조인(JOIN)



배재대학교 컴퓨터공학과 김 창 수

Contents

- 🦲 조인의 개념
- **칼럼 이름의 애매모호성,조인 조건식**
- 조인의 종류

❖ 조인의 개념

- 하나의 SQL 명령문에 의해 여러 테이블에 저장된 데이터를 한번에 조 회할수 있는 기능
- 관계형 데이터베이스 분야의 표준
- 두개 이상의 테이블을 '결합' 한다는 의미

❖ 조인의 필요성

- 조인을 사용하지 않는 일반적인 예
 - 사원 주소록을 출력하기 위해 사원들의 사번, 이름, 소속부서 이름을 검색
 - 사원에 대한 정보 검색하는 단계 필요
 - 사원 정보에서 소속부서번호 정보를 추출하여 소속부서 이름을 검색하는 단계 필요

■ 조인을 사용하지 일반적인 예

❖ 사용 예

■ 사번이 1014인 사원의 이름과 소속 부서 이름을 출력하여라.

SQL> 2 3	select empno, ename, from emp where empno = 1014;	deptno	먼저, 사원테이블에서 1014번 사원 이름과 소속 부 번호를 검색한다	서
	EMPNO ENAME	DEPTNO		
	1014 조인성	10		
	select dname from dept			
3	where deptno = 10;		부서 테이블에서 10번 부서 이름을 검색한다	
DNAME				
경리!	부			



조인을 사용하지 않은 경우

사원 테이블의 사번, 사원이름, 부서코드 검색 SELECT empno, ename, deptno FROM emp;

사원 테이블

EMPNO	ENAME	 DEPTNO
1001	김사랑	 20
1002	한예슬	 30
1013	박중훈	20
1014	조인성	10

부서 테이블의 부서코드, 부서명 검색

SELECT deptno, dname

FROM dept

WHERE deptno in (10,20, 30, 40);

부서 테이블

DEPTNO	DNAME
10	경리부
20	인사부
30	영업부
40	전산부

조인을 사용한 경우

조인을 이용한 사원이를과 부서이를 검색
SELECT empno, ename,
emp.deptno, dept.dname
FROM emp, dept
WHERE emp.deptno = dept.deptno;

시찬	사원 테이들							
EMF	NO	ENAME		PEPTNO				
10	01	김사랑		20				
10	02	한예슬		30				
10	13	박중훈		20				
10	14	조인성		10				
-								

TAI 41912				
PEPTNO	DNAME			
20	인사부			
30	명 업무			
20	인사부			
10	경리부			
	•			

부서 테이블

DNAME	DEPTNO	ENAME	EMPNO			
인사부	20	김사랑	1001			
영업부	30	한예슬	1002			
인사부	20	박중훈	1013			
경리부	10	조인성	1014			

사원 테이블

부서 테이들

EMPNO	ENAME	JOB	 DEPTNO	DEPTNO	DNAME	LOC
1001	김사랑	사원	 20 -	 > 10	경리부	서울
1002	한예슬	대리	 30	 → 20	인사부	인천
			 	> 30	영업부	용인
1013	박중훈	부장	20 /	40	전산부	수원
1014	조인성	사원	10 ′			

사원 테이블과 부서 테이블 조인

SELECT empno, ename, job, student.deptno,
FROM emp, dept
WHERE emp.deptno = dept.deptno

조인 결과

EMPNO	ENAME	JOB	 DEPTNO	DNAME	LOC
1001	김사랑	사원	 20	인사부	인천
1002	한예슬	대리	 30	영업부	용인
1013	박중훈	부장	20	인사부	인천
1014	조인성	사원	10	경리부	서울



