



Oracle 4강 –JOIN

양 명 속

[now4ever7@gmail.com]



목차

- Inner join
- Outer join
- SELF Join
- 카티션 곱(Cross join)



실습-group by

- panmae 테이블에서 판매일(p_date)별로 판매금액(p_total)의 합계 구하기
- panmae 테이블에서 수량(p_qty)이 3개 이상인 데이터에 대해 판매일(p_date)별, 판매점(p_store)별로 판매금액(p_total)의 합계, 건수 구하기
 - 그 결과에서 건수가 2 이상인 데이터만 조회
- panmae 테이블에서 상품코드(p_code)별로 판매금액(p_total)의 합계, 수량(p_qty)의 합계, 건수, 판매금액(p_total)의 평균 구하기
 - 그 결과에서 판매금액(p_total)의 합계가 10000 이상이고, 수량(p_qty)의 합계가 15 이상인 데이터만 조회



Join

■ Join

- 각각의 테이블에 분리되어 있는 연관성 있는 데이터들을 연결하거나 조합하는 일련의 작업들
- 여러 테이블에 흩어져 있는 정보 중에서 사용자가 필요한 정보만 가져와서 가상의 테이블간 **공통된 열을 기준으로** 검색
- 테이블의 논리적 관계를 이용하여 두 개 또는 그 이상의 테이블에서 결과 검색
- 논리적 관계의 요소는 공통된 열이 필요.

■ 관계형 데이터베이스

- 모든 정보들이 하나의 테이블에 몰려 있는 것이 아니라 여러 곳의 테이블에 정규화되어 흩어져 있다는 뜻
- 각 테이블끼리는 서로 특정한 규칙이나 관계가 있도록 설계됨
 - -> 사용자의 정보가 여러 곳에 흩어져 있기 때문에 조회를 여러 번 해야 한다는 문제 발생
 - => Join 기법 등장

Join

사원 테이블

사원번호	이름	부서번호
1000	홍길동	100
1001	김연아	100
1002	손연재	201

부서 테이블

부서번호	부서명	위치
100	영업부	서울
101	관리부	서울
201	생산부	부산

사원번호	이름	부서명
1000	홍길동	영업부
1001	김연아	영업부
1002	손연재	생산부



Join

■ Join

- 오라클용 Join - 오라클 제품에서 사용되는 join
- 표준 (ANSI) Join - 모든 제품들에서 공통적으로 사용할 수 있는 Join

■ Join의 종류

- 내부 조인(Inner Join)
- 외부 조인(Outer Join)
- 셀프 조인(Self Join)
- Cross 조인(Cross Join)

Inner join

- Inner Join – 등가 join
 - 일반적으로 많이 사용되는 조인으로 where 절에 기술되는 Join 조건으로 양쪽 테이블에 같은 조건이 존재할 경우의 값만을 가져오는 조인
 - Equal 연산자를 사용해서 EQUI Join 이라고 함
- 예제) 학생 테이블 (student)과 학과 테이블 (department)을 사용하여 학생이름, 1전공 학과번호 (deptno1), 1전공 학과이름을 출력하시오.

```
SQL> select s.name "학생이름", s.deptno1 "학과번호", d.dname "학과이름"
2   from student s, department d
3   where s.deptno1 = d.deptno;
```

학생이름	학과번호	학과이름
서진수	101	컴퓨터공학과
서재수	102	멀티미디어공학과
이미재	103	소프트웨어공학과
김재수	201	전자공학과
박동호	202	기계공학과
김신영	101	컴퓨터공학과
김신영	102	멀티미디어공학과
오나라	202	기계공학과
구유미	301	문헌정보학과
임세현	201	전자공학과
일지매	101	컴퓨터공학과
김진욱	102	멀티미디어공학과
아관훈	201	전자공학과
김문호	201	전자공학과
김문호	301	문헌정보학과
노정호	101	컴퓨터공학과
이윤나	201	전자공학과
이영민	201	전자공학과
김주현	102	멀티미디어공학과
허우	103	소프트웨어공학과

20 개의 행이 선택되었습니다.

```
select studno, name, grade, deptno1, profno
from student;
```

