/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQL: 대소문자 구분 안함 (값은 구분)

Select 기본 구문 - 연산자, 컬럼 별칭

select 컬럼명, 컬럼명 [,....] =>조회할 컬럼 지정

from 테이블명 =>조회할 테이블 지정

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

desc emp; --테이블의 컬럼 조회

--EMP 테이블의 모든 컬럼의 모든 항목을 조회.

select \* from emp;

select emp\_id, emp\_name, job, mgr\_id, hire\_date, salary, comm\_pct, dept\_name from emp;

--EMP 테이블의 직원 ID(emp\_id), 직원 이름(emp\_name), 업무(job) 컬럼의 값을 조회.

select emp\_id, emp\_name, job from emp;

--EMP 테이블의 업무(job) 어떤 값들로 구성되었는지 조회. - 동일한 값은 하나씩만 조회되도록 처리. boolean값

--select distinct 컬럼명[,...]

select distinct job from emp;

--EMP 테이블의 부서명(dept\_name)이 어떤 값들로 구성되었는지 조회 - 동일한 값은 하나씩만 조회되도록 처리.

select distinct dept\_name from emp;

select distinct job, dept\_name from emp order by 2; --job과 deptname을 하나로 보아 쌍이 중복되지 않는것들을 출력함

--EMP 테이블에서 emp\_id는 직원ID, emp\_name은 직원이름, hire\_date는 입사일, salary는 급여, dept\_name은 소속부서 별칭으로 조회한다.

--select 컬럼 as 별칭(alias); 컬럼에서 조회한 것을 별칭으로 보여줘. as는 생략 가능하다.

select emp\_id as 직원ID, emp\_name as 직원이름, hire\_date as 입사일, salary as 급여, dept\_name as 소속부서 from emp;

select emp\_id as "직원 ID", --별칭에 공백이 들어가면 " "로 묶어준다.

emp\_name as "직원 이름",

hire\_date as 입사일,

salary as 급여,

dept\_name as 소속부서

from emp;

select salary\*20 as "20개월급여" from emp;

select sum(salary) as "총급여" from emp;

/\*

연산자

산술연산자: +,-,\*,/

- date : +,- >>day를 +,-

- 피연산자가 null인 경우 결과는 null

연결연산자: 값들을 합칠(붙일) 때 사용

값||값

\*/

select 1+1, 2-1, 3\*5, 6/3, 10/3 , round(10/3, 2) from dual;

select sysdate from dual;

select sysdate, sysdate+10, sysdate-10 from dual;

select 10+null from dual;

select 20||'세' from dual;

select '$'||5000 as 가격 from dual;

--EMP 테이블에서 직원의 이름(emp\_name), 급여(salary) 그리고 급여 + 1000 한 값을 조회.

select emp\_name, salary, salary + 1000 from emp; --피연산자가 컬럼인 경우 행단위 연산.

--EMP 테이블에서 입사일(hire\_date)과 입사일에 10일을 더한 날짜를 조회.

select hire\_date, hire\_date+10 from emp;

--TODO: EMP 테이블에서 직원의 ID(emp\_id), 이름(emp\_name), 급여(salary), 커미션\_PCT(comm\_pct), 급여에 커미션\_PCT를 곱한 값을 조회.

select emp\_id, emp\_name, salary, comm\_pct, salary\*comm\_pct as 커미션 from emp;

--TODO: EMP 테이블에서 급여(salary)을 연봉으로 조회. (곱하기 12)

select salary 월급, salary\*12 as 연봉 from emp;

--TODO: EMP 테이블에서 직원이름(emp\_name)과 급여(salary)을 조회. 급여 앞에 $를 붙여 조회.

select emp\_name, '$'||salary as salary from emp;

--TODO: EMP 테이블에서 입사일(hire\_date) 30일전, 입사일, 입사일 30일 후를 조회

select hire\_date-30 as 입사30일전, hire\_date as 입사일, hire\_date+30 as 입사30일후 from emp;

select q'[I'm]' from dual;

select 'I''m' from dual;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Where 절을 이용한 행 제한

컬럼의 값들을 조건으로 연산식을 만들어준다.

where 행 제한조건식

컬럼=값a :컬럼의 값이 값a와 같은 행만 볼때

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

--EMP 테이블에서 직원\_ID(emp\_id)가 110인 직원의 이름(emp\_name)과 부서명(dept\_name)을 조회

select emp\_name, dept\_name from emp where emp\_id=110;

--EMP 테이블에서 'Sales' 부서에 속하지 않은 직원들의 ID(emp\_id), 이름(emp\_name), 부서명(dept\_name)을 조회.

select emp\_id, emp\_name, dept\_name

from emp

where dept\_name<>'Sales'; --!=,<>: 같지 않은 행 값은 대소문자 구별 'sales'와 'Sales'는 같지 않다.

