id =
'100');
=====
-- ■ 4. 업무별로 최소 급여를 받는 사원의 정보를 사원번호, 이름, 업무, 부서번호를 출력하여라. 단 업무별로
정렬하여라. ■
select employee_id, first_name, job_id, department_id from employees
where (job_id, salary) in(select job_id, min(salary) from employees group by job_id);
=====
-- ■ 5. EMPLOYEES 과 DEPARTMENTS 테이블에서 업무가 SA_MAN 사원의 정보를 이름, 업무, 부서명,
근무지를 출력하라. ■
select e.first_name, e.job_id, d.department_name, d.location_id
from employees e, departments d
where e.department_id = d.department_id
and e.job_id = 'SA_MAN';
=====
-- ■ 6. EMPLOYEES 테이블에서 가장 많은 사원을 갖는 MANAGER의 사원번호를 출력하라. ■
select manager_id from employees group by manager_id
having count(manager_id) in(select max(count(manager_id)) from employees group by manager_id);
=====
-- ■ 7. EMPLOYEES 테이블에서 가장 많은 사원이속해 있는 부서번호와 사원수를 출력하라. ■
select department_id, count(department_id) from employees
group by department_id
having count(department_id) in(select max(count(department_id)) from employees group by department_id);
=====
-- ■ 8. EMPLOYEES 테이블에서 사원번호가 123인 사원의 직업과 같고 사원번호가 192인 사원의
급여(SAL)보다 많은 사원의 사원번호, 이름, 직업, 급여를 출력하라. ■
select employee_id, first_name, job_id, salary from employees
where job_id = (select job_id from employees where employee_id = '123') and salary > (select salary from
employees where employee_id = '192');

=====

```



```
-- ■ 9. 직업(JOB)별로 최소 급여를 받는 사원의 정보를 사원번호, 이름, 업무, 부서명을 출력하라. -- 조건 1 :
직업별로 내림차순 정렬 ■
select employee_id, first_name, job_id, d.department_name
from employees e, departments d
where e.employee_id = d.department_id
=====
-- ■ 10. EMPLOYEES 테이블에서 50번 부서의 최소 급여를 받는 사원 보다 많은 급여를 받는 사원의 정보를
사원번호, 이름, 업무, 입사일자, 급여, 부서번호를 출력하라 단 50번은 제외 ■
select employee_id, first_name, job_id, hire_date, salary, department_id
from employees
where salary > (select min(salary) from employees where department_id = '50') and department_id != '50'
order by department_id;
=====
-- ■ 11. EMPLOYEES 테이블에서 50번 부서의 최고 급여를 받는 사원 보다 많은 급여를 받는 사원의 정보를
사원번호, 이름, 업무, 입사일자, 급여, 부서번호를 출력하라 단 50번은 제외 ■
select department_id, first_name, job_id, hire_date, salary, department_id
from employees
where salary > (select max(salary) from employees where department_id = '50'
group by department_id) and department_id != '50' order by salary;
=====
-- ■ 1. Steven King의 부서명을 출력하라. ■
select e.last_name, e.first_name, e.department_id, d.department_name
from employees e, departments d
where e.department_id = d.department_id and e.first_name = 'Steven' and e.last_name = 'King';
/*
select e.employee_id, e.first_name, e.last_name, e.department_id, d.department_id, department_name
from employees e, departments d
where e.department_id = d.department_id and e.first_name = 'Steven' and lower(e.last_name) = 'king';
*/
=====
-- ■ 2. IT 부서에서 근무하고 있는 사람들을 출력하라. ■
select e.last_name, e.first_name, e.department_id, d.department_name
from employees e, departments d
where e.department_id = d.department_id and d.department_id = 90;
/*
select d.department_name, e.employee_id, e.first_name, e.last_name
from departments d, employees e
where d.department_id = e.department_id
and d.department_name = 'IT';
*/
=====
-- ■ 3. EMPLOYEES 테이블과 DEPARTMENTS 테이블을 Cartesian Product(모든 가능한 행들의 Join)하여
-- 사원번호, 이름, 업무, 부서번호, 부서명, 근무지를 출력하여라. ■
select e.employee_id, e.last_name, e.first_name, e.job_id, e.department_id, d.department_name, d.location_id
from employees e cross join departments d;
/*
select * from employees e, departments d;
*/
=====
-- ■ 4. EMPLOYEES 테이블에서 사원번호, 이름, 업무, EMPLOYEES 테이블의 부서번호, DEPARTMENTS
테이블의 부서번호, 부서명, 근무지를 출력하여라. ■
select e.employee_id, e.last_name, e.first_name, e.job_id, e.department_id, d.department_id,
d.department_name, d.location_id
```

