

4장. JOIN

- ❖ Cartesian Product(카티션 곱)
- ❖ EQUI Join(등가조인)
- ❖ Non Equi Join(비등가 조인)
- ❖ Out Join(아우터 조인)
- ❖ SELF Join

JOIN

사원 테이블

사번	이름	부서번호
1000	홍길동	100
1001	일지매	101
1002	강감찬	201

부서 테이블

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



이름	부서명
홍길동	영업부
일지매	관리부
.....

Cartesian Product(카티션 곱, CROSS JOIN)

□ Cartesian Product

- ▣ 두 테이블 행수를 모두 곱한 값의 결과가 만들어 짐

□ Oracle Join 문법

```
SELECT e.ename , d.dname  
FROM emp e, dept d ;
```

□ ANSI Join 문법

```
SELECT e.ename , d.dname  
FROM emp e CROSS JOIN dept d ;
```

EQUI Join (등가 Join)

□ Equi join

- WHERE 절에 기술되는 JOIN 조건으로 양쪽 테이블에 같은 조건이 존재할 경우의 값만 가져 옴
- 예 1: 학생 테이블 (student) 과 학과 테이블 (department) 테이블을 사용하여 학생이름, 1 전공학과번호(deptno1) , 1전공 학과 이름을 출력하세요.
- Oracle Join 구문

```
SELECT s.name "학생이름", s.deptno1 "학과번호", d.dname "학과명"  
FROM student s, department d  
WHERE s.deptno1=d.deptno;
```

- Ansi Join 구문

```
SELECT s.name "학생이름", s.deptno1 "학과번호", d.dname "학과명"  
FROM student s JOIN department d  
ON(s.deptno1=d.deptno);
```

EQUI Join (등가 Join)

Equi Join 예 2:

학생 테이블 (student) 과 교수 테이블 (professor) 을 join 하여 학생의 이름과 지도교수번호, 지도교수 이름을 출력하세요.

학생이름	교수번호	교수이름
서진수	1001	조인형
서재수	2001	양선희
이미경	3002	한나연
김재수	4001	신원범
박동호	4003	박승원
김신호	1002	박영조
김은나	2002	김원범
오유미	4003	박원희
구세현	4007	허성준
임지매	4001	심원기
일진욱	1002	박기범
김광훈	2001	양선희
김문호	4002	최슬원
노정호	4003	박원희
	4007	허성준

15 rows selected.

