

# ch08. 여러 테이블을 하나의 테이블처럼 사용하는 조인

## 08-1. 조인 ( join )

조인은 두 개 이상의 테이블을 연결하여 하나의 테이블처럼 출력할 때 사용하는 방식

집합 연산자를 사용한 결과는 두 개 이상 SELECT문의 결과 값을 세로로 연결한 것이고, 조인을 사용한 결과는 두 개 이상의 테이블 데이터를 가로로 연결한 것이라고 볼 수 있다.

FROM절을 사용할 때는 하나의 테이블만 불러올 수 있는 게 아니라 여러 개도 지정 가능하다.

조인을 사용한 데이터를 뽑을 때 출력 행을 선정하는 조건식을 명시하는 WHERE절이 중요한 역할을 한다.

테이블 이름.열 이름

```
SELECT *  
  FROM EMP, DEPT  
 WHERE EMP.DEPTNO = DEPT.DEPTNO  
 ORDER BY DEPTNO;
```

### 테이블의 별칭 설정

FROM절에 지정한 테이블에는 SELECT절의 열에 사용한 것처럼 별칭 지정할 수 있다.

```
FROM 테이블이름1 별칭1, 테이블이름2 별칭2 ...
```

## 08-2. 조인 종류

### 등가 조인

등가 조인 ( equi join ) 은 테이블을 연결한 후에 출력 행을 각 테이블의 특정 열에 일치한 데이터를 기준으로 선정하는 방식

등가조인은 내부 조인 ( inner join ) 또는 단순 조인 ( simple join ) 으로 부르기도 한다.

- 여러 테이블의 열 이름이 같을 때 유의점

등가 조인을 사용할 때 조인 조건이 되는 각 테이블의 열 이름이 같을 경우 해당 열 이름을 테이블 구분 없이 명시하면 오류가 발생한다.

하여 어느 테이블에 속해 있는 열인지 명시를 해야한다.

```
SELECT E.EMPNO, E.ENAME, D.DEPTNO, D.DNAME, D.LOC  
FROM EMP E, DEPT D  
WHERE E.DEPTNO = D.DEPTNO  
ORDER BY D.DEMPNO, E.EMPNO
```

## 비등가 조인

비등가 조인 ( non-equi join ) 은 등가 조인 방식 외의 방식

등가 조인 방식에 비해 자주 쓰이는 건 아니지만 조인 조건이 특정 열의 일치 여부를 검사하는 방식 외에 다른 방식도 사용할 수 있다는 것

## 자체 조인

하나의 테이블을 여러 개의 테이블처럼 활용하여 조인하는 방식으로 앞에서 물리적으로 동일한 테이블 여러 개를 사용할 때 발생할 수 있는 문제점을 해결한다.

- FROM절에 같은 테이블을 여러 번 명시하되 테이블 별칭만 다르게 지정하는 방식으로 사용.

## 외부 조인

두 테이블간 조인 수행에서 조인 기준 열의 어느 한쪽이 NULL이어도 강제로 출력하는 방식.

- 외부 조인은 좌우로 따로 나누어 지정하는데 WHERE절에 조인 기준 열 중 한쪽에 + 기호를 붙여준다.

```

SELECT E1.EMPNO, E1.ENAME, E1.MGR,
       E2.EMPNO AS MGR_EMPNO,
       E2.ENAME AS MGR_ENAME
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO(+)
ORDER BY E1.EMPNO;