```
from employees e cross join departments d;
```

```
/*
```

```
select e.employee_id, e.first_name, e.job_id, e.department_id, d.department_id, d.department_name,  
d.location_id
```

```
from employees e, departments d
```

```
where e.department_id = d.department_id;
```

```
*/
```

```
=====
```

```
-- ■ 5. EMPLOYEES 테이블과 DEPARTMENTS 테이블의 부서번호를 조인하고 SA_MAN 직원만의 직원번호,  
이름, 급여, 부서명, 근무지를 출력하라. (Alias를 사용) ■
```

```
select e.employee_id, e.last_name, e.first_name, e.salary, d.department_name, d.location_id
```

```
from employees e join departments d
```

```
on e.department_id = d.department_id;
```

```
/*
```

```
select e.employee_id, e.first_name, e.salary, e.department_id, d.department_id, d.department_name,  
d.location_id
```

```
from employees e, departments d
```

```
where e.department_id = d.department_id
```

```
and e.job_id = 'SA_MAN';
```

```
*/
```

```
=====
```

```
-- ■ 6. EMPLOYEES 테이블과 DEPARTMENTS 테이블에서 DEPARTMENTS 테이블에 있는 모든 자료를  
직원번호, 이름, 업무, EMPLOYEES 테이블의 부서번호, DEPARTMENTS 테이블의 부서번호, 부서명,  
근무지를 출력하여라. ■
```

```
select e.employee_id, e.first_name, e.last_name, e.job_id, e.department_id, d.department_name, d.location_id
```

```
from employees e cross join departments d;
```

```
/*
```

```
select e.employee_id, e.first_name, e.job_id, e.department_id, d.department_id, d.department_name,  
d.location_id;
```

```
from employees e, departments d
```

```
where d.department_id = e.department_id(+);
```

```
*/
```

```
=====
```

```
-- ■ 7. EMPLOYEES 테이블에서 Self join하여 관리자(매니저)를 출력하여라. ■
```

```
select a.employee_id, a.last_name, a.manager_id, a.job_id, b.employee_id, b.first_name, b.job_id
```

```
from employees a, employees b
```

```
where a.manager_id = b.employee_id;
```

```
/*
```

```
SELECT a.first_name || '의 매니저는 ' || b.first_name || '입니다'
```

```
FROM employees a, employees b
```

```
WHERE a.manager_id = b.employee_id;
```

```
*/
```

```
-- ■ 8. EMPLOYEES 테이블에서 left join하여 관리자(매니저)를 출력하고 매니저 아이디가 없는 사람은  
배제하고 하향식으로 하며, 급여는 역순으로 출력하여라. ■
```

```
select a.employee_id, a.last_name, a.manager_id, a.job_id, a.salary, b.employee_id, b.first_name, b.job_id ,  
b.salary
```

```
from employees a join employees b
```

```
on a.manager_id(+) = b.employee_id
```

```
where a.manager_id is not null;
```

```
order by salary desc;
```

```
/*
```

```
select e.employee_id, a.first_name, a.manager_id, b.employee_id, b.first_name
```

```
from employees a, employees b
```

```

where a.manager_id = b.employee_id(+)
start with a.manager_id is not null
connect by a.manager_id = prior a.employee_id
order by a.salary desc;
*/

```

```

=====
-- ■ 9. EMPLOYEES 테이블에서 right join 하여 관리자(매니저)가 108번 상향식으로 급여는 역순으로
출력하라. ■

```

```

SELECT a.employee_id, a.first_name, a.manager_id,
       b.employee_id, b.first_name
FROM employees a, employees b
WHERE a.manager_id(+) = b.employee_id
      AND a.manager_id = 108
      CONNECT BY PRIOR a.manager_id = a.employee_id
ORDER BY a.salary DESC;
=====

```

```

-- ■ 10. 모든 사원의 이름, 부서번호, 부서이름을 표시하시오. (emp, dept) ■

```

```

select e.ename, e.deptno, d.deptno, d.dname
from emp e, dept d
where e.deptno = d.deptno;
=====

```

```

-- ■ 11. 업무가 MANAGER인 사원의 정볼르 이름, 업무, 부서명, 근무지 순으로 출력하시오. ( emp, dept) ■

```

```

SELECT e.ename, e.job, d.dname, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.job = 'MANAGER';
=====

```

```

-- ■ 12. 커미션을 받고 급여가 1,600이상인 사원의 사원이름, 부서명, 근무지를 출력하시오. ■

```

```

SELECT e.ename, d.dname, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.comm IS NOT NULL AND e.comm != 0
      AND e.sal >= 1600;
=====

```

