
SQL - JOIN

정종욱 jongukjeong@gmail.com

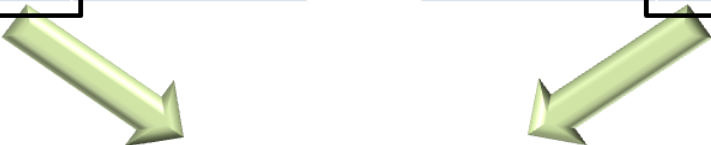
JOIN

사원 테이블

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

부서 테이블

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



이름	부서명
홍길동	영업부
이지매	관리부
.....

JOIN

1. Cartesian Product (카티션 곱 , CROSS Join)

- Oracle Join 문법

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

- ANSI Join 문법

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

JOIN

2. EQUI Join (등가 Join)

1) Equi join 예 1:

학생 테이블 (student) 과 학과 테이블 (department) 테이블을 사용하여 학생 이름, 1 전공학과번호(deptno1) , 1전공 학과 이름을 출력하세요.

학생이름	학과번호	학과이름
서지훈	101	컴퓨터공학
이윤지	101	컴퓨터공학
김신재	101	컴퓨터공학
김주현	102	멀티미디어공학
김지욱	102	멀티미디어공학
김은미	102	멀티미디어공학
이영경	103	소프트웨어공학
허우광	103	소프트웨어공학
안영세	201	전자공학
임현호	201	전자공학
김문수	201	전자공학
김은수	201	전자공학
오나라	202	기계공학
박동호	202	기계공학
구유미	301	문헌정보학
노정호	301	문헌정보학

20 rows selected.

SCOTT>

JOIN

- Oracle Join 문법

```
SCOTT>SELECT s.name "학생이름" , s.deptno1 "학과번호", d.dname "학과이름"  
2 FROM student s , department d  
3 WHERE s.deptno1 = d.deptno ;
```

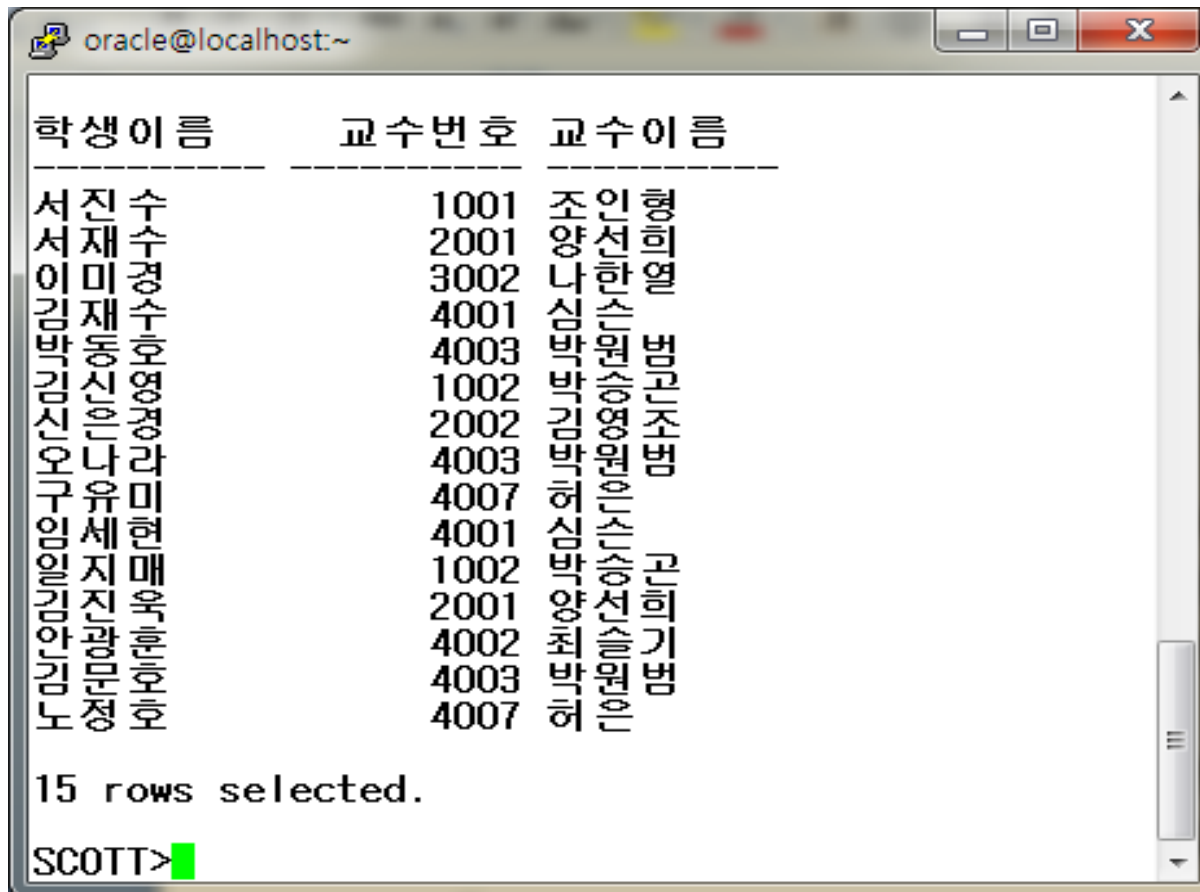
- ANSI Join 문법

```
SCOTT>SELECT s.name "학생이름" , s.deptno1 "학과번호", d.dname "학과이름"  
2 FROM student s JOIN department d  
3 ON (s.deptno1 = d.deptno) ;
```

