

Join

강사 : 강병준

Table Join

- JOIN 조건
- 테이블 Alias
- 여러 결과 집합의 결합
 - EquiJoin
 - Non.EquiJoin
 - Self Join
 - Outer Join

JOIN 조건

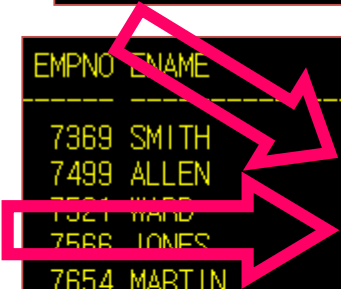
- 여러 테이블에서 특정 열 선택
 - Where절 안에 두 테이블의 공통되는 컬럼 비교
- 둘 이상의 테이블을 쿼리하여 결과 집합 생성
 - 기본 키(Primary Key) 및 외래 키(Foreign Key)를 조인 조건으로 사용
 - 테이블을 조인하려면 지정한 테이블에서 공통적으로 사용하는 열 사용

Join ?

- 조인(Join)? - 둘 이상의 테이블을 연결하여 데이터를 검색하는 방법

EMPNO	ENAME	JOB	DEPTNO
7369	SMITH	CLERK	20
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7788	SCOTT	ANALYST	20
7839	KING	PRESIDENT	10
7844	TURNER	SALESMAN	30
7876	ADAMS	CLERK	20
7900	JAMES	CLERK	30
7902	FORD	ANALYST	20
7934	MILLER	CLERK	10

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



EMPNO	ENAME	DEPTNO	DNAME	LOC
7369	SMITH	20	RESEARCH	DALLAS
7499	ALLEN	30	SALES	CHICAGO
7521	WARD	30	SALES	CHICAGO
7566	JONES	20	RESEARCH	DALLAS
7654	MARTIN	30	SALES	CHICAGO
7698	BLAKE	30	SALES	CHICAGO
7782	CLARK	10	ACCOUNTING	NEW YORK
7788	SCOTT	20	RESEARCH	DALLAS
7839	KING	10	ACCOUNTING	NEW YORK
7844	TURNER	30	SALES	CHICAGO
7876	ADAMS	20	RESEARCH	DALLAS
7900	JAMES	30	SALES	CHICAGO
7902	FORD	20	RESEARCH	DALLAS
7934	MILLER	10	ACCOUNTING	NEW YORK

조인의 필요성

- 특정 부서 번호에 대한 부서이름이 무엇인지는 부서(DEPT) 테이블에 있습니다. 특정 사원에 대한 부서명을 알아내기 위해서는 부서 테이블에서 정보를 얻어 와야 합니다.

```
C:\Windows\system32\cmd.exe
SQL> SELECT ENAME, DEPTNO
2 FROM EMP
3 ORDER BY DEPTNO;
```

ENAME	DEPTNO
CLARK	10
KING	10
MILLER	10
JONES	20
FORD	20
ADAMS	20
SMITH	20
SCOTT	20
WARD	30
TURNER	30
ALLEN	30
JAMES	30
BLAKE	30
MARTIN	30

14 개의 행이 선택되었습니다.

```
SQL>
```

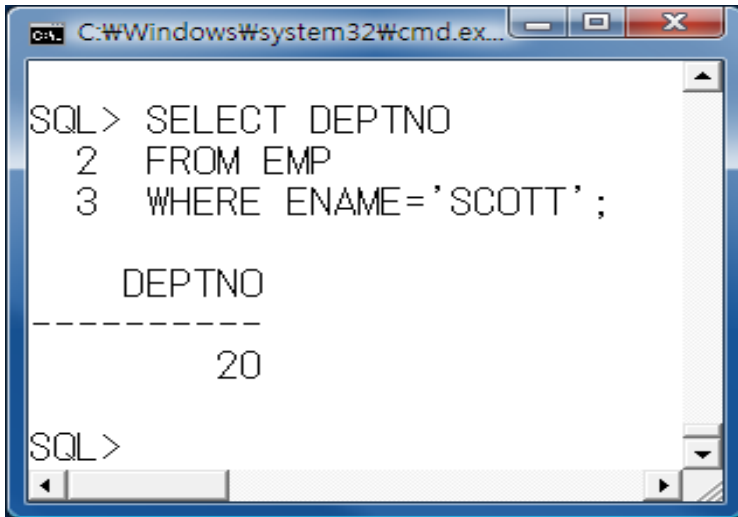
```
C:\Windows\system32\cmd.exe
SQL> SELECT DEPTNO, DNAME
2 FROM DEPT;
```

DEPTNO	DNAME
10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS

```
SQL>
```

조인의 필요성

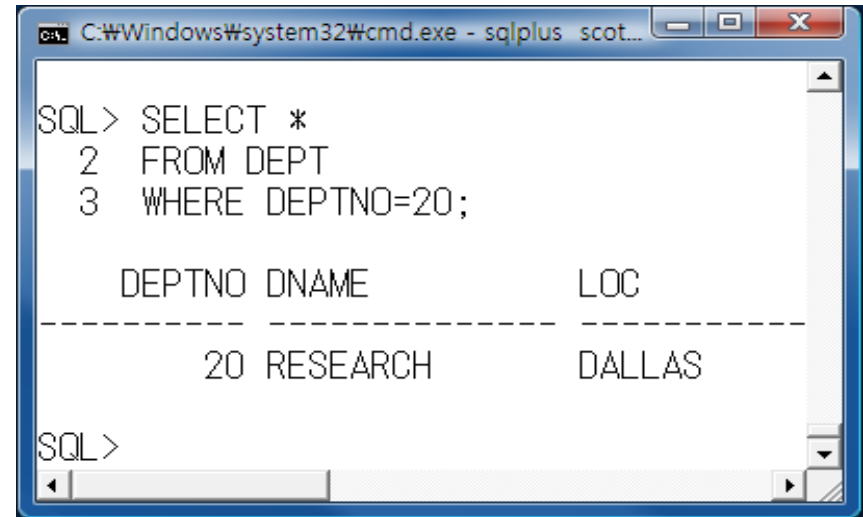
- SCOTT인 사원이 소속되어 있는 부서의 이름이 무엇인지 알아보려고 합니다.
- SCOTT이란 사원의 부서명을 알아내는 일 역시 사원 테이블에서 SCOTT이 소속된 부서 번호를 알아낸 후에 부서 테이블에서 해당 부서 번호에 대한 부서명을 얻어 와야 합니다.



```
SQL> SELECT DEPTNO
2 FROM EMP
3 WHERE ENAME='SCOTT';
```

DEPTNO
20

```
SQL>
```



```
SQL> SELECT *
2 FROM DEPT
3 WHERE DEPTNO=20;
```

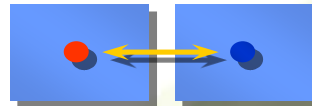
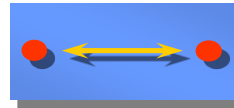
DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS

```
SQL>
```

- 실습에서처럼 원하는 정보가 두 개 이상의 테이블에 나누어져 있다면 위와 같이 여러 번 질의를 해야 할까요?
- 다행히도 SQL에서는 두 개 이상의 테이블을 결합해야만 원하는 결과를 얻을 수 있을 때 한 번의 질의로 원하는 결과를 얻을 수 있는 조인 기능을 제공합니다.

여러 테이블의 데이터 결합

- 테이블에 Alias 사용
- 카티션 곱
- Join 방법
 - **Equijoin** (동등 조인, 내부조인)
 - **Non-Equijoin**
 - **Self join**
 - **Outer Join**



테이블 이름에 별칭 사용

- 테이블 예명(Alias)
 - 테이블 alias로 column을 단순, 명확히 할 수 있음
 - 현재의 SELECT 문장에 대해서만 유효함
 - 테이블 alias는 길이가 30자 까지 가능하나 짧을수록 더욱 좋음
 - 테이블 alias는 의미가 있어야 함
 - FROM절에 테이블 alias설정 시 해당 테이블 alias는 SELECT문장에서 테이블 이름 대신에 사용해야 함

테이블 이름에 별칭 사용

- 예제 1

```
SELECT emp.ename, emp.job, emp.hiredate, dept.deptno,  
dept.dname  
FROM emp, dept  
WHERE emp.deptno = dept.deptno
```

예제 2

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

카티션 곱

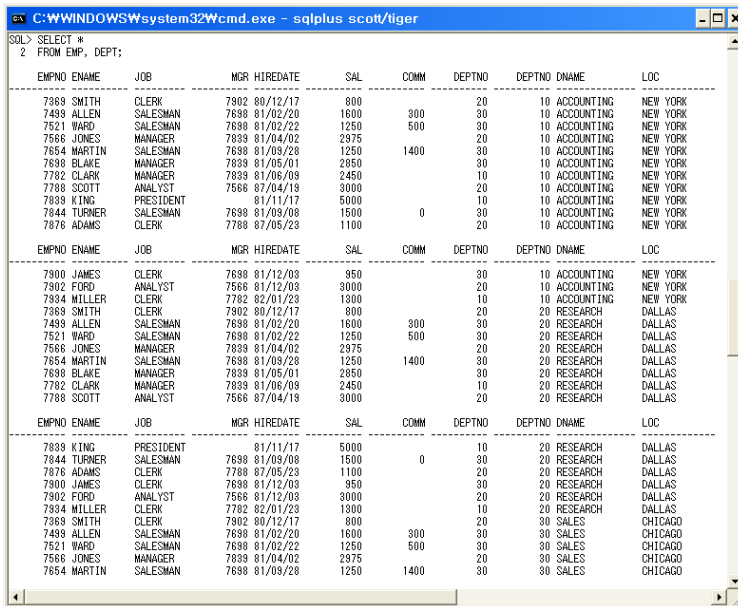
- Cartesian Product(카티션 곱)
 - 검색하고자 했던 데이터뿐 아니라 조인에 사용된 테이블들의 모든 데이터가 검색되는 현상
- Cartesian product는 다음과 같은 경우에 발생
 - 조인 조건을 정의하지 않았을 경우
 - 조인 조건이 잘못된 경우
 - 첫 번째 테이블의 모든 행들이 두 번째 테이블의 모든 행과 조인이 되는 경우
 - 테이블의 개수가 N 이라면 Cartesian product를 피하기 위해서는 적어도 $N-1$ 개의 등가