```

-- ■ 13. 근무지가 CHICAGO인 모든 사원의 이름, 업무, 부서번호 및 부서이름을 표시하시오. ■

```

```

SELECT e.ename, e.job, e.deptno, d.deptno, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND d.loc = 'CHICAGO';
=====

```

```

-- ■ 14. 근무지별로 근무하는 사원의 수가 5명 이하인 경우, 인원이 적은 도시순으로 정렬하시오. (근무
인원이 0 명인 곳도 표시 ) ■

```

```

SELECT d.loc, COUNT(e.empno)
FROM emp e, dept d
WHERE e.deptno(+) = d.deptno
GROUP BY d.loc
HAVING COUNT(e.empno) <= 5
ORDER BY COUNT(e.empno);
=====

```

```

-- ■ 15. 사원의 이름 및 사원 번호를 관리자의 이름과 관리자 번호와 함께 표시하고 각각의 열레이블은
employee, emp#, manager, mag#로 지정하시오. ■

```

```

SELECT a.ename "사원 이름", a.empno "사원 번호",
       b.ename "관리자 이름", b.empno "관리자 번호"
FROM emp a, emp b

```

```
WHERE a.mgr = b.empno;
```

```
=====
-- ■ 16. 관리자보다 먼저 입사한 모든 사원의 이름 및 입사일을 관리자의 이름 및 입사일과 함께 표시하고 열 레이블을 각각 employee, emp hired, manger, mgr hired로 지정 ■
```

```
SELECT a.ename "사원 이름", a.hiredate "사원의 입사일",
       b.ename "매니저 이름", b.hiredate "매니저 입사일"
```

```
FROM emp a, emp b
```

```
WHERE a.mgr = b.empno
```

```
AND a.hiredate < b.hiredate;
```

```
=====
-- ■ 17. 사원의 이름 및 사원번호를 관리자의 이름과 관리자 번호와 함께 표시하고 각각의 열 레이블은 employee, emp#, manager, mgr#로 지정하는데 King을 포함하여 관리자가 없는 모든 사원을 표시하도록 하고 결과를 사원번호를 기준으로 정렬 ■
```

```
SELECT a.ename "사원이름", a.empno "사원번호",
       b.ename "매니저이름", b.empno "매니저번호"
```

```
FROM emp a, emp b
```

```
WHERE a.mgr = b.empno(+)
```

```
ORDER BY a.empno ASC;
```

```
=====
-- ■ 18. 지정한 부서번호, 사원이름 및 지정한 사원과 동일한 부서에서 근무하는 모든 사원을 표시하도록 질의를 작성하고 부서번호는 department, 사원이름은 employee, 동일한 부서에서 근무하는 사원은 colleague로 표시하십시오.(부서번호, 사원이름,동료 순으로 오름차순 정렬) ■
```

```
SELECT a.ename, a.deptno, b.deptno, b.ename "colleague"
```

```
FROM emp a, emp b
```

```
WHERE a.deptno = b.deptno
```

```
AND a.empno <> b.empno
```

```
-- AND a.ename = 'SMITH'
```

```
ORDER BY a.deptno, a.ename, b.ename;
```

```
=====
-- ■ 19. 10번부서에서 근무하는 직원들의 부서번호, 부서이름, 사원이름,월급,급여등급을 출력하십시오. (emp,dept,salgrade) ■
```

```
SELECT e.deptno, d.deptno, d.dname, e.ename, e.sal, s.grade
```

```
FROM emp e, dept d, salgrade s
```

```
WHERE e.deptno = d.deptno
```

```
AND e.sal BETWEEN s.losal AND s.hisal
```

```
AND e.deptno = 10;
```

```
=====
-- ■ 1. Steven King의 부서명을 출력하라. ■
```

```
SELECT employee_id, first_name, last_name,
       e.department_id, d.department_id, department_name
```

```
FROM employees e, departments d
```

```
WHERE e.department_id = d.department_id
```

```
AND e.first_name = 'Steven'
```

```
AND lower(e.last_name) = 'king';
```

```
=====
-- ■ 2. IT부서에서 근무하고 있는 사람들을 출력하라. ■
```

```
SELECT d.department_name, e.employee_id,
```

```
e.first_name, e.last_name
```

```
FROM departments d, employees e
```

```
WHERE d.department_id = e.department_id
```

```
AND d.department_name = 'IT';
```

```
=====
-- ■ 3. EMPLOYEES 테이블과 DEPARTMENTS 테이블을 Cartesian Product(모든 가능한 행들의 Join)하여
사원번호,이름,업무,부서번호,부서명,근무지를 출력하여라. ■
```

