



15강.oracle

- OUT JOIN

1 out Join

목차

1 Outer Join

2 ANSI Join

1 out Join

목차

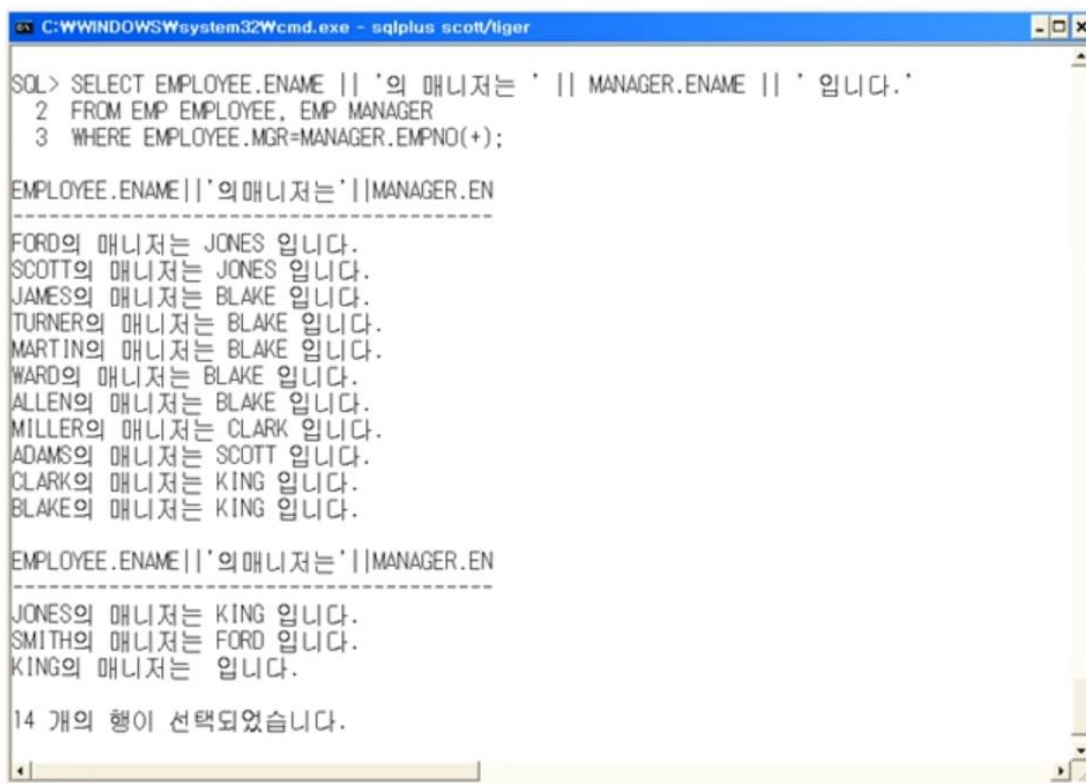
- ❖ Self Join을 학습하면 특정 사원의 매니저 이름을 구했습니다.
- ❖ 결과를 꼼꼼히 살펴보면 이름이 KING인 사원 한사람의 정보가 빠져 있음을 확인할 수 있습니다.
- ❖ KING은 이 회사의 사장(PRESIDENT)으로 매니저가 존재하지 않으므로 MGR 컬럼 값이 NULL 입니다. 사원 번호(EMPNO)가 NULL인 사원은 없으므로 조인 조건에 만족하지 않아서 KING은 Self Join의 결과에서 배제되었습니다.
- ❖ 조인 조건에 만족하지 못하였더라도 해당 로우를 나타내고 싶을 때에 사용하는 것이 외부 조인(Outer Join)입니다.
- ❖ 외부 조인은 NULL 값이기에 배제된 행을 결과에 포함시킬 수 있으며, “(+)” 기호를 조인 조건에서 정보가 부족한 컬럼 이름 뒤에 덧붙입니다.
- ❖ 사원 번호(EMPNO)가 NULL인 사원은 없으므로 manager.empno 뒤에 “(+)” 기호를 덧붙입니다.