Cross Join

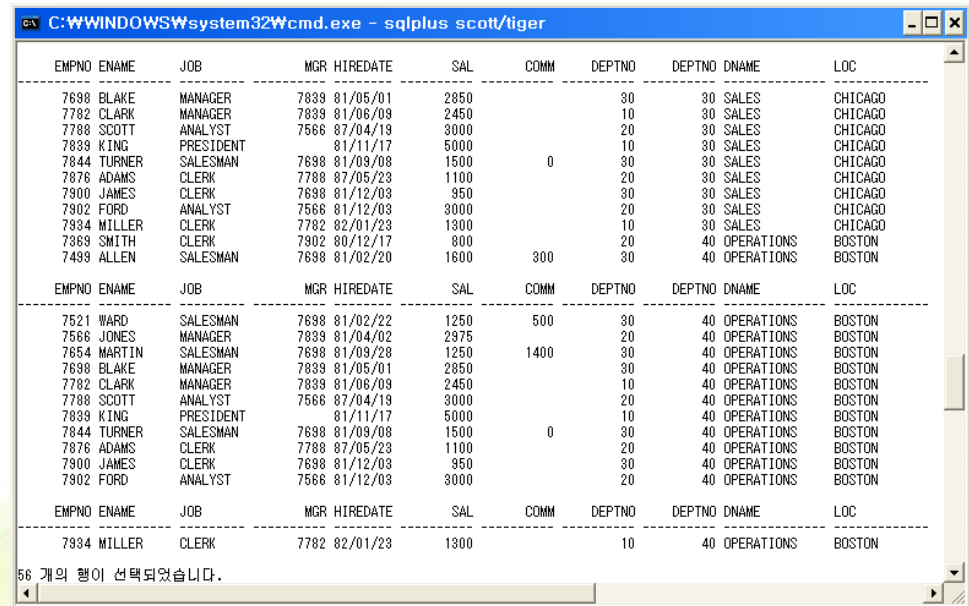
- 다음은 Cross Join으로 특별한 키워드 없이 SELECT 문의 FROM 절에 사원(EMP) 테이블과 부서(DEPT) 테이블을 콤마로 연결하여 연속하여 기술하는 것입니다. Cartesian Product 라고 합니다.

예

```
SELECT *  
FROM EMP, DEPT;
```



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO DNAME	LOC
7369	SMITH	CLERK	7902	80/12/17	800		20	10 ACCOUNTING NEW YORK	
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	10 ACCOUNTING NEW YORK	
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	10 ACCOUNTING NEW YORK	
7566	JONES	MANAGER	7839	81/04/02	2975		20	10 ACCOUNTING NEW YORK	
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	10 ACCOUNTING NEW YORK	
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	10 ACCOUNTING NEW YORK	
7782	CLARK	MANAGER	7839	81/06/09	2450		10	10 ACCOUNTING NEW YORK	
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	10 ACCOUNTING NEW YORK	
7839	KING	PRESIDENT		81/11/17	5000		10	10 ACCOUNTING NEW YORK	
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	10 ACCOUNTING NEW YORK	
7876	ADAMS	CLERK	7788	87/05/23	1100		20	10 ACCOUNTING NEW YORK	
7900	JAMES	CLERK	7698	81/12/03	950		30	10 ACCOUNTING NEW YORK	
7902	FORD	ANALYST	7566	81/12/03	3000		20	10 ACCOUNTING NEW YORK	
7934	MILLER	CLERK	7782	82/01/23	1300		10	10 ACCOUNTING NEW YORK	
7989	SMITH	CLERK	7902	80/12/17	800		20	10 ACCOUNTING NEW YORK	
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	40 OPERATIONS BOSTON	
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	40 OPERATIONS BOSTON	
7566	JONES	MANAGER	7839	81/04/02	2975		20	40 OPERATIONS BOSTON	
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	40 OPERATIONS BOSTON	
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	40 OPERATIONS BOSTON	
7782	CLARK	MANAGER	7839	81/06/09	2450		10	40 OPERATIONS BOSTON	
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	40 OPERATIONS BOSTON	
7839	KING	PRESIDENT		81/11/17	5000		10	40 OPERATIONS BOSTON	
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	40 OPERATIONS BOSTON	
7876	ADAMS	CLERK	7788	87/05/23	1100		20	40 OPERATIONS BOSTON	
7900	JAMES	CLERK	7698	81/12/03	950		30	40 OPERATIONS BOSTON	
7902	FORD	ANALYST	7566	81/12/03	3000		20	40 OPERATIONS BOSTON	
7934	MILLER	CLERK	7782	82/01/23	1300		10	40 OPERATIONS BOSTON	
7989	SMITH	CLERK	7902	80/12/17	800		20	40 OPERATIONS BOSTON	
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	40 OPERATIONS BOSTON	
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	40 OPERATIONS BOSTON	
7566	JONES	MANAGER	7839	81/04/02	2975		20	40 OPERATIONS BOSTON	
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	40 OPERATIONS BOSTON	



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO DNAME	LOC
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	30 SALES CHICAGO	
7782	CLARK	MANAGER	7839	81/06/09	2450		10	30 SALES CHICAGO	
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	30 SALES CHICAGO	
7839	KING	PRESIDENT		81/11/17	5000		10	30 SALES CHICAGO	
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	30 SALES CHICAGO	
7876	ADAMS	CLERK	7788	87/05/23	1100		20	30 SALES CHICAGO	
7900	JAMES	CLERK	7698	81/12/03	950		30	30 SALES CHICAGO	
7902	FORD	ANALYST	7566	81/12/03	3000		20	30 SALES CHICAGO	
7934	MILLER	CLERK	7782	82/01/23	1300		10	30 SALES CHICAGO	
7989	SMITH	CLERK	7902	80/12/17	800		20	40 OPERATIONS BOSTON	
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	40 OPERATIONS BOSTON	
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	40 OPERATIONS BOSTON	
7566	JONES	MANAGER	7839	81/04/02	2975		20	40 OPERATIONS BOSTON	
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	40 OPERATIONS BOSTON	
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	40 OPERATIONS BOSTON	
7782	CLARK	MANAGER	7839	81/06/09	2450		10	40 OPERATIONS BOSTON	
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	40 OPERATIONS BOSTON	
7839	KING	PRESIDENT		81/11/17	5000		10	40 OPERATIONS BOSTON	
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	40 OPERATIONS BOSTON	
7876	ADAMS	CLERK	7788	87/05/23	1100		20	40 OPERATIONS BOSTON	
7900	JAMES	CLERK	7698	81/12/03	950		30	40 OPERATIONS BOSTON	
7902	FORD	ANALYST	7566	81/12/03	3000		20	40 OPERATIONS BOSTON	
7934	MILLER	CLERK	7782	82/01/23	1300		10	40 OPERATIONS BOSTON	

56 개의 행이 선택되었습니다.

Cross Join

- Cross Join의 결과 얻어지는 컬럼의 수는 사원 테이블의 컬럼의 수(8)와 부서 테이블의 컬럼의 수(3)를 더한 것이므로 11이 됩니다. 행의 수는 사원 한 명에 대해서 DEPT 테이블의 모든 행과 결합되기에 2개 테이블의 행의 개수를 곱한 것 만큼 출력됩니다.
- Cross Join의 결과를 보면 사원 테이블에 부서에 대한 상세정보가 결합되긴 했지만, 조인 될 때 아무런 조건을 제시하지 않았기에 사원 한 명에 대해서 DEPT 테이블의 모든 행의 데이터와 결합된 형태이기 에 Cross Join의 결과는 아무런 의미를 갖지 못합니다.
- 조인 결과가 의미를 갖으려면 조인할 때 조건을 지정해야 합니다.

카디션 곱

- 카디션 곱 질의

```
SELECT ename, emp.deptno, dname  
FROM emp, dept;
```

CROSS JOIN
select ename, job
from emp **CROSS JOIN** dept;

– 결과 : 56 행 검출

Join 질의

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

참고

Cartesian Product는 많은 수의 행을 생성하는 경향이 있고 결과도 거의 유용하지 못하다. 그러므로 모든 테이블로부터 모든 행을 조합할 필요가 없을 경우 WHERE절에 조인 조건을 명확히 기술하여야 한다.

문제 1) EMP 테이블과 DEPT 테이블을 Cartesian Product하여
사원번호, 이름, 업무, 부서번호, 부서명, 근무지를 출력하여라.

SQL> **SELECT empno,ename,job,dept.deptno,dname,loc FROM dept,emp**
3 ORDER BY empno;

EMPNO	ENAME	JOB	DEPTNO	DNAME	LOC
7369	SMITH	CLERK	10	ACCOUNTING	NEW YORK
7369	SMITH	CLERK	20	RESEARCH	DALLAS
7369	SMITH	CLERK	30	SALES	CHICAGO
7369	SMITH	CLERK	40	OPERATIONS	BOSTON
7499	ALLEN	SALESMAN	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	40	OPERATIONS	BOSTON
7566	JONES	MANAGER	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	20	RESEARCH	DALLAS
7566	JONES	MANAGER	30	SALES	CHICAGO

.....
56 rows selected.

동등 조인

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

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

DEPT

Table Name: **EMP**

Table Name: **DEPT**

EMPNO	ENAME	JOB	DEPTNO	DEPTNO	DNAME	LOC
7839	KING	PRESIDENT	10	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	30	20	RESEARCH	DALLAS
7782	CLARK	MANAGER	10	30	SALES	CHICAGO
7566	JONES	MANAGER	20	40	OPERATIONS	BOSTON

Primary key

Foreign key

Primary key

Equi Join (동등조인)

- Equi Join

- 조건절 Equality Condition(=)에 의하여 조인이 이루어짐
- PK와 FK 관계가 있는 테이블에서 주로 사용

```
SQL> SELECT      e.empno, e.ename, e.deptno,  
2                d.deptno, d.loc  
3 FROM          emp e, dept d  
4 WHERE         e.deptno=d.deptno;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS

...
14 rows selected.

Equi Join (동등조인)

Equijoin이란 조인 조건에서 “=”을 사용하여 값들이 정확하게 일치하는 경우에 사용하는 조인을 말합니다. 대부분 PK와 FK의 관계를 이용하여 조인합니다. Equijoin은 다른 말로 단순 조인 또는 내부 조인 이라고도 합니다.

Syntax

```
SELECT table1.column1 [,table2.column2, . . . . .]  
FROM table1, table2  
WHERE table1.column1 = table2.column2;
```

table1.column1

- 조회할 자료가 있는 테이블과 열 이름을 기술

table1.column1=table2.column2

- 두 테이블들간에 논리적으로 연결하는 조인 조건 기술

Equijoin으로 자료 검색

- 1) SELECT절은 검색할 열 이름을 명시
- 2) FROM절은 데이터베이스가 Access해야 하는 두개의 테이블을 명시
- 3) WHERE절은 테이블의 조인 조건을 명시
- 4) 양쪽 테이블에 공통으로 존재하는 열 이름은 모호함을 피하기 위하여 열 이름 앞에 테이블명을 기술함

Equi Join (동등조인)

Equi join 의 방법

종업원의 부서 이름을 결정하기 위해 EMP Table 의 DEPTNO 와 DEPT Table 의 DEPTNO 와 값을 비교하여야 합니다. EMP Table 과 DEPT Table 사이의 관계는 양쪽 테이블의 DEPTNO 열이 같아야 합니다. 이들이 PK 와 FK 로 연결되어 있습니다...

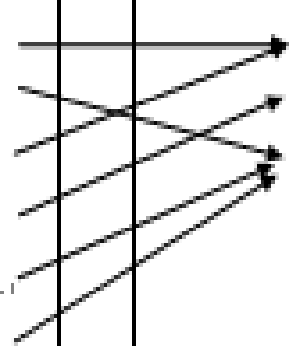
```
SQL> SELECT empno,ename,job,deptno  
2 FROM emp; 2 FROM dept;
```

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10
7698	BLAKE	MANAGER	30
7782	CLARK	MANAGER	10
7566	JONES	MANAGER	20
7654	MARTIN	SALESMAN	30
7499	ALLEN	SALESMAN	30

14 rows selected...