❖ 칼럼 이름의 애매모호성 해결방법

- 서로 다른 테이블에 있는 동일한 칼럼 이름을 연결할경우
- 컬럼 이름앞에 테이블 이름을 접두사로 사용
- 테이블 이름과 칼럼 이름은 점(.)으로 구분
- SQL 명령문에 대한 구문분석 시간(parsing time) 줄임
 - emp.deptno와 dept.deptno

```
SQL> SELECT emp.empno, emp.ename, emp.job, deptno,
                                               dname
    FROM
          emp, dept
    WHERE emp.deptno = dept.deptno;
SELECT emp.empno, emp.ename, emp.job, deptno, dname
1행에 오류:
DRA-00918: column ambiguously defined
SQL> SELECT emp.empno, emp.ename, emp.job, emp.deptno, dname
    FROM
            emp, dept
    WHERE | deptno = deptno;
                                 deptno 칼럼이 student 테이블과 department 테이
WHERE deptno = deptno
                                 불에 동일한 이름으로 정의되어 오류가 발생
3행에 오류:
DRA-00918: column ambiguously defined
```





칼럼 이름의 애매모호성

deptno **칼럼이름 앞에 테이블 이름을 사용하면 애매 모호성** 문제가 해결

2 FROM <u>em</u>	.empno, emp.ename, o, dept .deptno = dept.dept	emp.job, <mark>emp.deptno</mark> , no;	dname
EMPNO ENAM	E JOB	DEPTNO DNAME	
1001 김사 1002 한예 1003 오지 1004 이병 1005 신동 1006 장동 1007 이문 1009 안성 1010 조형 1012 강혜 1013 박중 1014 조인	슬 대 라 라 라 라 라 라 라 라 라 라 라 라 라 라 라 라 라 라	20 영영 인영영 기 기 기 기 기 기 기 기 기 기 기 기 기 기 기 기	





❖ 테이블 이름이 너무 긴 경우 사용

- 테이블 이름을 대신하는 별명 사용 가능
- FROM절에서 테이블 이름 다음에 공백을 두고 별명 정의
- 테이블 별명 작성 규칙
 - 테이블의 별명은 30자 까지 가능, 너무 길지 않게 작성
 - FROM 절에서 테이블 이름을 명시하고 공백을 둔 다음 테이블 별명지정
 - 하나의 SQL 명령문에서 테이블 이름과 별명을 혼용할 수 없다
 - 테이블의 별명은 해당 SQL 명령문내에서만 유효

테이블 별명

❖ 사용 예

■ 테이블 이름과 별명을 혼용하여 사용한 경우

```
SQL> SELECT emp.empno, emp.ename, emp.job, emp.deptno, dname
2 FROM emp e, dept d
3 WHERE e.deptno = dept.deptno;
WHERE e.deptno = dept.deptno
*
3행에 오류:
ORA-00904: "DEPT"."DEPTNO": invalid identifier
```

■ 테이블 별명을 정확하게 사용한 경우



AND 연산자를 사용한 검색 조건 추가

❖ 사용 예

'안성기'사원의 사번, 이름, 부서 이름 그리고 부서 위치를 출력 하여라.

```
SQL> select e.empno, e.ename, d.dname, d.loc
 2 from emp e, dept d
 3 where e.deptno = d.deptno
 4 and e.ename = '안성기';
    EMPNO ENAME DNAME
                               LOC
     1009 안성기 인사부 인천
```



AND 연산자를 사용한 검색 조건 추가

❖ 실습 예

■ 급여가 350이상인 사원의 사번, 이름, 급여, 부서 이름, 부서위치를 출력하여라.

```
SQL> select e.empno, e.ename, d.dname, d.loc
   from emp e, dept d
   where e.deptno = d.deptno
    and e.sal >= 350:
    EMPNO ENAME
                                LOC
                   DNAME
                   영업부
     1003 오지호
                                용인용용서용인처
     1004 이병헌
                   인자부
                   1005 신통협
     1006 장동건
     1007 이분제
     1008 감후성
                   인사부
     1009 안성기
                   경리부
     1010 이병헌
     1013 박중훈
                                이처
                   인사부
```



