

Sub Query

강사 : 강병준

서브 쿼리의 기본 개념

SCOTT의 부서명을 알아내기 위한 서브 쿼리문

메인 쿼리

서브 쿼리

```
SELECT DNAME  
FROM DEPT  
WHERE DEPTNO = ( SELECT DEPTNO  
                  FROM EMP  
                  WHERE ENAME='SCOTT' )
```

서브 쿼리(Sub Query) 소개

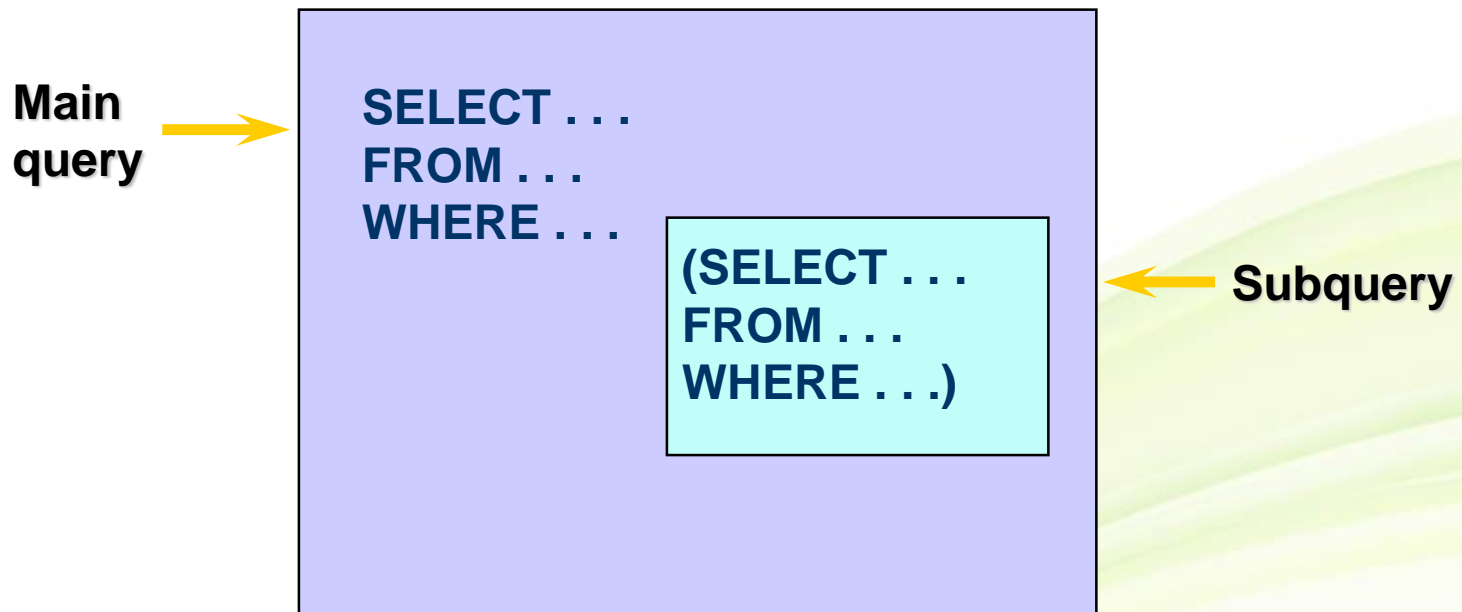
- SUBQUERY는 다른 하나의 SQL 문장의 절에 포함된 SELECT 문장
- SELECT, UPDATE, DELETE, INSERT와 같은 DML문과 CREATE TABLE 또는 VIEW에서 이용
- 알려지지 않은 조건에 근거한 값들을 검색하는 SELECT 문장을 작성하는데 유용
- SUBQUERY는 MAIN QUERY가 실행되기 이전에 한번 실행.

```
SELECT    select_list  
FROM      table  
WHERE     expr operator
```

```
(SELECT    select_list  
FROM      table);
```

서브 쿼리(Sub Query) 소개

- A subquery is a **SELECT** statement embedded in a clause of another SQL statement.



부질의 유형

- 단일 행(row)를 반환하는 부질의
 - 외부(outer) table로 1 또는 0 개의 로우(row)를 return 하는 경우.

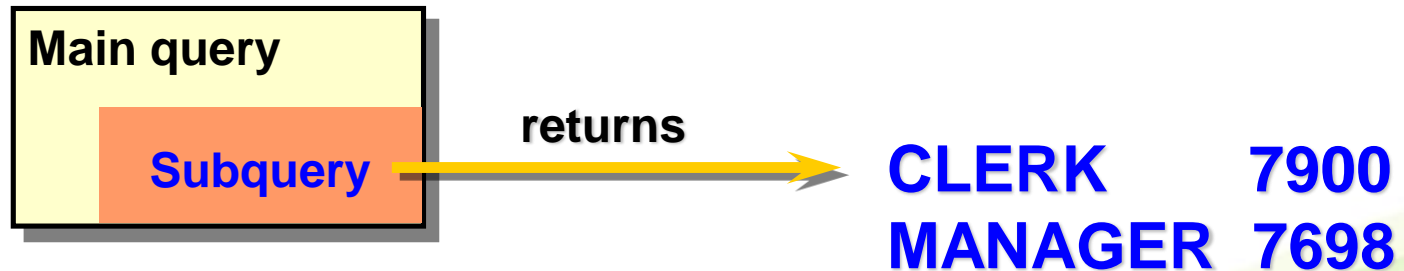


- 다중 행(row)를 반환하는 부질의
 - 외부(outer) table로 1 개 이상의 로우(row)를 return 하는 경우.



부질의의 유형

- 다중 열(column)을 반환하는 부질의
 - 외부(outer) table로 1 개 이상의 열(column)을 return 하는 경우.



- (상관관계)연관된 부질의
 - 내부(inner) 질의문에서 외부(outer) 질의문의 1 개 이상의 열(column)을 참조하는 경우.
- 내포된 부질의
 - 부질의 안에 다시 부질의가 있는 경우.