```
SQL> SELECT deptno,dname,loc.
```

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



Equi Join (동등조인)

EMP 테이블에서 사원번호,이름,업무,EMP 테이블의 부서번호,DEPT 테이블의 부서번호,부서명,근무지를 출력하더라...

```
SQL> SELECT empno,ename,job,emp.deptno,dept.deptno,dname,loc.  
2 FROM dept,emp.  
3 WHERE dept.deptno = emp.deptno.  
4 ORDER BY dept.deptno;
```

EMPNO	ENAME	JOB	DEPTNO	DEPTNO	DNAME	LOC.
7839	KING	PRESIDENT	10	10	ACCOUNTING	NEW YORK.
7782	CLARK	MANAGER	10	10	ACCOUNTING	NEW YORK.
7934	MILLER	CLERK	10	10	ACCOUNTING	NEW YORK.
7566	JONES	MANAGER	20	20	RESEARCH	DALLAS.
7788	SCOTT	ANALYST	20	20	RESEARCH	DALLAS.

.....
14 rows selected..

Equi Join (동등조인)

Guidelines ..

- 1) 테이블 Alias 는 30 자까지 사용 가능하지만 짧을수록 더 좋다...
- 2) FROM 절에서 Alias 가 사용되면 SELECT 문 전체에서 사용 가능하다...
- 3) 테이블의 Alias 에 가급적 의미를 부여..
- 4) 테이블은 현재 SELECT 문장에서만 유용..

AND 연산자를 사용하여 추가적인 검색 조건 ..

조인 이외의 WHERE 절에 추가적인 조건을 가질 수 있다...

SALESMAN 사원만 사원번호,이름,급여,부서명,근무지를 출력하여라..

```
SQL> SELECT e.empno,e.ename,e.sal,d.dname,d.loc,  
2 FROM dept d,emp e..  
3 WHERE d.deptno = e.deptno AND e.job = 'SALESMAN' ;
```

EMPNO	ENAME	SAL	DNAME	LOC..
7654	MARTIN	1250	SALES	CHICAGO..
7499	ALLEN	1600	SALES	CHICAGO..
7844	TURNER	1500	SALES	CHICAGO..
7521	WARD	1250	SALES	CHICAGO..

Equi Join (동등조인)

- 다음은 두 테이블을 조인한 결과입니다. 결과를 살펴보면 다음과 같이 부서 번호를 기준으로 같은 값을 가진 사원 테이블의 컬럼과 부서 테이블의 컬럼이 결합됩니다.

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

```
SQL> SELECT *
2 FROM EMP, DEPT
3 WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7782	CLARK	MANAGER	7839	81/06/09	2450		10	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT		81/11/17	5000		10	10	ACCOUNTING	NEW YORK
7934	MILLER	CLERK	7782	82/01/23	1300		10	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	81/04/02	2975		20	20	RESEARCH	DALLAS
7902	FORD	ANALYST	7566	81/12/03	3000		20	20	RESEARCH	DALLAS
7876	ADAMS	CLERK	7788	87/05/23	1100		20	20	RESEARCH	DALLAS
7369	SMITH	CLERK	7902	80/12/17	800		20	20	RESEARCH	DALLAS
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	20	RESEARCH	DALLAS
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	30	SALES	CHICAGO
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	30	SALES	CHICAGO
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	30	SALES	CHICAGO
7900	JAMES	CLERK	7698	81/12/03	950		30	30	SALES	CHICAGO
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	30	SALES	CHICAGO
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	30	SALES	CHICAGO

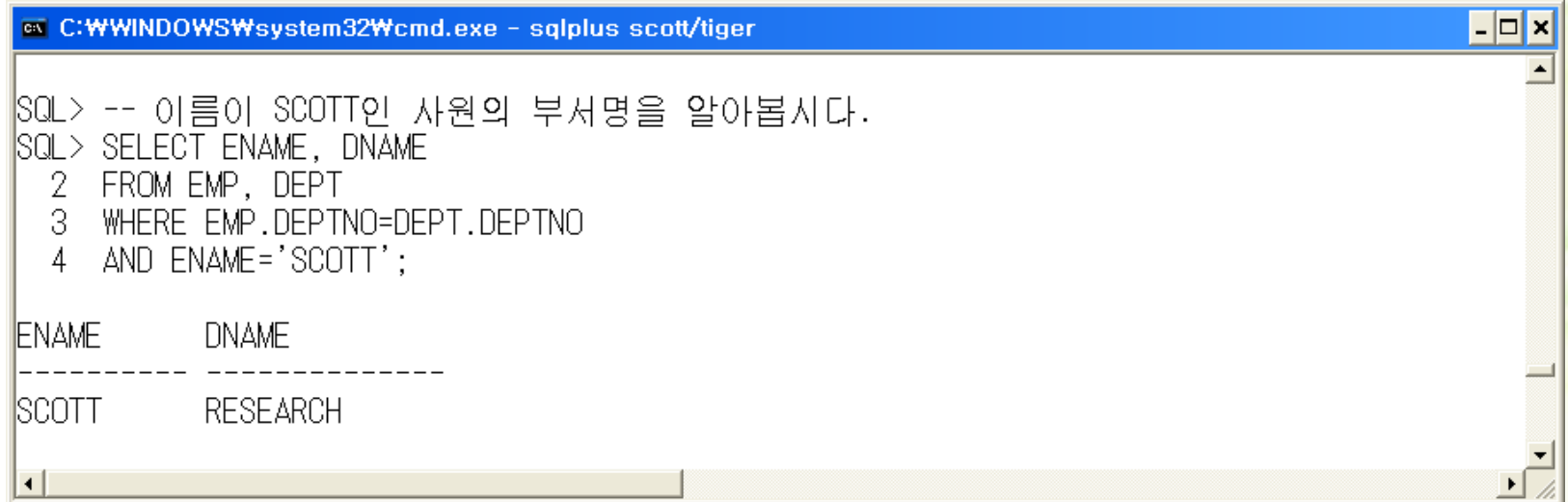
14 개의 행이 선택되었습니다.

Equi Join에 AND 연산

- emp 테이블의 ename이 SCOTT인 사원의 ename과 dept 테이블의 dname을 출력

예

```
SELECT ENAME, DNAME  
FROM EMP, DEPT  
WHERE EMP.DEPTNO=DEPT.DEPTNO  
AND ENAME='SCOTT';
```



```
C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> -- 이름이 SCOTT인 사원의 부서명을 알아봅시다.  
SQL> SELECT ENAME, DNAME  
2 FROM EMP, DEPT  
3 WHERE EMP.DEPTNO=DEPT.DEPTNO  
4 AND ENAME='SCOTT';  
  
ENAME      DNAME  
-----  
SCOTT      RESEARCH
```

컬럼명의 모호성 해결

- 두 테이블에 동일한 이름의 컬럼을 사용하면 어느 테이블 소속인지 불분명하기에 애매모호한 상태라는 오류 메시지가 출력됩니다.

예

```
SELECT ENAME, DNAME, DEPTNO  
FROM EMP, DEPT  
WHERE EMP.DEPTNO = DEPT.DEPTNO  
AND ENAME='SCOTT';
```

```
C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> -- 이름이 SCOTT인 사원의 부서명을 알아내어 사원이름 부서명 부서번호를 출력하시오.  
SQL> SELECT ENAME, DNAME, DEPTNO  
2 FROM EMP, DEPT  
3 WHERE EMP.DEPTNO=DEPT.DEPTNO  
4 AND ENAME='SCOTT';  
SELECT ENAME, DNAME, DEPTNO  
      *  
1행에 오류:  
ORA-00918: 열의 정의가 애매합니다
```

컬럼명의 모호성 해결

- 이러한 문제를 해결하기 위한 방법이 있어야 합니다. 이렇게 동일한 이름의 컬럼은 컬럼 명 앞에 테이블 명을 명시적으로 기술함으로써 컬럼이 어느 테이블 소속인지 구분할 수 있게 됩니다.