카티션 곱

❖ 카티션 곱

- 두 개 이상의 테이블에 대해 연결 가능한 행을 모두 결합
- WHERE 절에서 조인 조건절을 생략하거나 잘못 설정한 경우
- 대용량 테이블에서 발생할 경우 SQL명령문의 처리속도 저하
- 개발자가 시뮬레이션을 위한 대용량의 실험용 데이터를 생성하기 위해 의도적으로 사용 가능
- 오라클 9i 이후 버전에서 FROM절에 CROSS JOIN 키워드 사용

사원 테이블(14건)

EMPNO	ENAME	 DEPTNO
1001	김사랑	 20
1002	한예슬	 30
1013	박증훈	20
1014	조인성	10

카티션 곱

SELECT empno, ename, e.deptno, d.deptno, dname FROM emp e, dep d

부서 테이블(4건)

DEPTNO	DNAME
10	경리부
20	인사부
30	영업부
40	전산부

EMPNO	ENAME	E.DEPTNO	D.DEPTNO	DNAME
1001	전인하	20	10	경리부
1002	이동훈	30	10	경리부
1013	박중훈	20	10	경리부
1014	조인성	10	10	경리부
1001	전인하	20	20	인사부
1002	이동훈	30	20	인사부
1013	박중훈	20	20	인사부
1014	조인성	10	20	인사부
1001	전인하	20	30	영업부
1002	이동훈	30	30	영업부
1013	박중훈	20	30	영업부
1014	조인성	10	30	영업부
1001	전인하	20	40	전산부
1002	이동훈	30	40	전산부
1013	박중훈	20	40	전산부
1014	조인성	10	40	전산부



카티션 곱

❖ CROSS JOIN 사용법

```
SELECT table.column, table2.column
       [table, table2 | table1 CROSS JOIN table2];
FROM
```

❖ 사용 예

사원 테이블과 부서 테이블을 카티션 곱을 한 결과를 출력하여라

```
SQL> select ename, emp.deptno, dname, loc
  2 from emp. dept:
ENAME
                DEPTNO DNAME
                                        LOC
                                        서울
서울
서울
                    20 경리부
김사랑
                    30 경리부
30 경리부
30 경리부
20 경리부
한예슬
오지호
이병헌
                                                두 질의문은 동일한 결과
SQL> select ename, emp.deptno, dname, loc
    from emp CROSS JOIN dept:
                DEPTNO DNAME
                                        LOC
ENAME
                    20 경리부
30 경리부
30 경리부
20 경리부
                                        서울
서울
서울
김사랑
한예슬
오지호
이별菌
```





❖ 개념

- 조인 대상 테이블에서 공통 칼럼을 '='(equal) 비교를 통해 같은 값을 가지는 행을 연결하여 결과를 생성하는 조인 방법
- SQL 명령문에서 가장 많이 사용하는 조인 방법
- 조인 애트리뷰트(join attribute(속성))

❖ WHERE 절을 이용한 EQUI JOIN 사용법

```
SELECT table.column, table2.column
FROM table, table2
WHERE table1.column1 = table2.column2;
```

- FROM : 조인 대상 테이블을 기술한다. 테이블은 콤마(,)로 구분
- WHERE : 조인 애트리뷰트와 '='연산자를 사용하여 조인 조건을 기술



EQUI JOIN

❖ 사용 예

• 학생 테이블과 부서 테이블을 EQUI JOIN하여 학번, 이름, 학과 번 호, 소속 학과 이름, 학과 위치를 출력하여라

•	s.name, s.deptno, d.dname , department d		
	= d.deptno;	학생 테이블의 학과번호에 의해 소속 학과 과 학과 위치를 조회하기 위해 student터	· —
STUDNO NAME	DEPTNO DNAME	deptno 칼럼과 department 테이블의 dept 럼을 EQUI JOIN 한다	tno 칼
19191 전하 19192 전비 지역 19194 지역 19196 유민과 지역 19198 유민과	101 101 101 101 101 101 101 101 101 101	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	