```
SELECT
```

```
*
```

```
FROM employees e, departments d;
```

```
=====
-- ■ 4. EMPLOYEES 테이블에서 사원번호,이름,업무 EMPLOYEES 테이블의 부서번호, DEPARTMENTS
테이블의 부서번호,부서명,근무지를 출력하여라. ■
```

```
SELECT e.employee_id, e.first_name, e.job_id,
       e.department_id, d.department_id, d.department_name, d.location_id
FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

```
=====
-- ■ 5. EMPLOYEES 테이블과 DEPARTMENTS 테이블의 부서번호를 조인하고 SA_MAN 사원만의
사원번호,이름,급여,부서명,근무지를 출력하라. ■
```

```
SELECT e.employee_id, e.first_name, e.salary, e.job_id,
       e.department_id, d.department_id, d.department_name, d.location_id
FROM employees e, departments d
WHERE e.department_id = d.department_id
      AND e.job_id = 'SA_MAN';
```

```
=====
-- ■ 6. EMPLOYEES 테이블과 DEPARTMENTS 테이블에서 DEPARTMENTS 테이블에 있는 모든 자료를
사원번호,이름,업무, EMPLOYEES 테이블의 부서번호, DEPARTMENTS 테이블의 부서번호,부서명,근무지를
출력하여라 ■
```

```
SELECT e.employee_id, e.first_name, e.job_id,
       e.department_id, d.department_id, d.department_name, d.location_id
FROM departments d, employees e
WHERE d.department_id = e.department_id(+);
```

```
=====
-- ■ 7. EMPLOYEES 테이블에서 Self join하여 관리자(매니저)를 출력하여라. ■
```

```
SELECT a.first_name || '의 매니저는 ' || b.first_name || '입니다'
FROM employees a, employees b
WHERE a.manager_id = b.employee_id;
```

```
=====
-- ■ 8. EMPLOYEES 테이블에서 left join하여 관리자(매니저)를 출력하고 매니저 아이디가 없는 사람은
배제하고 하향식으로 하며, PRIOR 급여는 역순으로 ORDER BY 출력하라. ■
```

```
SELECT a.employee_id, a.first_name,
       a.manager_id, b.employee_id, b.first_name
FROM employees a, employees b
WHERE a.manager_id = b.employee_id(+)
      START WITH a.manager_id IS NOT NULL
      CONNECT BY a.manager_id = PRIOR a.employee_id
ORDER BY a.salary DESC;
```

```
=====
-- ■ 9. EMPLOYEES 테이블에서 right join하여 관리자(매니저)가 108번 상향식으로 급여는 역순으로
출력하라. ■
```

```
SELECT a.employee_id, a.first_name, a.manager_id,
       b.employee_id, b.first_name
FROM employees a, employees b
WHERE a.manager_id(+) = b.employee_id
      AND a.manager_id = 108
      CONNECT BY PRIOR a.manager_id = a.employee_id
```

ORDER BY a.salary DESC;

-- ■ 1. 모든 사원의 이름, 부서번호, 부서이름을 표시하시오.(emp,dept) ■

```
SELECT e.ename, e.deptno, d.deptno, d.dname
FROM emp e, dept d
WHERE e.deptno = d.deptno;
```

-- ■ 2. 업무가 MANAGER인 사원의 정보를 이름,업무,부서명,근무지 순으로 출력하시오.(emp,dept) ■

```
SELECT e.ename, e.job, d.dname, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.job = 'MANAGER';
```

-- ■ 3. 커미션을 받고 급여가 1,600이상인 사원의 사원이름,부서명,근무지를 출력하시오 ■

```
SELECT e.ename, d.dname, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.comm IS NOT NULL AND e.comm != 0
      AND e.sal >= 1600;
```

-- ■ 4. 근무지가 CHICAGO인 모든 사원의 이름,업무,부서번호 및 부서이름을 표시하시오. ■

```
SELECT e.ename, e.job, e.deptno, d.deptno, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND d.loc = 'CHICAGO';
```

-- ■ 5. 근무지별로 근무하는 사원의 수가 5명 이하인 경우, 인원이 적은 도시순으로 정렬하시오. (근무 인원이 0명인 곳도 표시) ■

```
SELECT d.loc, COUNT(e.empno)
FROM emp e, dept d
WHERE e.deptno(+) = d.deptno
GROUP BY d.loc
HAVING COUNT(e.empno) <= 5
ORDER BY COUNT(e.empno);
```

