**2장**

- 테이블 구조를 확인하기 위해서 사용하는 명령어

describe emp;

- 전체 테이블 내용 조회

select ename, sal

from emp;

- emp 테이블의 모든 레코드 조회

select \*

from emp;

- 사원들의 직급(job) 조회

select job

from emp;

- 컬럼 명에 별칭 설정 (as를 써도 되고 안 써도 됨)

select empno as 사원번호, ename, mgr 관리자번호

from emp;

- 문자열 결합 연산자 ||

select empno || ‘의 관리자 번호는 ‘ || mgr 사원\_관리자

from emp;

- SMITH(CLERK)와 같이 출력되도록 사원명(직급) 검색

select ename || ‘(‘ || job || ‘)’ “name and job”

from emp;

- SMITH’s sal is $800 같이 출력되도록 emp 테이블 조회

select ename || ‘’’ sal is $’ || sal “name and sal”

from emp;

- SQL 기본 구문

select distinct 컬럼명1, [컬럼명2, …, 컬럼명n] \* as alias

from 테이블명

where 컬럼명 연산자 값

order by 컬럼명 [asc|desc]

- where 연산자

- 비교 연산자: =, !=, ><, >=, <=, >, <

- 논리 연산자: and, or, not

- 범위 연산자: between a and b

- 포함 연산자: in, not in

- 문자열 연산자: like(%, \_)

- null 여부 판별 연산자: is null, is not null

- 직급(job)이 manager인 사원명 jator

select ename

from emp

where lower(job) = ‘manager’;

- 10번 부서에 소속된 사원명 검색

select ename

from emp

where deptno = 10;

- 1985년 1월 1일 이후 입사한 사원명 검색

select ename, hiredate

from emp

where hiredate >= ‘85/01/01’;

- 봉급이 3000 이상 5000 이하인 사원명 검색

select ename, sal

from emp

where sal between 3000 and 5000;

- 직급이 manager이고 10번 부서에 소속된 사원명 검색

select ename, job, deptno

from emp

where lower(job = ‘manager’ and deptno = 10;

- 직급이 manager이거나 salesman을 가진 사원명 검색

select ename, job

from emp

where lower(job) in (‘manager’, ‘salseman’);

- smith 사원의 직급 검색

select job

from emp

where lower(ename) = ‘smith’;

- 이름 중에 ‘S’ 문자를 포함하고 있는 사원들의 직급

select job, ename

from emp

where ename like ‘%S%’;

- 이름이 ‘S’로 시작하고 4글자가 있는 사원들의 직급

select job, ename

from emp

where ename like ‘%S\_\_\_\_’;

- %: 0개 이상의 임의의 문자열

- \_: 1개의 임의의 문자

- mgr이 null인 사원명 조회 (값이 없기에 비교 연산(=)이 불가능해서 is 연산자를 사용

select ename, comm

from emp

where mgr is null;

- and가 or보다 우선순위가 높다

select ename, comm, sal

from emp

where sal > 1000 amd (comm < 100 or comm is null);

- &: 사용자로부터 매개변수 값을 입력 받기

select ename

from emp

where deptno = &부서번호;

- 입사일을 기준(오름차순)으로 사원명 검색 (asc 오름차순, desc 내림차순)

select ename, hiredate

from emp

order by hiredate asc;

- 사원명, 봉급, 봉급\*12(연봉) 검색, 연봉을 기준으로 내림차순으로 정렬 출력

select ename, sal, sal\*12 연봉

from emp

order by 3 desc;

- 1번째 기준 값이 동일할 경우 2번째 기준 값으로 출력

select ename, sal, deptno

from emp

order by deptno, sal desc;

**3장**

- 문자 함수

- initcan() 앞 글자만 대문자로 출력, upper() 대문자로 출력, lower() 소문자로 출력

select ename, initcap(ename), upper(ename), lower(ename)

from emp;

- length() 문자의 길이, lengthb() 문자의 길이의 바이트 값

select ename, length(ename), lengthb(ename)

from emp;

- 더미 테이블

select length(‘김동현’), ‘lengthb(‘김동현’)

from dual;

- concap() 문자열 결합

select concat(ename, sal), ename || sal

from emp;

- substr(문자, 첫 번째 값, 출력할 개수)

select ename, substr(ename, 1, 3), substr(hiredate, 1, 2) year, substr(hiredate, 4, 2) month, substr(hiredate, -2, 2) day

from emp;

select substr(‘김동현’, 2, 2)

from dual;