SCOTT>

- 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);
```

3) Equi Join 예 3:

학생 테이블(student)과 학과 테이블(department) , 교수 테이블(professor) 을 Join하여 학생의 이름과 학과이름, 지도교수 이름을 출력하세요.

학생이름	학과이름	교수이름
매영수	컴퓨터공학과	고인형
김진욱	컴퓨터공학과	조희열
김진호	컴퓨터공학과	김영선
김진재	컴퓨터공학과	박원슬
김민호	컴퓨터공학과	박원슬
김광현	컴퓨터공학과	박원슬
김세현	컴퓨터공학과	박원슬
김재나	컴퓨터공학과	박원슬
박동호	컴퓨터공학과	박원슬
박정호	컴퓨터공학과	박원슬
유미	컴퓨터공학과	박원슬
김민준	컴퓨터공학과	박원슬
김민준	컴퓨터공학과	박원슬
김민준	컴퓨터공학과	박원슬
김민준	컴퓨터공학과	박원슬

15 rows selected.

SCOTT>

- Oracle Join 문법

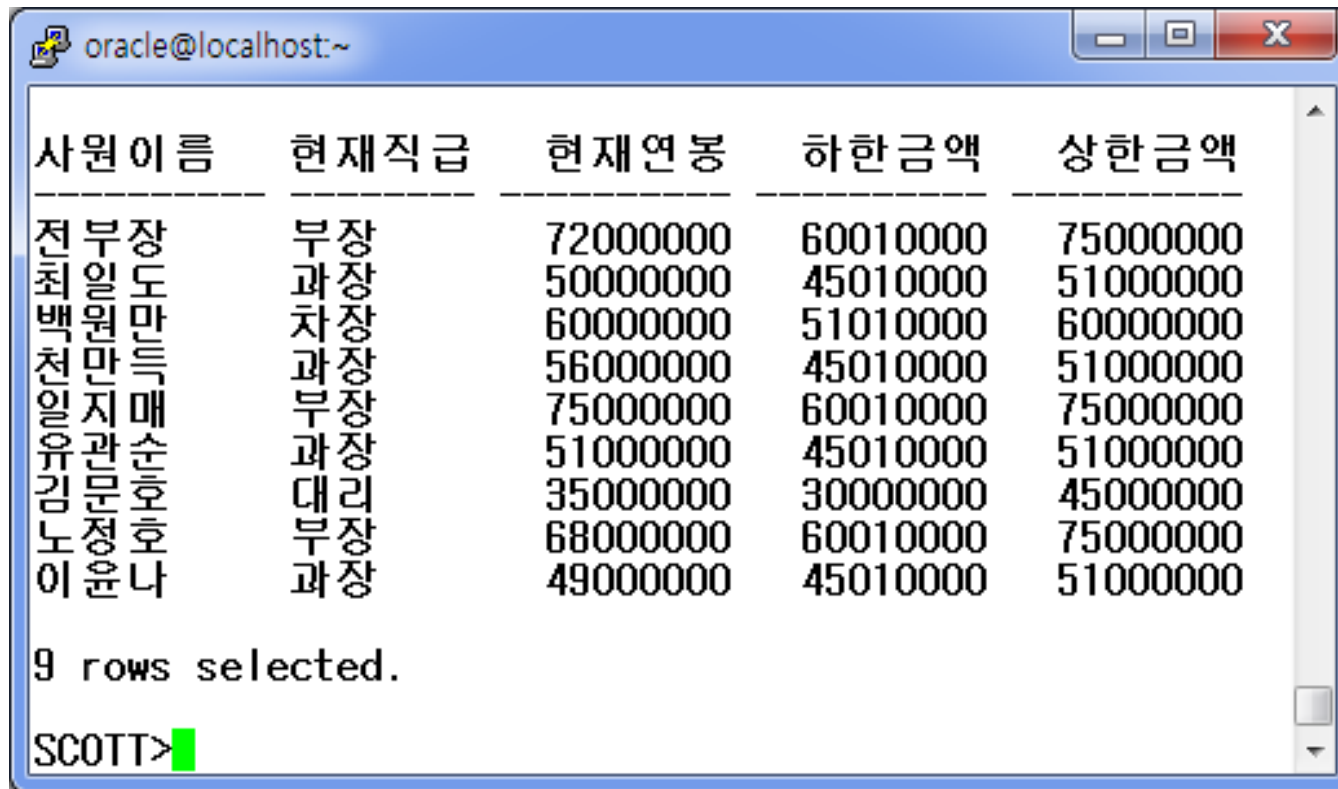
```
SELECT s.name "학생이름" , d.dname "학과이름" , p.name "교수이름"  
  FROM student s , department d , professor p  
  WHERE s.deptno1 = d.deptno  
  AND s.profno = p.profno ;
```

- ANSI Join 문법

```
SELECT s.name "학생이름" , d.dname "학과이름" , p.name "교수이름"  
  FROM student s JOIN department d  
  ON s.deptno1 = d.deptno  
  JOIN professor p  
  ON s.profno = p.profno ;
```

4) Equi Join 예 4 :

emp2 테이블과 p_grade 테이블을 조회하여 사원의 이름과 직급, 현재 연봉, 해당 직급의 연봉의 하한금액과 상한 금액을 출력하세요.



The screenshot shows a terminal window titled 'oracle@localhost:~'. It displays the result of an SQL query. The output is a table with 5 columns: '사원 이름' (Employee Name), '현재직급' (Current Job), '현재연봉' (Current Salary), '하한금액' (Lower Bound Salary), and '상한금액' (Upper Bound Salary). There are 9 rows of data. Below the table, it says '9 rows selected.' and the prompt 'SCOTT>' is visible.

사원 이름	현재직급	현재연봉	하한금액	상한금액
전부장	부장	72000000	60010000	75000000
최일도	과장	50000000	45010000	51000000
백원만	차장	60000000	51010000	60000000
천만득	과장	56000000	45010000	51000000
일지매	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 rows selected.

SCOTT>

- Oracle Join 문법

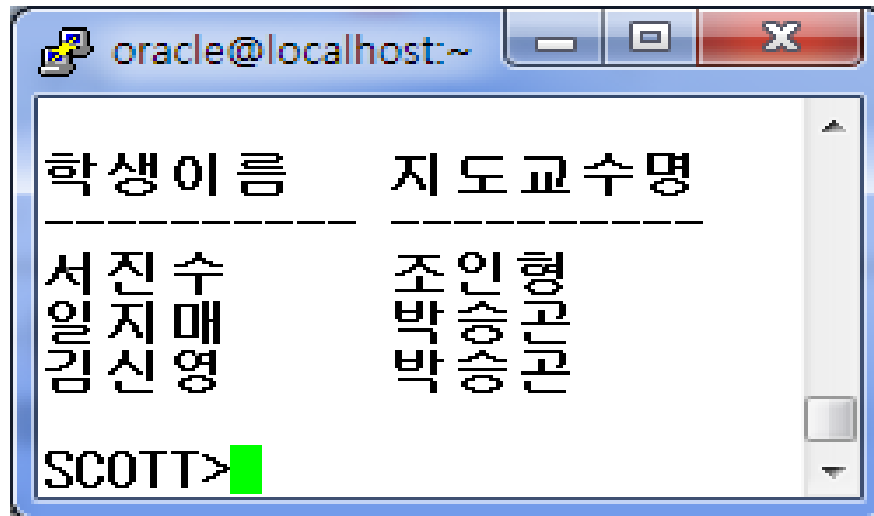
```
SELECT e.name "사원이름" ,  
        e.position "현재직급" ,  
        e.pay "현재연봉",  
        p.s_pay "하한금액" ,  
        p.e_pay "상한금액"  
FROM emp2 e , p_grade p  
WHERE e.position = p.position ;
```

- ANSI Join 문법

