**[ III ] join : 2개 이상의 테이블을 연결하여 데이터를 검색하는 방법**

1. Join의 필요성
   * 특정부서번호에 대한 부서이름은 무엇인지 dept 테이블에 있고 특정 사원에 대한 부서명을 알아내기 위해서는 부서 테이블에서 정보를 얻어와야 합니다. Join(두 개 이상의 테이블을 결합) 을 하지 않는 경우 다음의 두 SQL문을 통해 알아내야 하는 번거로움이 있다.
2. Cross Join
   * Cross Join은 특별한 키워드 없이 FROM절에 두 개 이상의 테이블을 기술하는 것.
   * Cross Join의 결과 얻어지는 컬럼의 수는 사원 테이블의 컬럼의 수(8)와 부서 테이블의 컬럼의 수(3)를 더한 것이므로 11이 됩니다. 로우 수는 사원 한 명에 대해서 DEPT 테이블의 4개의 로우와 결합되기에 56개(14\*4).
   * Cross Join의 결과를 보면 사원 테이블에 부서에 대한 상세정보가 결합되긴 했지만, 조인 될 때 아무런 조건을 제시하지 않았기에 사원 한명에 대해서 DEPT 테이블의 4개의 로우와 결합된 형태이기에 Cross Join의 결과는 아무런 의미를 갖지 못 한다.
   * 조인 결과가 의미를 갖으려면 조인할 때 조건을 지정. 조인 조건에 따라 조인의 종류가 결정.
     + **Equi Join : 동일한 컬럼을 기준으로 조인**
     + **Non-Equi Join : 동일한 컬럼없이 다른 조건을 사용하여 조인**
     + **Outer Join : 조인 조건에 만족하지 않는 행도 나타나는 조인**
     + **Self Join : 한 테이블 내에서 조인.**
3. Equi Join
   * EQUI JOIN은 가장 많이 사용하는 조인 방법으로서 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치되는 행을 연결하여 결과를 생성하는 조인 방법
   * 다음은 사원 정보를 출력할 때 각 사원들이 소속된 부서의 상세 정보를 출력하기 위해서 두 개의 테이블을 조인한 예
     + SELECT \* FROM emp, dept WHERE emp.deptno=dept.deptno;
   * 사원(EMP) 테이블과 부서(DEPT) 테이블의 공통 컬럼인 DEPTNO의 값이 일치(=)되는 조건을 WHERE 절에 기술하여 사용.
   * 테이블을 조인하려면 일치되는 공통 컬럼을 사용해야 하고 컬럼의 이름이 같게 되면 혼동이 오기 때문에 컬럼 이름 앞에 테이블 이름을 기술.
   * 이름이 SCOTT인 사람의 부서명을 출력해 봅시다
     + SELECT ename, dname FROM emp, dept

WHERE emp.deptno=dept.deptno AND ename=’SCOTT’;

* + 사원이름, 부서이름, 부서번호를 출력하기 위해 다음의 SQL문을 수행하는 경우, 두 테이블에 동일한 이름의 컬럼을 사용하면 어느 테이블소속인지 불분명하기에 애매모호한 상태라는 **오류 메시지가 출력**됩니다.
    - SELECT ename, dname, deptno FROM emp, dept

WHERE emp.deptno = dept.deptno AND ename=’SCOTT’; -error

* + - SELECT ename, dname, emp.deptno FROM emp, dept

WHERE emp.deptno = dept.deptno AND ename=’SCOTT’;

* + 이러한 문제를 해결하기 위해, 동일한 이름의 컬럼은 반드시 컬럼명앞에 테이블명을 명시적으로 기술함으로서 컬럼이 어느 테이블 소속인지 구분할 수 있게 합니다
    - SELECT emp.ename, dept.dname, **emp.deptno** FROM emp, dept

WHERE emp.deptno=dept.deptno AND ename=’SCOTT’;

* + 테이블 이름에 별칭을 붙이는 방법은 FROM 절 다음에 테이블 이름을 명시하고 공백을 둔 다음에 별칭을 지정하면 됩니다 **테이블 별명은 30자이하의 의미있는 단어 사용을 추천**한다
    - SELECT e.ename, d.dname, e.deptno FROM emp e, dept d

WHERE e.deptno=d.deptno AND ename=’SCOTT’;

* + SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, E.DEPTNO, DNAME, LOC FROM EMP E, DEPT D WHERE E.DEPTNO=D.DEPTNO;
  + SELECT E.\*, DNAME, LOC FROM EMP E, DEPT D WHERE E.DEPTNO=D.DEPTNO;
  + SELECT ENAME, LOC FROM EMP E, DEPT D WHERE E.DEPTNO=D.DEPTNO;
  + 사번, 이름, 급여, 부서코드, 부서명
    - select empno 사번, ename 이름, sal 급여, e.deptno 부서코드, dname 부서명 from emp e, dept d where e.deptno = d.deptno;
  + 이름, 업무, 급여, 부서명, 근무지. 급여가 2000이상
    - select e.ename 이름, e.job 업무, to\_char(e.sal, 'l9,999') 급여, d.dname 부서명, d.loc 근무지 from emp e, dept d where e.sal>=2000 and e.deptno=d.deptno;
  + 이름, 업무, 부서명, 근무지. 근무지가 CHICAGO인 사람
    - select e.ename 이름, job 업무, dname 부서명, loc 근무지 from emp e, dept d where e.deptno=d.deptno and d.loc='CHICAGO';
  + 이름, 업무, 근무지. deptno이 10 또는 20인 경우, 급여순
    - select e.ename 이름, job 업무, loc 근무지 from emp e, dept d where e.deptno=d.deptno and e.deptno in (10,20) order by sal;
  + 이름, 급여, comm, 연봉(별칭), 부서명, 근무지. 연봉=(급여+comm)\*12 단 comm이 Null이면 0
    - select ename 이름, sal 급여, comm 보너스, to\_char((sal+NVL(comm,0))\*12, 'l999,999') 연봉, d.dname 부서명, loc 근무지 from emp e, dept d e.deptno=d.deptno;
  + 위에서 job이 salesman 또는 manager 대상이고 연봉이 큰순 정렬
    - select ename 이름, sal 급여, comm 보너스, to\_char((sal+NVL(comm,0))\*12, 'l99,999') 연봉, d.dname 부서명, loc 근무지 from emp e, dept d