- instr(‘문자열’ 또는 컬럼, 찾는 문자, 시작 위치, 몇 번째인지(1))

select instr(‘a-b-c-d-e’, ‘c’, 1), instr(‘a-b-c-d-e’, ‘-‘, 1, 2)

from dual;

select instr(hiredate, ‘/’, 1, 2)

from emp;

- lpad(‘문자열’ 또는 컬럼, ‘자릿수’, ‘채울 문자’), rpad(‘문자열’ 또는 컬럼, ‘자릿수’, ‘채울 문자’)

select lpad(‘dhkim’, 10, ‘\*’), rpad(‘dhkim’, 10, ‘#’)

from dual;

- ltrim(‘문자열’ 또는 컬럼, ‘제거할 문자’)

select ltrim(‘ dhkim d’, ‘ ‘), rtrim(‘ dhkim d’, ‘ ‘)

from dual;

- replace(‘문자열’ 또는 컬럼, ‘문자1’, ‘문자2’)

select replace(‘dhkim’, ‘kim’, ‘lee’)

from dual;

select replace(ename, substr(ename, 1, 1), ‘\*’)

from emp;

- 숫자 함수

- round(숫자, 소수점 자릿수(1이면 소수점 첫 번째 자리, -1이면 10의자리)) 반올림

select round(123.456, 0), round(123.456, 1), round(123.456, -1)

from dual;

- trunc(숫자, 소수점 자릿수(1이면 소수점 첫 번째 자리, -1이면 10의 자리)) 버림

select trunc(123.456, 0), trunc(123.456, 1), trunc(123.456, -1)

from dual;

- 수의 연산

select 10+2, 10-2, 10\*2, 10/2

from dual;

- mod() 나머지, ceil() 가장 가까운 큰 정수, floor() 가장 가까운 작은 정수

select mod(11, 2) 나머지, ceil(123.456), floor(123.456)

from dual;

- power() 제곱 구하기

from power(2, 8)

from dual;

- 날짜 함수

select sysdate 현재날짜, sysdate+100, sysdate-100

from dual;

- 사원명, 입사일(hiredate), 근무일수

select, ename, hiredate 입사일, round(sysdate-hiredate, 0) 근무일수, round((sysdate-hiredate)/365, 0) 근무년수

from emp;

- months\_between() 개월수

select ename, round(months\_between(sysdate, hiredate), 0) 근무개월수

from emp;

- add\_months() 오늘 날짜를 기준으로 몇 개월 뒤, last\_Day() 오늘 날짜를 기준으로 달의 마지막에 해당하는 날짜

- next\_day() 다음으로 오는 날짜

select add\_months(sysdate, 27), last\_day(sysdate), next\_day(sysdate, ‘금’)

from emp;

- 오후에 작업했다면 round에서 다음 날짜가 나옴

select sysdate, round(sysdate), trunc(sysdate)

from dual;

- 형변환 함수

- 오라클이 문자를 자동으로 숫자로 변환 묵시적으로 형변환

select 2 + ‘3’

from dual;

- 명시적으로 형변환 (TO\_CHAR, TO\_NUMBER, TO\_DATE)

select 2 + to\_number(’10’)

from dual;

- to\_char(날짜 -> 문자)

- yyyy/year/yy, mm/month, day(요일), dd/ddth, hh/hh24 (시간, 24시간 기준 시간), mi (분), ss (초)

select sysdate, to\_char(sysdate, ‘yyyy’), to\_char(sysdate, ‘mm’), to\_char(sysdate, ‘dd’)

from dual;

select ename, to\_char(hiredate, ‘yyyy’), to\_char(hiredate, ‘mm’), to\_char(hiredate, ‘day’)

from emp;

- to\_char(숫자 -> 문자), l = \ 표시

select to\_char(1234.56, ‘9,999’), to\_char(1234.56, ‘$9,999.9’), to\_char(1234.56, ’19,999’)

from dual;

select ename, sal, to\_char(sal, ‘9,999’)

from emp;

- to\_number()

select 2 + to\_number(‘3’)

from dual;

- to\_date()

select to\_date(‘21/10/10’), extract(year from to\_date(‘21/10/10’))

from dual;

- extract() 지정된 날짜 영역의 값 추출

select ename, hiredate, extract(year from hiredate), extract(month from hiredate), extract(day from hiredate)

from emp;

