JOIN

[정 의]

[필요성]
Display data from multiple tables
(한 개 이상의 테이블로부터 데이터를 읽어야 할때) (수평적 결합)

[종 류]
EQUI JOIN .
NON EQUI JOIN
OUTER JOIN
SELF JOIN

[내부 알고리즘]
NESTED LOOP JOIN
SORT MERGE JOIN
HASH JOIN

// 실습환경

```
SELECT * FROM TAB;
DESC  EMP              -- DEPTNO 컬럼 관찰
DESC  DEPT             -- DEPTNO 컬럼 관찰
DESC  SALGRADE         -- LOWSAL,HIGHSAL 컬럼 관찰
```

[JOIN 필요성]

사원번호,이름,급여,부서번호 정보가 필요한 경우
➔ SELECT EMPNO,ENAME,SAL,DEPTNO FROM EMP;

부서번호,부서명,부서위치 정보가 필요한경우
➔ SELECT DEPTNO,DNAME,LOC FROM DEPT;

부서명,사원번호,이름,급여 정보가 필요한경우
➔ 부서명은 DEPT 테이블에서 정의,
사원번호,이름,급여는 EMP 테이블에서 정의
이러한 정보를 조회하기 위해서는 2개의 테이블 결합(JOIN)해서 SELECT 해야 합니다.

여러 테이블에 각각 나누어져서 정의된 속성(컬럼)들을 동시에 조회할경우 JOIN이 필요 합니다.

[EQUI-JOIN(SIMPLE JOIN ,INNER JOIN)]

정의: JOIN에 사용되는 테이블의 컬럼간에 정확이 일치(EQUAL)하는 데이터를 RETURN

EQUAL(=) 연산자를 사용하여 JOIN

① SELECT DNAME,ENAME,JOB,SAL FROM EMP, DEPT WHERE DEPTNO = DEPTNO;
➔ ORA-00918: 열의 정의가 애매합니다.. 이유는 ?

② SELECT DNAME,ENAME,JOB,SAL FROM EMP, DEPT WHERE EMP.DEPTNO = DEPT.DEPTNO;

③
```
SELECT  DNAME,ENAME,JOB,SAL FROM EMP, DEPT
WHERE  EMP.DEPTNO = DEPT.DEPTNO AND
       EMP.JOB IN ('MANAGER','CLERK')
ORDER BY DNAME;
```
// 실행순서? 1. JOIN 처리 ➔ EMP.JOB
2. EMP.JOB ➔ JOIN처리

④
```
SELECT DEPT.DNAME,EMP.ENAME,EMP.JOB,EMP.SAL  FROM  SCOTT.EMP, SCOTT.DEPT
WHERE  EMP.DEPTNO = DEPT.DEPTNO;
```
// OBJECT NAME 표기법 : [SCHEMA.]OBJECT_NAME

EX) SCOTT.EMP , EMP
// COLUMN NAME 표기법 : [TABLE_NAME.]COLUMN_NAME
EX) EMP.EMPNO, EMPNO

① SELECT D.DNAME,E.ENAME,E.JOB,E.SAL FROM EMP E,DEPT D WHERE E.DEPTNO = D.DEPTNO;
// TABLE ALIAS 용도 1. 편의성
2. 가독성(의미있는 이름사용) EX) EMP E , EMP A
3. SELF JOIN시에는 필수적으로 사용
4. 동일 컬럼명이 존재하는 경우

② [ANSI/ISO] Equi-Join
SELECT D.DNAME,E.ENAME,E.JOB,E.SAL
FROM EMP E INNER JOIN DEPT D
ON E.DEPTNO = D.DEPTNO // ON JOIN 조건 표기

③ SELECT D.DNAME,E.ENAME,E.JOB,E.SAL
FROM EMP E INNER JOIN DEPT D
ON E.DEPTNO = D.DEPTNO
WHERE E.DEPTNO IN (10,20) AND D.DNAME = 'RESEARCH' // WHERE 조건

[NON EQUI-JOIN]

정의: EQUAL(=) 이외의 연산자를 사용하여 JOIN

④ SELECT E.ENAME, E.JOB,E.SAL,S.GRADE FROM EMP E,SALGRADE S
WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL;

⑤ SELECT DNAME,ENAME,JOB,SAL,GRADE
FROM EMP E, DEPT D, SALGRADE S // 3개테이블 JOIN , 최소 JOIN조건: N(테이블개수) - 1
WHERE E.DEPTNO = D.DEPTNO AND E.SAL BETWEEN S.LOSAL AND S.HISAL;

⑥ SELECT E.ENAME, E.JOB,E.SAL,E.GRADE FROM EMP E, SALGRADE S
WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL AND E.DEPTNO IN (10,30);
ORDER BY E.ENAME;

⑦ SELECT E.ENAME, E.JOB,E.SAL,S.GRADE FROM EMP E, SALGRADE S
WHERE E.SAL < S.LOSAL AND E.DEPTNO IN (10,30)
ORDER BY E.ENAME; // ??