❖ 자연조인을 이용한 EQUI JOIN

- 오라클 9i 버전부터 EQUI JOIN을 자연조인이라 명명
- WHERE 절을 사용하지 않고 NATURAL JOIN 키워드 사용
- 오라클에서 자동적으로 테이블의 모든 칼럼을 대상으로 공통 칼럼을 조사 후, 내부적으로 조인문 생성

❖ 사용법

```
SELECT table.column, table2.column
FROM table1
NATURAL JOIN table2;
```

- 주의
 - 조인 애트리뷰트에 테이블 별명을 사용하면 오류가 발생





❖ 조인 애트리뷰트에서 테이블 별명을 사용한 경우

```
SQL> SELECT  s.studno, s.name,(s.deptno) d.dname
 2 FROM
           student s
           NATURAL JOIN department d;
SELECT s.studno, s.name, s.deptno, d.dname
1행에 오류:
DRA-25155: NATURAL 조인에 사용된 열은 식별자를 가질 수 없음
```

❖ 조인 애트리뷰트에서 테이블 별명을 사용하지 않은 경우

```
SQL> SELECT s.studno, s.name( deptno,)d.dname
     FROM
            student s
           NATURAL JOIN department d;
   STUDNO NAME
                      DEPTNO DNAME
                                        조인 애트리뷰트는 테이블 별명을 사용하지 않거
    18181 저이하
                        101 커프디고하다
                                        나 아래 예처럼 모든 칼럼에서 별명을 사용하지
                                        않는다
SQL> SELECT studno, name, deptno, dname
     FROM
            student
            NATURAL JOIN department;
   STUDNO NAME
                      DEPTNO DNAME
    <u> 18181 저이하</u>
                        <u> 101 커프디고</u>하다
```



❖ 사용 예

NATURAL JOIN을 이용하여 교수 번호, 이름, 학과 번호, 학과 이름 을 출력하여라.

```
SQL> SELECT p.profno, p.name, deptno, d.dname
                                      FROM
                                                                                                  professor p
                                                                                                    NATURAL JOIN department d;
                                 PROFNO NAME
                                                                                                                                                                                                               DEPTNO DNAME
                                             9901 김도훈 101 컴퓨터공학과 9903 성연회 101 컴퓨터공학과 9907 전은지 101 컴퓨터공학과 9906 이만식 101 컴퓨터공학과 9904 염일웅 102 멀티미디어학과 100명 기취인 100명 기취인 100명 기취인 100명 기계 100명 
                                              9903 성전만식
9907 전만식
9906 이만일
9904 연권
9905 이자
9902 이자
                                                                                                                                                          182 멀티미디어학과
                                                                                                                                                                                                                                                  201 전자공학과
                                                  9908 남은혁
                                                                                                                                                                                                                                                  202 기계공학과
                개의 행이 선택되었습니다.
```





❖ 사용 예

NATURAL JOIN을 이용하여 4학년 학생의 이름, 학과 번호, 학과이 름을 출력하여라.

```
SQL> SELECT  s.name, s.qrade, s.deptno, d.dname
  2 FROM
             student s, department d
     WHERE
             s.deptno = d.deptno
             s.qrade = '4';
     AND
NAME
                 DEPTHO DNAME
전인하
이광훈
오유석
                 101 컴퓨터공학과
101 컴퓨터공학과
102 멀티미디어학과
SQL> SELECT  s.name, s.grade, deptno, d.dname
    FROM
             student s
             NATURAL JOIN department d
             grade = '4':
     WHERE
                 DEPTNO DNAME
name
전인하
이광훈
오유석
                 101 컴퓨터공학과
101 컴퓨터공학과
                     102 멀티미디어학과
```



EQUIJOIN - JOIN ~ USING

❖ JOIN ∼ USING 절을 이용한 EQUI JOIN

- USING절에 조인 대상 칼럼을 지정
- 칼럼 이름은 조인 대상 테이블에서 동일한 이름으로 정의되어 있어 야함

❖ 사용법

```
SELECT table.column, table2.column
FROM table1 JOIN table2
USING (column);
```