예

```
SELECT EMP.ENAME, DEPT.DNAME, EMP.DEPTNO  
FROM EMP, DEPT  
WHERE EMP.DEPTNO=DEPT.DEPTNO  
AND ENAME='SCOTT';
```

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

```
SQL> -- 컬럼명 앞에 테이블 명을 명시하기  
SQL> SELECT EMP.ENAME, DEPT.DNAME, EMP.DEPTNO  
2 FROM EMP, DEPT  
3 WHERE EMP.DEPTNO=DEPT.DEPTNO  
4 AND ENAME='SCOTT';
```

ENAME	DNAME	DEPTNO
SCOTT	RESEARCH	20

테이블에 별칭 부여하기

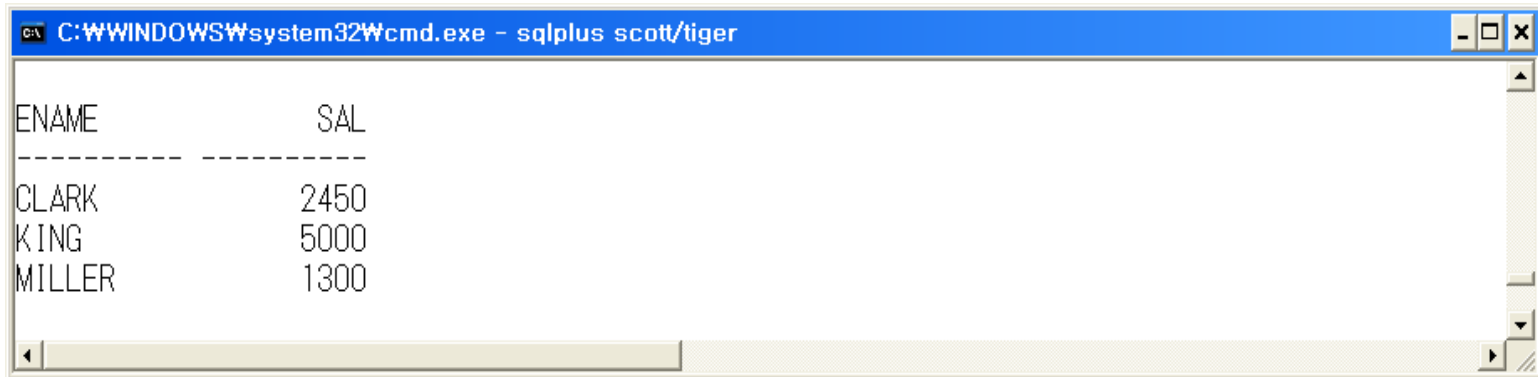
- 테이블 이름에 별칭을 붙이는 방법은 FROM 절 다음에 테이블 이름을 명시하고 공백을 둔 다음에 별칭을 지정하면 됩니다.

예

```
SELECT E.ENAME, D.DNAME, E.DEPTNO, D.DEPTNO  
FROM EMP E, DEPT D  
WHERE E.DEPTNO = D.DEPTNO  
AND E.ENAME='SCOTT';
```

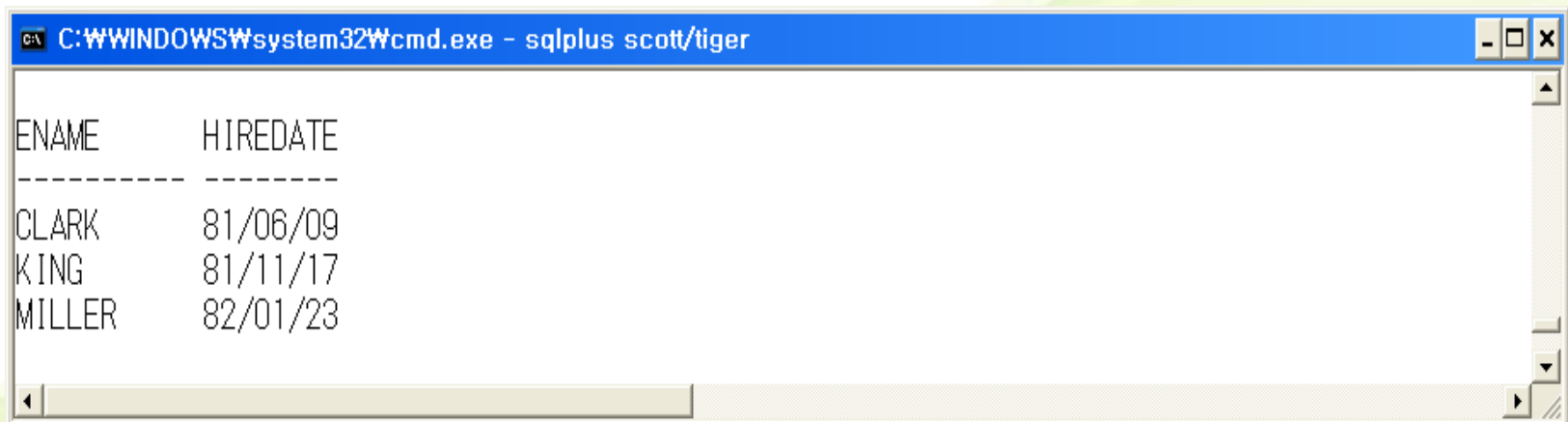
연습문제

1. DEPT 테이블의 LOC가 'NEW YORK' 인 사원의 EMP 테이블의 이름(ename)과 급여(sal)를 출력하시오.



ENAME	SAL
CLARK	2450
KING	5000
MILLER	1300

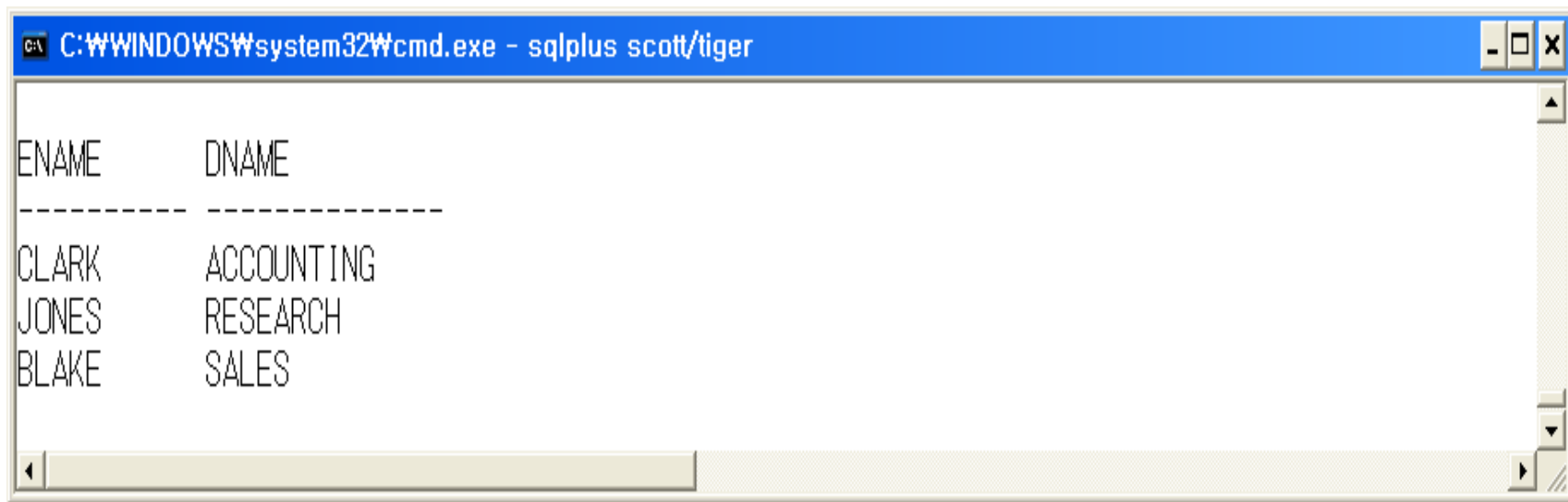
2. DEPT 테이블의 DNAME 컬럼의 값이 'ACCOUNTING' 인 사원의 EMP 테이블의 이름(ENAME)과 입사일(HIREDATE)을 출력하시오.



ENAME	HIREDATE
CLARK	81/06/09
KING	81/11/17
MILLER	82/01/23

연습문제

3. EMP 테이블의 JOB이 'MANAGER'인 사원의 EMP 테이블의 ename, DEPT 테이블의 dname을 출력하시오.



The screenshot shows a Windows command prompt window with the title bar "C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger". The window displays the output of a SQL query, which is a table with two columns: ENAME and DNAME. The table is separated by a dashed line. The data rows are: CLARK ACCOUNTING, JONES RESEARCH, and BLAKE SALES.

ENAME	DNAME
CLARK	ACCOUNTING
JONES	RESEARCH
BLAKE	SALES

Joining More Than Two Tables

CUSTOMER

NAME	CUSTID
-----	-----
JOCKSPORTS	100
TKB SPORT SHOP	101
VOLLYRITE	102
JUST TENNIS	103
K+T SPORTS	105
SHAPE UP	106
WOMENS SPORTS	107
...	...
9 rows selected.	

ORD

CUSTID	ORDID
-----	-----
101	610
102	611
104	612
106	601
102	602
106	
106	
...	...
21 rows selected.	

ITEM

ORDID	ITEMID
-----	-----
610	3
611	1
612	1
601	1
602	1
...	...
64 rows selected.	

Joining More Than Two Tables

두개 이상의 테이블 조인..

때로는 두개 이상의 테이블을 조인 할 경우가 있다...

고객의 TKB SPORT SHOP 의 이름,주문처,항목수,각 항목의 합계,각 주문의 합계를 출력하여라...

<pre>SQL> SELECT name,custid, 2 FROM customer;..</pre>	<pre>SQL> SELECT ordid,itemid, 2 FROM item;..</pre>	<pre>SQL> SELECT custid,ordid 2 FROM ord; ..</pre>																								
<table><tr><th>NAME</th><th>CUSTID..</th></tr><tr><td>JOCKSPORTS</td><td>100..</td></tr><tr><td>TKB SPORT SHOP</td><td>101..</td></tr><tr><td>VOLLYRITE</td><td>102..</td></tr></table> <p>9 rows selected. ..</p>	NAME	CUSTID..	JOCKSPORTS	100..	TKB SPORT SHOP	101..	VOLLYRITE	102..	<table><tr><th>ORDID</th><th>ITEMID..</th></tr><tr><td>610</td><td>3..</td></tr><tr><td>611</td><td>1..</td></tr><tr><td>612</td><td>1..</td></tr></table> <p>64 rows selected..</p>	ORDID	ITEMID..	610	3..	611	1..	612	1..	<table><tr><th>CUSTID</th><th>ORDID ..</th></tr><tr><td>101</td><td>610 ..</td></tr><tr><td>102</td><td>611 ..</td></tr><tr><td>104</td><td>612 ..</td></tr></table> <p>21 rows selected. ..</p>	CUSTID	ORDID ..	101	610 ..	102	611 ..	104	612 ..
NAME	CUSTID..																									
JOCKSPORTS	100..																									
TKB SPORT SHOP	101..																									
VOLLYRITE	102..																									
ORDID	ITEMID..																									
610	3..																									
611	1..																									
612	1..																									
CUSTID	ORDID ..																									
101	610 ..																									
102	611 ..																									
104	612 ..																									
<pre>SQL> SELECT c.name,o.ordid,i.itemid,i.itemtot,o.total, 2 FROM customer c,ord o,item i, 3 WHERE c.custid = o.custid AND o.ordid = i.ordid, 4 AND c.name = 'TKB SPORT SHOP';..</pre>																										
<table><tr><th>NAME</th><th>ORDID</th><th>ITEMID</th><th>ITEMTOT</th><th>TOTAL..</th></tr><tr><td>TKB SPORT SHOP</td><td>610</td><td>3</td><td>58</td><td>101.4..</td></tr><tr><td>TKB SPORT SHOP</td><td>610</td><td>1</td><td>35</td><td>101.4..</td></tr><tr><td>TKB SPORT SHOP</td><td>610</td><td>2</td><td>8.4</td><td>101.4..</td></tr></table>			NAME	ORDID	ITEMID	ITEMTOT	TOTAL..	TKB SPORT SHOP	610	3	58	101.4..	TKB SPORT SHOP	610	1	35	101.4..	TKB SPORT SHOP	610	2	8.4	101.4..				
NAME	ORDID	ITEMID	ITEMTOT	TOTAL..																						
TKB SPORT SHOP	610	3	58	101.4..																						
TKB SPORT SHOP	610	1	35	101.4..																						
TKB SPORT SHOP	610	2	8.4	101.4..																						