[OUTER-JOIN (OUT SIDER 정보를 보겠다)]

정의: JOIN조건에 직접 만족되지 않는 정보도 조회.
INNER JOIN

⑧ SELECT D.DNAME,E.ENAME,E.JOB,E.SAL FROM EMP E,DEPT D
WHERE E.DEPTNO = D.DEPTNO
ORDER BY D.DNAME;

⑨ SELECT D.DEPTNO,D.DNAME,E.ENAME,E.JOB,E.SAL
FROM EMP E,DEPT D
WHERE E.DEPTNO = D.DEPTNO
ORDER BY D.DEPTNO;

// 40번부서에 근무하는 직원이 없기 때문에 EQUI JOIN에서는 …

⑩ SELECT D.DNAME,E.ENAME,E.JOB,E.SAL FROM EMP E,DEPT D
WHERE E.DEPTNO(+) = D.DEPTNO
ORDER BY D.DNAME;

⑪ SELECT D.DEPTNO,D.DNAME,E.ENAME,E.JOB,E.SAL
FROM EMP E,DEPT D
WHERE E.DEPTNO(+) = D.DEPTNO
ORDER BY D.DEPTNO;

⑫ SELECT D.DNAME,E.ENAME,E.JOB,E.SAL FROM EMP E,DEPT D
WHERE E.DEPTNO = D.DEPTNO(+)
ORDER BY D.DNAME; // EQUI JOIN과 결과가 동일,효율성(?)

⑬ SELECT D.DNAME,NVL(E.ENAME,'비상근 부서'),E.JOB,E.SAL FROM EMP E,DEPT D
WHERE E.DEPTNO(+) = D.DEPTNO
ORDER BY D.DNAME; // 정말 NULL로 채우나?

⑭ SELECT D.DNAME,E.ENAME,E.JOB,E.SAL FROM EMP E,DEPT D
WHERE E.DEPTNO(+) = D.DEPTNO(+)
ORDER BY D.DNAME;

// ORA-01468: outer-join된 테이블은 1개만 지정할 수 있습니다,
ORACLE은 양방향 OUTER JOIN을 허용하지 않는다.
SQL 1999 에서는 양방향 OUTER JOIN을 허용

[요구]
ⓐ 아래의 2개 SQL을 실행하여 데이터를 비교 한후 원하는 결과집합이 나오도록 2번째 SQL을 수정 하십시요

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

SELECT  D.DEPTNO,D.DNAME,E.ENAME,E.JOB,E.SAL FROM EMP E,DEPT D
WHERE   E.DEPTNO(+) = D.DEPTNO AND
        E.SAL > 2000
ORDER BY D.DNAME;
```

[ANSI-SQL : LEFT OUTER JOIN , RIGHT OUTER JOIN , FULL OUTER JOIN]
---

① SELECT E.DEPTNO,D.DNAME,E.ENAME
FROM SCOTT.EMP E LEFT OUTER JOIN SCOTT.DEPT D
ON E.DEPTNO = D.DEPTNO
ORDER BY E.DEPTNO;

// DEPT을 기준 테이블(Driving Table)으로 JOIN 연산 수행 ,40번 부서의 정보 표기? E.DEPTNO 에 나타나는 값
② SELECT E.DEPTNO,D.DNAME,E.ENAME
FROM SCOTT.EMP E RIGHT OUTER JOIN SCOTT.DEPT D
ON E.DEPTNO = D.DEPTNO
ORDER BY E.DEPTNO;

// DEPT을 기준 테이블(Driving Table)으로 JOIN 연산 수행 ,40번 부서의 정보 표기? D.DEPTNO 에 나타나는 값
③ SELECT D.DEPTNO,D.DNAME,E.ENAME
FROM SCOTT.EMP E RIGHT OUTER JOIN SCOTT.DEPT D
ON E.DEPTNO = D.DEPTNO
ORDER BY E.DEPTNO;

// 양방향 FULL OUTER JOIN !!!!!
④ SELECT D.DEPTNO,D.DNAME,E.ENAME
FROM SCOTT.EMP E **FULL OUTER** JOIN SCOTT.DEPT D

```
   ON       E.DEPTNO = D.DEPTNO
   ORDER   BY E.DEPTNO;
```

[SELF-JOIN]