- 주의
 - 조인 애트리뷰트에 테이블 별명을 사용하면 오류가 발생





EQUI JOIN - JOIN ~ USING

❖ 사용 예

JOIN ~ USING 절을 이용하여 학번, 이름, 학과번호, 학과이름, 학 과위치를 출력하여라.

```
SQL> SELECT | s.studno, s.name, deptno, d.dname, d.loc
            student s JOIN department d
 2 FROM
            USING (deptno);
   STUDNO NAME
                        DEPTNO DNAME
                                                LOC
                                                1호관
1호관
1호관
    10101 전인하101 컴퓨터공학과10102 박미경101 컴퓨터공학과
                            101 컴퓨터공학교
    10104 지은경
```

```
SQL> SELECT | studno, name, deptno, dname, loc
   FROM
          student JOIN department
           USING (deptno);
   STUDNO NAME DEPTNO DNAME
                                            LOC
    10101 전인하 101 컴퓨터공학과
10102 박미경 101 컴퓨터공학과
                                          1호관
                                            1호환
```





EQUIJOIN - JOIN ~ USING

❖ 실습 예

- EQUI JOIN의 3가지 방법을 이용하여 성이 '김' 씨인 학생들의 이름, 학과이름, 학과위치를 출력하여라.
- WHERE 절을 사용한 방법

```
SQL> SELECT
            name, dname, loc
     FROM
            student s, department d
            s.deptno = d.deptno
            name like '김%';
```

NATURAL JOIN절을 사용한 방법

```
SQL> SELECT s.name, d.dname, d.loc
   FROM
            student s
            NATURAL JOIN department d
            name like '김%';
    WHERE
```

JOIN~USING절을 사용한 방법

```
SQL> SELECT name, dname, loc
            student JOIN department
    FROM
            USING(deptno)
            name like '김%';
    WHERE
```



NON-EQUI JOIN

NON-EQUI JOIN

■ '<',BETWEEN a AND b 와 같이 '=' 조건이 아닌 연산자 사용

❖ 사용 예

교수 테이블과 급여 등급 테이블을 NON-EQUI JOIN하여 교수별로 급여 등급을 출력하여라.

```
SQL> SELECT p.profno, p.name, p.sal, s.grade
    FROM
            professor p, salgrade s
           p.sal BETWEEN s.losal AND s.hisal;
    WHERE
    PROFNO NAME
                              SAL
                                       GRADE
     9904 염일웅
9907 전은지
                              240
                              210
      9902 이재우
                              320
      9903 성연희
                              360
     9908 남은혁
9901 김도훈
9905 권혁일
                              400
                              500
                              450
      9906 이만식
                              420
```



NON-EQUI JOIN

❖ 실습 예

101번 학과 교수의 교수번호, 이름, 급여, 급여 등급을 출력하여라.

```
SQL> SELECT p.profno, p.name, p.sal, s.grade
    FROM
           professor p, salgrade s
    WHERE p.sal BETWEEN s.losal AND s.hisal
  4 AND
           p.deptno = 101;
   PROFNO NAME
                          SAL
                                   GRADE
     9907 전은지
                          210
     9903 정편회
                          360
     9901 김도훈
                          500
     9906 이만식
                          420
```

PROFESSOR 테이블

PROFNO	NAME	POSITION	SAL	GRADE: 3
9901	김도훈	교수	500	400
9902	이재우	조교수	320	GRADE: 2
9903	성연희	조교수	360	300
9904	염일웅	전임강사	240	
9905	권혁일	교수	450	200
9906	이만식	부교수	420	GRADE: 1
9907	전은지	전임강사	210	100
9908	남은혁	부교수	400	ı

PROFESSOR 테이블과 SALGRADE 테이블 NON-EQUI JOIN

500 XX

SELECT p.profno, p.name, p.sal, s.grade FROM professor p, salgrade s

WHERE p.sal BETWEEN s.losal AND s.hisal

_				
PROFNO	NAME	POSITION	SAL	GRADE
9901	김도훈	교수	500	3
9902	이재우	조교수	320	2
9907	전은지	전임강사	210	1
9908	남은혁	부교수	400	2