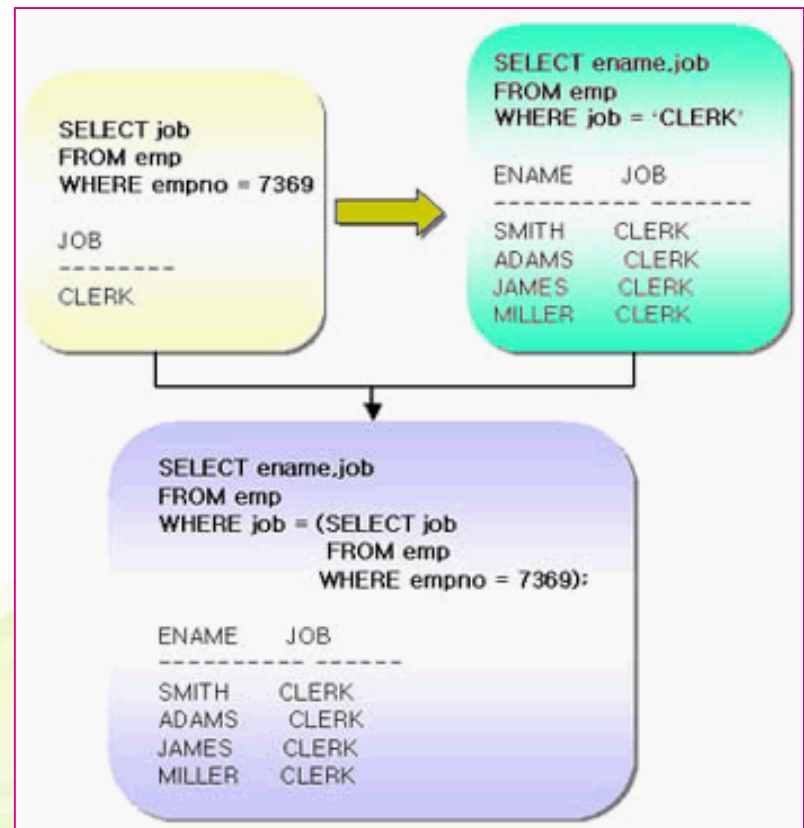
서브 쿼리 설정 권장 사항

- SUBQUERY는 괄호로 묶어서 사용
- SUBQUERY는 연산자의 오른쪽 적용
- SUBQUERY는 FROM, WHERE, HAVING에서 사용
- 두 종류의 비교 연산자들이 SUBQUERY에 사용
 - 단일 행 연산자
=, >, >=, <, <=, <>, !=
 - 복수 행 연산자
IN, NOT IN, ANY, ALL, EXISTS

단일 행 (Single-Row) 서브 쿼리

- 오직 한 개의 행(값)을 반환 (Return only one row)
- 단일 행 연산자(=, >, >=, <, <=, <>, !=) 만 사용

```
SELECT ename, job
FROM emp
WHERE job=(SELECT job
            FROM emp
            WHERE empno = 7369);
```



단일 행 (Single-Row) 서브 쿼리

■ Where절에 내포된 SubQuery

```
SQL> SELECT      ename, job
  2  FROM          emp
  3  WHERE         job =
  4                (SELECT      job
  5                  FROM        emp
  6                  WHERE        empno = 7369)
  7  AND          sal >
  8                (SELECT      sal
  9                  FROM        emp
 10                  WHERE        empno = 7876) ;
```

CLERK

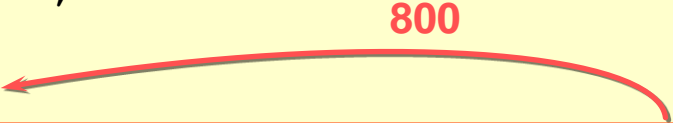
1100

ENAME	JOB
-----	-----
MILLER	CLERK

단일 행 (Single-Row) 서브 쿼리

■ Group함수가 내포된 SubQuery

```
SQL> SELECT  ename, job, sal
2  FROM      emp
3  WHERE     sal =
4             (SELECT  MIN(sal)
5             FROM      emp) ;
```



A red arrow points from the value **800** to the **MIN(sal)** subquery in the SQL statement.

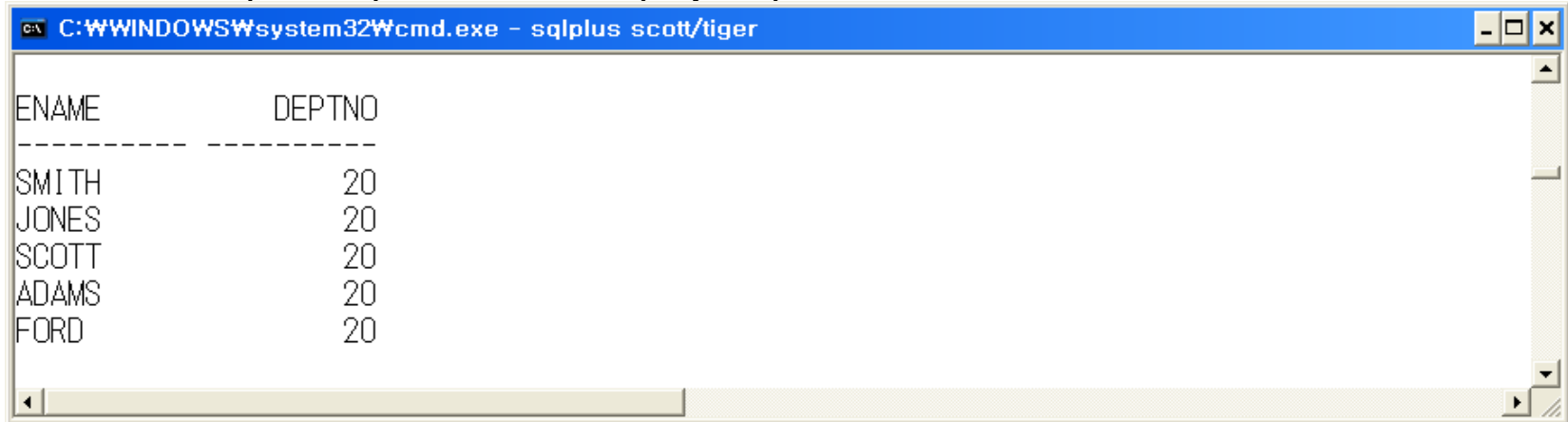
ENAME	JOB	SAL
-----	-----	-----
SMITH	CLERK	800

연습문제

1. 사원 테이블에서 BLAKE 보다 급여가 많은 사원들의 사번, 이름, 급여를 검색하시오.
2. 사원 테이블에서 MILLER보다 늦게 입사한 사원의 사번, 이름, 입사일을 검색하시오.
3. 사원 테이블에서 사원 전체 평균 급여보다 급여가 많은 사원들의 사번, 이름, 급여를 검색하시오.
4. 사원 테이블에서 CLARK와 같은 부서이며, 사번이 7698인 직원의 급여보다 많은 급여를 받는 사원들의 사번, 이름, 급여를 검색하시오.
5. 사원 테이블에서 부서별 최대 급여를 받는 사원들의 사번, 이름, 부서코드, 급여를 검색하시오.