STUDNO	NAME	GRADE	DEPTNO1	PROFNO
9411	서진수	4	101	1001
9412	서재수	4	102	2001
9413	이미경	4	103	3002
9414	김재수	4	201	4001
9415	박동호	4	202	4003
9511	김신영	3	101	1002
9512	신은경	3	102	2002
9513	오나라	3	202	4003
9514	구유미	3	301	4007
9515	임세현	3	201	4001
9611	일지매	2	101	1002
9612	김진욱	2	102	2001
9613	안광훈	2	201	4002
9614	김문호	2	201	4003
9615	노정호	2	301	4007
9711	이윤나	1	101	
9712	안은수	1	201	
9713	인영민	1	201	
9714	김주현	1	102	
9715	허우	1	103	

```
select deptno, dname, build from department;
```

DEPTNO	DNAME	BUILD
101	컴퓨터공학과	정보관
102	멀티미디어공학과	멀티미디어관
103	소프트웨어공학과	소프트웨어관
201	전자공학과	전자제어관
202	기계공학과	기계실험관
203	화학공학과	화학실습관
301	문헌정보학과	인문관
100	컴퓨터정보학부	
200	메카트로닉스학부	
300	인문사회학부	
10	공과대학	
20	인문대학	


```
select s.name "학생이름", s.deptno1 "학과번호", d.dname "학과이름"
from student s, department d
where s.deptno1 = d.deptno;
```

학생이름	학과번호	학과이름
서진수	101	컴퓨터공학과
서재수	102	멀티미디어공학과
이미경	103	소프트웨어공학과
김재수	201	전자공학과
박동호	202	기계공학과
김신영	101	컴퓨터공학과
신은경	102	멀티미디어공학과
오나라	202	기계공학과
구유미	301	문헌정보학과
임세현	201	전자공학과
일지매	101	컴퓨터공학과
김진욱	102	멀티미디어공학과
안광훈	201	전자공학과
김문호	201	전자공학과
노정호	301	문헌정보학과
이윤나	101	컴퓨터공학과
안은수	201	전자공학과
인영민	201	전자공학과
김주현	102	멀티미디어공학과
허우	103	소프트웨어공학과



Inner join

Oracle join 구문

```
select s.name "학생이름", s.deptno1 "학과번호", d.dname "학과이름"  
from student s, department d  
where s.deptno1 = d.deptno;
```

ANSI join 구문

```
select s.name "학생이름", s.deptno1 "학과번호", d.dname "학과이름"  
from student s join department d  
On s.deptno1 = d.deptno;
```

Inner join

- 예제) 학생 테이블(student)과 교수 테이블(professor)을 join하여 학생이름, 지도교수 번호, 지도교수 이름을 출력하시오

```
SQL> select s.name "학생이름", s.profno "교수번호", p.name "교수이름"
2  from student s, professor p
3  where s.profno = p.profno;
```

학생이름	교수번호	교수이름
서진수	1001	조인형
서재수	2001	양선희
이미경	3002	나한열
김재수	4001	심신원
박동호	4003	박원범
김신영	1002	박승민
김은영	2002	김영준
오나라	4003	박원범
구유미	4007	허이신
임세현	4001	심신원
일지매	1002	박승민

학생이름	교수번호	교수이름
김진욱	2001	양선희
안광훈	4002	최슬기
김문호	4003	박원범
노정호	4007	허이신

15 개의 행이 선택되었습니다.

```
select studno, name, grade, deptno1, profno
from student;
```

STUDNO	NAME	GRADE	DEPTNO1	PROFNO
9411	서진수	4	101	1001
9412	서재수	4	102	2001
9413	이미경	4	103	3002
9414	김재수	4	201	4001
9415	박동호	4	202	4003
9511	김신영	3	101	1002
9512	신은경	3	102	2002
9513	오나라	3	202	4003
9514	구유미	3	301	4007
9515	임세현	3	201	4001
9611	일지매	2	101	1002
9612	김진욱	2	102	2001
9613	안광훈	2	201	4002
9614	김문호	2	201	4003
9615	노정호	2	301	4007
9711	이윤나	1	101	
9712	안은수	1	201	
9713	인영민	1	201	
9714	김주현	1	102	
9715	허우	1	103	

```
select profno, name, position, pay, deptno
from professor;
```