```
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 ;
```

5) Equi Join 예 5 :

1전공(deptno1)이 101번인 학생들의 학생 이름과 지도교수 이름을 출력하세요.



The screenshot shows a terminal window titled 'oracle@localhost:~'. It displays the result of an SQL query. The output is a table with two columns: '학생이름' (Student Name) and '지도교수명' (Advisor Name). The data is as follows:

학생이름	지도교수명
서진수	조인형
일지매	조인형
김신영	조인형

Below the table, the prompt 'SCOTT>' is visible with a green cursor.

- Oracle Join 문법

```
SCOTT>SELECT s.name "학생이름" , p.name "지도교수명"  
2 FROM student s , professor p  
3 WHERE s.profno = p.profno  
4 AND s.deptno1 = 101 ;
```

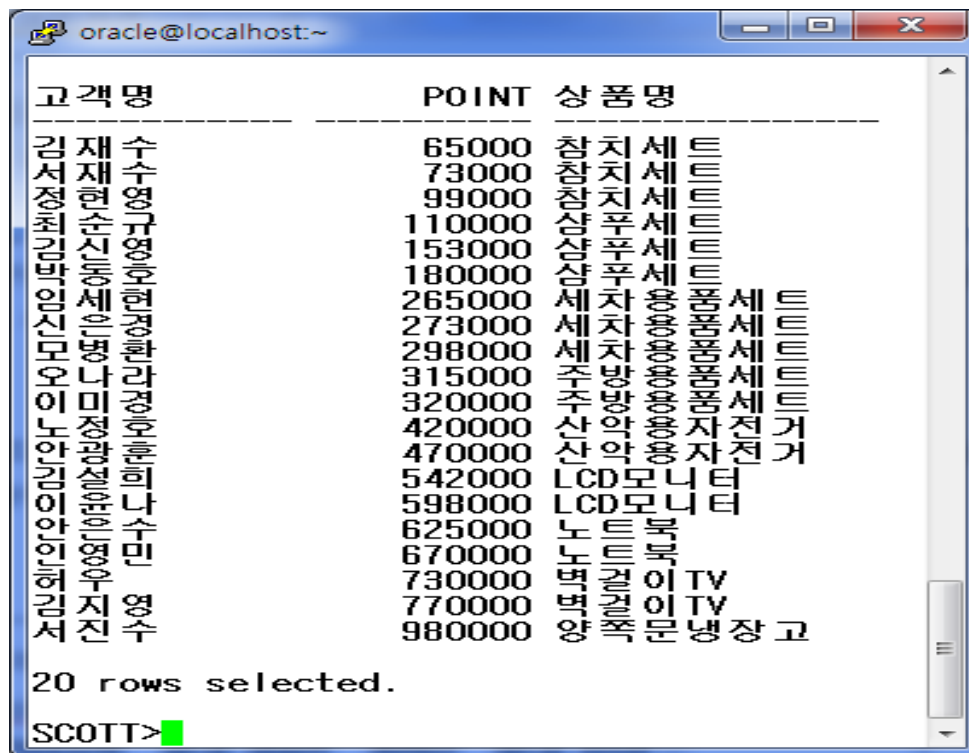
- ANSI Join 문법