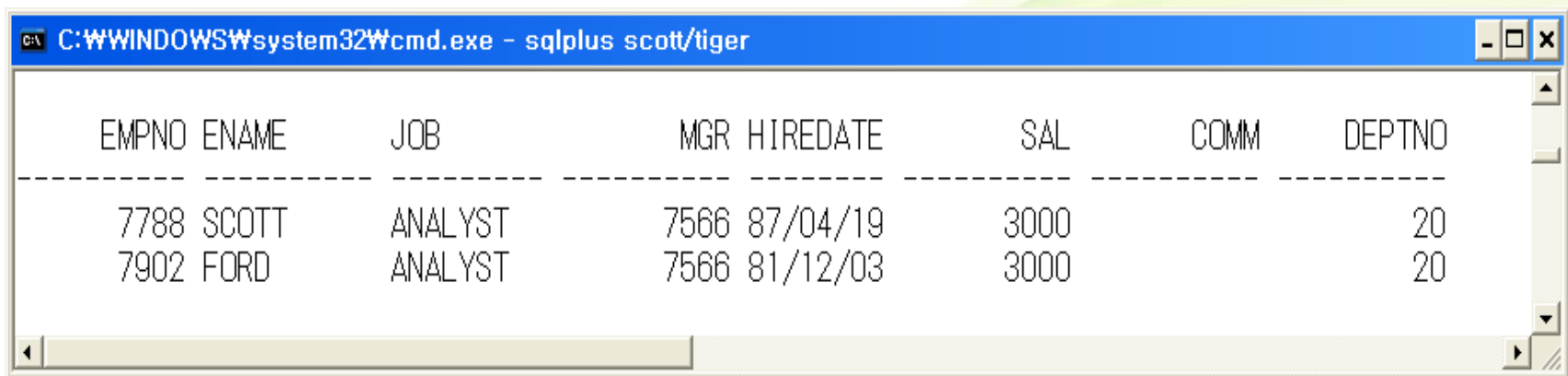
연습문제

1. EMP 테이블에서 ename이 SCOTT인 데이터와 같은 부서(deptno)에서 근무하는 사원의 이름(ename)과 부서 번호(deptno)를 출력하는 SQL 문을 작성해 보시오.



ENAME	DEPTNO
SMITH	20
JONES	20
SCOTT	20
ADAMS	20
FORD	20

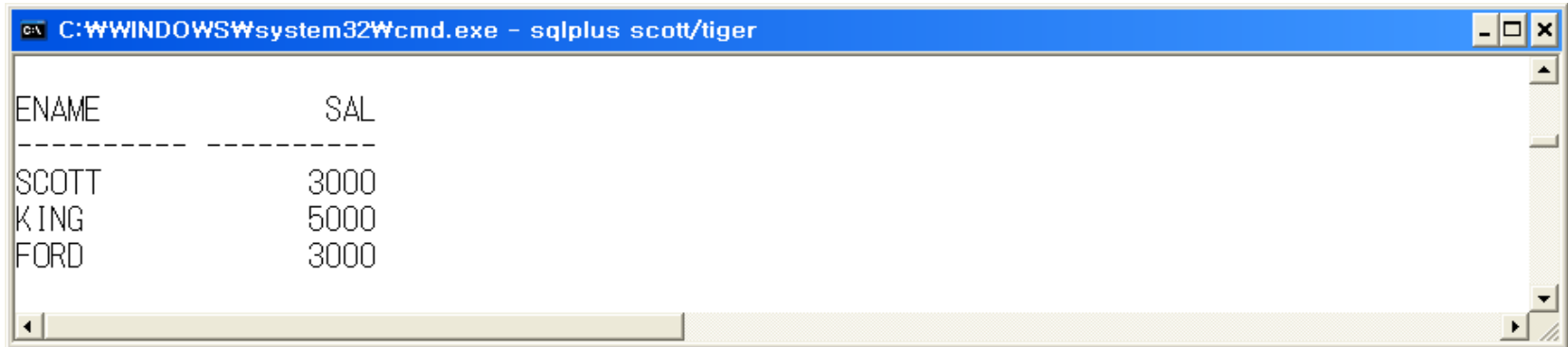
1. EMP 테이블에서 ename이 SCOTT와 데이터와 동일한 직급(JOB)을 가진 사원의 모든 컬럼을 출력하는 SQL 문을 작성해 보시오.



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7902	FORD	ANALYST	7566	81/12/03	3000		20