Non-Equi Join

- Non-equijoin은 테이블의 어떤 column도 join할 테이블의 column에 일치하지 않을 때 사용하고 조인조건은 동등(=) 이외의 연산자를 갖는다.
 - **BETWEEN AND**
 - **IS NULL, IS NOT NULL**
 - **IN, NOT IN**
- 급여 등급 테이블(salgrade)에는 급여에 대한 등급을 다음과 같이 나누어 놓았습니다.
- 급여의 등급은 총 5등급으로 나누어져 있으며, 1등급은 급여가 700부터 1200 사이이고, 2등급은 1201부터 1400 사이이고, 3등급은 1401부터 2000 사이이고, 4등급은 2001부터 3000사이이고, 5등급이면 3001부터 9999사이입니다.
- 급여 등급을 5개로 나누어 놓은 salgrade에서 정보를 얻어 와서 각 사원의 급여 등급을 지정해보도록 합니다. 이를 위해서 사원(emp) 테이블과 급여 등급(salgrade) 테이블을 조인하도록 합니다. 다음은 사원의 급여가 몇 등급인지 살펴보는 예제입니다.

Non-Equi Join

EMP

EMPNO	ENAME	SAL

7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250
7499	ALLEN	1600
7844	TURNER	1500
7900	JAMES	950
...		
14 rows selected.		

SALGRADE

GRADE	LOSAL	HISAL

1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

“salary in the EMP table is between low salary and high salary in the SALGRADE table”

Non-Equi Join

```
SQL>  SELECT    e.ename, e.sal, s.grade
      2  FROM      emp e,  salgrade s
      3  WHERE     e.sal
      4  BETWEEN  s.losal AND s.hisal;
```

ENAME	SAL	GRADE
-----	-----	
JAMES	950	1
SMITH	800	1
ADAMS	1100	1

...

14 rows selected.

Non-Equi Join

Non-Equi join

EMP 와 SALGRADE 사이의 관련성은 EMP 테이블의 어떠한 column 도 직접적으로 SALGRADE 테이블의 한 column 에 상응하지 않기 때문에 Non-Equijoin 이다. 두 테이블 사이의 관련성은 EMP 테이블의 SAL 열이 SALGRADE 테이블의 LOSAL 과 HISAL 열 사이에 있다는 것이다. 조인 조건은 등등(=) 이외의 연산자(BETWEEN ~ AND ~)를 갖는다...

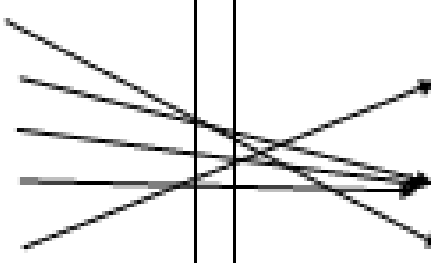
```
SQL> SELECT empno,ename,sal  
2 FROM emp;
```

EMPNO	ENAME	SAL
7839	KING	5000
7698	BLAKE	2850
7782	CLARK	2450
7566	JONES	2975
7654	MARTIN	1250

14 rows selected...

```
SQL> SELECT grade,losal,hisal  
2 FROM salgrade;
```

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999



Non-Equi Join

EMP 테이블에서 사원번호,이름,업무,급여,급여의 등급,하한값,상한값을 출력하여라...

```
SQL> SELECT e.empno,e.ename,e.job,e.sal,s.grade,s.losal,s.hisal,
2 FROM salgrade s,emp e,
3 WHERE e.sal BETWEEN s.losal AND s.hisal AND e.deptno = 10;
```

EMPNO	ENAME	JOB	SAL	GRADE	LOSAL	HISAL
7839	KING	PRESIDENT	5000	5	3001	9999
7782	CLARK	MANAGER	2450	4	2001	3000
7934	MILLER	CLERK	1300	2	1201	1400

❁ 참고

위 질의가 실행될 때 한번만 조인되는 것을 알 수 있다. 이에 대한 두가지 이유가 있다...

- 1) SALGRADE 테이블에서 중복되는 등급을 포함하는 행이 없다...
- 2) EMP 테이블에 있는 SAL의 값은 SALGRADE 테이블에서 제공하는 값 범위에 있다...

☞ Guidelines

<= 및 >= 같은 다른 연산자를 사용 가능하나 BETWEEN 이 가장 단순하다. 또한 테이블에 Alias 를 사용하였는데 이는 보호성 때문이 아니라 성능 때문에 사용하였다. BETWEEN 사용 시 하한값을 먼저 명시하고 상한값을 나중에 명시한다는 것을 명심하라...

Outer Join (외부조인)

- 정상적으로 조인 조건을 만족하지 못하는 행들을 보기 위해 outer join을 사용. Outer join 연산자 "(+)"
- 조인조건에 만족되지 않는 값들을 구할 때 해당 위치에 "(+)"를 위치 시킴
- 일반적으로 OUTER JOIN이란 두 개 이상의 테이블을 JOIN할 경우에 테이블 중에서 기준이 되는 테이블의 데이터가 다른 테이블의 데이터와 일치 되지 않는 경우에도 JOIN을 할 수 있도록 하기 위해서 사용.
- OUTER JOIN을 일반적인 JOIN문으로 표현하면, SQL문이 길어지며 또한 실행계획이 인덱스를 이용하지 못하여 수행시간이 늘어남.
- 이러한, 경우에 OUTER JOIN으로 간단하게 표현할 수 있으며, 수행시간 또한 일반적인 JOIN과 거의 비슷하게 진행

Outer Join (외부조인)

- 일반적인 Join

```
SQL> SELECT DISTINCT(e.deptno), d.deptno  
FROM emp e, dept d  
WHERE e.deptno = d.deptno
```

DEPTNO	DEPTNO
10	10
20	20
30	30

Outer Join

```
SQL> SELECT DISTINCT(e.deptno), d.deptno  
FROM emp e, dept d  
WHERE e.deptno(+) = d.deptno
```

DEPTNO	DEPTNO
10	10
20	20
30	30
	40

Oracle Outer Join

- Left Outer Join

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

Right Outer Join

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

Full Outer Join

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

Oracle Outer Join

Guidelines ..

- 1) 행인 조건을 만족하지 않을 시 해당 행은 질의 결과에 나타나지 않는다...
- 2) Outer join 연산자를 조인 조건에 사용시 조인 조건을 만족하지 않는 행들도 결과에 나타날 수 있다...
- 3) 연산자는 괄호로 묶인 플러스 기호(+)이며 조인 시킬 값이 없는 조인 측에 "(+)"를 위치 시킨다...
- 4) (+)연산자는 한 개 이상의 NULL 행을 생성하고 정보가 충분한 테이블의 한 개 이상의 행들이 이런 NULL 행에 조인된다...
- 5) Outer join 연산자는 표현식의 한 편에만 올 수 있다...
- 6) Outer join 을 포함하는 조건은 IN 연산자, OR 연산자를 사용하여 다른 하나의 조건 연결될 수 없다..

1.6.1) Syntax..

```
SELECT          table1.column1 [,table2.column2, . . . . .]..
FROM            table1, table2..
WHERE           table1.column1 = table2.column2(+);..
```

```
SELECT          table1.column1 [,table2.column2, . . . . .]..
FROM            table1, table2..
WHERE           table1.column1(+) = table2.column2;..
```

Oracle Outer Join

table1.column 테이블을 함께 조인(또는 관련)시키는 조건입니다...

table2.column(+) (+)는 outer join 기호입니다. WHERE 절 조건의 양쪽이 아니라,
어느 한쪽에 둘 수 있습니다. 즉 양쪽 모두에는 둘 수 없습니다...
일치하는 행이 없는 테이블의 열 이름 뒤에 outer join 연산자,
를 사용합니다...

1.6.2) Outer Join 제약 사항..

- 1) Outer Join 연산자는 정보가 부재하는 쪽의 표현식 한 쪽에만 둡니다. 다른 테이블의 어떠한 열과도 직접적으로 일치하는 것이 없는 한 테이블의 행을 리턴합니다...
- 2) Outer Join 을 포함하는 조건은 IN 연산자를 사용할 수 없고, OR 연산자에 의해 다른 조건과 연결될 수 없습니다...

```
SQL> select * from emp,dept..
      2  where dept.deptno(+) = emp.deptno(+);..
where dept.deptno(+) = emp.deptno(+)..
      *..

ERROR at line 2:..
ORA-01468: a predicate may reference only one outer-joined table..
..
SQL> ..
```

Oracle Outer Join

EMP 테이블과 DEPT 테이블에서 DEPT 테이블에 있는 모든 자료를 사원번호,이름,업

무,EMP 테이블의 부서번호,DEPT 테이블의 부서번호,부서명,근무지를 출력하여라.

```
SQL> SELECT e.empno,e.ename,e.job,e.deptno,
2      d.deptno,d.dname,d.loc,
3      FROM dept d,emp e,
4      WHERE d.deptno = e.deptno(+);
```

EMPNO	ENAME	JOB	DEPTNO	DEPTNO	DNAME	LOC.
7900	JAMES	CLERK	30	30	SALES	CHICAGO.
7521	WARD	SALESMAN	30	30	SALES	CHICAGO.
				40	OPERATIONS	BOSTON.

15 rows selected...