- 1981년도에 입사한 사원명 검색

select ename, hiredate

from emp

where hiredate between ‘81/01/01’ and ‘81/12/31’;

where extract(year from hiredate) = 1981;

where to\_char(hiredat, ‘yyyy’) = ‘1981’;

**4장**

- 일반 함수

- nvl(값, null일 경우 출력할 값), nvl2(값, null이 아닐 경우 출력 값, null일 경우 출력 값)

- 사원명, 봉급, 커미션, 연봉(sal\*12+comm), comm이 null인 경우 0으로 계산

select ename, sal, comm, sal\*12+nvl(comm, 0) 연봉, sal\*12+nvl2(comm, comm, 0) 연봉

from emp;

select ename, comm, nvl(to\_char(comm), ‘no\_comm’), nvl2(comm, ‘O’, ‘X’)

from emp;

- 사원번호, 관리자번호 검색(관리자번호가 null인 경우 ‘CEO’로 출력)

select empno, nvl(to\_char(mgr), ‘CEO’, nvl2(to\_char(mgr), ‘job’, ‘CEO’)

from emp;

- decode()

- 사원명, 소속부서명

select ename, deptno, decode(deptno, 10, ‘10번부서소속’, 20, ‘20번부서소속’, ‘그외부서소속’) 소속부서명

from emp;

- case 문

select ename, deptno, case when deptno=10 then ‘10번부서소속’ when deptno=20 then ‘20번부서소속’ else ‘그외부서소속’ end 소속부서명

from emp;

- 사원번호 홀수/짝수 ‘odd’/’even’ 0이면 even 아니면 odd

select ename, empno, decode(mod(empno, 2), 0, ‘even’, ‘odd’) 비고1,

case when mod(empno, 2)=0 then ‘even’

else ‘odd’

end 비고2

from emp;

- 사원명, 입사일, 입사시기(분기)

- 1~3월: 1분기, 4~6월: 2분기, 7~9월: 3분기, 10~12월” 4분기

select ename, hiredate, case when extract(month from hiredate) >= 1 and extract(month from hiredate) <= 3 then ‘1분기’

when extract(month from hiredate) between 4 and 6 then ‘2분기’

when extract(month from hiredate) between 7 and 9 then ‘3분기’

when extract(month from hiredate) between 10 and 12 then ‘4분기’

end 분기

from emp;

- 테이블 삭제 명령어

drob table reg;

- 테이블 생성 명령어

CREATE TABLE reg

(text varchar2(10);

INSERT INTO reg VALUES('ABC123');

INSERT INTO reg VALUES('XYZ123');

INSERT INTO reg VALUES('xyz123');

INSERT INTO reg VALUES('ABC 123');

INSERT INTO reg VALUES('ABC 123');

INSERT INTO reg VALUES('abc 123');

INSERT INTO reg VALUES('abc 123');

INSERT INTO reg VALUES('a1b2c3');

INSERT INTO reg VALUES('aabbcc123');

INSERT INTO reg VALUES('?/!@#$\*&');

INSERT INTO reg VALUES('\~\*();.,');

INSERT INTO reg VALUES(123123);

INSERT INTO reg VALUES('123abc');

INSERT INTO reg VALUES('123ABC');

INSERT INTO reg VALUES('abc');

INSERT INTO reg VALUES('ABC');

- 정규식 함수

- ^(캐럿): 해당 문자로 시작하는 line 출력 – ‘^pattern’

- $(달러): 해당 문자로 끝나는 line 출력 – ‘pattern$’

- . : S로 시작하여 E로 끝나는 line ( . -> 1 character) – ‘S….E’

- \* : 모든이라는 뜻, 글자수가 0일 수도 있음 – [a-z]\*’

- [ ] : 해당 문자에 해당하는 한 문자 – ‘[Pp]attern’

- [ ^ ] : 해당 문자에 해당하지 않는 한 문자 – [^a-m]attern’

- [[:alpha:]] : [a-zA-Z]

- [[:digit:]] : [0-9]

- [[:space:]] : ‘ ‘

- regexp\_like 함수

select text from reg

where regexp\_like(text, ‘[A-Z]’);

- ‘a’ 문자를 포함하고 있는 행 검색

select text from reg

where regexp\_like(text, ‘a’);

- 대/소문자를 포함하는 행 검색

select text from reg

where regexp\_like(text, ‘[a-zA-Z]’);

where regexp\_like(text, ‘[[:alpha:]]’);

- 소문자로 뒤에 공백을 포함하는 행 검색

select text from reg

where regexp\_like(text, ‘[a-z] ‘);