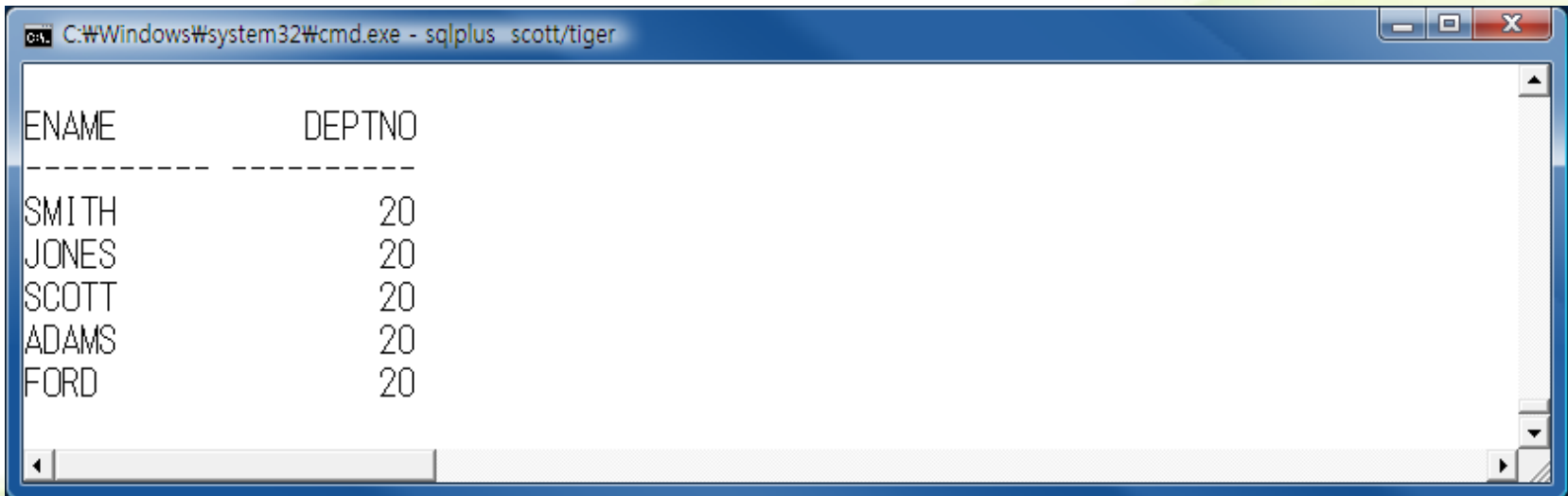
연습문제

3. EMP 테이블에서 ename이 SCOTT인 데이터의 급여(SAL)와 동일하거나 더 많이 받는 사원명(ename)과 급여(sal)를 출력하시오.



ENAME	SAL
SCOTT	3000
KING	5000
FORD	3000

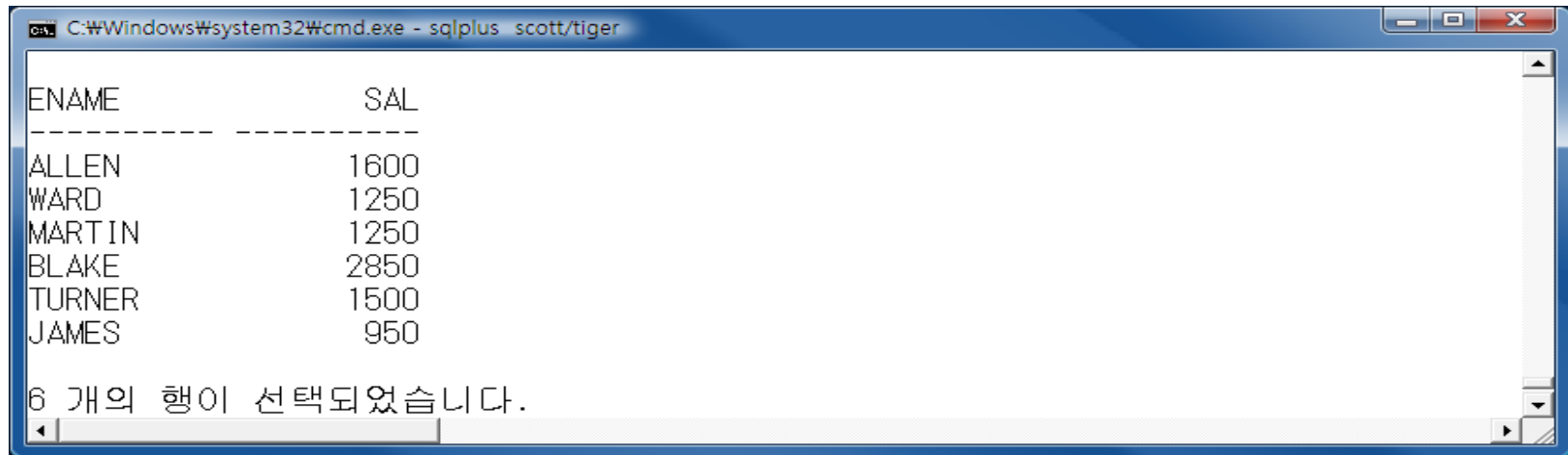
4. EMP 테이블에서 DEPT 테이블의 LOC가 DALLAS인 사원의 이름(ename), 부서 번호(deptno)를 출력하시오.



ENAME	DEPTNO
SMITH	20
JONES	20
SCOTT	20
ADAMS	20
FORD	20

연습문제

5. EMP 테이블에서 DEPT 테이블의 dname이 SALES(영업부)인 부서에서 근무하는 사원의 이름(ename)과 급여(sal)를 출력하시오.

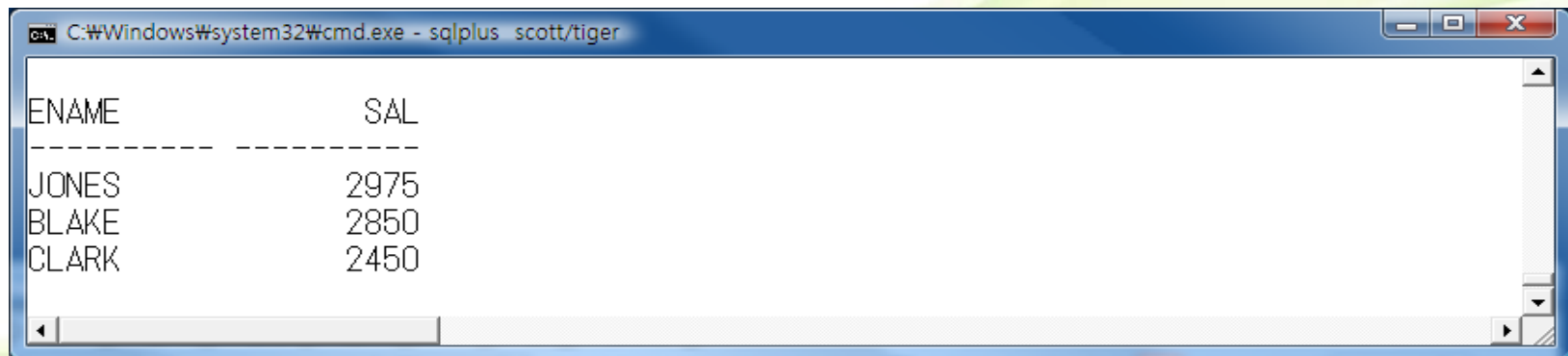


C:\Windows\system32\cmd.exe - sqlplus scott/tiger

ENAME	SAL
-----	-----
ALLEN	1600
WARD	1250
MARTIN	1250
BLAKE	2850
TURNER	1500
JAMES	950

6 개의 행이 선택되었습니다.

6. EMP 테이블에서 직속상관(mgr)의 이름이 KING인 사원의 이름(ename)과 급여(sal)를 출력하시오.



C:\Windows\system32\cmd.exe - sqlplus scott/tiger

ENAME	SAL
-----	-----
JONES	2975
BLAKE	2850
CLARK	2450

Having 절과 서브쿼리

- The Oracle Server executes subqueries first.
- The Oracle Server returns results into the HAVING clause of the main query.

```
SQL> SELECT      deptno, MIN(sal)
  2  FROM          emp
  3  GROUP BY      deptno
  4  HAVING        MIN(sal) >
  5                (SELECT      MIN(sal)
  6                FROM          emp
  7                WHERE         deptno = 20) ;
```

The diagram illustrates the execution of the SQL query. A red arrow points from the subquery result '800' to the 'HAVING' clause condition, indicating that the subquery is executed first and its result is used to filter the main query results.

단일 행(row)을 반환하는 부질의

- FROM절에 내포된 부질의 (inline view)

```
SQL> select d.deptno, d.dname, d.loc,  
2         number_of_emp.emp_count  
3       from dept d,  
4         (select deptno, COUNT(deptno) emp_count  
5           from emp  
6           group by deptno ) number_of_emp  
7       where d.deptno = number_of_emp.deptno
```

DEPTNO	DNAME	LOC	EMP_COUNT
10	ACCOUNTING	NEW YORK	3
20	RESEARCH	DALLAS	5
30	SALES	CHICAGO	6

단일 행(row)을 반환하는 부질의

- SubQuery에서는 ORDER BY절을 사용할 수 없다.

```
SQL> SELECT empno, ename, sal
2      FROM emp
3      WHERE sal > (SELECT AVG(sal)
4                  FROM emp )
5      ORDER BY sal DESC, empno DESC ;
```

EMPNO	ENAME	SAL
7839	KING	5000
7902	FORD	3000
7788	SCOTT	3000
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450

잘못 사용된 서브쿼리

```
SQL> SELECT empno, ename  
2 FROM emp  
3 WHERE sal =  
4  
5  
6
```

*Single-row operator with
multiple-row subquery*

```
(SELECT MIN(sal)  
FROM emp  
GROUP BY deptno);
```

ERROR:

ORA-01427: single-row subquery returns more than
one row

no rows selected

다중 행 (Multiple-Row) 서브 쿼리

- 하나 이상의 행을 RETURN하는 SUBQUERY를 다중 행 SUBQUERY라고 한다
- 복수 행 연산자 사용
 - **IN, NOT IN**
 - **ANY**
 - **ALL**
 - **EXISTS**

다중 행 (Multiple-Row) 서브 쿼리

■ IN 연산자를 사용한 다중 행 서브 쿼리

조건>

부서별로 가장 급여를 많이 받는 사원의 정보를 출력하시오.