PROFNO	NAME	POSITION	PAY	DEPTNO
1001	조인형	정교수	550	101
1002	박승곤	조교수	380	101
1003	송도권	전임강사	270	101
2001	양선희	전임강사	250	102
2002	김영조	조교수	350	102
2003	주승재	정교수	490	102
3001	김도형	정교수	530	103
3002	나한열	조교수	330	103
3003	김현정	전임강사	290	103
4001	심슨	정교수	570	201
4002	최슬기	조교수	330	201
4003	박원범	조교수	310	202
4004	차범철	전임강사	260	202
4005	바비	정교수	500	203
4006	전민	전임강사	220	301
4007	허은	조교수	290	301

```
select s.name "학생이름", s.profno "교수번호", p.name "교수이름"
from student s, professor p
where s.profno = p.profno;
```

학생이름	교수번호	교수이름
서진수	1001	조인형
서재수	2001	양선희
이미경	3002	나한열
김재수	4001	심슨
박동호	4003	박원범
김신영	1002	박승곤
신은경	2002	김영조
오나라	4003	박원범
구유미	4007	허은
임세현	4001	심슨
일지매	1002	박승곤
김진욱	2001	양선희
안광훈	4002	최슬기
김문호	4003	박원범
노정호	4007	허은



Inner join

Oracle join 구문

```
select s.name "학생이름", s.profno "교수번호", p.name "교수이름"  
from student s, professor p  
where s.profno = p.profno;
```

ANSI join 구문

```
select s.name "학생이름", s.profno "교수번호", p.name "교수이름"  
from student s join professor p  
On s.profno = p.profno;
```

양쪽 테이블에 모두 데이터가 존재해야 결과가 나옴

5명의 학생은 지도교수가 결정이 안 된 상태라서 결과에서 빠졌음

Inner join

- 예제) 학생 테이블(student)과 학과 테이블(department), 교수 테이블(professor)을 join하여 학생이름, 학과 이름, 지도교수 이름을 출력하시오

```
SQL> select s.name "학생이름", d.dname "학과이름", p.name "교수이름"
2   from student s, department d, professor p
3   where s.deptno1 = d.deptno
4   and s.profno = p.profno;
```

학생이름	학과이름	교수이름
서진수	컴퓨터공학과	조인형
서재수	멀티미디어공학과	양선희
이미경	소프트웨어공학과	나한열
김재수	전자공학과	심승원
박동호	기계공학과	박원범
김신영	컴퓨터공학과	박승영
신은영	멀티미디어공학과	김영원
오나라	기계공학과	박원준
구유미	문헌정보학과	허신준
임세현	전자공학과	심승원
일지매	컴퓨터공학과	박승원

학생이름	학과이름	교수이름
김진욱	멀티미디어공학과	양선희
안광호	전자공학과	최슬기
김문호	전자공학과	박원범
노정호	문헌정보학과	허신준

15 개의 행이 선택되었습니다.

select studno, name, grade, deptno1, profno
from **student**;

STUDNO	NAME	GRADE	DEPTNO1	PROFNO
9411	서진수	4	101	1001
9412	서재수	4	102	2001
9413	이미경	4	103	3002
9414	김재수	4	201	4001
9415	박동호	4	202	4003
9511	김신영	3	101	1002
9512	신은경	3	102	2002
9513	오나라	3	202	4003
9514	구유미	3	301	4007
9515	임세현	3	201	4001
9611	일지매	2	101	1002
9612	김진욱	2	102	2001
9613	안광훈	2	201	4002
9614	김문호	2	201	4003
9615	노정호	2	301	4007
9711	이윤나	1	101	
9712	안은수	1	201	
9713	인영민	1	201	
9714	김주현	1	102	
9715	허우	1	103	

select deptno, dname, build from **department**;

DEPTNO	DNAME	BUILD
101	컴퓨터공학과	정보관
102	멀티미디어공학과	멀티미디어관
103	소프트웨어공학과	소프트웨어관
201	전자공학과	전자제어관
202	기계공학과	기계실험관
203	화학공학과	화학실습관
301	문헌정보학과	인문관
100	컴퓨터정보학부	
200	메카트로닉스학부	
300	인문사회학부	
10	공과대학	
20	인문대학	