- 소문자로 시작하는 행 검색

select text from reg

where regexp\_like(text, ‘^[a-z]’);

- 소문자로 끝나는 행 검색

select text from reg

where regexp\_like(text, ‘[a-z]$’);

- 공백을 포함하고 있는 행 검색

select text from reg

where regexp\_like(text, ‘[[:space:]]’);

- 대문자가 연속으로 3글자 이상 있는 행

select text from reg

where regexp\_like(text, ‘[A-Z]{3}’);

- 첫번째 문자가 대문자이거나 숫자인 행

select text from reg

where regexp\_like(text, ‘^[A-Z0-9]’);

- 대문자로 시작하지 않는 행 검색

select text from reg

where regexp\_like(text, ‘^[^A-Z]’);

where not regexp\_like(text, ‘^[A-Z]’);

- 메타문자

- ? = 0 또는 1번

- \* = 0개 이상의 문자

- . = 1개의 문자

- ‘ ! ’ 특수문자를 포함하는 행 검색

select text from reg

where regexp\_like(text, ‘\!’);

- ‘ ? ‘ 특수문자를 포함하는 행 검색

select text from reg

where regexp\_like(text, ‘\?);

- regexp\_replace(): 정규식 패턴을 검색하여 대체 문자열로 치환

- 모든 숫자를 ‘ \* ‘ 기호로 변경

select text, regexp\_replace(text, ‘[0-9]’, ‘\*’), replace(text, ‘1’, ‘\*’), regexp\_replace(text, ‘[[:digit:]]’, ‘\*’)

from reg;

select regexp\_replace(‘127.0.0.1’, ‘\.’, ‘/’), regexp\_replace(‘127.0.0.1’, ‘\.’, ‘/’, 1, 1)

from dual;

- 특정 패턴의 공백 제거 (2개 이상의 공백 제거)

select regexp\_replace(‘aaa bb bb’, ‘( ){2,}’, ‘’)

from dual;

- 특정 문자열 형태를 다른 형태로 변환

select regexp\_replace(‘20211122’, ‘([[:digit:]]{4})([[:digit:]]{2})([[:digit:]]{2})’, ‘\1-\2-\3’)

- regexp\_count(): 정규식 패턴을 검색하여 특정 문자의 개수를 반환

select text, regexp\_count(text, ‘[0-9]’), regexp\_count(text, ‘A’), regexp\_count(text, ‘A’, 1, ‘I’)

from reg;

**5장**

- 그룹합수(집단함수): count, sum, max, min, avg, stddev, variance

- 사원수, 평균봉급, 최대봉급

select count(ename), count(comm), count(\*)

from emp;

select round(avg(sal)), sum(sal), max(sal), min(sal), round(stddev(sal)), round(variance(sal))

from emp;

select avg(comm), avg(nvl(comm, 0)), count(comm), count(nvl(comm, 0))

from emp;

select max(hiredate), min(hiredate)

from emp;

- group by 절

- 전체 사원들의 평균 봉급

select avg(sal)

from emp;

- 10번 부서에 소속된 사원들의 평균 봉급

select avg(sal)

from emp

where deptno=10;

- 부서번호를 기준으로 정렬

select deptno, ename

from emp

order by 1, ename;

- 부서번호별 평균 봉급

select deptno, round(avg(sal)), sum(sal), count(\*), max(sal)

from emp

group by deptno

order by 1;

- 부서번호별 평균 봉급이 2000 이상인 부서에 한하여 평균 봉급 검색

select deptno, round(avg(sal), 5), sum(sal), count(\*), max(sal)

from emp

group by deptno

having avg(sal) >= 2000

order by 1;

- 분석함수(윈도함수)

- rollup(), cube(), grouping sets()

select deptno, count(\*)

from emp

group by deptno

order by 1

- 부서별 인원수 + 전체 인원수 검색

select deptno, count(\*)

from emp

group by rollup(deptno)

group by cube(deptno)

order by 1;

- 부서별 직급별 인원수 검색

select deptno, job, count(\*)

from emp

group by deptno, job

order by 1;

- 부서별 직급별 인원수 검색 + 부서별 인원수 + 전체 인원수

select deptno, job, count(\*)

from emp

group by rollup(deptno, job)

order by 1;

- 부서별 직급별 인원수 검색 + 부서별 인원수 + 직급별 인원수 + 전체 인원수

select deptno, job, count(\*)

from emp

group by cube(deptno, job)

order by 1;