EMPNO	ENAME	SAL	DEPTNO
7698	BLAKE	2850	30
7788	SCOTT	3000	20
7902	FORD	3000	20
7839	KING	5000	10

다중 행(row)을 반환하는 부질의

■ IN

```
SQL> SELECT deptno, ename, sal
2      FROM emp
3      WHERE deptno IN ( SELECT deptno
4                        FROM dept
5                        WHERE dname in
                          ('ACCOUNTING', 'SALES'));
```

DEPTNO	ENAME	SAL
-----	-----	-----
30	ALLEN	1600
30	WARD	1250
30	MARTIN	1250
30	BLAKE	2850
10	CLARK	2450
10	KING	5000
30	TURNER	1500
30	JAMES	950
10	MILLER	1300

다중 행(row)을 반환하는 부질의

■ ANY

- ANY 조건은 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상만 일치하면 참입니다.
- >ANY는 찾아진 값에 대해서 하나라도 크면 참이 되는 셈이 됩니다. 그러므로 찾아진 값 중에서 가장 작은 값 즉, 최소값 보다 크면 참이 됩니다.

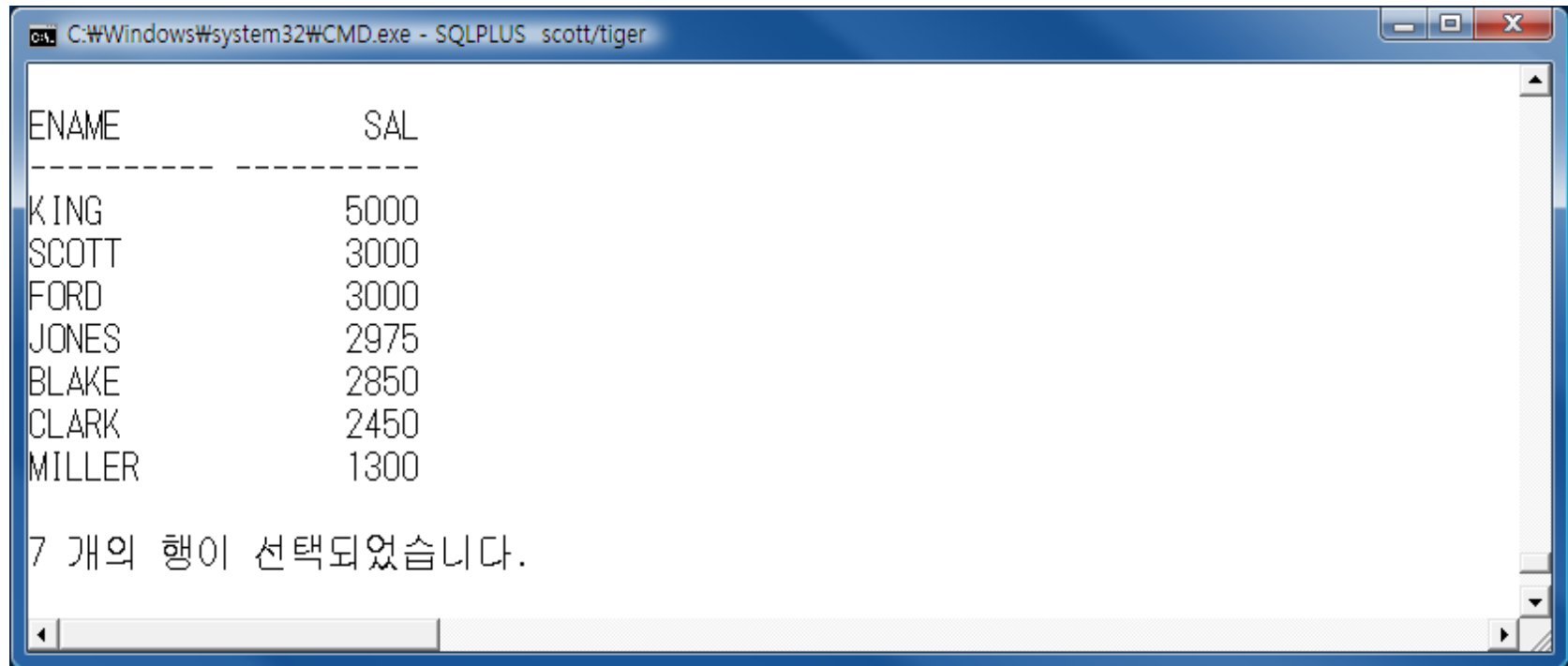
```
SQL> SELECT employee_id, first_name
2      FROM employees
3      WHERE salary > ANY ( SELECT low_salary
4                           FROM salary_grades ) ;
```

```
EMPLOYEE_ID FIRST_NAME
-----
```

```
1 James
2 Ron
3 Fred
4 Susan
```

연습문제

- EMP 테이블에서 JOB이 SALESMAN 인 사원의 최소 SAL 보다 SAL이 많은 사원들의 이름과 급여와 직급(담당 업무)를 출력하되 영업 사원(SALESMAN)은 출력하지 않습니다.



```
C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger

ENAME          SAL
-----
KING            5000
SCOTT           3000
FORD            3000
JONES           2975
BLAKE           2850
CLARK           2450
MILLER          1300

7 개의 행이 선택되었습니다.
```

다중 행(row)을 반환하는 부질의

■ ALL

- ALL 조건은 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참입니다.
- 찾아진 값에 대해서 AND 연산을 해서 모두 참이면 참이 되는 셈이 됩니다. > ALL 은 “모든 비교값 보다 크냐”고 묻는 것이 되므로 최대값보다 더 크면 참이 됩니다.