❖ 개요

- EQUI JOIN의 조인 조건에서 양측 칼럼 값 중, 어느 하나라도 NULL 이면 '=' 비교 결과가 거짓이 되어 NULL 값을 가진 행은 조인 결과 로 출력 불가
 - NULL 에 대해서 어떠한 연산을 적용하더라고 연산 결과는 NULL
- 일반적인 EQUI JOIN 의 예 :
 - 학생 테이블의 학과번호 칼럼과 부서 테이블의 부서번호 칼럼에 대한 EQUI JOIN (student.deptno = department.deptno) 한 경우
 - 학생 테이블의 deptno 칼럼이 NULL 인 경우 해당 학생은 결과로 출력되지
- EQUI JOIN에서 양측 칼럼 값증의 하나가 NULL 이지만 조인 결과로 출력할 필요가 있는 경우 OUTER JOIN 사용
- OUTER JOIN의 예 :
 - 학생 테이블과 교수 테이블을 EQUI JOIN하여 학생의 지도 교수 이름 출력
 - 조건 : 지도 학생을 한 명도 배정받지 못한 교수 이름도 반드시 함께 출력

❖ (+) 기호를 사용한 OUTER JOIN

- WHERE 절의 조인 조건에서 OUTER JOIN 연산자인 '(+)' 기호 사용
- 조인 조건문에서 NULL 이 출력되는 테이블의 칼럼에 (+) 기호 추 가

❖ 사용법

```
table.column, table2.column
SELECT
FROM
              table1, table2
            table1.column(+) = table2.column
WHERE
      또는 ; table1.column = table2.column(+)
```



❖ 사용 예

학생 테이블과 교수 테이블을 조인하여 이름, 학년, 지도교수의 이 름, 직급을 출력하여라. 단, 지도교수가 배정되지 않은 학생이름도 함께 출력하여라.

```
SQL> SELECT s.name, s.grade, p.name, p.position
            student s, professor p
    FROM
    WHERE
            s.profno = p.profno(+)
    ORDER
            BY p.profno;
NAME
           G NAME
                       POSITION
경하훈영석욱단경정지
진이광지유지영인미유
김전이김오윤김지류임
          2 이재우
                        조교수
          조교수
조교수
                       표수
부교수
전임강사
전임강사
전임강사
이동훈
박동진
하나리
                                     OUTER JOIN에 의해 지도 교수가 배정되지
                                     않은 학생도 조인 결과로 출력된다
```



❖ OUTER JOIN의 제약사항

- OUTER JOIN 연산자(+)는 NULL이 존재하는 칼럼쪽에 표시
- OUTER JOIN에서는 IN 연산자를 사용 불가
- 다른 조건과 OR 연산자로 결합 불가

DEPTNO

101

102 102

101

101

202

ION

. . .

					The same of the same of			12000
STUDNO	NAME	GRADE	 DEPTNO	PROFNO		PROFNO	NAME	POSITION
10101	전인하	4	 101	9903		9901	김도훈	교수
20101	이동훈	1	 201			∌ 902	이재우	조교수
			 	\		7002		- W 1
10103	김영균	3	 101	9906		9 903	성연희	조교수
20102	박동진	1	 201	`,		9904	염일웅	전임강사
10201	김진영	2	 102	9905	X	>9 905	권혁일	교수
			 			20000	ОПРА	ᆸᅴᄉ
20103	김진경	2	201	9902		29 906	이만식	부교수
				3302		9 907	전은지	전임강사
20104	조명훈	1	 201	****		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
10108	류민정	2	 101	9907	``	`` 9908	남은혁	부교수
		•				Z (+)		

학생 테이블과 교수 테이블 OUTER JOIN

SELECT s.name, s.grade, p.name, p.position FROM student s, professor p

WHERE s.profno = p.profno(+);