-- ■ 6. 사원의 이름 및 사원 번호를 관리자의 이름과 관리자 번호와 함께 표시하고 각각의 열 레이블(label)은 employee, emp#, manager, mgr#로 지정하시오. ■

```
SELECT a.ename "사원 이름", a.empno "사원 번호",
       b.ename "관리자 이름", b.empno "관리자 번호"
FROM emp a, emp b
WHERE a.mgr = b.empno;
```

-- ■ 7. 관리자보다 먼저 입사한 모든 사원의 이름 및 입사일을 관리자의 이름 및 입사일과 함께 표시하고 열 레이블을 각각 employee, emp hired, manager, mgr hired로 지정 ■

```
SELECT a.ename "사원 이름", a.hiredate "사원의 입사일",
       b.ename "매니저 이름", b.hiredate "매니저 입사일"
FROM emp a, emp b
WHERE a.mgr = b.empno
      AND a.hiredate < b.hiredate;
```

-- ■ 8. 사원의 이름 및 사원번호를 관리자의 이름과 관리자 번호와 함께 표시하고 각각의 열 레이블은 employee, emp#, manager, mgr#로 지정하는데 King을 포함하여 관리자가 없는 모든 사원을 표시하도록 하고 결과를 사원번호를 기준으로 정렬 ■

```
SELECT a.ename "사원이름", a.empno "사원번호",
```

```
    b.ename "매니저이름", b.empno "매니저번호"  
FROM emp a, emp b  
WHERE a.mgr = b.empno(+)  
ORDER BY a.empno ASC;
```

```
=====
-- ■ 9. 지정한 부서번호, 사원이름 및 지정한 사원과 동일한 부서에서 근무하는 모든 사원을 표시하도록
질의를 작성하고 부서번호는 department, 사원이름은 employee, 동일한 부서에서 근무하는 사원은
colleague로 표시하시오. (부서번호, 사원이름, 동료 순으로 오름차순 정렬) ■
```

```
SELECT a.ename, a.deptno, b.deptno, b.ename "colleague"
FROM emp a, emp b
WHERE a.deptno = b.deptno
      AND a.empno <> b.empno
      -- AND a.ename = 'SMITH'
ORDER BY a.deptno, a.ename, b.ename;
```

```
=====
-- ■ 10. 10번 부서에서 근무하는 직원들의 부서번호, 부서이름, 사원이름, 월급, 급여등급을 출력하시오. ■
```

```
SELECT e.deptno, d.deptno, d.dname, e.ename, e.sal, s.grade
FROM emp e, dept d, salgrade s
WHERE e.deptno = d.deptno
      AND e.sal BETWEEN s.losal AND s.hisal
      AND e.deptno = 10;
```

```
=====
-- ■ 1. EMPLOYEES 테이블에서 (Kochhar의 급여)보다 많은 직원의 정보를 직원번호,이름,담당업무,급여를
출력하라. ■
```

```
SELECT employee_id, last_name, job_id, salary
FROM employees
-- ORDER BY salary DESC
WHERE salary > (SELECT salary
                FROM employees
                WHERE last_name = 'Kochhar');
```

```
=====
-- ■ 2. EMPLOYEES 테이블에서 (급여의 평균)보다 적은 직원의 정보를
직원번호,이름,담당업무,급여,부서번호를 출력하여라. ■
```

```
SELECT employee_id, last_name, job_id, salary, department_id
FROM employees
WHERE salary < (SELECT AVG(salary)
                FROM employees)
ORDER BY salary DESC;
```

```
=====
-- ■ 3. EMPLOYEES 테이블에서 (100번 부서의 최소 급여)보다 최소 급여가 많은 다른 모든 부서를 출력하라
■
```

```
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) > (SELECT MIN(salary)
                     FROM employees
                     WHERE department_id = 100);
```

```
=====
-- ■ 4. (업무별로 최소 급여)를 받는 직원의 정보를 직원번호,이름,업무,부서번호를 출력하여라. 단 업무별로
정렬하여라. ■
```

```
SELECT employee_id, first_name, job_id, salary, department_id
FROM employees
WHERE (job_id, salary) in (SELECT job_id, MIN(salary)
```



```
FROM employees
GROUP BY job_id);
```

```
=====
-- ■ 5. EMPLOYEES 과 DEPARTMENTS 테이블에서 업무가 SA_MAN 사원의 정보를
이름,업무,부서명,근무지를 출력하라. ■
```

```
SELECT first_name, job_id, e.department_id, d.department_id,
       d.department_name, d.location_id
FROM ( SELECT first_name, job_id, department_id
      FROM employees
      WHERE job_id = 'SA_MAN') e, departments d
WHERE e.department_id = d.department_id;
```

```
=====
-- ■ 6. EMPLOYEES 테이블에서 (가장 많은 사원을 갖는 MANAGER)의 사원번호를 출력하라. ■
```

```
SELECT manager_id, COUNT(*)
FROM employees
GROUP BY manager_id
HAVING COUNT(manager_id) = (SELECT MAX(COUNT(*))
                             FROM employees
                             GROUP BY manager_id);
```

```
=====
-- ■ 7. EMPLOYEES 테이블에서 (가장 많은 사원이 속해 있는 부서 번호)와 사원수를 출력하라. ■
```

```
SELECT department_id, COUNT(*)
FROM employees
GROUP BY department_id
HAVING COUNT(department_id) = (SELECT MAX(COUNT(*))
                                FROM employees
                                GROUP BY department_id);
```

```
=====
-- ■ 8. EMPLOYEES 테이블에서 (사원번호가 123인 사원의 직업과 같고) (사원번호가 192인 사원의
급여(SAL)보다 많은) 사원의 사원번호,이름,직업,급여를 출력하라.■
```

```
SELECT employee_id, first_name, job_id, salary
FROM employees
WHERE job_id = (SELECT job_id from employees WHERE employee_id=123)
  and
  salary > (SELECT salary from employees WHERE employee_id=192);
```

```
=====
-- ■ 9. (직업(JOB)별로 최소급여)를 받는 사원의 정보를 사원번호,이름,업무,부서명을출력하라. 조건1
:직업별로내림차순정렬 ■
```

```
SELECT e.employee_id, e.first_name, salary, e.job_id, d.department_name
FROM employees e, departments d
WHERE (job_id, salary) IN ( SELECT job_id, MIN(salary)
                           FROM employees
                           GROUP BY job_id)
  and e.department_id = d.department_id
ORDER BY e.job_id;
```

```
=====
-- ■ 10. EMPLOYEES 테이블에서 (50번 부서의 최소 급여)를 받는 사원보다 많은 급여를 받는 사원의 정보를
사원번호,이름,업무,입사일자,급여,부서번호를 출력하라. 단 50번은 제외 ■
```

```
SELECT employee_id, first_name, job_id, hire_date, salary, department_id
FROM employees
WHERE salary > (SELECT MIN(salary)
                from employees
```

```
WHERE department_id = 50)  
AND department_id != 50;
```