- grouping sets() 함수

- 부서별 인원수 + 직급별 인원수

select deptno, job, count(\*)

from emp

group by grouping sets(deptno, job);

- 부서별 인원수 + 직급별 인원수 + 전체 인원수

select deptno, job, count(\*)

from emp

group by grouping sets(deptno, job, ());

- 부서번호별 사원명

select deptno, ename

from emp

order by deptno;

- listagg(): 데이터를 가로로 출력, group by()절 필수

select deptno 부서번호, listagg(ename, ‘,’) within group(order by hiredate) 사원명

from emp

group by deptno;

- 집합 연산자: uniomn, union all, intersect, minus (컬럼 개수 동일, 컬럼 데이터형 일치)

select deptno, count(\*)

from emp

group by deptno

union

select null deptno, count(\*)

from emp;

**6장**

- 부서별 급여 총액

select deptno, sum(sal)

from emp

group by deptno

order by deptno;

- 부서별 급여 총액 (가로로 출력: decode() 함수)

select sum(decode(deptno, 10, sal)) “10번부서총봉급”,

sum(decode(deptno, 20, sal)) “20번부서총봉급”,

sum(decode(deptno, 30, sal)) “30번부서총봉급”

from emp;

- 부서별 급여 총액 (가로로 출력: pivot() 함수)

select \*

from (select deptno, sal from emp)

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

- 부서별/직급별 인원 수 검색 (decode())

select deptno, count(decode(job, ‘SALESMAN’, empno)) salesman,

count(decode(job, ‘CLERK’, empno)) clerk,

count(decode(job, ‘MANAGER’, empno)) manager

from emp

group by deptnp

order by deptno;

- 부서별/직급별 인원 수 검색 (pivot())

select \*

from (select deptno, job from emp)

pivot (count(\*) for job in (‘SALESMAN’, ‘CLERK’, ‘MANAGER’, ‘PRESIDENT’, ‘ANALYST’))

order by deptno;

- 그룹함수 (집단함수, 집계함수) – sum, count, max, min, avg…

- 그룹함수는 하나의 값

- 분석함수 (윈도함수) over()

- 분석함수는 행 단위로

- 그룹 내 순위 관련: rank(), dense\_rank(), row\_number()

- 그룹 내 집계 관련: sum, count, max, min, avg

- 그룹 내 순서 관련: lag(), lead()

- 그룹 내 비율 관련: ratio\_to\_report()

- 사원명, 봉급 순위

select ename, sal, hiredate

rank() over(order by sal desc) 봉급순위1,

dense\_rank() over(order by sal desc) 봉급순위2,

row\_number() over(order by sal desc) 봉급순위3

from emp;

- 동일부서내 봉급 순위

select ename, sal, hiredate, deptno,

rank() over(order by sal desc) 봉급순위,

rank() over(partition by deptno order by sal desc) 부서내봉급순위

from emp

order by deptno;

- 사원명, 봉급, 전체봉급

select ename, sal, deptno,

sum(sal) over() 전체봉급,

sum(sal) over(order by empno) 누적봉급,

sum(sal) over(partition by deptno) 부서별봉급총액,

round(avg(sal) over(partition by deptno)) 부서별봉급평균

from emp

order by deptno;

- lag(): 이전 행의 값 출력

- lead(): 다음 행의 값 출력

select ename, hiredate, lag(sal, 1, 0) over(order by hiredate),

lead(sal, 2, 0) over(order by hiredate)

from emp;

- 사원명, 봉급, 봉급총액, 비율(봉급/전체봉급합계)

select ename, sal, sum(sal) over(),

round(sal/sum(sal) over()\*100, 3) || ‘%’ 비율,

round(ratio\_to\_report(sal) over(), 3) 비율

from emp;

- 사원명, 봉급, 봉급총액, 비율(봉급/동일부서내봉급총액)

select ename, sal, deptno,

sum(sal) over() 전체봉급총액,

sum(sal) over(partition by deptno) 부서별봉급총액,

round(sal/sum(sal) over(partition by deptno)\*100, 3) || ‘%’ 부서별비율,

round(ratio\_to\_report(sal) over(partition by deptno), 3) 부서별비율

from emp

order by deptno;

- 부서별 직원급여총액 비율(group by절 사용)

select deptno, sum(sal) 부서별봉급총액,

sum(sum(sal)) over() 봉급총액,

round(sum(sal) / sum(sum(sal)) over(), 3) “2”,

round(ratio\_to\_report(sum(sal)) over(), 3) “3”

from emp

group by deptno;