... Self Join

때때로 자체적으로 테이블을 조인할 필요가 있습니다. 각 종업원의 관리자 명을 알기 위해서 자체적으로 EMP 테이블을 조인하는 것이 필요합니다...

- 1) ENAME 열을 검사하여 EMP 테이블에서 Blake를 검색한다...
- 2) MGR 열을 검사하여 Blake에 대한 관리자 번호를 검색한다.(Blake 관리자 번호:7839)..
- 3) ENAME 열을 검사하여 EMPNO 가 7839 인 관리자를 검색한다.7839 는 King 이므로 Blake의 관리자는 King이다...

Self Join

- Equi Join과 같으나 하나의 테이블에서 조인이 일어나는 것이 다름
- 같은 테이블에 대해 두 개의 alias를 작성함으로 FROM 절에 두 개의 테이블을 사용 하는 것과 같이 함
- 조인은 두 개 이상의 서로 다른 테이블을 서로 연결하는 것뿐만 아니라, 하나의 테이블 내에서 조인을 해야만 원하는 자료를 얻는 경우가 생깁니다.
- Self Join이란 말 그대로 자기 자신과 조인을 맺는 것을 말합니다.
- Self Join을 보다 구체적인 예를 통해서 알아보도록 합시다.
- SMITH의 매니저 이름이 무엇인지 알아내려면 어떻게 구해야 할까요?

Self Join

```
C:\Windows\system32\cmd.exe
SQL> SELECT ENAME, MGR
2 FROM EMP;
```

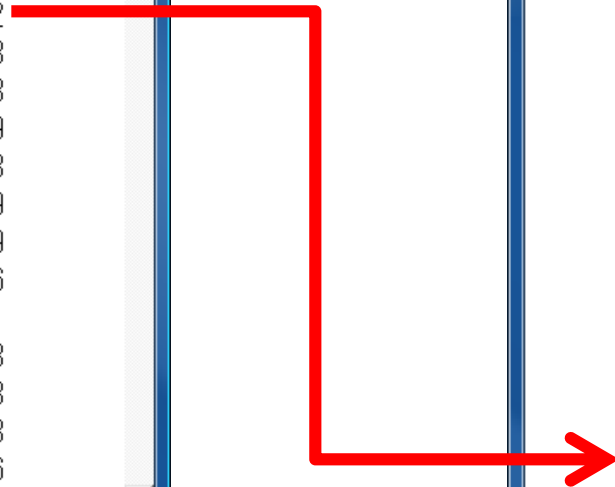
ENAME	MGR
SMITH	7902
ALLEN	7698
WARD	7698
JONES	7839
MARTIN	7698
BLAKE	7839
CLARK	7839
SCOTT	7566
KING	
TURNER	7698
ADAMS	7788
JAMES	7698
FORD	7566
MILLER	7782

14 개의 행이 선택되었습니다.

```
C:\Windows\system32\cmd.exe
SQL> SELECT EMPNO, ENAME
2 FROM EMP;
```

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

14 개의 행이 선택되었습니다.



Self Join

EMP (WORKER)

EMPNO	ENAME	MGR
-----	-----	----
7839	KING	
7698	BLAKE	7839
7782	CLARK	7839
7566	JONES	7839
7654	MARTIN	7698
7499	ALLEN	7698

EMP (MANAGER)

EMPNO	ENAME
-----	-----
7839	KING
7839	KING
7839	KING
7698	BLAKE
7698	BLAKE



**“MGR in the WORKER table is equal to EMPNO in the
MANAGER table”**

Self Join

```
SQL> SELECT worker.ename||'의 상사는 '||manager.ename|| '입니다'
2 FROM      emp worker, emp manager
3 WHERE      worker.mgr = manager.empno;
```

```
WORKER.ENAME||'WORKSFOR'||MANAG
```

```
-----
BLAKE의 상사는 KING 입니다.
CLARK의 상사는 KING 입니다.
JONES의 상사는 KING 입니다.
MARTIN의 상사는 BLAKE 입니다.
```

```
...
```

```
13 rows selected.
```

Self Join

```
SQL> SELECT empno,ename,mgr  
2 FROM emp;
```

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

14 rows selected.

```
SQL> SELECT empno,ename,mgr  
2 FROM emp;
```

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

14 rows selected...

Self Join

Guidelines ..

- 1) Self join을 사용하여 한 테이블의 행들을 같은 테이블의 행들과 조인한다...
- 2) 같은 테이블에 대해 두 개의 alias 를 작성(테이블 구분)함으로써 FROM 절에 두개의 테이블을 사용하는 것과 같이 한다...
- 3) Column에 대해서도 어떤 테이블에서 왔는지 반드시 Alias명을 기술하여야 한다...
- 4) 테이블 하나를 두개 또는 그 이상으로 Self join할 수 있다...

문제 7) EMP 테이블에서 Self join하여 관리자를 출력하여라...

```
SQL> SELECT worker.ename || '의 관리자는 ' || manager.ename || '이다'..  
2 FROM emp worker, emp manager..  
3 WHERE worker.mgr = manager.empno;..
```

```
..  
WORKER.ENAME||'의관리자는'||MANAGER..  
-----..
```

```
BLAKE의 관리자는 KING이다..  
CLARK의 관리자는 KING이다..  
..  
..  
..  
13 rows selected..
```

Self Join

Set Operators

하나 이상의 테이블로부터 자료를 검색하는 또 다른 방법은 SET 연산자를 이용하는 방법이 있다. 즉 SET 연산자를 이용하여 여러 개의 SELECT 문장을 연결하여 작성할 수 있다...

1.8.1) Syntax..

```
SELECT      * | column1[, column2, column3, . . . . ]..
FROM        table1..
. . . . .
SET         operator..
SELECT      * | column1[, column2, column3, . . . . ]..
FROM        table2..
. . . . .
[ORDER BY   column | expression];..
```

[☞] Guidelines ..

- 1) 첫번째 SELECT 구문에서 기술된 열과 두번째 SELECT 구문에서 기술된 열들은 좌측부터 1대1 대응하며 그 개수와 타입이 일치해야 한다...
- 2) FROM 절 뒤에 기술되는 테이블은 같을 수도 있고 다를 수도 있다...
- 3) 출력되는 HARDING 을 첫번째 SELECT 구문에서 기술된 열이 출력된다...
- 4) ORDER BY 는 단 한번만 기술 가능하고 SELECT 구문의 마지막에 기술한다...
- 5) SELECT 문장은 위에서 아래로 수행되고 이를 변경하고자 할 경우는 괄호를 사용한다...

Self Join

Set Operators

하나 이상의 테이블로부터 자료를 검색하는 또 다른 방법은 SET 연산자를 이용하는 방법
· 즉 SET 연산자를 이용하여 여러 개의 SELECT 문장을 연결하여 작성할 수 있다...

1.8.1) Syntax.

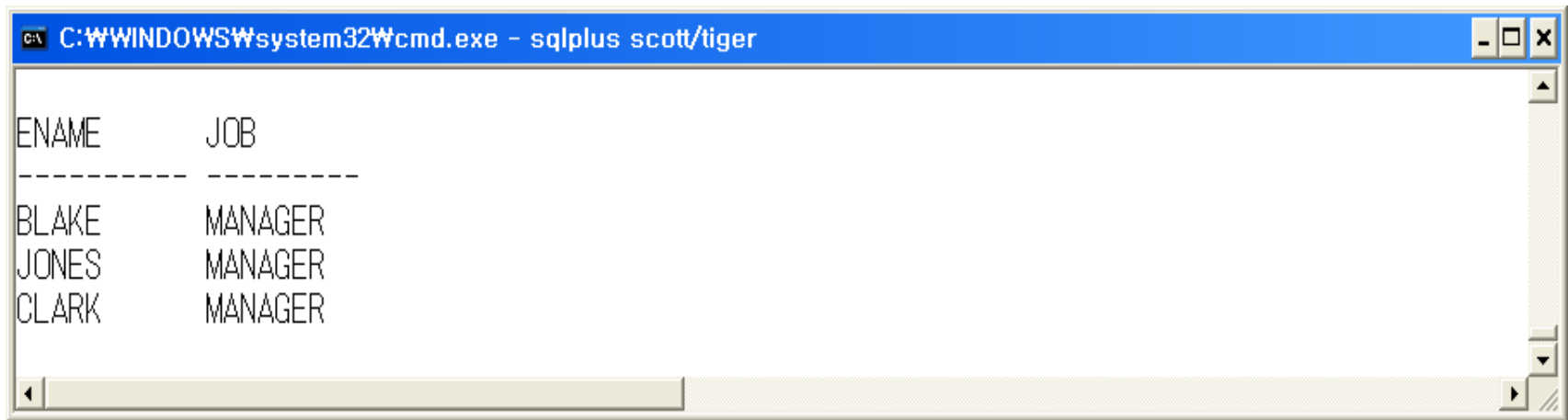
```
SELECT      * | column1[, column2, column3, . . . . . ]  
FROM        table1..  
.  
.  
.  
SET         operator..  
SELECT      * | column1[, column2, column3, . . . . . ]  
FROM        table2..  
.  
.  
.  
[ORDER BY   column | expression];
```

☞ Guidelines |

- 1) 첫번째 SELECT 구문에서 기술된 열과 두번째 SELECT 구문에서 기술된 열들은 좌측부터 1대1 대응하며 그 개수와 타입이 일치해야 한다...
- 2) FROM 절 뒤에 기술되는 테이블은 같을 수도 있고 다를 수도 있다...
- 3) 출력되는 HARDING 을 첫번째 SELECT 구문에서 기술된 열이 출력된다...
- 4) ORDER BY 는 단 한번만 기술 가능하고 SELECT 구문의 마지막에 기술한다...
- 5) SELECT 문장은 위에서 아래로 수행되고 이를 변경하고자 할 경우는 괄호를 사용한다...

연습문제

4. Emp 테이블에서 manager가 'KING'인 직원들의 이름(ename)과 직급(job)을 출력하시오.



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

ENAME      JOB
-----
BLAKE      MANAGER
JONES      MANAGER
CLARK      MANAGER
```

집합 연산

SET 연산자의 종류..

종 류..	설 명..
UNION..	각 결과의 합(합집합: 중복되는 값은 한번 출력) ..
UNION ALL..	각 결과의 합(합집합)..
INTERSET..	각 결과의 중복되는 부분만 출력(교집합)..
MINUS..	첫번째 결과에서 두번째 결과를 뺀(차집합)..

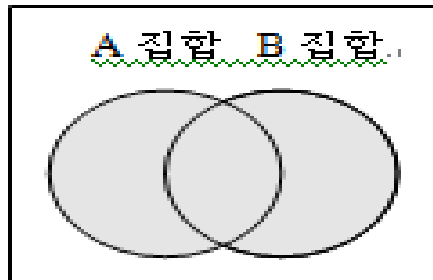
UNION과 UNION ALL 의 차이..

양쪽에서 검색된 결과를 모두 출력한다. 아래의 두개의 SELECT 문장을 참조하여라..