=====

-- ■ 11. EMPLOYEES 테이블에서 50번 부서의 최고 급여를 받는 사원 보다 많은 급여를 받는 사원의 정보를
사원번호,이름,업무,입사일자,급여,부서번호를 출력하라. 단50번은 제외 ■

```
SELECT employee_id, first_name, job_id, hire_date, salary, department_id
FROM employees
WHERE salary > (SELECT MAX(salary)
                from employees
                WHERE department_id = 50)
AND department_id <> 50;
```

=====

-- ■ 12. (BLAKE와 같은 부서)에 있는 사원들의 이름과 입사일을 구하는데 BLAKE는 제외하고
출력하시오.(BLAKE가 여러명일 수 있음) ■

```
SELECT ename, hiredate, deptno
FROM emp
WHERE deptno in(SELECT deptno from emp WHERE ename = 'BLAKE')
AND ename != 'BLAKE';
```

=====

-- ■ 13. (평균급여)보다 많은 급여를 받는 사원들의 사원번호, 이름, 월급을 출력하는데 월급이 높은
사람순으로 출력하시오.■

```
SELECT empno, ename, sal
FROM emp
WHERE sal > (SELECT AVG(sal) from emp)
ORDER BY sal DESC;
```

=====

-- ■ 14. (10번부서에서 급여를 가장 적게 받는 사원)과 동일한 급여를 받는 사원의 이름을 출력하시오. ■

```
SELECT ename, sal
FROM emp
WHERE sal = (SELECT MIN(sal) FROM emp WHERE deptno = 10);
```

=====

-- ■ 15. 사원수가 3명이 넘는 부서의 부서명과 사원수를 출력하시오. ■

```
SELECT d.dname, COUNT(e.empno)
FROM emp e, dept d
WHERE e.deptno = d.deptno
GROUP BY d.dname
HAVING count(e.empno) > 3;
```

=====

-- ■ 16. (사원번호가 7844인 사원)보다 빨리 입사한 사원의 이름과 입사일을 출력하시오. ■

```
SELECT ename, hiredate
FROM emp
WHERE hiredate < (SELECT hiredate FROM emp WHERE empno = 7844);
```