**7장**

- ‘smith’ 사원의 봉급 순위 검색

select ename, sal, rank() over(order by sal desc)

from emp;

- sal 안에서 2000이라는 값의 순위

select ank(‘2000’) within group (order by sal)

from emp;

- 이름순으로 정렬 후 smith의 위치

select rack(‘SMITH’) within group (order by ename)

from emp;

select ename

from emp

ordet by ename;

- ntile() 특정 기준으로 분할

select ename, sal, ntile(4) over(order by sal)

from emp;

- keep() 함수

- min/max() keep (dense\_rank first/last order by)

- 부서별 최대급여, 최소급여

select deptno, max(sal), min(sal)

from emp

group by deptno;

- 부서번호별 최대급여, 최대급여수급자

select deptno, max(sal) 최대급여,

min(ename) keep (dense\_rank first order by sal desc) 최대급여수급자

from emp

group by deptno;

- 최대급여, 최대급여수급자

select max(sal) 최대급여, max(ename) keep (dense\_rank first order by sal desc) 최대급여수급자

from emp;

- 부서번호별 최소급여, 최소급여수급자

select deptno, min(sal) 최소급여,

max(ename) keep (dense\_rank last order by sal desc) 최소급여수급자

from emp

group by deptno;

- keep() + over(): 분석함수 모든 사원들의 값을 보려면 분석함수를 써야 함

- 모든 사원들의 사원명, 봉급, 최대봉급, 최대봉급수급자, 소속부서내최대봉급, 소속부서내최대봉급수급자

- select ename, sal, max(sal) (집계함수라 오류가 남, 하나의 값만 리턴)

select eame 사원명, sal 봉급, max(sal) over() 최대봉급,

max(ename) keep(dense\_rank first order by sal desc) over() 최대봉급수급자,

max(sal) over(parition by deptno) 소속부서내최대봉급,

max(ename) keep(dense\_rank first order by sal desc) over(partition by deptno) 부서내최대봉급수급자

from emp;

- first\_value(), last\_value() + over()

select ename, sal, deptno,

max(sal) over(),

first\_value(ename) over(order by sal desc) 최대봉급수급자,

last\_value(job) over(order by sal rows between current row and unbounded following) 최대봉급수급자의직급,

max(sal) over(partition by deptno) 부서내최대봉급,

first\_value(ename) over(partition by deptno order by sal desc) 부서내최대봉급수급자

from emp

order by deptno;

- 조인

- 사원명, 소속부서명 검색

- emp: 사원명, 소속부서번호

- dept: 부서번호, 부서명

- cross 조인

select ename, e.deptno, d.deptno, dname

from emp e, dept d;

select ename, e.deptno, d.deptno, dname

from emp e cross join dept d;

- 사원명, 소속부서명 검색

- 등가(equit) 조인

select ename, dname

from emp, dept

where emp.deptno = dept.deptno;

- ANSI 표준 표기법

select ename, dname

from emp e inner join dept d

on e.deptno = d.deptno;

--

select ename, dname

from emp e natural join dept d;

--

select ename, dname

from emp e join dept d

using (deptno);

- self 조인

- 사원명, 관리자명 검색

select e.name 사원명, m.ename 관리자명

from emp e, emp m

where e.mgr = m.empno;

select e.ename 사원명, m.ename 관리자명

from emp e join emp m

on e.mgr = m.empno;

- 관리자명이 ‘KING’인 사원들의 이름, 직급 검색

select e.ename 사원명, m.ename 관리자명, e.job 직급

from emp e, emp m

where e.mgr = m.empno and m.ename=’KING’;

select e.ename 사원명, m.ename 관리자명

from emp e join emp m on e.mgr = m.empno

where m.ename=’KING’;

- 외부조인

- 사원명, 소속부서명 검색, 소속 사원이 없는 부서명도 출력되도록

select ename, dname

from emp, dept

where emp.deptno(+) = dept.deptno;

- outer join: left / right / full

select ename, dname

from emp e right outer join dept d

on e.deptno = d.deptno;

- 관리자명이 ‘KING’인 사람들의 이름, 직급 (관리자가 없는 사원명도 출력되도록)

select e.ename 사원명, m.ename 관리자명, e.job

from emp e, emp m

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

where e.mgr = m.empno(+) and m.ename = ‘KING’;

select e.ename 사원명, m.ename 관리자명, e.job

from emp e left outer join emp m on e.mgr = m.empno;