where e.deptno=d.deptno and job in (upper('salesman'), upper('manager')) order by 연봉 desc;

* + 이름, 급여, 입사일, 부서코드, 부서명 comm이 null이고 급여가 1200이상인 경우. 부서명순 정렬. 부셔명이 같으면 급여큰순
    - select ename 이름, sal 급여, to\_char(hiredate, 'yyyy"년 "MM"월" dd"일" dy"요일"') 입사일, d.deptno 부서코드, dname 부서명 from emp e, dept d

where e.deptno=d.deptno and comm is Null and sal>=1200 order by dname asc, sal desc;

1. Non-Equi Join
   * Non-Equi Join은 조인 조건에 특정 범위 내에 있는지를 조사하기 위해서 WHERE 절에 조인 조건을 = 연산자 이외의 비교 연산자를 사용합니다. Non-Equi Join을 학습하기 전에 급여 등급 테이블(SALGRADE )을 살펴봅시다.
     + SELECT \* FROM salgrade;
   * 급여 등급을 5개로 나누어 놓은 salgrade에서 정보를 얻어 와서 각 사원의 급여 등급을 지정해보도록 합시다. 이를 위해서 사원(emp) 테이블과 급여 등급(salgrade) 테이블을 조인하도록 합시다. 다음은 사원의 급여가 몇 등급인지 살펴보는 예제.
     + SELECT ename, sal, grage FROM emp, salgrage WHERE sal BETWEEN losal AND sal between losal AND hisal;
     + SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, GRADE||'등급', NVL(COMM,0), DEPTNO FROM EMP E, SALGRADE S WHERE SAL BETWEEN LOSAL AND HISAL;
     + SELECT E.\*, GRADE FROM EMP E, SALGRADE S WHERE SAL BETWEEN LOSAL AND HISAL;
2. Self Join
   * 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 것. Self Join을 보다 구체적인 예를 통해서 알아보도록 합시다. SMITH의 매니저 이름이 무엇인지 알아내려면 어떻게 구해야 할까요?

SELECT W.ENAME, W.MGR, M.EMPNO, M.ENAME FROM EMP W, EMP M;

SELECT W.ENAME, W.MGR, M.EMPNO, M.ENAME FROM EMP W, EMP M WHERE W.ENAME='SMITH';

SELECT W.ENAME, W.MGR, M.EMPNO, M.ENAME FROM EMP W, EMP M

WHERE W.ENAME='SMITH' AND W.MGR=M.EMPNO;

* + 이런 경우 Self Join을 씁니다.
  + Slef Join은 Equi Join과 같으나 하나의 테이블에 조인이 일어나는 것이 다름. 같은 테이블에 대해 두 개의 Alias를 작성 함으로서 from 절에 두 개의 테이블을 사용하는 것과 같이 함
  + WORKER.MGR 과 MANAGER.EMPNO은 같다

SELECT WORKER.ENAME || ‘의 상사는 ‘ || MANAGER.ENAME

FROM EMP WORKER, EMP MANAGER

WHERE WORKER.MGR=MANAGER.EMPNO

1. Outer Join
   * Seif Join을 통해 특정 사원의 매니저 이름을 구한 결과를 꼼꼼히 살펴보면 이름이 KING인 사원 한 사람의 정보가 빠져 있음을 확인할 수 있다.
   * 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 조인(Outer Join)은 NULL 값이기에 배제된 행을 결과에 포함시킬 수 있으며 다음과 같이 “(+)” 기호를 조인 조건에서 정보가 부족한 칼럼 이름 뒤에 덧붙입니다.
   * 사원 번호(EMPNO)가 NULL인 사원은 없으므로 manager.empno 뒤에 “(+)” 기호를 덧붙입니다.
     + SELECT WORKER.ENAME, MANAGER.ENAME 상사

FROM EMP WORKER, EMP MANAGER

WHERE WORKER.MGR=MANAGER.EMPNO(+) ;

* 말단사원들의 리스트를 출력하시오
  + SELECT W.EMPNO, W.MGR, W.ENAME, M.ENAME FROM EMP W, EMP M

WHERE W.MGR(+)=M.EMPNO AND W.ENAME IS NULL ORDER BY W.ENAME;

* 이름, 업무, 부서명
  + select e.ename, e.job, d.dname from emp e, dept d where e.deptno=d.deptno;
* 이름, 업무, 부서코드, 부서명 (직원이 없는 부서명 출력)
  + select e.ename, e.job, d.deptno, d.dname from emp e, dept d where e.deptno(+)=d.deptno;
* 이름, 업무, 부서명, 급여 (직원이 없는 부서명 출력) 부서명순 정렬, 부서가 같으면 급여 큰순
  + select e.ename, e.job, d.dname, to\_char(sal, 'l9,999') from emp e, dept d where e.deptno(+)=d.deptno order by d.dname, e.sal desc;