=====

-- ■ 17. (직속상사(mgr)가 KING)인 모든 사원의 이름과 급여를 출력하시오. ■

```
SELECT ename, sal, mgr
FROM emp
WHERE mgr = (SELECT empno FROM emp WHERE ename = 'KING');
```

=====

-- ■ 18. (20번 부서에서 가장 급여를 많이 받는 사원)과 동일한 급여를 받는 사원의 이름과 부서명,급여,
급여등급을 출력하시오.(emp, dept, salgrade) ■

```
SELECT e.ename, d.dname, e.sal, s.grade
FROM emp e, dept d, salgrade s
WHERE e.deptno = d.deptno
AND e.sal = (SELECT MAX(sal) FROM emp WHERE deptno = 20)
```

AND e.sal BETWEEN s.losal AND s.hisal;

=====

-- ■ 19. (총급여(sal+comm))가 (평균 급여)보다 많은 급여를 받는 사람의 부서번호, 이름, 총급여, 커미션을 출력하시오.(커미션은 유(O),무(X)로 표시하고 컬럼명은 "comm유무" 출력) ■

```
SELECT deptno, ename, sal + NVL(comm, 0), comm, NVL2(comm, 'O', 'X')
FROM emp
WHERE sal + NVL(comm, 0) > (SELECT AVG(sal) FROM emp);
```

=====

-- ■ 20. (CHICAGO 지역에서 근무하는 사원의 평균 급여)보다 높은 급여를 받는 사원의 이름과 급여, 지역명을 출력하시오. ■

```
SELECT e.ename, e.sal, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND sal > ( SELECT AVG(sal)  -- 부서의 평균 급여
                  FROM emp
                  WHERE deptno = ( SELECT deptno -- 지역번호
                                   FROM dept
                                   WHERE loc = 'CHICAGO')
                  GROUP BY deptno);
```

=====

-- ■ 21. (업무가 SALESMAN인 직원이 2명 이상)인 부서의 이름, 근무하는 사원의 이름, 업무를 출력하시오.(컬럼명은 부서명, 사원명, 업무로 출력) ■

```
SELECT d.dname "부서명", e.ename "사원명", job 업무
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.deptno in ( SELECT deptno
                        FROM emp
                        WHERE job = 'SALESMAN'
                        GROUP BY deptno
                        HAVING COUNT(job) >= 2);
```

=====

-- ■ 22. (커미션이 없는 사원들 중 월급이 가장 높은 사원)의 이름과 급여등급을 출력하시오. ■

```
SELECT e.ename, s.grade
FROM emp e, salgrade s
WHERE e.sal = (SELECT MAX(sal)
               FROM emp
               WHERE comm IS NULL)
      AND e.sal >= s.losal AND e.sal <= s.hisal;
```

=====

-- ■ 23. (SMITH의 관리자(mgr))의 이름과 부서명, 근무지역을 출력하시오. ■

```
SELECT e.empno, e.ename, d.dname, d.loc
FROM emp e, dept d
WHERE e.deptno = d.deptno
      AND e.empno = (SELECT mgr FROM emp WHERE ename = 'SMITH');
```

=====

-- ■ 1. EMPLOYEES 테이블에서 부서별로 인원수, 평균, 급여, 급여의 합, 최소 급여, 최대 급여를 포함하는 EMP_DEPTNO 테이블을 생성하라. ■

```
CREATE TABLE EMP_DEPTNO( DEPNO, E_COUNT, E_AVG, E_SUM, E_MIN, E_MAX )
AS (
  SELECT DEPARTMENT_ID DEPTNO, COUNT(DEPARTMENT_ID), TRUNC(AVG(SALARY), 3),
  SUM(SALARY), MIN(SALARY), MAX(SALARY)
  FROM EMPLOYEES
  GROUP BY DEPARTMENT_ID
```

);

=====

-- ■ 2. EMP_DEPTNO 테이블에 ETC COLUMN을 추가하라. 단 자료형은 VARCHAR2(50) 사용하라. ■

ALTER TABLE EMP_DEPTNO ADD ETC VARCHAR2(50);

DESC EMP_DEPTNO;

=====

-- ■ 3. EMP_DEPTNO 테이블에 ETC COLUMN을 수정하라. 자료형은 VARCHAR2(15)로 하라. ■

```
ALTER TABLE EMP_DEPTNO MODIFY ETC VARCHAR(15);
DESC EMP_DEPTNO;
```

=====

-- ■ 4. EMP_DEPTNO 테이블에 있는 ETC를 삭제하고 확인하라. ■

```
ALTER TABLE EMP_DEPTNO DROP COLUMN ETC;
DESC EMP_DEPTNO;
```