PROFNO	NAME	POSITION	PAY	DEPTNO
1001	조인형	정교수	550	101
1002	박승곤	조교수	380	101
1003	송도권	전임강사	270	101
2001	양선희	전임강사	250	102
2002	김영조	조교수	350	102
2003	주승재	정교수	490	102
3001	김도형	정교수	530	103
3002	나한열	조교수	330	103
3003	김현정	전임강사	290	103
4001	심슨	정교수	570	201
4002	최슬기	조교수	330	201
4003	박원범	조교수	310	202
4004	차범철	전임강사	260	202
4005	바비	정교수	500	203
4006	전민	전임강사	220	301
4007	허은	조교수	290	301

select profno, name,
position, pay, deptno
from **professor**;


```

select s.name "학생이름", d.dname "학과이름", p.name "교수이름"
from student s, department d, professor p
where s.deptno1 = d.deptno
and s.profno = p.profno;

```

학생이름	학과이름	교수이름
서진수	컴퓨터공학과	조인형
서재수	멀티미디어공학과	양선희
이미경	소프트웨어공학과	나한열
김재수	전자공학과	심슨
박동호	기계공학과	박원범
김신영	컴퓨터공학과	박승곤
신은경	멀티미디어공학과	김영조
오나라	기계공학과	박원범
구유미	문헌정보학과	허은
임세현	전자공학과	심슨
일지매	컴퓨터공학과	박승곤
김진욱	멀티미디어공학과	양선희
안광훈	전자공학과	최슬기
김문호	전자공학과	박원범
노정호	문헌정보학과	허은



Inner join

Oracle join 구문

```
select s.name "학생이름", d.dname "학과이름", p.name "교수이름"  
from student s, department d, professor p  
where s.deptno1 = d.deptno  
and s.profno = p.profno;
```

ANSI join 구문

```
1 select s.name "학생이름", d.dname "학과이름", p.name "교수이름"  
2 from student s join department d  
3 on s.deptno1 = d.deptno  
4 Join professor p  
5 on s.profno = p.profno;
```

3개의 테이블 조인

ANSI Join – 쿼리의 2번과 3번 라인의 조건으로 먼저 조인을 수행 후 나온 결과 값을 가지고 4번과 5번 라인의 조건으로 조인을 수행함
조인 테이블이 더 늘어날 경우 계속 **Join ~ on** 조건을 추가하면 됨

Inner join

- 예제) emp2 테이블과 학과 p_grade 테이블을 join하여 사원이름, 직급, 현재연봉, 해당 직급의 연봉의 하한 금액과 상한 금액을 출력하시오

```
SQL> select e.name "사원이름", e.position "현재직급", e.pay "현재연봉",  
2 p.s_pay "하한금액", p.e_pay "상한금액"  
3 from emp2 e, p_grade p  
4 where e.position = p.position;
```

사원이름	현재직급	현재연봉	하한금액	상한금액
전부장	부장	72000000	60010000	75000000
최일도	과장	50000000	45010000	51000000
백원만	차장	60000000	51010000	60000000
천만득	과장	56000000	45010000	51000000
이지매	부장	75000000	60010000	75000000
유관순	과장	51000000	45010000	51000000
김문호	대리	35000000	30000000	45000000
노정호	부장	68000000	60010000	75000000
이윤나	과장	49000000	45010000	51000000

9 개의 행이 선택되었습니다.



Inner join

Oracle join 구문

```
select e.name "사원이름", e.position "현재직급", e.pay "현재연봉",  
p.s_pay "하한금액", p.e_pay "상한금액"  
from emp2 e, p_grade p  
where e.position = p.position;
```

ANSI join 구문

```
select e.name "사원이름", e.position "현재직급", e.pay "현재연봉",  
p.s_pay "하한금액", p.e_pay "상한금액"  
from emp2 e join p_grade p  
on e.position = p.position;
```

Inner join

예제) 1전공(deptno1)이 101번인 학생들의 학생이름과 지도교수 이름을 출력하시오.

```
SQL> select s.name "학생이름", p.name "지도교수명"  
2   from student s, professor p  
3   where s.profno = p.profno  
4   and s.deptno1 = 101;
```

학생이름	지도교수명
서진수	조인형
김신영	박승근
일지매	박승근

조인 조건과 검색 조건이 동시에 있을 경우 검색 조건을 먼저 검색해서 데이터의 검색 범위를 줄여 놓고, 조인 작업을 수행함

Oracle join 구문

```
select s.name "학생이름", p.name "지도교수명"  
from student s, professor p  
where s.profno = p.profno  
and s.deptno1 = 101;
```