--EMP 테이블에서 급여(salary)가 $10,000를 초과인 직원의 ID(emp\_id), 이름(emp\_name)과 급여(salary)를 조회

select emp\_id, emp\_name, salary

from emp

where salary > 10000;

--EMP 테이블에서 커미션비율(comm\_pct)이 0.2~0.3 사이인 직원의 ID(emp\_id), 이름(emp\_name), 커미션비율(comm\_pct)을 조회.

select emp\_id, emp\_name, comm\_pct

from emp

where comm\_pct between 0.2 and 0.3; --between <=> not between

--EMP 테이블에서 커미션을 받는 직원들 중 커미션비율(comm\_pct)이 0.2~0.3 사이가 아닌 직원의 ID(emp\_id), 이름(emp\_name), 커미션비율(comm\_pct)을 조회.

select emp\_id, emp\_name, comm\_pct

from emp

where comm\_pct not between 0.2 and 0.3; --null은 알수 없는 값이기 때문에 조회조건에서 빠짐

--EMP 테이블에서 업무(job)가 'IT\_PROG' 거나 'ST\_MAN' 인 직원의 ID(emp\_id), 이름(emp\_name), 업무(job)을 조회.

select emp\_id, emp\_name, job

from emp

--where job='IT\_PROG' or job='ST\_MAN';

where job in('IT\_PROG', 'ST\_MAN');

--EMP 테이블에서 업무(job)가 'IT\_PROG' 나 'ST\_MAN' 가 아닌 직원의 ID(emp\_id), 이름(emp\_name), 업무(job)을 조회.

select emp\_id, emp\_name, job

from emp

--where job<>'IT\_PROG' and job<>'ST\_MAN';

where job not in('IT\_PROG','ST\_MAN'); --in<=>not in

--EMP 테이블에서 직원 이름(emp\_name)이 S로 시작하는 직원의 ID(emp\_id), 이름(emp\_name)

select emp\_id, emp\_name

from emp

where emp\_name like 'S%'; --S뒤의 글자수 제한X: %, 글자수 제한: \_를 원하는 수만큼 붙임 ex)김씨면서 외자인 이름: like '김\_';

/\*

xxx로 시작하는: xxx%

xxx로 끝나는: %xxx

xxx가 들어간: %xxx%

글자수: \_

3번째 글자가 x인: \_\_x%

\*/

--EMP 테이블에서 직원 이름(emp\_name)이 S로 시작하지 않는 직원의 ID(emp\_id), 이름(emp\_name)

select emp\_id, emp\_name

from emp

where emp\_name not like 'S%';

--EMP 테이블에서 직원 이름(emp\_name)이 en으로 끝나는 직원의 ID(emp\_id), 이름(emp\_name)을 조회

select emp\_id, emp\_name

from emp

where emp\_name like '%en';

--EMP 테이블에서 직원 이름(emp\_name)의 세 번째 문자가 “e”인 모든 사원의 이름을 조회

select emp\_name

from emp

where emp\_name like '\_\_e%';

-- EMP 테이블에서 직원의 이름에 '%' 가 들어가는 직원의 ID(emp\_id), 직원이름(emp\_name) 조회

select emp\_name

from emp

where emp\_name like '%\_%%' escape '\_'; --%를 원래 like에서 쓰는 의미가 아닌 그냥 문자열로 사용하고 싶을 때 사용

--like에서 %,\_ (패턴문자) 앞에 특수 문자를 붙이면 %,\_ 자체 문자를 가리킨다.