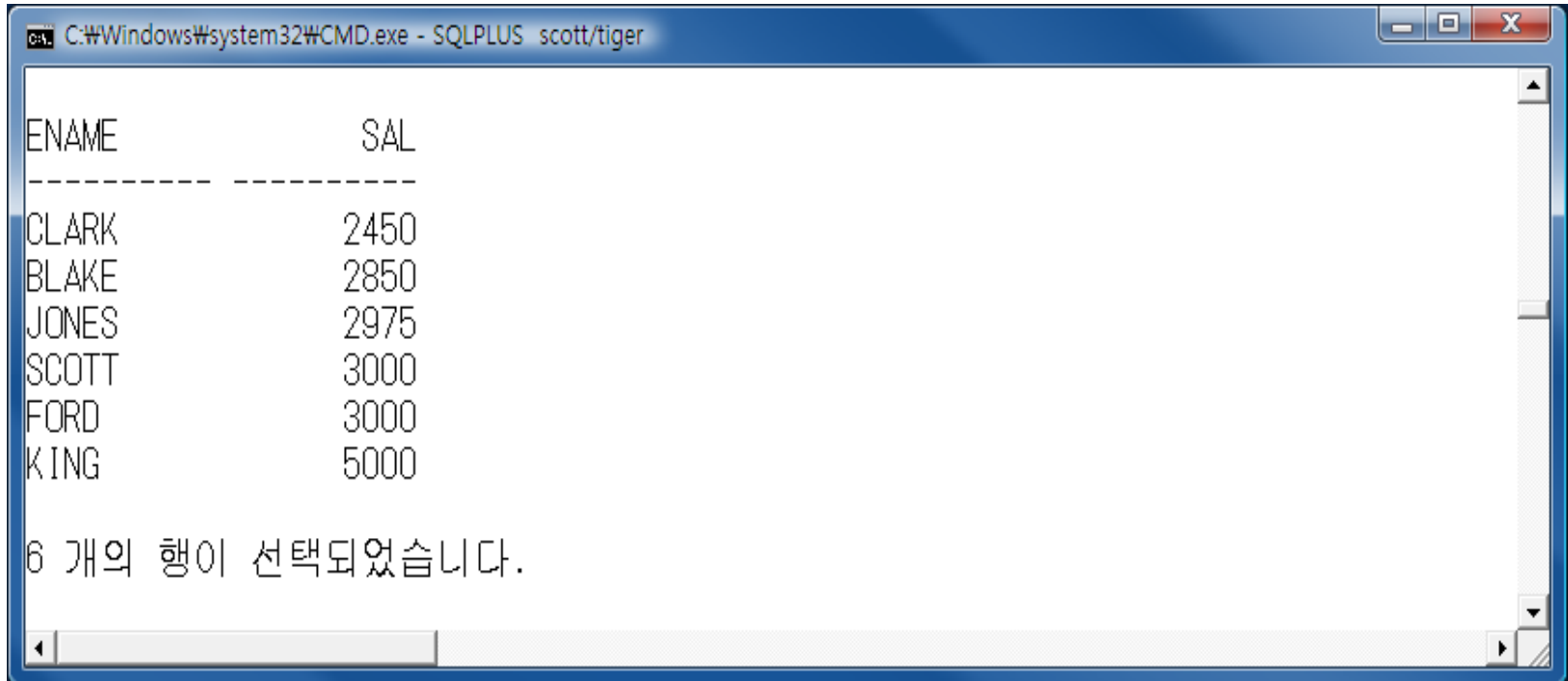
```
SQL> SELECT employee_id, first_name
2      FROM employees
3      WHERE salary < ALL ( SELECT high_salary
4                           FROM salary_grades ) ;
```

```
EMPLOYEE_ID FIRST_NAME
-----
```

```
3 Fred
```


연습문제

- EMP 테이블에서 JOB이 SALESMAN 인 모든 사원들보다 급여를 많이 받는 사원들의 이름과 급여와 직급(담당 업무)를 출력.



```
C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger

ENAME          SAL
-----
CLARK          2450
BLAKE          2850
JONES          2975
SCOTT          3000
FORD           3000
KING           5000

6 개의 행이 선택되었습니다.
```

다중 열 (Multiple-Column) 서브 쿼리

- 다중 열 서브쿼리란 서브쿼리의 결과값이 두개 이상의 컬럼을 반환하는 서브쿼리
 - **Pairwise(쌍비교) 서브 쿼리**
서브쿼리가 한번 실행되면서 모든 조건을 검색해서 주 쿼리로 넘김
 - **NonPairwise(비쌍비교) 서브 쿼리**
서브쿼리가 여러 조건별로 사용 되어서 결과값을 주 쿼리로 넘김

다중 열 (Multiple-Column) 서브 쿼리

■ Pairwise(쌍비교) 서브 쿼리

```
SELECT empno, sal, deptno
FROM emp
WHERE (sal, deptno) IN ( SELECT sal, deptno
                        FROM emp
                        WHERE deptno = 30
                        AND comm is NOT NULL );
```

EMPNO	SAL	DEPTNO
7521	1250	30
7654	1250	30
7844	1500	30
7499	1600	30

다중 열 (Multiple-Column) 서브 쿼리

■ NonPairwise(비쌍비교) 서브 쿼리

```
SELECT empno, sal, deptno
FROM emp
WHERE sal IN ( SELECT sal
                FROM emp
                WHERE deptno = 30
                  AND comm is NOT NULL )
AND deptno IN ( SELECT deptno
                 FROM emp
                 WHERE deptno = 30
                   AND comm is NOT NULL );
```

EMPNO	SAL	DEPTNO
7521	1250	30
7654	1250	30
7844	1500	30
7499	1600	30

다중 열 (Multiple-Column) 서브 쿼리

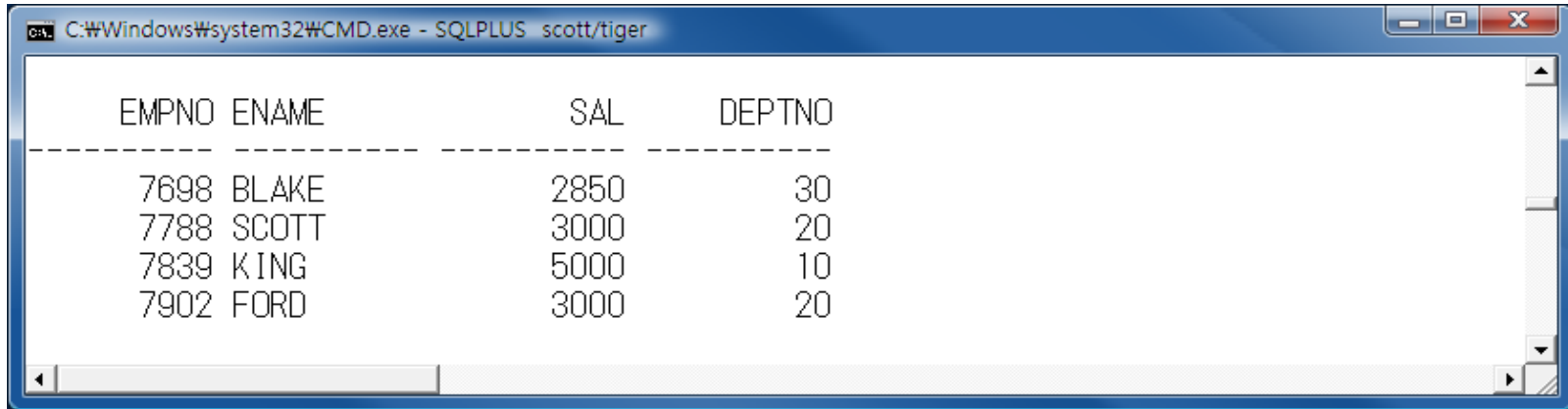
복수 컬럼 서브쿼리는 비교조건의 형태의 따라 Pairwise 비교와 Nonpairwise 비교로 구분된다. Pairwise 비교조건은 다음과 같이 사원 테이블에서 사번이 7369 또는 7499번인 직원과 직급 및 부서코드가 동일한 사원을 검색하는 경우이다.

```
SQL> SELECT EMPNO, JOB, DEPTNO
2  FROM EMP
3  WHERE (JOB, DEPTNO) IN
4          (SELECT JOB, DEPTNO
5           FROM EMP
6           WHERE EMPNO IN (7369, 7499))
7  AND EMPNO NOT IN (7369, 7499);
```