NAME	GRADE	NAME	POSITION				
김진경	2	이재우	조교수				
전인하	4	성연희	조교수				
김진영	2	권혁일	교수				
김영균	3	이만식	부교수				
류민정	2	전은지	전임강사				
이동훈	1						
박동진	1						
조명훈	1						



실습 예

학생 테이블과 교수 테이블을 조인하여 이름, 학년, 지도교수 이름, 직급을 출력하여라. 단, 지도학생을 배정받지 않은 교수 이름도 함 께 출력하여라.

```
SQL> SELECT s.name, s.grade, p.name, p.position
      FROM
                  student s, professor p
        WHERE
                   s.profno(+) = p.profno
        ORDER
                   BY p.profno;
NAME
               G NAME
                                  POSITION
               호우히희왕의의의의식지지지하
도재정의의해휴휴휴다의의의의의
김이성성영관권권권이전전전남
                                  교수
김진경
전인하
이광훈
                                  조조조전교교교부전전전부 수수수강 수강하다 수강하다
김오유석욱 균경
지유진영인민유
김지류임
```



35

OUTER JOIN

❖ OUTER JOIN ~ON 절을 사용한 OUTER JOIN

오라클 9i 버전부터 OUTER JOIN 절 사용

❖ 사용법

```
table.column, table2.column
SELECT
FROM
               table1[RIGHT | LEFT | FULL] OUTER JOIN table2
               ON table1.column = table2.column;
```

LEFT OUTER JOIN

- FROM 절의 왼쪽에 위치한 테이블이 NULL 을 가질 경우에 사용
- WHERE절의 오른편에 '(+)' 기호를 추가한 것과 동일
- 예 : 지도교수가 배정되지 않은 학생 명단을 출력하여라

SELECT studno, s.name, s.profno, p.name FROM students **LEFT OUTER JOIN** professor p **ON** s.profno = p.profno;



SELECT studno, s.name, s.profno, p.name FROM student s, professor p WHERE s.profno = p.profno(+);



RIGHT OUTER JOIN

- FROM 절의 오른쪽에 위치한 테이블이 NULL 을 가질 경우, 사용
- WHERE 절의 왼편'(+)'기호를 추가한 것과 동일
- 예
 - 지도학생이 배정되지 않은 교수 명단을 출력하여라.

SELECT studno, s.name, s.profno, p.name FROM student s **RIGHT OUTER JOIN** professor p **ON** s.profno = p.profno;

SELECT studno, s.name, s.profno, p.name FROM student s, professor p WHERE **s.profno(+)** = p.profno;



FULL OUTER JOIN

- LEFT OUTER JOIN 과 RIGHT OUTER JOIN 을 동시에 실행한 결 과를 출력
- 예
 - 지도교수가 없는 학생과 지도학생이 배정되지 않은 교수를 함께 출력하 라.

SELECT studno, s.name, s.profno, p.name

FROM student s

FULL OUTER JOIN professor p **ON** s.profno = p.profno;



STUDNO	NAME	GRADE	 DEPTNO	PROFNO	7
10101	전인하	4	 101	9903	ı
20101	이동훈	1	 201		
10103	김영균	3	 101	9906	
20102	박동진	1	 201	×.	``
10201	김진영	2	 102	9905	
			 		١,
20103	김진경	2	 201	9902	
20104	조명훈	1	 201	*****	
10108	류민정	2	 101	9907	K .

			The second second second		
	PROFNO	NAME	POSITION		DEPTNO
	_/ 9901	김도훈	교수		101
	9902	이재목	조교수	I	201
	9903	성연희	조교수		101
	/9904	염일용	전임강사		102
	9905	권혁일	교수		102
	9906	이만식	부교수		101
\ \	9907	전은지	전임강사		101
`,	`√9908	남은혁	부교수		202

학생 테이블과 교수 테이블 FULL OUTER JOIN

(+)

SELECT studno, s.name, grade, s.deptno, s.profno, p.name
FROM student's **FULL OUTER JOIN** professor p
ON s.profno = p.profno;

(+)