--ctrl+/ >> 한줄 주석처리 단축키

--EMP 테이블에서 부서명(dept\_name)이 null인 직원의 ID(emp\_id), 이름(emp\_name), 부서명(dept\_name)을 조회.

select emp\_id, emp\_name, dept\_name

from emp

where dept\_name is null; --null은 정해지지 않은 값이기때문에 =연산자를 이용해서 비교할 수 없기 때문에 =연산자를 사용하지 않고 is null을 이용한다.

--부서명(dept\_name) 이 NULL이 아닌 직원의 ID(emp\_id), 이름(emp\_name), 부서명(dept\_name) 조회

select emp\_id, emp\_name, dept\_name

from emp

where dept\_name is not null;

--TODO: EMP 테이블에서 업무(job)가 'IT\_PROG'인 직원들의 모든 컬럼의 데이터를 조회.

select \*

from emp

where job = 'IT\_PROG';

--TODO: EMP 테이블에서 업무(job)가 'IT\_PROG'가 아닌 직원들의 모든 컬럼의 데이터를 조회.

select \*

from emp

where job<>'IT\_PROG';

--TODO: EMP 테이블에서 이름(emp\_name)이 'Peter'인 직원들의 모든 컬럼의 데이터를 조회

select \*

from emp

where emp\_name='Peter';

--TODO: EMP 테이블에서 급여(salary)가 $10,000 이상인 직원의 ID(emp\_id), 이름(emp\_name)과 급여(salary)를 조회

select emp\_id, emp\_name, salary

from emp

where salary >=10000;

--TODO: EMP 테이블에서 급여(salary)가 $3,000 미만인 직원의 ID(emp\_id), 이름(emp\_name)과 급여(salary)를 조회

select emp\_id, emp\_name, salary

from emp

where salary<3000;

--TODO: EMP 테이블에서 급여(salary)가 $3,000 이하인 직원의 ID(emp\_id), 이름(emp\_name)과 급여(salary)를 조회

select emp\_id, emp\_name, salary

from emp

where salary<=3000;

--TODO: 급여(salary)가 $4,000에서 $8,000 사이에 포함된 직원들의 ID(emp\_id), 이름(emp\_name)과 급여(salary)를 조회

select emp\_id, emp\_name, salary

from emp

where salary between 4000 and 8000;

--TODO: 급여(salary)가 $4,000에서 $8,000 사이에 포함되지 않는 모든 직원들의 ID(emp\_id), 이름(emp\_name), 급여(salary)를 select emp\_id, emp\_name, salary

select emp\_id, emp\_name, salary

from emp

where salary not between 4000 and 8000;

--TODO: EMP 테이블에서 2007년 이후 입사한 직원들의 ID(emp\_id), 이름(emp\_name), 입사일(hire\_date)을 조회.

select emp\_id, emp\_name, hire\_date

from emp

where hire\_date >= '2007-01-01';

--where to\_char(hire\_date,'yyyy')>='2007'

--TODO: EMP 테이블에서 2004년에 입사한 직원들의 ID(emp\_id), 이름(emp\_name), 입사일(hire\_date)을 조회.

select emp\_id, emp\_name, hire\_date

from emp

where hire\_date like '2004%';

--TODO: EMP 테이블에서 2005년 ~ 2007년 사이에 입사(hire\_date)한 직원들의 ID(emp\_id), 이름(emp\_name), 업무(job), 입사일(hire\_date)을 조회.

select emp\_id, emp\_name, job, hire\_date

from emp

where hire\_date between '2005-01-01' and '2007-12-31';

--TODO: EMP 테이블에서 직원의 ID(emp\_id)가 110, 120, 130 인 직원의 ID(emp\_id), 이름(emp\_name), 업무(job)을 조회

select emp\_id, emp\_name, job

from emp

where emp\_id in (110,120,130);

--TODO: EMP 테이블에서 부서(dept\_name)가 'IT', 'Finance', 'Marketing' 인 직원들의 ID(emp\_id), 이름(emp\_name), 부서명(dept\_name)을 조회.

select emp\_id, emp\_name, dept\_name

from emp

where dept\_name in('IT','Finance','Marketing');

--TODO: EMP 테이블에서 'Sales' 와 'IT', 'Shipping' 부서(dept\_name)가 아닌 직원들의 ID(emp\_id), 이름(emp\_name), 부서명(dept\_name)을 조회.