ANSI join 구문

```
select s.name "학생이름", p.name "지도교수명"  
from student s join professor p  
on s.profno = p.profno --조인 조건  
and s.deptno1 = 101; -- 검색 조건
```

Inner 조인

- emp2, dept2 테이블을 이용해서 사원 이름(name), 급여(pay), 직급(position), 부서명(dname)을 조회하시오

```
select e.name, e.pay, e.position, d.dname
from emp2 e, dept2 d
where E.DEPTNO = D.DCODE;
```

- emp2, dept2 테이블에서 부서이름별 pay의 평균 구하기

```
select d.dname, avg(e.pay)
from emp2 e, dept2 d
where E.DEPTNO = D.DCODE
group by d.dname;
```

DNAME	AVG(E.PAY)
재무관리팀	60000000
S/W지원	27500000
H/W지원	35500000
영업기획팀	49000000
영업3팀	26000000
영업4팀	25000000
총무팀	56000000
영업1팀	26000000
사장실	100000000
영업2팀	26000000
경영지원부	61000000
기술부	75000000
영업부	68000000

[실습] 위의 예제의 결과에서 부서 이름이 영업으로 시작하는 부서의 평균만 조회하시오.

DNAME	AVG(E.PAY)
영업기획팀	49000000
영업3팀	26000000
영업4팀	25000000
영업1팀	26000000
영업2팀	26000000
영업부	68000000

실습

- emp, dept 테이블에서 부서번호,사원명,직업,부서명,지역 조회 단, 직업(job)이 CLERK인 사원 데이터만 조회
- emp, dept 테이블에서 부서번호,사원명,직업,부서명,지역 조회 단, 직업(job)이 CLERK인 사원이거나 Manager인 사원만 조회

(비교 연산자 우선순위 - and가 우선순위가 높다.)

- emp, dept 테이블에서 지역(loc)별 급여(sal)의 평균 조회
 - Join, group by 모두 이용

LOC	평균급여
NEW YORK	2916.7
BOSTON	9500
CHICAGO	1566.7
DALLAS	2175



실습

- student 테이블과 exam_01 테이블을 조회하여 학생들의 학번, 이름, 점수, 학점을 출력하시오 (학점은 dcode나 case이용- 90 이상이면 'A', 80이상이면 'B', 70이상이면 'C', 60이상이면'D' 60미만이면 'F')

STUDNO	NAME	TOTAL
9411	서진수	97
9412	서재수	78
9413	이미경	83
9414	김재수	62
9415	박동호	88
9511	김신영	92
9512	신은경	87
9513	오나라	81



실습 - hr 계정

- employees , jobs 테이블에서 first_name, salary, job_title 조회하기
- employees , departments, locations 테이블에서 city, department_name, job_id, salary 조회하기
- employees , departments, locations 테이블에서 city별, department_name 별, job_id 별로 그룹화하여 city, department_name, job_id, 인원수, 급여 (salary)합계 구하기



OUTER Join

- Inner Join

- join에 참여하는 모든 테이블에 데이터가 존재하는 경우에만 결과 값을 출력함

- Outer Join

- Inner join 과는 반대로 한쪽 테이블에는 데이터가 있고 한쪽 테이블에 없는 경우에 데이터가 있는 쪽 테이블의 내용을 전부 출력하게 하는 방법
- DB 성능에 아주 나쁜 영향을 줄 수 있으므로 주의
 - A 테이블과 B 테이블을 Outer Join을 수행해서 A 테이블에 있는 데이터를 모두 검색하는 경우, 만약 A 테이블에 인덱스가 있어도 인덱스를 쓰지 않고 Full Scan 을 하기 때문

OUTER Join

- 예제) student 테이블과 professor 테이블을 조인하여 학생이름과 지도교수 이름을 출력하시오.
단, 지도교수가 결정되지 않은 학생의 명단도 함께 출력하시오.
(학생 데이터는 전부 출력되도록)

```
SQL> set null *****;
SQL> select s.name "학생이름", p.name "교수이름"
2   from student s, professor p
3   where s.profno=p.profno(+);
```

학생이름	교수이름
서진수	조인형
서재수	양선희
이미경	나한열
김재수	심승원
박동호	박원범
김신영	김영준
김시은	김원범
오나라	허은
구유미	심승원
임세현	박승곤
일지매	
학생이름	교수이름
김진욱	양선희
안광훈	최기범
김민호	박원범
김민정	허은
노정호	*****
이준호	*****
이영민	*****
이영주	*****
김민우	*****
허우	*****

20 개의 행이 선택되었습니다.