EMPNO	JOB	DEPTNO
7876	CLERK	20
7521	SALESMAN	30
7654	SALESMAN	30
7844	SALESMAN	30

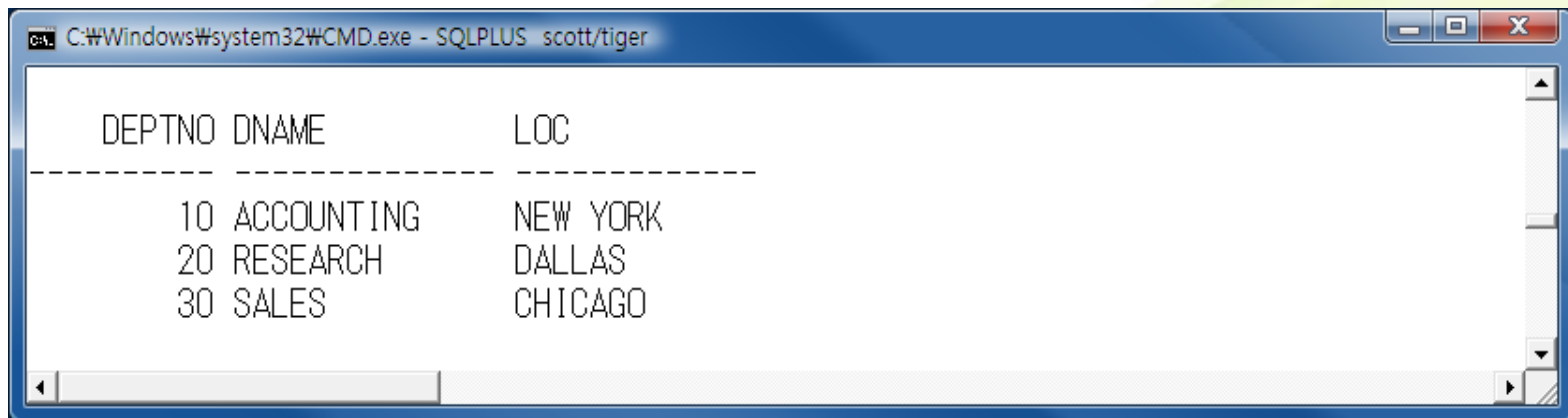
연습문제

7. emp 테이블에서 부서별로 가장 급여를 많이 받는 사원의 사원 번호(empno), 사원 이름(ename), 급여(sal), 부서번호(deptno)를 출력하시오.(IN 연산자 이용)



EMPNO	ENAME	SAL	DEPTNO
7698	BLAKE	2850	30
7788	SCOTT	3000	20
7839	KING	5000	10
7902	FORD	3000	20

8. emp 테이블에서 직급(JOB)이 MANAGER인 사람이 속한 부서의 부서 번호(deptno)와 부서명(dname)과 지역(loc)을 출력하시오.

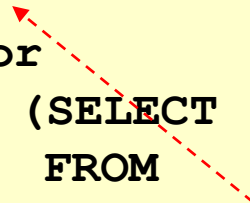


DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO

상관관계 서브 쿼리

- 상관관계 서브쿼리란 바깥쪽 쿼리의 컬럼 중의 하나가 안쪽 서브쿼리의 조건에 이용되는 처리 방식
- 이는 주 쿼리에서 서브 쿼리를 참조하고 이 값을 다시 주 쿼리로 반환한다는 것이다

```
SELECT outer1, outer2, ...  
FROM   table1 alias1  
WHERE  outer1 operator  
        (SELECT inner1  
           FROM   table2 alias2  
           WHERE  alias1.expr1 =  
                  alias2.expr2) ;
```




상관관계 서브 쿼리

- Find all employees who earn more than the average salary in their department.

```
SQL> SELECT empno, sal, deptno
      2 FROM emp outer
      3 WHERE sal > (SELECT AVG(sal)
      4                  FROM emp inner
      5                  WHERE outer.deptno = inner.deptno);
```

Each time the outer query is processed the inner query is evaluated.



EMPNO	SAL	DEPTNO
7839	5000	10
7698	2850	30
7566	2975	20
...		

6 rows selected.

다중 행(row)을 반환하는 부질의

■ Exists

- SQL문에서 Exists를 사용하는 것은 어떤 조건을 만족하는 집합의 존재 여부를 확인할 경우에 사용.
- Exists는 주어진 조건을 만족하는 첫 번째 열을 만나면 트랜잭션을 멈추게 됨. 즉, 조건을 만족하면 처리를 멈추고 다음의 처리를 수행.
- 그러나 Exists를 사용하지 않은 경우는 일반적으로 만족하는 조건을 모두 Search한 이후에 다음 처리를 수행함으로써 Access하는 횟수가 많아지게 됨.

```
SQL> SELECT empno, ename, sal
2   FROM emp e
3   WHERE EXISTS (SELECT empno
4                  FROM emp
5                  where e.empno=mgr) ;
```

◆ 연습문제 ◆

1. EMP 테이블에서 Blake와 같은 부서에 있는 모든 사원의 이름과 입사일자를 출력하는 SELECT문을 작성하시오.
2. EMP 테이블에서 평균 급여 이상을 받는 모든 종업원에 대해서 종업원 번호와 이름을 출력하는 SELECT문을 작성하시오. 단 급여가 많은 순으로 출력하여라.
3. EMP 테이블에서 이름에 “T”가 있는 사원이 근무하는 부서에서 근무하는 모든 종업원에 대해 사원 번호, 이름, 급여를 출력하는 SELECT문을 작성하시오. 단 사원번호 순으로 출력하여라.
4. EMP 테이블에서 부서 위치가 Dallas인 모든 종업원에 대해 이름, 업무, 급여를 출력하는 SELECT문을 작성하시오.
5. EMP 테이블에서 King에게 보고하는 모든 사원의 이름과 급여를 출력하는 SELECT문을 작성하시오.
6. EMP 테이블에서 SALES부서 사원의 이름, 업무를 출력하는 SELECT문을 작성하시오.
7. EMP 테이블에서 월급이 부서 30의 최저 월급보다 높은 사원을 출력하는 SELECT문을 작성하시오.
8. EMP 테이블에서 부서 10에서 부서 30의 사원과 같은 업무를 맡고 있는 사원의 이름과 업무를 출력하는 SELECT문을 작성하시오.
9. EMP 테이블에서 FORD와 업무도 월급도 같은 사원의 모든 정보를 출력하는 SELECT문을 작성하시오.