1 out Join

매니저가 null 인 경우도 출력

예

```
SELECT employee.ename || '의 매니저는 '
       || manager.ename || '입니다.'
FROM emp employee, emp manager
WHERE employee.mgr = manager.empno(+);
```



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

SQL> SELECT EMPLOYEE.ENAME || '의 매니저는 ' || MANAGER.ENAME || '입니다.'
2  FROM EMP EMPLOYEE, EMP MANAGER
3  WHERE EMPLOYEE.MGR=MANAGER.EMPNO(+);

EMPLOYEE.ENAME||'의매니저는' ||MANAGER.EN
-----
FORD의 매니저는 JONES 입니다.
SCOTT의 매니저는 JONES 입니다.
JAMES의 매니저는 BLAKE 입니다.
TURNER의 매니저는 BLAKE 입니다.
MARTIN의 매니저는 BLAKE 입니다.
WARD의 매니저는 BLAKE 입니다.
ALLEN의 매니저는 BLAKE 입니다.
MILLER의 매니저는 CLARK 입니다.
ADAMS의 매니저는 SCOTT 입니다.
CLARK의 매니저는 KING 입니다.
BLAKE의 매니저는 KING 입니다.

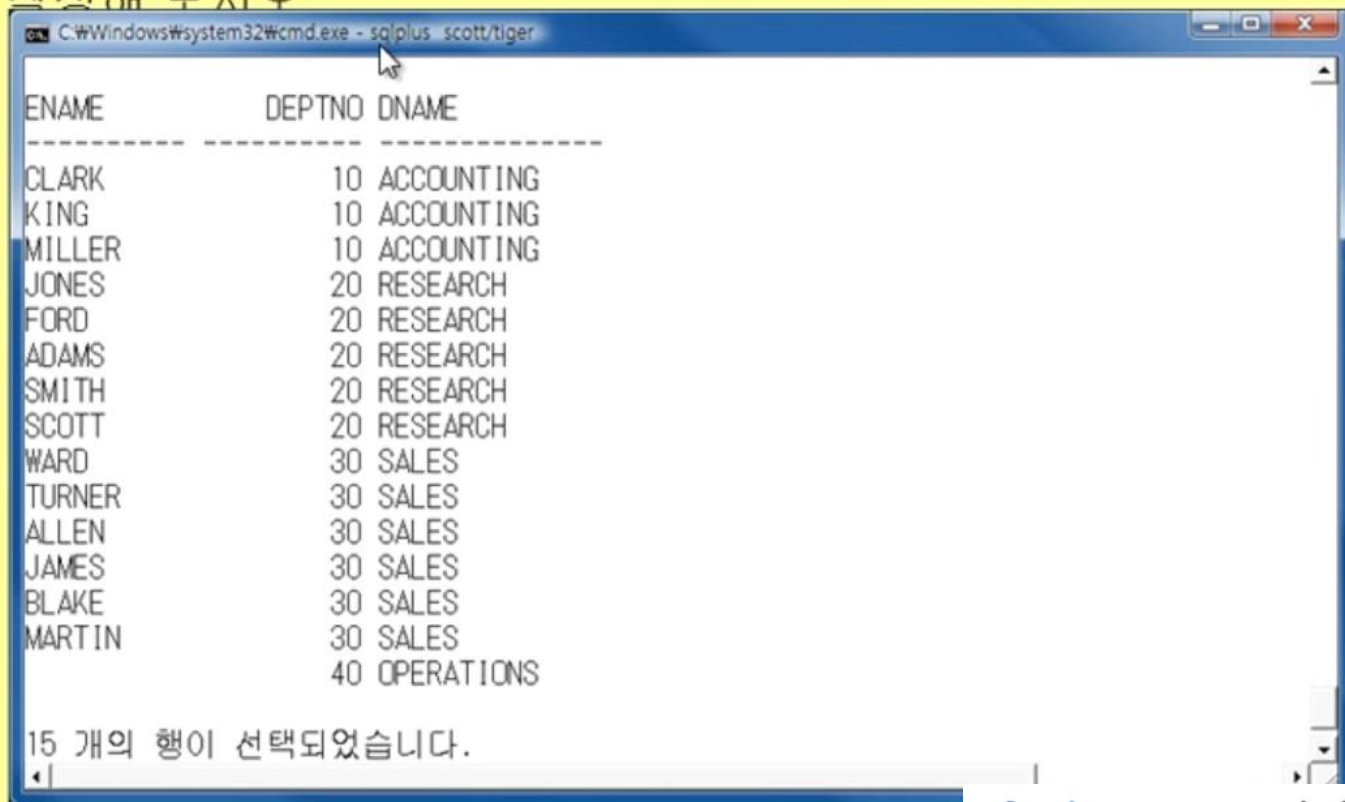
EMPLOYEE.ENAME||'의매니저는' ||MANAGER.EN
-----
JONES의 매니저는 KING 입니다.
SMITH의 매니저는 FORD 입니다.
KING의 매니저는  입니다.