```
SCOTT>SELECT s.name "학생이름" , p.name "지도교수명"  
2 FROM student s JOIN professor p  
3 ON s.profno = p.profno  
4 AND s.deptno1 = 101 ;
```

Non-Equi Join (비등가 Join)

1) Non-Equi Join 예제 1:

Gogak 테이블과 gift 테이블을 Join하여 고객의 마일리지 포인트별로 받을 수 있는 상품을 조회하여 고객의 이름과 상품 명을 출력하세요.



고객명	POINT	상품명
김재수	65000	치세트
김재수	73000	치세트
김재수	99000	치세트
김재수	110000	치세트
김재수	153000	치세트
김재수	180000	치세트
김재수	265000	치세트
김재수	273000	치세트
김재수	298000	치세트
김재수	315000	치세트
김재수	320000	치세트
김재수	420000	치세트
김재수	470000	치세트
김재수	542000	치세트
김재수	598000	치세트
김재수	625000	치세트
김재수	670000	치세트
김재수	730000	치세트
김재수	770000	치세트
김재수	980000	치세트

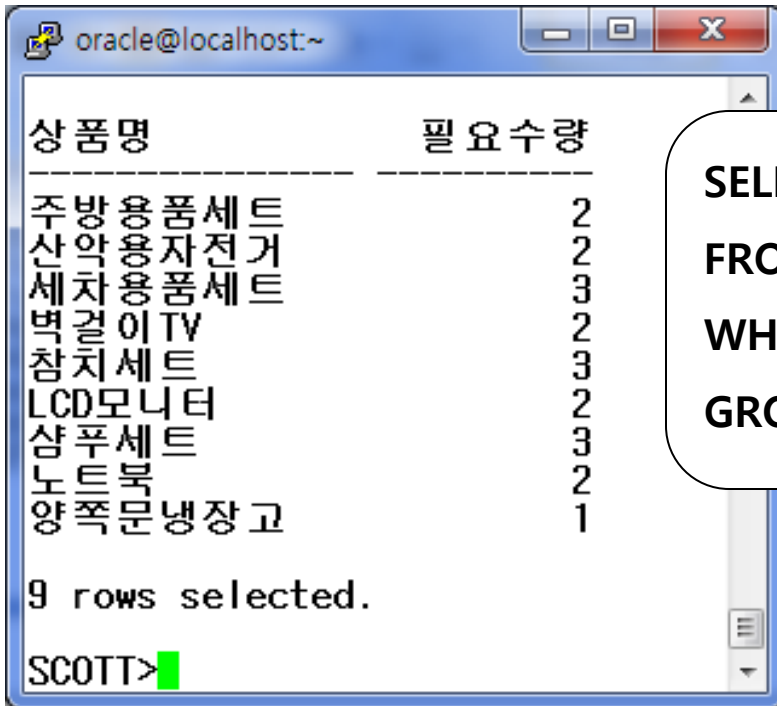
20 rows selected.

SCOTT>