select emp\_id, emp\_name, dept\_name

from emp

where dept\_name not in('Sales','IT','Shipping');

--TODO: EMP 테이블에서 급여(salary)가 17,000, 9,000, 3,100 인 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 급여(salary)를 조회.

select emp\_id, emp\_name, job, salary

from emp

where salary in (17000,9000,3100);

--TODO EMP 테이블에서 업무(job)에 'SA'가 들어간 직원의 ID(emp\_id), 이름(emp\_name), 업무(job)를 조회

select emp\_id, emp\_name, job

from emp

where job like '%SA%';

--TODO: EMP 테이블에서 업무(job)가 'MAN'로 끝나는 직원의 ID(emp\_id), 이름(emp\_name), 업무(job)를 조회

select emp\_id, emp\_name, job

from emp

where job like '%MAN';

--TODO. EMP 테이블에서 커미션이 없는(comm\_pct가 null인) 모든 직원의 ID(emp\_id), 이름(emp\_name), 급여(salary) 및 커미션비율(comm\_pct)을 조회

select emp\_id, emp\_name, salary, comm\_pct

from emp

where comm\_pct is null;

--TODO: EMP 테이블에서 커미션을 받는 모든 직원의 ID(emp\_id), 이름(emp\_name), 급여(salary) 및 커미션비율(comm\_pct)을 조회

select emp\_id, emp\_name, salary, comm\_pct

from emp

where comm\_pct is not null;

--TODO: EMP 테이블에서 관리자 ID(mgr\_id) 없는 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 소속부서(dept\_name)를 조회

select emp\_id, emp\_name, job, dept\_name

from emp

where mgr\_id is null;

--TODO : EMP 테이블에서 연봉(salary \* 12) 이 200,000 이상인 직원들의 모든 정보를 조회.

select \*

from emp

where salary\*12>=200000;

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WHERE 조건이 여러개인 경우

AND 참 and 참 => (나머진 거짓)

OR 거짓 or 거짓=> (나머진 참)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-- EMP 테이블에서 업무(job)가 'SA\_REP' 이고 급여(salary)가 $9,000 인 직원의 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 급여(salary)를 조회.

select emp\_id,emp\_name, job, salary

from emp

where job='SA\_REP' and salary=9000;

-- EMP 테이블에서 업무(job)가 'FI\_ACCOUNT' 거나 급여(salary)가 $8,000 이상인인 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 급여(salary)를 조회.

select emp\_id,emp\_name, job, salary

from emp

where job='FI\_ACCOUNT' or salary>=8000;

select emp\_id,emp\_name, job, salary

from emp

where not(job='FI\_ACCOUNT' or salary>=8000); -- = job<>'FI\_ACCOUNT' and salary<8000

--TODO: EMP 테이블에서 부서(dept\_name)가 'Sales이'고 업무(job)가 'SA\_MAN' 이고 급여가 $13,000 이하인

-- 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 급여(salary), 부서(dept\_name)를 조회

select emp\_id, emp\_name, job, salary, dept\_name

from emp

where dept\_name='Sales' and job ='SA\_MAN' and salary<=13000;

--TODO: EMP 테이블에서 업무(job)에 'MAN'이 들어가는 직원들 중에서 부서(dept\_name)가 'Shipping' 이고 2005년이후 입사한

-- 직원들의 ID(emp\_id), 이름(emp\_name), 업무(job), 입사일(hire\_date), 부서(dept\_name)를 조회

select emp\_id, emp\_name, job, hire\_date, dept\_name

from emp

where job like '%MAN%' and dept\_name='Shipping' and hire\_date >='2005-01-01';

--TODO: EMP 테이블에서 입사년도가 2004년인 직원들과 급여가 $20,000 이상인

-- 직원들의 ID(emp\_id), 이름(emp\_name), 입사일(hire\_date), 급여(salary)를 조회.

select emp\_id, emp\_name, hire\_date, salary

from emp

where hire\_date like '2004%' or salary>=20000;

--TODO : EMP 테이블에서, 부서이름(dept\_name)이 'Executive'나 'Shipping' 이면서 급여(salary)가 6000 이상인 사원의 모든 정보 조회.

select \*

from emp

where dept\_name in('Executive','Shipping') and salary >=6000;