14 개의 행이 선택되었습니다.
```

1 실습.

아래와 같이 출력하시오.

6. 사원 테이블과 부서 테이블을 조인하여 사원이름과 부서번호와 부서명을 출력하도록 합시다. 부서 테이블의 40번 부서와 조인할 사원 테이블의 부서번호가 없지만, 아래 그림과 같이 40번 부서의 부서 이름도 출력되도록 쿼리문을 작성해 보시오



ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
KING	10	ACCOUNTING
MILLER	10	ACCOUNTING
JONES	20	RESEARCH
FORD	20	RESEARCH
ADAMS	20	RESEARCH
SMITH	20	RESEARCH
SCOTT	20	RESEARCH
WARD	30	SALES
TURNER	30	SALES
ALLEN	30	SALES
JAMES	30	SALES
BLAKE	30	SALES
MARTIN	30	SALES
	40	OPERATIONS

15 개의 행이 선택되었습니다.

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


1 ANSI Join

ANSI Cross Join

```
SELECT *  
FROM EMP CROSS JOIN DEPT;
```

```
select * from emp, dept;
```

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

SOL> SELECT *
2 FROM EMP CROSS JOIN 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									
7521	WARD									
7566	JONES									
7654	MARTIN									
7698	BLAKE									
7782	CLARK									
7788	SCOTT									
7839	KING									
7844	TURNER									
7876	ADAMS									
7900	JAMES									
7902	FORD									
7934	MILLER									
7369	SMITH									

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

7934	MILLER	CLERK	7782	82/01/23	1300		10	30	SALES	CHICAGO
7369	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.

1 ANSI Join

ANSI Inner Join

- ❖ 새로운 ANSI 조인은 FROM 다음에 INNER JOIN 이란 단어를 사용하여 조인할 테이블 이름을 명시하고 ON 절을 사용하여 조인 조건을 명시하여 다음과 같이 작성합니다.

```
SELECT * FROM table1 INNER JOIN table2  
ON table1.column1 = table2.column2
```

- ❖ ANSI 조인에서는 조인 정보를 ON절에 기술하여 조인 조건을 명확하게 지정하고, 다른 조건에 대해서는 WHERE 구문에서 지정하면 됩니다.

```
SELECT ENAME, DNAME  
FROM EMP INNER JOIN DEPT  
ON EMP.DEPTNO=DEPT.DEPTNO  
WHERE ENAME='SCOTT';
```

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

1 ANSI Join

ANSI Inner Join - using

❖ USING을 이용한 조인 조건 지정하기

- 두 테이블에 각각 조인을 정의한 컬럼의 이름이 동일하다면 USING 절에서 조인할 컬럼을 지정하여 구문을 더 간단하게 표현할 수 있습니다.

```
SELECT * FROM table1 JOIN table2  
USING (공통컬럼)
```

- EMP와 DEPT에 DEPTNO 라는 같은 이름의 컬럼이 있기 때문에 다음과 같이 간단하게 조인문을 기술할 수 있습니다.

```
SELECT EMP.ENAME, DEPT.DNAME  
FROM EMP INNER JOIN DEPT  
USING (DEPTNO);
```

```
select ename, dname FROM emp inner join dept  
on emp.deptno = dept.deptno ;
```


1 ANSI Join

ANSI Inner Join – natural join

❖ NATURAL Join

- 두 테이블에 각각 조인을 정의한 컬럼의 이름이 동일하다면 USING 절에서 조인할 컬럼을 지정하여 구문을 더 간단하게 표현할 수 있습니다.

```
SELECT * FROM table1 NATURAL JOIN table2
```

- EMP와 DEPT에 DEPTNO 라는 같은 이름의 컬럼이 있기 때문에 다음과 같이 간단하게 조인문을 기술할 수 있습니다.

```
SELECT EMP.ENAME, DEPT.DNAME  
FROM EMP NATURAL JOIN DEPT;
```

1 ANSI Join

ANSI outer join

- ❖ 새로운 ANSI 구문에서 Outer Join은 LEFT Outer Join, RIGHT Outer Join 그리고 FULL Outer Join 세 가지 타입의 조인을 제공합니다.