```
SELECT go.gname "고객명", go.point "POINT" , gi.gname "상품명"  
FROM gogak go , gift gi  
WHERE go.point BETWEEN gi.g_start AND gi.g_end ;
```

2) Non-Equi Join 예제 2:

위 예제 1번에서 조회한 상품의 이름과 필요 수량이 몇 개 인지 조회하세요.



The screenshot shows an Oracle SQL*Plus window with the title 'oracle@localhost:~'. The query result is displayed in a table with two columns: '상품명' (Product Name) and '필요수량' (Required Quantity). The table contains 9 rows of data. Below the table, it says '9 rows selected.' and the prompt 'SCOTT>' is visible.

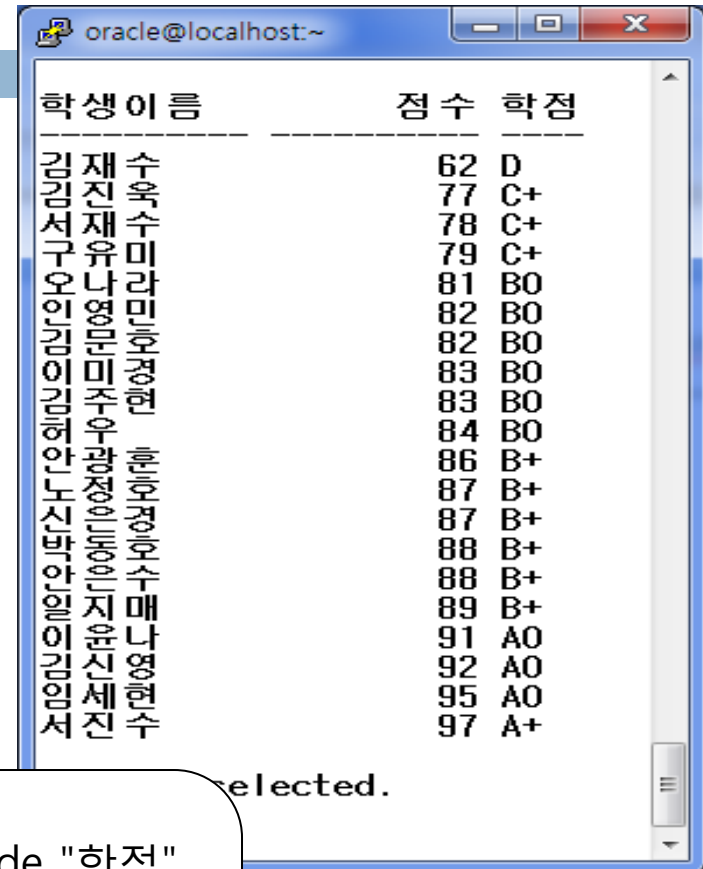
상품명	필요수량
주방용품세트	2
산악용자전거	2
세차용품세트	3
벽걸이TV	2
참치세트	3
LCD모니터	2
삼푸세트	3
노트북	2
양쪽문냉장고	1

9 rows selected.
SCOTT>

```
SELECT gi.gname "상품명" , COUNT(*) "필요수량"  
FROM gogak go , gift gi  
WHERE go.point BETWEEN gi.g_start AND g_end  
GROUP BY gi.gname ;
```

3) Non-Equi Join 예제 3:

Student 테이블과 exam_01
테이블 , hakjum 테이블을
조회하여 학생들의 이름과
점수와 학점을 출력하세요.



oracle@localhost:~

학생이름	점수	학점
김재수	62	D
김진욱	77	C+
서재수	78	C+
구유미	79	C+
오나라	81	B0
이영민	82	B0
김민호	82	B0
김지민	83	B0
김주우	83	B0
허광호	84	B0
안정호	86	B+
노정호	87	B+
신동호	87	B+
박지훈	88	B+
안지훈	88	B+
이재민	89	B+
김영수	91	A0
김세진	92	A0
김임서	95	A0
	97	A+

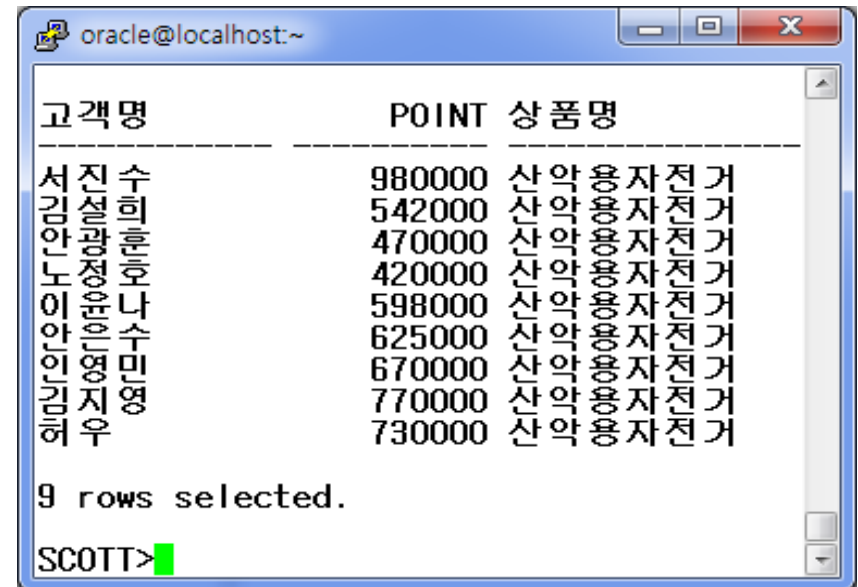
selected.