STUDNO	NAME	GRADE	DEPTN O	PROFNO	NAME	POSITION
10101	전인하	4	101	9903	성연희	조교수
20101	이동훈	1	201			
10103	김영균	3	101	9906	이만식	부교수
20102	박동진	1	201			
10201	김진영	2	102	9905	권혁일	교수
10108	류민정	2	101	9907	전은지	전임강사
				9901	김도훈	山 全
				9904	염일용	전임강사

❖ 실습 예

- 학생 테이블과 교수 테이블을 조인하여 학년, 이름, 지도교수 이름, 직급을 출력하여라. 단, 지도교수가 배정되지 않은 학생 명단과 지 도 학생이 배정되지 않은 교수 명단도 함께 출력하여라.
- (+)기호를 사용한 OUTER JOIN

```
SQL> SELECT s.name, s.grade, p.name, p.position
     FROM
            student s, professor p
           s.profno(+) = p.profno(+);
 3 WHERE
       s.profno(+) = p.profno(+)
3행에 오류:
DRA-01468: outer-join된 테이블은 1개만 지정할 수 있습니다
```

OUTER JOIN ~ON절을 사용한 OUTER JOIN

```
SQL>
     SELECT s.name, s.grade, p.name, p.position
     FROM
             student s FULL OUTER JOIN professor p
  3
             ON s.profno = p.profno;
          G NAME
                       POSITION
NAME
간지경
          2 이재우
                       조교수
```



나서 1885

SELF JOIN

❖ 개요

- 하나의 테이블내에 있는 칼럼끼리 연결하는 조인이 필요한 경우 사용
- 조인 대상 테이블이 자신 하나라는 것 외에는 EQUI JOIN과 동일

❖ WHERE 절을 사용한 SELF JOIN

- 한 테이블에서 두 개의 칼럼을 연결하여 EQUI JOIN
- FROM절에서 하나의 테이블에 테이블 별명지정
- 예
 - 부서 테이블에서 WHERE 절을 사용하여 SELF JOIN을 처리

부서 테이블

부서 테이블

DEPTNO	DNAME	COLLEGE	LOC	
101	컴퓨터공학과	100	1호관	
102	멀티어미디어학과	100	2호관	
				ŀ
100	정보미디어학부	(10)-		
200	메카트로닉스학부	10		ŀ
10	공과대학			

DEPTNO	DNAME	COLLEGE	LOC
101	컴퓨터공약과	100	1호관
102	멀티어미디어학과	100	2 호 관
100	정보미디어학부	10	
200	메카트로닉스학부	10	
10	공과대학		

부서 테이블의 SELF JOIN

SELECT c.deptno, c.dname, c.college d.dname college_name, ... FROM department c, department d

WHERE c.college = d.deptno;

조인 결과

DEPTNO	DNAME	COLLEGE	COLLEGE_NAME	LOC
101	컴퓨터공학과	100	정보미디어학부	1호관
102	멀티미디어학과	100	정보미디어학부	2호관
100	정보미디어학부	10	공과대학	
200	메카트로닉스학부	10	공과대학	
10	공과대학			



❖ JOIN ∼ ON 절을 사용한 SELF JOIN

오라클 9i에서 JOIN ~ ON 지원

❖ 사용법

```
SELECT table.column, table2.column
```

FROM table1 JOIN table2

ON table1.column = table2.column;





❖ 사용 예

- 부서 테이블에서 SELF JOIN을 이용하여 부서 이름과 상위 부서의 이름을 출력하여라.
- WHERE절을 사용한 SELF JOIN방법

```
SQL> SELECT dept.dname || '의 소속은 ' || org.dname
2 FROM department dept, department org
3 WHERE dept.college = org.deptno;
```

■ JOIN~ON절을 사용한 SELF JOIN방법

SELF JOIN

❖ 사용 예

- 부서 번호가 201 이상인 부서 이름과 상위 부서의 이름을 출력하여라.
- WHERE절을 사용한 SELF JOIN방법

■ JOIN~ON절을 사용한 SELF JOIN방법