```
SELECT * FROM table1  
[LEFT | RIGHT | FULL] Outer Join table2
```

- ❖ Outer Join은 이미 설명했듯이 어느 한쪽 테이블에는 해당하는 데이터가 존재하는데 다른 쪽 테이블에는 데이터가 존재하지 않을 경우
그 데이터가 출력되지 않는 문제점을 해결하기 위해 사용하는 조인 기법입니다.

1 실습.

아래와 같이 출력하시오.

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

```
DROP TABLE DEPT01;
```

2. 부서번호와 부서명을 컬럼으로 갖는 DEPT01 테이블을 생성합니다.

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

3. 데이터를 추가합니다.

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

아래와 같이 출력하시오.

4. 동일한 방법으로 DEPT02 테이블을 생성합니다.

```
DROP TABLE DEPT01;
```

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

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

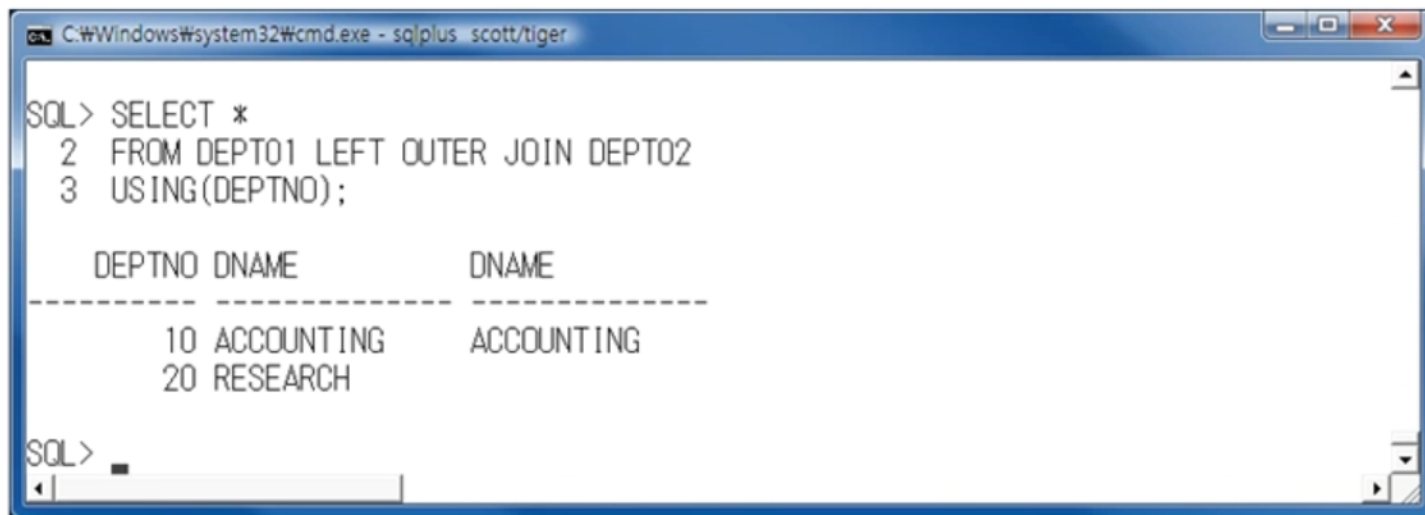
```
SELECT * FROM DEPT02;
```

1 ANSI Join

Left outer join

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

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



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT *  
2 FROM DEPT01 LEFT OUTER JOIN DEPT02  
3 USING(DEPTNO);  
  
DEPTNO DNAME DNAME  
-----  
10 ACCOUNTING ACCOUNTING  
20 RESEARCH  
  
SQL>
```


1 ANSI Join

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 SQL*Plus window with the following content:

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT *  
2 FROM DEPT01 RIGHT OUTER JOIN DEPT02  
3 USING(DEPTNO);  
  
DEPTNO DNAME      DNAME  
-----  
10 ACCOUNTING    ACCOUNTING  
30              SALES  
  
SQL>
```

DEPTNO	DNAME	DNAME
10	ACCOUNTING	ACCOUNTING
30		SALES

1 ANSI Join

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". Inside the window, the following SQL query is entered and executed:

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

The output of the query is displayed as follows:

DEPTNO	DNAME	DNAME
10	ACCOUNTING	ACCOUNTING
30		SALES
20	RESEARCH	

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

THANK YOU