```
SELECT s.name "학생이름" , e.total "점수" , h.grade "학점"  
FROM student s , exam_01 e , hakjum h  
WHERE s.studno = e.studno  
AND e.total BETWEEN h.min_point AND h.max_point ;
```

4) Non-Equi Join 예제 4 :

Gogak 테이블과 gift 테이블을 Join하여 고객이 자기 포인트보다 낮은 포인트의 상품 중 한가지를 선택할 수 있다고 할 때 산악용 자전거를 선택할 수 있는 고객명과 포인트, 상품명을 출력하세요.

```
SELECT go.gname "고객명" ,  
       go.point "POINT" ,  
       gi.gname "상품명"  
FROM   gogak go , gift gi  
WHERE  gi.g_start <= go.point  
AND    gi.gname ='산악용자전거';
```



고객명	POINT	상품명
서진수	980000	산악용자전거
김설희	542000	산악용자전거
안광훈	470000	산악용자전거
노정호	420000	산악용자전거
이윤나	598000	산악용자전거
안희수	625000	산악용자전거
이영민	670000	산악용자전거
김지영	770000	산악용자전거
허우	730000	산악용자전거

9 rows selected.

SCOTT>

5) Non-Equi Join 예제 5:

Emp 2 테이블과 p_grade 테이블을 조회하여 직원들의 이름과 나이, 현재 직급, 예상 직급을 출력하세요. 예상 직급은 나이로 계산하며 해당 나이가 받아야 하는 직급을 의미합니다. 나이는 오늘(sysdate)를 기준으로 하되 trunc로 소수점 이하는 절삭해서 계산하세요.

```
SELECT e.name "이름",
       trunc((sysdate-e.birthdate)/365,0) "현재나이" ,
       e.position "현재직급" ,
       g.position "예상직급"
FROM emp2 e , p_grade g
WHERE trunc((sysdate-e.birthdate)/365,0)
      BETWEEN g.s_age AND g.e_age ;
```

이름	현재나이	현재직급	예상직급
이태백	27		대리
김설지	29		과장
박지오	30		과장
최오대	30		과장
강영남	31		과장
장영남	31		과장
정유민	31		과장
윤주하	31		과장
나정호	31		과장
노백일	35	부장	차장
최일남	35	과장	차장
	37	과장	부장
	38	과장	부장
	39	과장	부장
	39	과장	부장
	39	과장	부장
	40	대리	부장
	48	대표이사	이사

OUTER Join (아우터 조인) vs INNER Join (이너 조인)

1) Outer Join 예제 1:

Student 테이블과 Professor 테이블을 Join하여 학생이름과 지도교수 이름을 출력하세요. 단 지도교수가 결정되지 않은 학생의 명단도 함께 출력하세요.

- Oracle Outer Join 문법

```
SELECT s.name "학생이름", p.name "교수이름"  
FROM student s, professor p  
WHERE s.profno = p.profno(+);
```

- Oracle Outer Join 문법

```
SELECT s.name "학생이름", p.name "교수이름"  
FROM student s LEFT OUTER JOIN professor p  
ON s.profno = p.profno;
```

학생이름	교수이름
서일김김서신이임김아김오박노구하김이안이	조박박양양김나심심최박박하하
수매영옥수경경현수훈호라호호미	이승승선선영한스스슬원원원이이
진지신진재은미세재광문나동정유우주영이윤	형고고히히조열기범범범
현민수나	*****

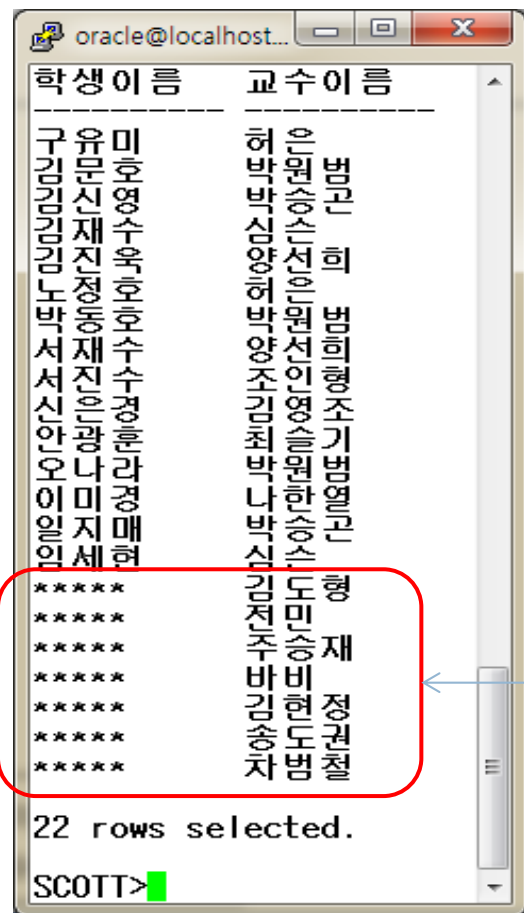
20 rows selected.

SCOTT>

지도교수가 없는 학생들입니다.
교수이름에 * 가 나오는 이유는
SQL> **set null ******* ; 로 설정했
기 때문입니다.

2) Outer Join 예제 2 :

Student 테이블과 Professor 테이블을 Join하여 학생이름과 지도교수 이름을 출력하세요. 단 **지도학생이 결정되지 않은 교수**의 명단도 함께 출력하세요.



학생이름	교수이름
구유미	허원범
김문신	박승선
김재진	심승선
김정호	양원선
김동재	양원선
노재진	조영수
박광나	최경훈
서미지	박나영
신안	박나영
오일	김도민
일세	김재
*****	정권
*****	철

22 rows selected.

SCOTT>