```

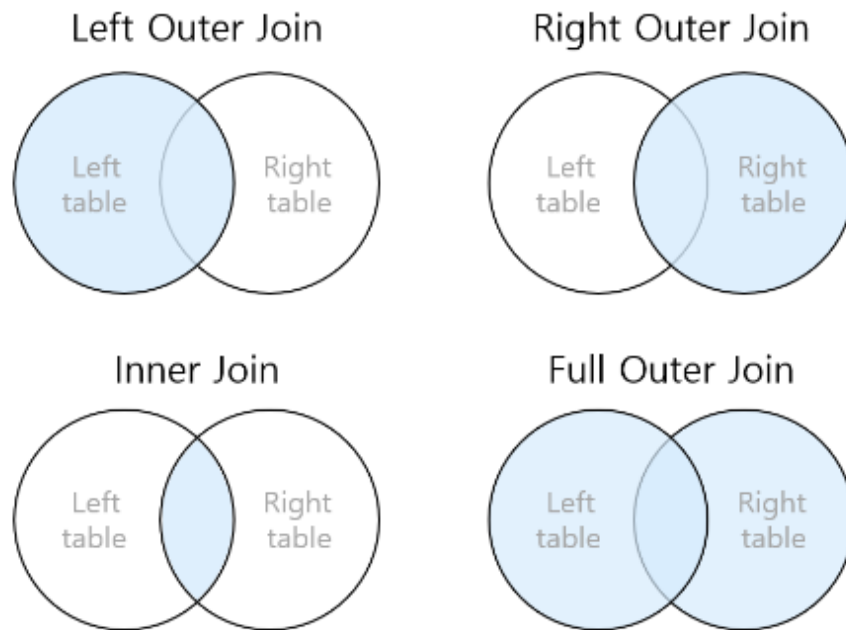
- 왼쪽 열을 기준으로 오른쪽 열의 데이터 존재 여부에 상관없이 출력하라는 뜻이다.
- 따라서 SELECT문의 WHERE절 조건식에서 KING은 E1.MGR이 NULL로서 E2.EMPNO와 일치한 데이터가 없음에도 출력한다.

```

SELECT E1.EMPNO, E1.ENAME, E1.MGR,
       E2.EMPNO AS MGR_EMPNO,
       E2.ENAME AS MGR_ENAME
FROM EMP E1, EMP E2
WHERE E1.MGR(+) = E2.EMPNO
ORDER BY E1.EMPNO;

```

- 오른쪽 열을 기준으로 왼쪽 열 데이터의 존재와 상관없이 데이터를 출력하라는 뜻이다.



조인(join)의 벤 다이어그램

## 08-3. SQL-99 표준 문법으로 배우는 조인

### NATURAL JOIN

등가 조인을 대신해 사용할 수 있는 조인 방식

조인 대상이 되는 두 테이블에 이름과 자료형이 같은 열을 찾은 후 그 열을 기준으로 등가 조인을 해주는 방식.

```
SELECT E.EMPNO, E.ENAME, E.JOB, E.MGR, E.HIREDATE, E.SAL, E.COMM,
       DEPTNO, D.DNAME, D.LOC
FROM EMP E NATURAL JOIN DEPT D
ORDER BY DEPTNO, E.EMPNO;
```

- 기존 등가 조인과는 다르게 조인 기준 열인 DEPTNO를 SELECT절에 명시할 때 테이블 이름을 붙이면 안 되는 특성이 있다.

## JOIN ~ USING

NATURAL JOIN이 자동으로 조인 기준 열을 지정하는 것과 달리 USING 키워드에 조인 기준으로 사용할 열을 명시하여 사용.

```
FROM TABLE1 JOIN TABLE2 USING ( 조인에 사용한 기준열 )
```

```
SELECT E.EMPNO, E.ENAME, E.JOB, E.MGR, E.HIREDATE, E.SAL, E.COMM,  
       DEPTNO, D.DNAME, D.LOC  
FROM EMP E JOIN DEPT D USING (DEPTNO)  
WHERE SAL >= 3000  
ORDER BY DEPTNO, E.EMPNO;
```

## JOIN ~ ON

가장 범용성 있는 JOIN ~ ON 키워드를 사용한 조인 방식에서는 기존 WHERE절에 있는 조인 조건식을 ON 키워드 옆에 작성한다. 조인 기준 조건식은 ON에 명시하고 그 밖에 출력행을 걸러 내기 위해 WHERE 조건식을 따로 사용하는 방식이다.

```
FROM TABLE1 JOIN TABLE2 ON ( 조인 조건식 )
```

```
SELECT E.EMPNO, E.ENAME, E.JOB, E.MGR, E.HIREDATE, E.SAL, E.COMM,  
       E.DEPTNO,  
       D.DNAME, D.LOC  
FROM EMP E JOIN DEPT D ON (E.DEPTNO = D.DEPTNO)  
WHERE SAL <= 3000  
ORDER BY E.DEPTNO, EMPNO;
```

## OUTER JOIN

외부요인에 사용하며 WHERE절이 아닌 FROM절에서 외부 조인을 선언.