```
SQL> SELECT deptno
2 FROM dept
3 UNION
4 SELECT deptno
5 FROM emp;
```

DEPTNO

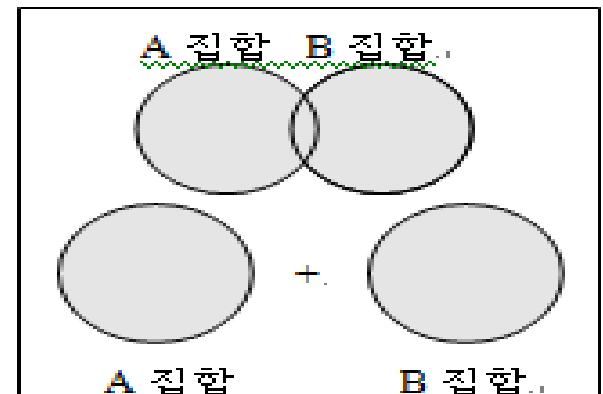
10
20
30
40



```
SQL> SELECT deptno..
2 FROM dept..
3 UNION ALL..
4 SELECT deptno..
5 FROM emp..
```

DEPTNO..

10..
20..
30..
40..
10..
30..



18 rows selected..

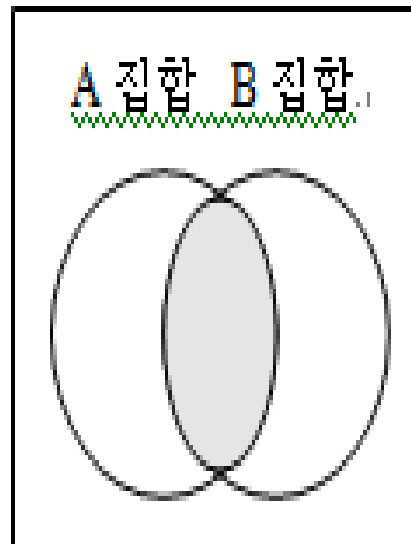
집합 연산

INTERSECT 연산자

양쪽에서 검색된 자료만 출력한다. 아래의 SELECT 문장을 참조하여라...

```
SQL> SELECT deptno.,  
2 FROM dept.,  
3 INTERSECT,  
4 SELECT deptno.,  
5 FROM emp;
```

```
DEPTNO.,  
-----  
10.,  
20.,  
30.,
```



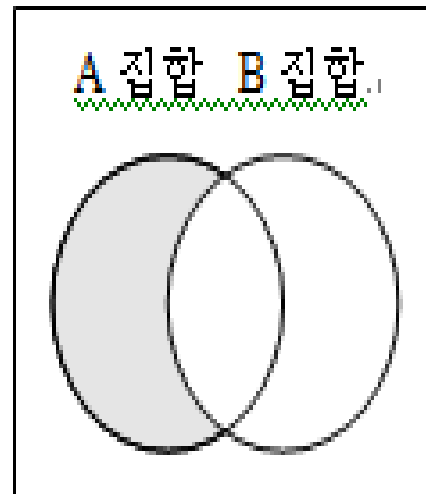
집합 연산

MINUS 연산자

두번째 SELECT 문장에서 검색되지 않았던 값을 첫번째 SELECT 문장에서 출력한다. 즉 첫번째 SELECT 문장에서 두번째 SELECT 문장에의 값을 뺀것을 출력한다. 아래의 SELECT 문장을 참조하여라...

```
SQL> SELECT deptno,  
2 FROM dept,  
3 MINUS  
4 SELECT deptno,  
5 FROM emp;
```

```
DEPTNO,  
-----  
40,
```



집합 연산

집합 연산자의 종류

집합 연산자는 두 개의 쿼리 문장을 결합하며 다음과 같은 연산을 할 수 있다.

집합 연산자

연산자	결과 값
UNION	각 쿼리에 의해서 선택된 결과 중에 중복 행을 제거하고 출력
UNION ALL	각 쿼리에 의해서 선택된 모든 결과를 출력, 중복 허용
INTERSECT	두 개의 쿼리를 모두 만족하는 결과 중에 중복 행을 제거하고 출력
MINUS	첫 번째 쿼리에 의해서 선택된 결과에서 두 번째 쿼리에 의해서 선택된 결과를 제거하고 출력

집합 연산자의 사용방법을 학습하기 위해 다음과 같은 임시 테이블을 작성하자.

```
CREATE TABLE A(VAL CHAR);
INSERT INTO A VALUES ('A');
INSERT INTO A VALUES ('B');
INSERT INTO A VALUES ('C');
INSERT INTO A VALUES ('D');
INSERT INTO A VALUES ('E');

CREATE TABLE B(NUM NUMBER, VAL CHAR);
INSERT INTO B VALUES (1, 'C');
INSERT INTO B VALUES (2, 'D');
INSERT INTO B VALUES (3, 'E');
INSERT INTO B VALUES (4, 'F');
INSERT INTO B VALUES (5, 'G');
```

ANSI SQL(1999) Join

- Left Outer Join

```
SELECT e.ename, e.deptno, d.dname  
FROM emp e LEFT OUTER JOIN dept d  
ON e.deptno = d.deptno;
```

Right Outer Join

```
SELECT e.ename, e.deptno, d.dname  
FROM emp e RIGHT OUTER JOIN dept d  
ON e.deptno = d.deptno;
```

Full Outer Join

```
SELECT e.ename, e.deptno, d.dname  
FROM emp e FULL OUTER JOIN dept d  
ON e.deptno = d.deptno;
```

ANSI SQL(1999) JOIN

- Natural Join

```
select deptno, dname, ename, sal  
from emp natural join dept;
```

natural join 의 조건

1. 두 테이블의 공통 컬럼이 반드시 1개만 있어야 가능.
2. 알아서 공통 컬럼을 찾으므로 테이블 alias를 주면 오류.

Using 절 사용

```
select deptno, dname, ename, sal  
from emp join dept  
Using (deptno)
```

다양한 Outer Join

1. DEPT 테이블과 닮은 DEPT01 테이블을 만들어 보겠습니다. 혹시 DEPT01 테이블이 존재한다면 DEPT01 테이블이 생성되지 않으므로 DROP 명령어로 삭제 후 생성하도록 합니다.

```
DROP TABLE DEPT01;
```

2. 부서번호와 부서명을 컬럼으로 갖는 DEPT01 테이블을 생성합니다.

```
CREATE TABLE DEPT01(  
  DEPTNO NUMBER(2),  
  DNAME VARCHAR2(14));
```

1. 데이터를 추가합니다.

```
INSERT INTO DEPT01 VALUES(10, 'ACCOUNTING');  
INSERT INTO DEPT01 VALUES (20, 'RESEARCH');
```


다양한 Outer Join

4. 동일한 방법으로 DEPT02 테이블을 생성합니다.

```
DROP TABLE DEPT02;
```

```
CREATE TABLE DEPT02(  
  DEPTNO NUMBER(2),  
  DNAME VARCHAR2(14));
```

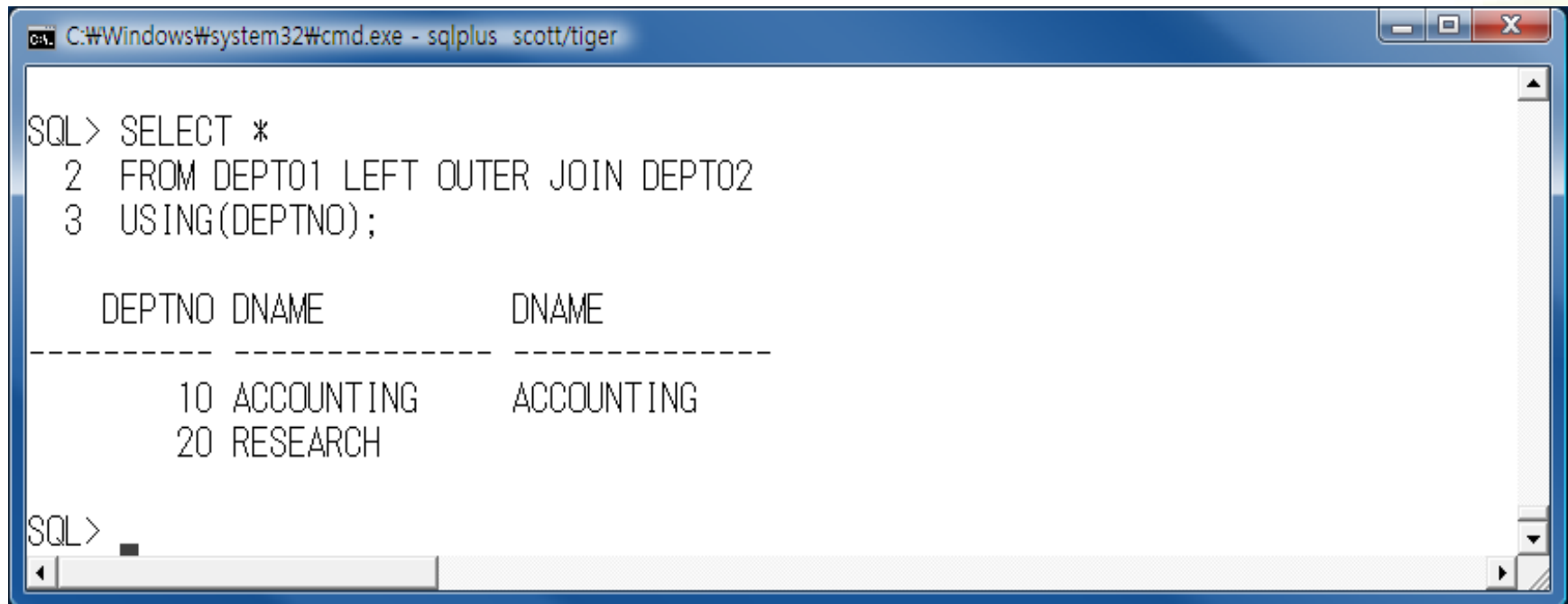
```
INSERT INTO DEPT02 VALUES(10, 'ACCOUNTING');  
INSERT INTO DEPT02 VALUES (30, 'SALES');
```

```
SELECT * FROM DEPT02;
```

LEFT OUTER JOIN

- DEPT01 테이블의 20번 부서와 조인할 부서번호가 DEPT02에는 없지만, 20번 부서도 출력되도록 하기 위해서 DEPT01 테이블이 왼쪽에 존재하기에 LEFT OUTER JOIN을 사용합시다.