```
SELECT s.name "학생이름", p.name "교수이름"  
FROM student s , professor p  
WHERE s.profno(+) = p.profno  
ORDER BY 1 ;
```

지도학생이 결정
안 된 교수들입니다.

3) Outer Join 예제 3 :

**Student 테이블과 Professor 테이블을 Join하여
학생이름과 지도교수 이름을 출력하세요.
단 지도학생이 결정 안 된 교수 명단과
지도 교수가 결정 안된 학생 명단을
한꺼번에 출력하세요.**

지도 교수가 없는 학생들입니다

지도 학생이 없는 교수들입니다

[illegible]

- Oracle Outer Join 문법

```
SCOTT>SELECT s.name "학생이름", p.name "교수이름"  
2 FROM student s , professor p  
3 WHERE s.profno(+) = p.profno  
4 UNION  
5 SELECT s.name "학생이름", p.name "교수이름"  
6 FROM student s , professor p  
7 WHERE s.profno = p.profno(+);
```

- Oracle Outer Join 문법

```
SCOTT>SELECT s.name "학생이름", p.name "교수이름"  
2 FROM student s FULL OUTER JOIN professor p  
3 ON s.profno = p.profno ;
```

SELF Join

```
oracle@localhost:~  
SCOTT>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

```
13 rows selected.  
SCOTT>
```

“경영지원부의 상위부서는
사장실이다”

```
oracle@localhost:~  
SCOTT>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

13 rows selected.
SCOTT>

```
oracle@localhost:~  
SCOTT>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

13 rows selected.
SCOTT>

- Oracle Join 문법

```
SCOTT> SELECT a.dname “부서명” , b.dname “상위부서명”  
2  FROM dept2 a , dept2 b  
3  WHERE a.pdept = b.dcode ;
```

- ANSI Join 문법

```
SCOTT> SELECT a.dname “부서명” , b.dname “상위부서명”  
2  FROM dept2 a JOIN dept2 b  
3  ON a.pdept = b.dcode ;
```


** Join 연습문제 1 :

professor 테이블에서 교수의 번호, 교수이름, 입사일, 자신보다 입사일 빠른 사람 인원수를 출력하세요. 단 자신보다 입사일이 빠른 사람수를 오름차순으로 출력하세요.



교수번호	교수명	입사일	빠른사람
1001	조인형	23-JUN-80	0
4001	심슨	23-OCT-81	1
3001	김도형	23-OCT-81	1
2003	주승재	29-APR-82	3
4005	바비	18-SEP-85	4
2002	김영조	30-NOV-85	5
1002	박승곤	30-JAN-87	6
3002	나한열	01-JUL-97	7
1003	송도권	22-MAR-98	8
4003	박원범	01-DEC-99	9
4007	허은선	23-MAY-01	10
2001	양선희	01-SEP-01	11
3003	김현정	24-FEB-02	12
4004	차범철	28-JAN-09	13
4002	최슬기	30-AUG-09	14
4006	전민	28-JUN-10	15

16 rows selected.

SCOTT>