```
// 사원의 이름과 매니저의 이름을 한줄에 표현하는 방법은 ?
⑤ SELECT E.ENAME||' ''S   MANAGER IS '||M.ENAME
   FROM    EMP E,EMP M
   WHERE   E.MGR = M.EMPNO
   ORDER BY M.ENAME;
```

[요구]

ⓐ ⑤ SQL에서는 사장님의 정보는 누락되었다. SQL을 수정하여 작성 하십시요

- 매니저가 없는 경우 매니저의 이름은 NOBODY로 표기

ⓑ ⓐ를 ANSI-SQL로 변환 하십시요

[CARTESIAN PRODUCT]

[정 의] 데카르트의 곱집합, 수학의 곱집합
[원 인] (1) JOIN 조건 생략시 (2) 잘못된 JOIN 조건
[문제점] 유용하지 않은 대량의 데이터 생성
[용 도] 1.테스트용 샘플데이타 생성
2.곱집합 기능을 이용한 빠른 연산에 응용

```
① SELECT ENAME,JOB,DNAME   FROM EMP, DEPT;

② SELECT ENAME,JOB,DNAME   FROM   EMP , DEPT
   WHERE   EMP.SAL > 2000   and   DEPT.DEPTNO IN (10,20);

③ SELECT ENAME,JOB,DNAME   FROM   EMP , DEPT
   WHERE   EMP.SAL > 2000   or   DEPT.DEPTNO IN (10,20);

④ SELECT    E.ENAME, E.JOB,E.SAL,S.GRADE   FROM   EMP E, SALGRADE S
   WHERE    E.SAL   <   S.LOSAL AND E.DEPTNO IN (10,30)
   ORDER BY E.ENAME;
```

[요구]

ⓒ C:₩03_SQL₩MAKE_ENV.SQL SQL SCRIPT 파일을 생성하여 아래의 실습 환경을 구성 한후 아래의 결과가 출력되는 JOIN 구문을 작성 하십시요 (캔 캔 ~)

MAKE_ENV.SQL

```
CREATE   TABLE SYSTEM(  SYSTEM_ID     VARCHAR2(5),
                        SYSTEM_NAME VARCHAR2(10)
);
INSERT INTO SYSTEM   VALUES('XXX','혜화DB');
INSERT INTO SYSTEM   VALUES('YYY','강남DB');
INSERT INTO SYSTEM   VALUES('ZZZ','영등포DB');

CREATE TABLE RESOURCE_USAGE(SYSTEM_ID          VARCHAR2(5),
                            RESOURCE_NAME  VARCHAR2(10)
);
INSERT INTO RESOURCE_USAGE   VALUES('XXX','FTP');
INSERT INTO RESOURCE_USAGE   VALUES('YYY','FTP');
INSERT INTO RESOURCE_USAGE   VALUES('YYY','TELNET');
INSERT INTO RESOURCE_USAGE   VALUES('YYY','EMAIL');
COMMIT;
```

SYSTE	SYSTEM_NAME	FTP	TELNET	EMAIL_
XXX	혜화DB	사용	미사용	미사용
YYY	강남DB	사용	미사용	사용
ZZZ	영등포DB	미사용	미사용	미사용

* SQL DEVELOPER 와 SQL*PLUS 각각에서 SQL SCRIPT를 실행 하십시요
* SELECT S.SYSTEM_ID,S.SYSTEM_NAME,R.RESOURCE_NAME
FROM SYSTEM S, RESOURCE_USAGE R
WHERE S.SYSTEM_ID = R.SYSTEM_ID;
SELECT S.SYSTEM_ID,S.SYSTEM_NAME,R.RESOURCE_NAME
FROM SYSTEM S,RESOURCE_USAGE R
WHERE S.SYSTEM_ID = R.SYSTEM_ID(+);
* JOIN,DECODE,COUNT 를 사용하여….

ⓓ JOIN을 사용하여 부서별 급여 지급 순위를 구하십시오 (CAN !!!)

부서번호,	이름,	직업,	급여,	급여순위
10	CLARK	MANAGER	5000	1
10	KING	PRESIDENT	3500	2
10	MILLER	CLERK	2750	3
20	SCOTT	ANALYST	3000	1
20	FORD	ANALYST	3000	1
20	JONES	MANAGER	2975	2

ⓔ RANK, DENSE_RANK 함수를 사용하여 ⓓ와 동일한 결과를 나타내는 SQL을 작성 하십시요

ⓕ 부서번호,사번,이름,급여,급여비율(소수점이하2자리)을 출력하는 SQL을 CARTESIAN PRODUCT를 응용하여 작성 하십시요

	DEPTNO	ENAME	SAL	SAL_RATE
1	20	SMITH	800	2.76%
2	30	ALLEN	1600	5.51%
3	30	WARD	1250	4.31%
4	20	JONES	2975	10.25%
5	30	MARTIN	1250	4.31%
6	30	BLAKE	2850	9.82%
7	10	CLARK	2450	8.44%
8	20	SCOTT	3000	10.34%
9	10	KING	5000	17.23%
10	30	TURNER	1500	5.17%
11	20	ADAMS	1100	3.79%
12	30	JAMES	950	3.27%
13	20	FORD	3000	10.34%
14	10	MILLER	1300	4.48%