OUTER Join

Oracle Outer join 구문

(+) 기호

- 학생은 존재하지만 교수가 없는 내용을 출력해야 하기에 교수 조건 쪽에 (+) 기호를 붙인 것.
- 데이터가 없는 쪽에 표시를 함

```
select s.name "학생이름", p.name "교수이름"  
from student s, professor p  
where s.profno=p.profno(+);
```

ANSI Outer join 구문

학생 데이터는 전부 출력

```
select s.name "학생이름", p.name "교수이름"  
from student s left outer join professor p  
on s.profno=p.profno;
```

Oracle Outer Join과 반대의 표현법 사용

LEFT Outer Join 구문

- 데이터가 있는 쪽에 표시를 함
- On 조건절 중에 학생은 존재하고 교수가 없으니 왼쪽 조건에는 데이터가 있는 행을 출력하게 하기 위해 Left Outer Join 구문 사용

OUTER Join

- 예제) student 테이블과 professor 테이블을 조인하여 학생이름과 지도교수 이름을 출력하시오.

단, 지도학생이 결정되지 않은 교수의 명단도 함께 출력하시오.

(교수 데이터는 전부 출력되도록)

```
SQL> select s.name "학생이름", p.name "교수이름"
2   from student s, professor p
3   where s.profno(+) = p.profno
4   order by 1;
```

학생이름	교수이름
구유미	허원식
김문신	박승진
김재진	박승진
김진욱	서희원
김정호	허원식
노동호	박희정
서재수	양조기
서진수	김영수
신광호	최기
안광훈	최기
학생이름	교수이름
오나라	박원식
이미경	나한식
일지매	박승진
임세현	김승진
*****	김승진
*****	김승진
*****	김승진
*****	김승진
*****	김승진
*****	김승진
*****	김승진
*****	김승진

22 개의 행이 선택되었습니다.



OUTER Join

Oracle Outer join 구문

교수 데이터는 전부 출력

```
select s.name "학생이름", p.name "교수이름"  
from student s, professor p  
where s.profno(+) = p.profno  
order by 1;
```

ANSI Outer join 구문

```
select s.name "학생이름", p.name "교수이름"  
from student s right outer join professor p  
on s.profno = p.profno  
order by 1;
```

OUTER Join

- 예제) student 테이블과 professor 테이블을 조인하여 학생이름과 지도교수 이름을 출력하시오.

단, 지도학생이 결정되지 않은 교수의 명단과 지도교수가 결정 안된 학생 명단을 한꺼번에 출력하시오.

- 두 가지 outer join의 결과를 합쳐서 만들어야 함
- Oracle outer join에서는 지원하지 않음 => 두 outer join을 각각 수행한 후 Union을 사용하여 결과를 인위적으로 합쳐서 출력함
- Ansi outer join에서는 훨씬 간단한 방법을 제공함

```
SQL> select s.name "학생이름", p.name "교수이름"
2   from student s FULL outer join professor p
3   on s.profno=p.profno;
```

학생이름	교수이름
------	------

서진수	조인형
서재수	양선희
이미경	나한열
김재수	심원섭
박동호	박영준
김신영	김영준
신나라	박희승
오유미	허신
구세현	박승곤
일지매	

학생이름	교수이름
------	------

김진욱	양선희
아광호	최슬기
김문호	박원범
노정호	허영민
이준호	*****
안영민	*****
김주현	*****
허우	*****
*****	송도권
*****	주승재

학생이름	교수이름
------	------

*****	김도형
*****	김현정
*****	차범철
*****	바비
*****	전민

27 개의 행이 선택되었습니다.