### 기존 방식과의 차이점

- 왼쪽 외부 조인

- 기존 : WHERE TABLE1.COL1 = TABLE2.COL1(+)
- SQL-99 : FROM TABLE1 LEFT OUTER JOIN TABLE2 ON (조인 조건식)

```
SELECT E1.EMPNO, E1.ENAME, E1.MGR,
       E2.EMPNO AS MGR_EMPNO,
       E2.ENAME AS MGR_ENAME
FROM EMP E1 LEFT OUTER JOIN EMP E2 ON (E1.MGR = E2.EMPNO)
ORDER BY E1.EMPNO;
```

- 오른쪽 외부 조인

- 기존 : WHERE TABLE1.COL1(+) = TABLE2.COL1
- SQL-99 : FROM TABLE1 RIGHT OUTER JOIN TABLE2 ON (조인 조건식)

```
SELECT E1.EMPNO, E1.ENAME, E1.MGR,
       E2.EMPNO AS MGR_EMPNO,
       E2.ENAME AS MGR_ENAME
FROM EMP E1 RIGHT OUTER JOIN EMP E2 ON (E1.MGR = E2.EMPNO)
ORDER BY E1.EMPNO;
```

- 전체 외부 조인

- 기존 : 기존 문법은 없는 (UNION 집합 연산자를 활용)
- SQL-99 : FROM TABLE1 FULL OUTER JOIN TABLE2 ON (조인 조건식)

```
SELECT E1.EMPNO, E1.ENAME, E1.MGR,
       E2.EMPNO AS MGR_EMPNO,
       E2.ENAME AS MGR_ENAME
FROM EMP E1 FULL OUTER JOIN EMP E2 ON (E1.MGR = E2.EMPNO)
ORDER BY E1.EMPNO;
```

## SQL-99 조인 방식에서 세 개 이상의 테이블을 조인할 때

FROM절에 두 개 테이블을 키워드로 조인한 바로 옆에 SQL-99 방식의 조인 내용을 추가로 작성.

```
FROM TABLE1 JOIN TABLE2 ON (조건식)
JOIN TABLE2 ON (조건식)
```

## | Q

1.

```
SELECT D.DEPTNO, D.DNAME, E.EMPNO, E.ENAME, E.SAL
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO
AND E.SAL > 2000;
```

```
SELECT DEPTNO, D.DNAME, E.EMPNO, E.ENAME, E.SAL
FROM EMP E NATURAL JOIN DEPT D
WHERE E.SAL > 2000;
```

2.

```
SELECT D.DEPTNO,
       D.DNAME,
       TRUNC(AVG(SAL)) AS AVG_SAL,
       MAX(SAL) AS MAX_SAL,
       MIN(SAL) AS MIN_SAL,
       COUNT(*) AS CNT
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO
GROUP BY D.DEPTNO, D.DNAME;
```

```
SELECT DEPTNO,
       D.DNAME,
       TRUNC(AVG(SAL)) AS AVG_SAL,
       MAX(SAL) AS MAX_SAL,
       MIN(SAL) AS MIN_SAL,
       COUNT(*) AS CNT
FROM EMP E JOIN DEPT D USING (DEPTNO)
GROUP BY DEPTNO, D.DNAME;
```

3.

```
SELECT D.DEPTNO, D.DNAME, E.EMPNO, E.ENAME, E.JOB, E.SAL
FROM EMP E, DEPT D
```

```
WHERE E.DEPTNO(+) = D.DEPTNO
ORDER BY D.DEPTNO, E.ENAME;
```

```
SELECT D.DEPTNO, D.DNAME, E.EMPNO, E.ENAME, E.JOB, E.SAL
FROM EMP E RIGHT OUTER JOIN DEPT D ON (E.DEPTNO = D.DEPTNO)
ORDER BY D.DEPTNO, E.ENAME;
```

4.

```
SELECT D.DEPTNO, D.DNAME,
       E.EMPNO, E.ENAME, E.MGR, E.SAL, E.DEPTNO,
       S.LOSAL, S.HISAL, S.GRADE,
       E2.EMPNO AS MGR_EMPNO, E2.ENAME AS MGR_ENAME
FROM EMP E, DEPT D, SALGRADE S, EMP E2
WHERE E.DEPTNO(+) = D.DEPTNO
      AND E.SAL BETWEEN S.LOSAL(+) AND S.HISAL(+)
      AND E.MGR = E2.EMPNO(+)
ORDER BY D.DEPTNO, E.EMPNO;
```

```
SELECT D.DEPTNO, D.DNAME,
       E.EMPNO, E.ENAME, E.MGR, E.SAL, E.DEPTNO,
       S.LOSAL, S.HISAL, S.GRADE,
       E2.EMPNO AS MGR_EMPNO, E2.ENAME AS MGR_ENAME
FROM EMP E RIGHT OUTER JOIN DEPT D
      ON (E.DEPTNO = D.DEPTNO)
LEFT OUTER JOIN SALGRADE S
      ON (E.SAL BETWEEN S.LOSAL AND S.HISAL)
LEFT OUTER JOIN EMP E2
      ON (E.MGR = E2.EMPNO)
ORDER BY D.DEPTNO, E.EMPNO;
```