```
SELECT *  
FROM DEPT01 LEFT OUTER JOIN DEPT02  
ON DEPT01.DEPTNO = DEPT02.DEPTNO;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". Inside the window, the following SQL query is entered and executed:

```
SQL> SELECT *  
2 FROM DEPT01 LEFT OUTER JOIN DEPT02  
3 USING(DEPTNO);
```

The output of the query is displayed as follows:

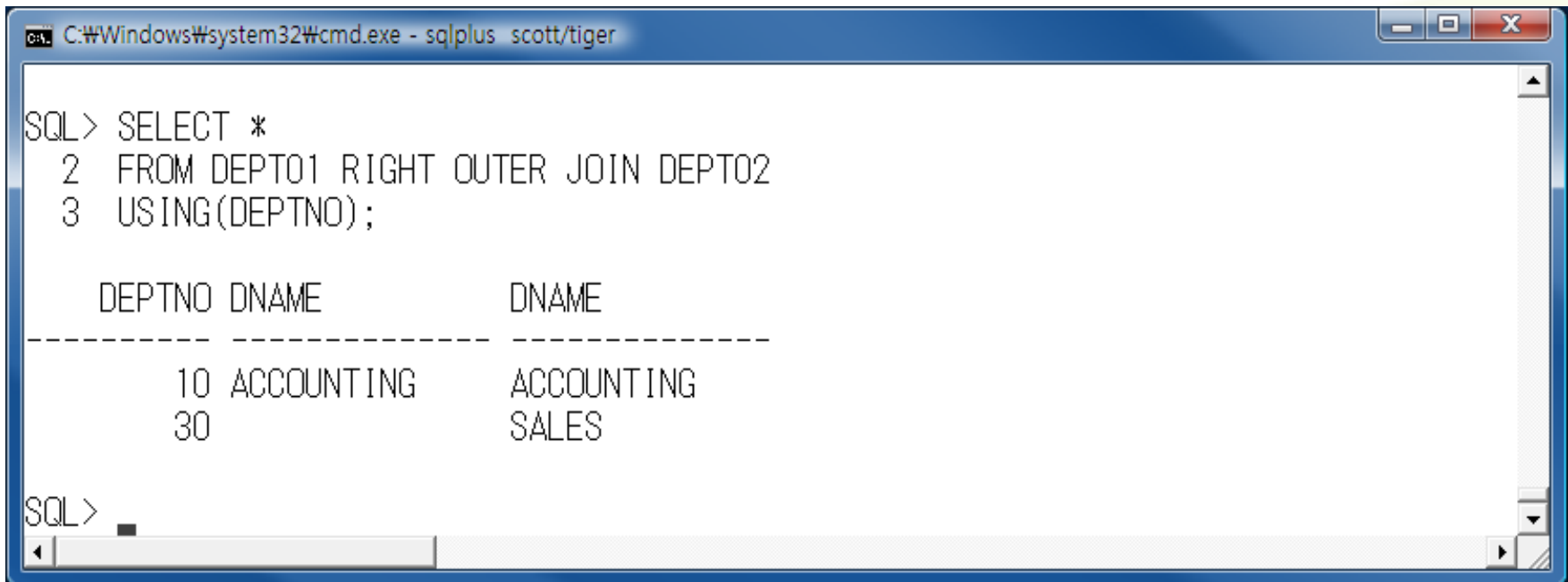
DEPTNO	DNAME	DNAME
10	ACCOUNTING	ACCOUNTING
20	RESEARCH	

The command prompt shows the prompt "SQL>" at the bottom left, indicating the query has been executed.

RIGHT OUTER JOIN

- DEPT02 테이블에만 있는 30번 부서까지 출력되도록 하기 위해서 RIGHT OUTER JOIN을 사용합시다.

```
SELECT *  
FROM DEPT01 RIGHT OUTER JOIN DEPT02  
USING(DEPTNO);
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The user has entered the following SQL query:

```
SQL> SELECT *  
2 FROM DEPT01 RIGHT OUTER JOIN DEPT02  
3 USING(DEPTNO);
```

The query result is displayed as a table with three columns: DEPTNO, DNAME, and DNAME. The output shows two rows of data, corresponding to department 10 (ACCOUNTING) and department 30 (SALES).

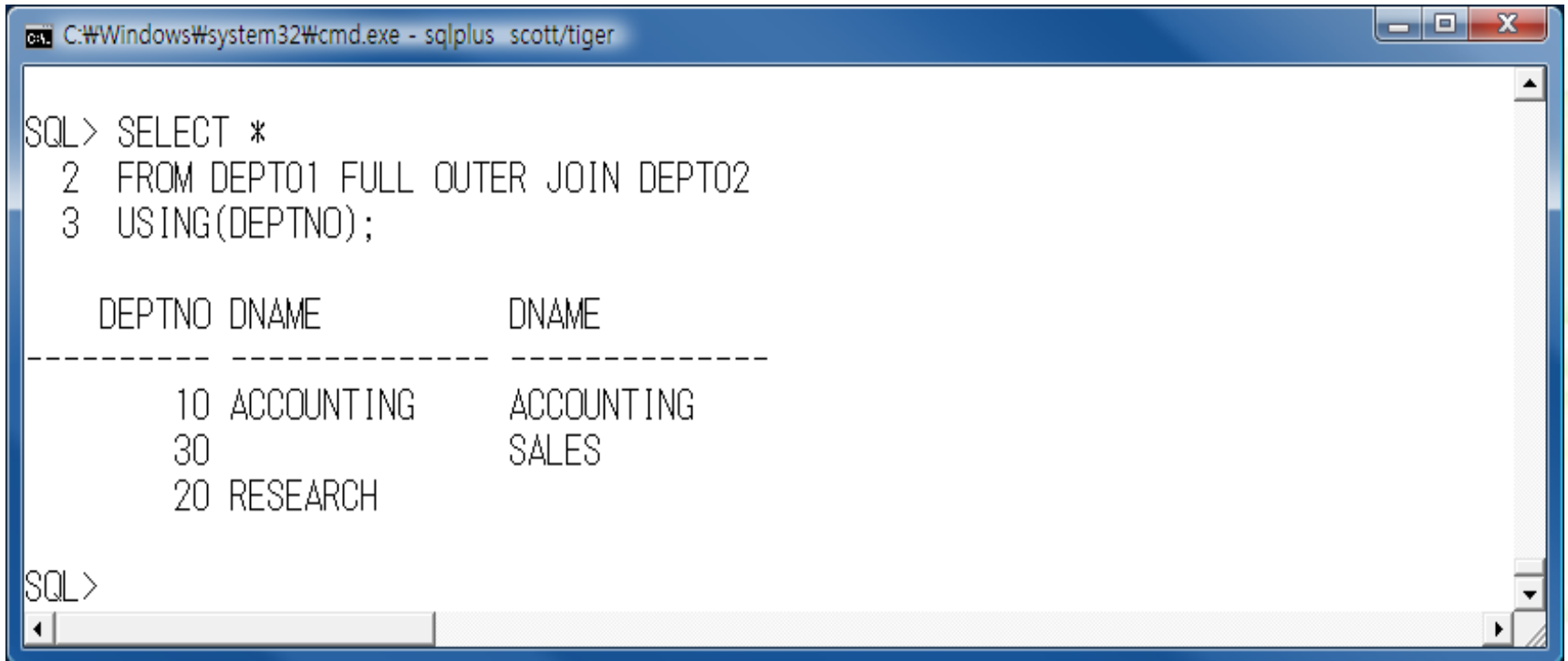
DEPTNO	DNAME	DNAME
10	ACCOUNTING	ACCOUNTING
30		SALES

The command prompt shows the prompt "SQL>" at the bottom left, indicating the query has been executed.

FULL OUTER JOIN

- FULL OUTER JOIN은 LEFT OUTER JOIN, RIGHT OUTER JOIN 을 합한 형태라고 볼 수 있습니다.

```
SELECT *  
FROM DEPT01 FULL OUTER JOIN DEPT02  
USING(DEPTNO);
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The user has entered the following SQL query:

```
SQL> SELECT *  
2 FROM DEPT01 FULL OUTER JOIN DEPT02  
3 USING(DEPTNO);
```

The query results are displayed in a table format:

DEPTNO	DNAME	DNAME
10	ACCOUNTING	ACCOUNTING
30		SALES
20	RESEARCH	

The command prompt shows the "SQL>" prompt at the bottom left, and the window has standard Windows window controls (minimize, maximize, close) at the top right.

연습 문제

1. EMP 테이블에서 모든 사원에 대한 이름, 부서번호, 부서명을 출력하는 SELECT 문장을 작성하여라.
2. EMP 테이블에서 NEW YORK에서 근무하고 있는 사원에 대하여 이름, 업무, 급여, 부서명을 출력하는 SELECT 문장을 작성하여라.
3. EMP 테이블에서 보너스를 받는 사원에 대하여 이름, 부서명, 위치를 출력하는 SELECT 문장을 작성하여라.
4. EMP 테이블에서 이름 중 L자가 있는 사원에 대하여 이름, 업무, 부서명, 위치를 출력하는 SELECT 문장을 작성하여라.
5. 사번, 사원명, 부서코드, 부서명을 검색하라. 사원명기준으로 오름차순정렬
6. 사번, 사원명, 급여, 부서명을 검색하라. 단 급여가 2000이상인 사원에 대하여 급여를 기준으로 내림차순으로 정렬하시오
7. 사번, 사원명, 업무, 급여, 부서명을 검색하시오. 단 업무가 MANAGER이며 급여가 2500이상인 사원에 대하여 사번을 기준으로 오름차순으로 정렬하시오.
8. 사번, 사원명, 업무, 급여, 등급을 검색하시오. 등급은 급여가 하한값과 상한값 범위에 포함되고 급여기준 내림차순으로 정렬하시오
9. 사원테이블에서 사원명, 사원의 관리자명을 검색하시오
10. 사원명, 관리자명, 관리자의 관리자명을 검색하시오
11. 위의 결과에서 상위 관리자가 없는 모든 사원의 이름도 사원명에 출력되도록 수정하시오