OUTER Join

Oracle Outer join 구문

```
select s.name "학생이름", p.name "교수이름"  
from student s, professor p  
where s.profno(+) = p.profno  
UNION  
select s.name "학생이름", p.name "교수이름"  
from student s, professor p  
where s.profno = p.profno(+);
```

ANSI Outer join 구문

```
select s.name "학생이름", p.name "교수이름"  
from student s FULL outer join professor p  
on s.profno = p.profno;
```


SELF Join

- Join - 원하는 데이터들이 여러 테이블에 흩어져 있어서 여러 테이블을 조인해야 함
- 원하는 데이터가 하나의 테이블에 다 들어 있을 경우
 - 예) dept2 테이블
 - Dname은 부서명, pdept는 상위부서
 - ‘경영지원부의 상위부서는 사장실이다’ 형태로 출력하려면, self join 사용
 - 하나의 테이블을 메모리상에서 별명을 두 개로 사용해서 가상으로 2개의 테이블로 만든 후 조인작업을 수행

select * from dept2;

DCODE	DNAME	PDEPT	AREA
0001	사장실		포항본사
1000	경영지원부	0001	서울지사
1001	재무관리팀	1000	서울지사
1002	총무팀	1000	서울지사
1003	기술부	0001	포항본사
1004	H/W지원	1003	대전지사
1005	S/W지원	1003	경기지사
1006	영업부	0001	포항본사
1007	영업기획팀	1006	포항본사
1008	영업1팀	1007	부산지사
1009	영업2팀	1007	경기지사
1010	영업3팀	1007	서울지사
1011	영업4팀	1007	울산지사

select * from dept2;

DCODE	DNAME	PDEPT	AREA
0001	사장실		포항본사
1000	경영지원부	0001	서울지사
1001	재무관리팀	1000	서울지사
1002	총무팀	1000	서울지사
1003	기술부	0001	포항본사
1004	H/W지원	1003	대전지사
1005	S/W지원	1003	경기지사
1006	영업부	0001	포항본사
1007	영업기획팀	1006	포항본사
1008	영업1팀	1007	부산지사
1009	영업2팀	1007	경기지사
1010	영업3팀	1007	서울지사
1011	영업4팀	1007	울산지사

왼쪽의 pdept번호와 오른쪽 테이블의 dcode 번호가 같은 부서가 상위부서명

SELF Join

Oracle join 구문

```
select a.dname "부서명", b.dname "상위부서명"
from dept2 a, dept2 b
where a.pdept = b.dcode;
```

ANSI join 구문

```
select a.dname "부서명", b.dname "상위부서명"
from dept2 a join dept2 b
on a.pdept = b.dcode;
```

부서명	상위부서명
경영지원부	사장실
재무관리팀	경영지원부
총무팀	경영지원부
기술부	사장실
H/W지원	기술부
S/W지원	기술부
영업부	사장실
영업기획팀	영업부
영업1팀	영업기획팀
영업2팀	영업기획팀
영업3팀	영업기획팀
영업4팀	영업기획팀

- 사장실은 상위부서가 없어서 부서명에서 조회되지 않음
- 사장실까지 출력하려면 **outer join** 사용

실습 – outer join, self join

- 1. emp2, p_grade 테이블에서 name(사원이름), position(직급), 시작연도(s_year), 끝연도(e_year)을 조회
 - 단, emp2 테이블의 데이터는 전부 출력되도록 할 것
- 2. EMP Table에 있는 EMPNO와 MGR을 이용하여 서로의 관계를 다음과 같이 출력하라. ‘FORD의 매니저는 JONES’

사원이름	매니저 이름
FORD	JONES
SCOTT	JONES
JAMES	BLAKE
TURNER	BLAKE
MARTIN	BLAKE
WARD	BLAKE
ALLEN	BLAKE
MILLER	CLARK
ADAMS	SCOTT
CLARK	KING
BLAKE	KING
JONES	KING
SMITH2	FORD



카티션 곱(Cartesian Product)

■ 카티션 곱- CROSS JOIN

- join 쿼리 중에 **where 절**에 기술하는 join조건이 잘못 기술되었거나 아예 없을 경우 발생
- ANSI Join에서는 **CROSS JOIN** 이라고도 부름
- Join 작업에 참조되는 테이블 행수를 모두 곱한 값의 결과가 만들어짐
- 예) 사원 테이블, 부서 테이블을 join 해서 사원이름과 부서명 출력

Oracle join 구문

```
select e.ename, d.dname  
from emp e, dept d;
```

ANSI join 구문

```
select e.ename, d.dname  
from emp e CROSS JOIN dept d;
```

결과 : 60건의 데이터가 출력됨
Emp 테이블 15건, dept 테이블 4건 => $15 * 4 = 60$ 건