--TODO: EMP 테이블에서 업무(job)에 'MAN'이 들어가는 직원들 중에서 부서이름(dept\_name)이 'Marketing' 이거나 'Sales'인

-- 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 부서(dept\_name)를 조회

select emp\_id, emp\_name, job, dept\_name

from emp

where job like '%MAN%' and dept\_name in ('Marketing','Sales');

--TODO: EMP 테이블에서 업무(job)에 'MAN'이 들어가는 직원들 중 급여(salary)가 $10,000 이하이 거나 2008년 이후 입사한

-- 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 입사일(hire\_date), 급여(salary)를 조회

--연산자 우선순위 AND>OR

select emp\_id, emp\_name, job, hire\_date, salary

from emp

where job like '%MAN%' and (salary<=10000 or hire\_date >='2008-01-01');

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

order by를 이용한 정렬

order by절은 select절의 마지막에 온다.

order by 정렬기준컬럼 정렬방식[,...]

정렬기준컬럼 : 컬럼이름, 컬럼의순번(select절의 선언 순서), 별칭(alias)

select salary 급여, hire\_date from emp

order by salary 또는 1 또는 급여 asc/desc;

정렬방식:

ASC : 오름차순( 작은것 > 큰것) 기본방식이기 때문에 생략가능

DESC: 내림차순( 큰것 > 작은것)

문자열 오름차순: 숫자 => 대문자 => 소문자 => 한글

Date 오름차순: 과거 => 미래

null 오름차순: 맨마지막에 나옴.

order by salary asc, emp\_id desc

salary로 전체 정렬을 하고 salary가 같은 행은 emp\_id로 정렬.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

-- 직원들의 전체 정보를 직원 ID(emp\_id)가 큰 순서대로 정렬해 조회

select \*

from emp

order by emp\_id desc;

-- 직원들의 id(emp\_id), 이름(emp\_name), 업무(job), 급여(salary)를

-- 업무(job) 순서대로 (A -> Z) 조회하고 업무(job)가 같은 직원들은 급여(salary)가 높은 순서대로 2차 정렬해서 조회.

select emp\_id, emp\_name, job, salary

from emp

order by job asc, salary desc;

--order by 3, 4 desc;

--부서명을 부서명(dept\_name)의 오름차순으로 정렬해 조회하시오.

select distinct dept\_name

from emp

--ORDER BY 1;

order by dept\_name asc;

--오름차순을 유지하면서 null만 먼저 볼 때 nulls first를 붙임

--order by dept\_name nulls first;

--내림차순을 유지하면서 null을 마지막에 볼 때 nulls last를 붙임

--order by dept\_name desc nulls last;

--TODO: 급여(salary)가 $5,000을 넘는 직원의 ID(emp\_id), 이름(emp\_name), 급여(salary)를 급여가 높은 순서부터 조회

select emp\_id, emp\_name, salary

from emp

where salary>5000

order by salary desc;

--TODO: 급여(salary)가 $5,000에서 $10,000 사이에 포함되지 않는 모든 직원의 ID(emp\_id), 이름(emp\_name), 급여(salary)를 이름(emp\_name)의 오름차순으로 정렬

select emp\_id, emp\_name, salary

from emp

where salary not between 5000 and 10000

order by emp\_name asc;

--TODO: EMP 테이블에서 직원의 ID(emp\_id), 이름(emp\_name), 업무(job), 입사일(hire\_date)을 입사일(hire\_date) 순(오름차순)으로 조회.

select emp\_id, emp\_name, job, hire\_date

from emp

order by hire\_date asc;

--TODO: EMP 테이블에서 ID(emp\_id), 이름(emp\_name), 급여(salary), 입사일(hire\_date)을 급여(salary) 오름차순으로 정렬하고 급여(salary)가 같은 경우는 입사일(hire\_date)가 오래된 순서로 정렬.

select emp\_id, emp\_name, salary, hire\_date

from emp

order by salary asc, hire\_date asc;

--치환변수

select \* from emp

where emp\_id=&id;

select \* from emp

where dept\_name = &dept\_name

and salary > &salary;

--oracle명령어 치환변수를 사용하지 않겠다.

set define off;

--oracle명령어 치환변수를 사용하겠다.

set define on;

select \* from emp

where dept\_name = '&dept\_name'

and salary > 6000;