=====

-- ■ 5. 이전에 생성한 EMP_DEPTNO 테이블의 이름을 EMP_DEPT로 변경하라. ■

```
ALTER TABLE EMP_DEPTNO RENAME TO EMP_DEPT;
```

=====

-- ■ 6. EMP_DEPT 테이블을 삭제하라. ■

```
DROP TABLE EMP_DEPT;
```

=====

-- ■ 7. EMPLOYEES 테이블을 EMP 테이블을 생성하고 복제하도록 하라. (데이터 포함) ■

```
CREATE TABLE EMP AS (SELECT * FROM EMPLOYEES);
```

=====

-- ■ 8. EMP 테이블에 ROW를 추가해 봅니다. 다만, 반드시 데이터를 기입을 안해도 되면, NULL로 설정하도록 한다. ■

```
INSERT INTO EMP
VALUES (1222, 'HEELO', 'WORLD', 'EMAIN', NULL, SYSDATE, 'SA_PKS', NULL, NULL, NULL, NULL);
```

```
SELECT * FROM EMP WHERE EMPLOYEE_ID = 1222;
```

=====

-- ■ 9. EMPLOYEES 테이블에서 EMPNO, ENAME, SAL, HIREDATE의 COLUMN만 선택하여 EMP_10 테이블을 생성(데이터 미포함) 한 후 10번

-- 부서만 선택하여 이에 값을 EMP_10테이블에 입력하라. ■

```
DROP TABLE EMP_10;
CREATE TABLE EMP_10 AS (
    SELECT EMPLOYEE_ID, LAST_NAME, SALARY, HIRE_DATE
    FROM EMPLOYEES
    WHERE DEPARTMENT_ID = 10
);
```

=====

-- ■ 11. EMPLOYEES 테이블에서 사원 번호가 180인 사원의 부서를 20, 급여를 3500으로 변경하여라. ■

```
UPDATE EMPLOYEES
SET DEPARTMENT_ID =20, SALARY=3500
WHERE EMPLOYEE_ID = 180;
```

=====

-- ■ 12. EMPLOYEES 테이블에서 Smith의 업무와 급여가 Hall의 업무와 급여와 일치하도록 수정하라. ■

```
UPDATE EMPLOYEES
SET (JOB_ID, SALARY) = (SELECT JOB_ID, SALARY FROM EMPLOYEES WHERE LAST_NAME='Hall')
WHERE LAST_NAME='Smith';
```

=====

-- ■ 1. EMPLOYEES 테이블에서 20번 부서의 세부 사항을 포함하는 EMP_20 VIEW를 생성하라 ■

```
CREATE OR REPLACE VIEW EMP_20 AS SELECT * FROM EMPLOYEES;
SELECT * FROM EMP_20;
```

=====

-- ■ 2. EMPLOYEES 테이블에서 30번 부서만 EMPLOYEE_ID 를 emp_no로 LAST_NAME을 name으로 SALARY를 sal로 바꾸어 EMP_30 VIEW 를 생성하라 ■

```
DESC EMPLOYEES;
```

```
CREATE OR REPLACE VIEW EMP_30(  
    EMP_NO,  
    FIRST_NAME,  
    NAME,  
    EMAIL,  
    PHONE_NUMBER,  
    HIRE_DATE,  
    JOB_ID,  
    SAL,  
    COMMISSION_PCT,  
    MANAGER_ID,  
    DEPARTMENT_ID  
) AS (SELECT * FROM EMPLOYEES);
```

=====

-- ■ 3. 부서별로 부서명, 최소 급여, 최대 급여, 부서의 평균 급여를 포함하는 DEPT_SUM VIEW을
생성하여라. ■

```
CREATE OR REPLACE VIEW DEPT_SUM(  
    DEPARTMENT_NAME,  
    MIN_SAL,  
    MAX_SAL,  
    SUM_SAL,  
    AVG_SAL  
)  
AS (  
    SELECT D.DEPARTMENT_NAME, MIN(E.SALARY), MAX(E.SALARY), SUM(E.SALARY),  
    AVG(E.SALARY)  
    FROM EMPLOYEES E LEFT JOIN DEPARTMENTS D ON E.DEPARTMENT_ID=D.DEPARTMENT_ID  
    GROUP BY D.DEPARTMENT_NAME  
);
```

=====

-- ■ 4. 앞에서 생성한 EMP_20, EMP_30 VIEW을 삭제하여라. ■

```
DROP VIEW EMP_20;  
DROP VIEW EMP_30;
```