10. EMP 테이블에서 업무가 JONES와 같거나 월급이 FORD이상인 사원의 정보를 이름,업무,부서번호,급여를 출력하는 SELECT문을 작성 단 업무별, 월급이 많은 순으로 출력하여라.
11. EMP 테이블에서 SCOTT 또는 WARD와 월급이 같은 사원의 정보를 이름,업무,급여를 출력하는 SELECT문을 작성하시오.
12. EMP 테이블에서 CHICAGO에서 근무하는 사원과 같은 업무를 하는 사원의 이름,업무를 출력하는 SELECT문을 작성하시오.
13. EMP 테이블에서 부서별로 월급이 평균 월급보다 높은 사원을 부서번호,이름,급여를 출력하는 SELECT문을 작성하시오.
14. EMP 테이블에서 업무별로 월급이 평균 월급보다 낮은 사원을 부서번호,이름,급여를 출력하는 SELECT문을 작성하시오.
15. EMP 테이블에서 적어도 한명 이상으로부터 보고를 받을 수 있는 사원을 업무,이름,사원번호,부서번호를 출력하는 SELECT문을 작성하시오.
16. EMP 테이블에서 말단 사원의 사원번호,이름,업무,부서번호를 출력하는 SELECT문을 작성하시오.

대표적인 시스템 변수

시스템 변수	설명
LINESIZE	검색 결과의 출력 너비를 지정
PAGESIZE	검색 결과의 한 페이지에 출력될 라인 수를 지정
HEADING	검색 결과에서 컬럼명의 표시 여부
FEEDBACK	SQL 문장 실행 후, 결과 메시지 표시 여부
SQLPROMPT	프롬프트 지정

포맷 명령

SQL*Plus 및 iSQL*Plus의 포맷 명령은 SQL 문장에 의한 검색 결과를 보고서로 작성할 수 있도록 도와주는 기능이다. 포맷 명령어는 다음과 같은 것들이 있다.

표 8-2. 포맷 명령

포맷 명령	설명
COL[UMN] [<i>column option</i>]	컬럼 포맷을 설정
TTI[TLE] [<i>text</i> OFF ON]	검색 결과의 각 페이지 최상단에 표시될 제목
BTI[TLE] [<i>text</i> OFF ON]	검색 결과의 각 페이지 최하단에 표시될 제목
BRE[AK] [ON <i>report_element</i>]	중복된 값의 출력을 억제

■ COLUMN

COLUMN 명령을 사용하는 방법은 다음과 같다.

```
COL[UMN] [{column|alias} [option]]
```

위에서 *option*으로 지정할 수 있는 항목은 다음과 같다.

표 8-3. COLUMN 명령 *option*

Option	설명
CLE[AR]	컬럼 포맷을 삭제
HEA[DING] <i>text</i>	컬럼명을 <i>text</i> 로 지정(컬럼명의 는 두줄에 걸쳐서 출력)
FOR[MAT] <i>format</i>	컬럼 데이터의 출력 형식
NOPRI[NT]	컬럼을 출력하지 않음
NUL[L] <i>text</i>	컬럼값이 NULL이면 <i>text</i> 출력
PRI[NT]	컬럼을 출력

컬럼 *format*

항목	설명	예	결과
9	숫자 1자리	999999	1234
0	숫자 1자리. 숫자가 지정되지 않으면 0으로 출력	099999	001234
\$	달러 표시	\$9999	\$1234
L	지역 화폐단위	L9999	W1234
.	소수점	9999.99	1234.00
,	천단위 구분자	9,999	1,234

COLUMN 명령을 이용하여 검색 결과를 보기 좋게 수정해보자. 포맷 명령을 사용하기 전의 출력결과가 다음과 같다고 가정한다.

```
SQL> SELECT EMPNO, ENAME, JOB, MGR, SAL FROM EMP;
```

EMPNO	ENAME	JOB	MGR	SAL
7369	SMITH	CLERK	7902	800
7499	ALLEN	SALESMAN	7698	1600
7521	WARD	SALESMAN	7698	1250
7566	JONES	MANAGER	7839	2975
7654	MARTIN	SALESMAN	7698	1250

```

SQL> COLUMN EMPNO HEADING '사번'
SQL> COLUMN ENAME HEADING '사원명'
SQL> COLUMN JOB HEADING '업무'
SQL> COLUMN MGR HEADING '관리자|사번'
SQL> COLUMN SAL HEADING '급여'
SQL> COLUMN EMPNO FORMAT 99999
SQL> COLUMN MGR FORMAT 99999
SQL> COLUMN ENAME FORMAT A7
SQL> COLUMN ENAME JUSTIFY RIGHT
SQL> COLUMN SAL FORMAT 999,999
SQL> COLUMN MGR NULL '사장'
SQL> SELECT EMPNO, ENAME, JOB, MGR, SAL FROM EMP;

```

사번	사원명	업무	관리자 사번	급여
7369	SMITH	CLERK	7902	800
7499	ALLEN	SALESMAN	7698	1,600
7521	WARD	SALESMAN	7698	1,250
7566	JONES	MANAGER	7839	2,975
7902	FORD	ANALYST	7566	3,000
...				
7934	MILLER	CLERK	7782	1,300

■ BREAK

BREAK 명령은 중복값의 출력을 억제하는 명령이다.

```
BREAK ON column [alias|row]
```

BREAK 명령은 대상이 되는 컬럼이 ORDER BY에 의해 정렬된 후 검색해야만 의미가 있다.

```
SQL> BREAK ON JOB
SQL> SELECT EMPNO, ENAME, JOB
2 FROM EMP
3 ORDER BY JOB;
```

EMPNO	ENAME	JOB
7788	SCOTT	ANALYST
7902	FORD	
7369	SMITH	
7876	ADAMS	CLERK
7934	MILLER	
7900	JAMES	
7566	JONES	MANAGER
7782	CLARK	
7698	BLAKE	
7839	KING	PRESIDENT
7499	ALLEN	
7654	MARTIN	
7844	TURNER	SALESMAN
7521	WARD	

■ TTITLE, BTITLE

TTITLE, BTITLE은 각각 페이지의 최상단, 최하단에 출력할 제목을 지정하는 것이다.

```
TTI[TLE]|BTI[TLE] [text|OFF|ON]
```

```
SQL> TTITLE '부서 리스트'  
SQL> BTITLE '대외비'  
SQL> SELECT * FROM DEPT;
```

토 Jun 19

부서 리스트

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

대외비