JOIN

2) Equi Join 예 2:

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



```
oracle@localhost:~  
-----  
학생이름      교수번호  교수이름  
-----  
서진수        1001     조인형  
서재수        2001     양선한  
이재수        3002     나한  
김재수        4001     심원  
박동호        4003     박승원  
김신영        1002     박영원  
김신호        2002     김박희  
오나미        4003     박희성  
구유현        4007     허시원  
임세지        4001     심박  
일광호        1002     양최  
김광호        2001     박  
아문호        4002     허  
김정호        4003  
노정호        4007  
  
15 rows selected.  
SCOTT>
```

JOIN

- Oracle Join 문법

```
SCOTT>SELECT s.name "학생이름" , s.profno "교수번호" , p.name "교수이름"  
2  FROM student s , professor p  
3  WHERE s.profno = p.profno ;
```

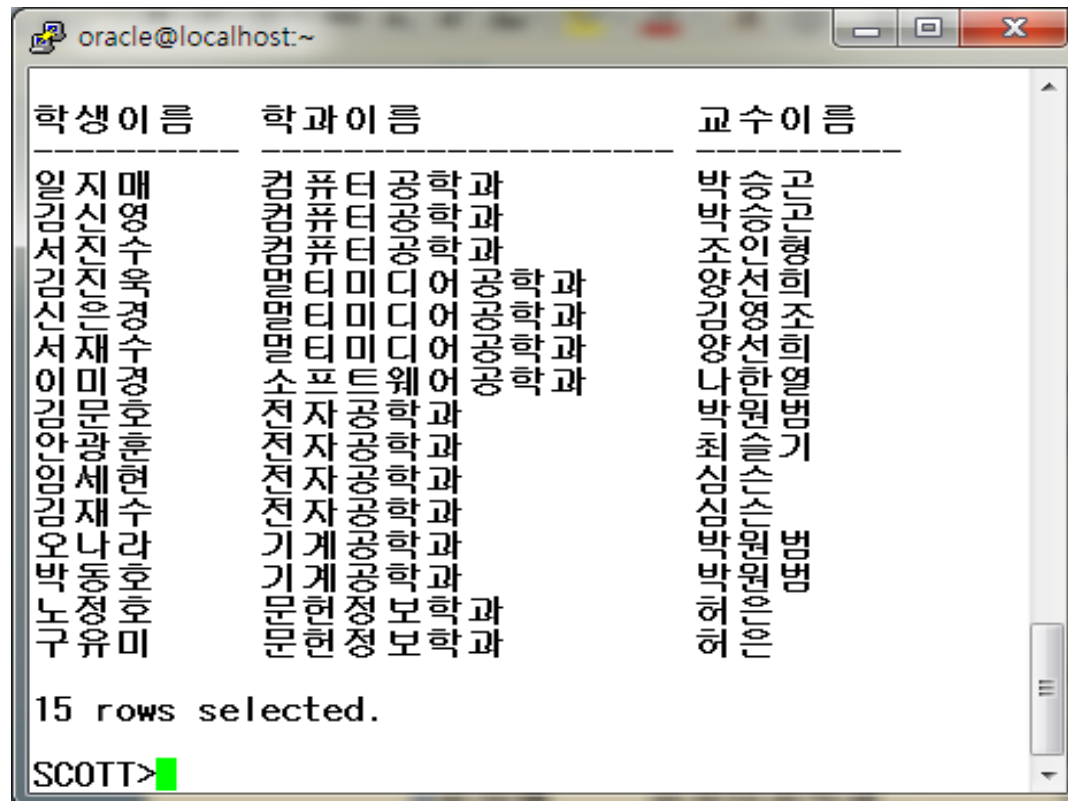
- ANSI Join 문법

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

JOIN

3) Equi Join 예 3:

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



```
oracle@localhost:~  
-----  
학생이름      학과이름      교수이름  
-----  
매영수      컴퓨터공학과      고근형  
지신재      컴퓨터공학과      송승인  
김진우      컴퓨터공학과      이영선  
김진우      컴퓨터공학과      조영한  
김진우      컴퓨터공학과      김양나  
김진우      컴퓨터공학과      박최심  
서재민      컴퓨터공학과      박심박  
이광세      컴퓨터공학과      박하영  
김재나      컴퓨터공학과      박하영  
오동호      컴퓨터공학과      박하영  
박정유      컴퓨터공학과      박하영  
노구미      컴퓨터공학과      박하영  
15 rows selected.  
SCOTT>
```


JOIN

- Oracle Join 문법

```
SCOTT> 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 ;
```

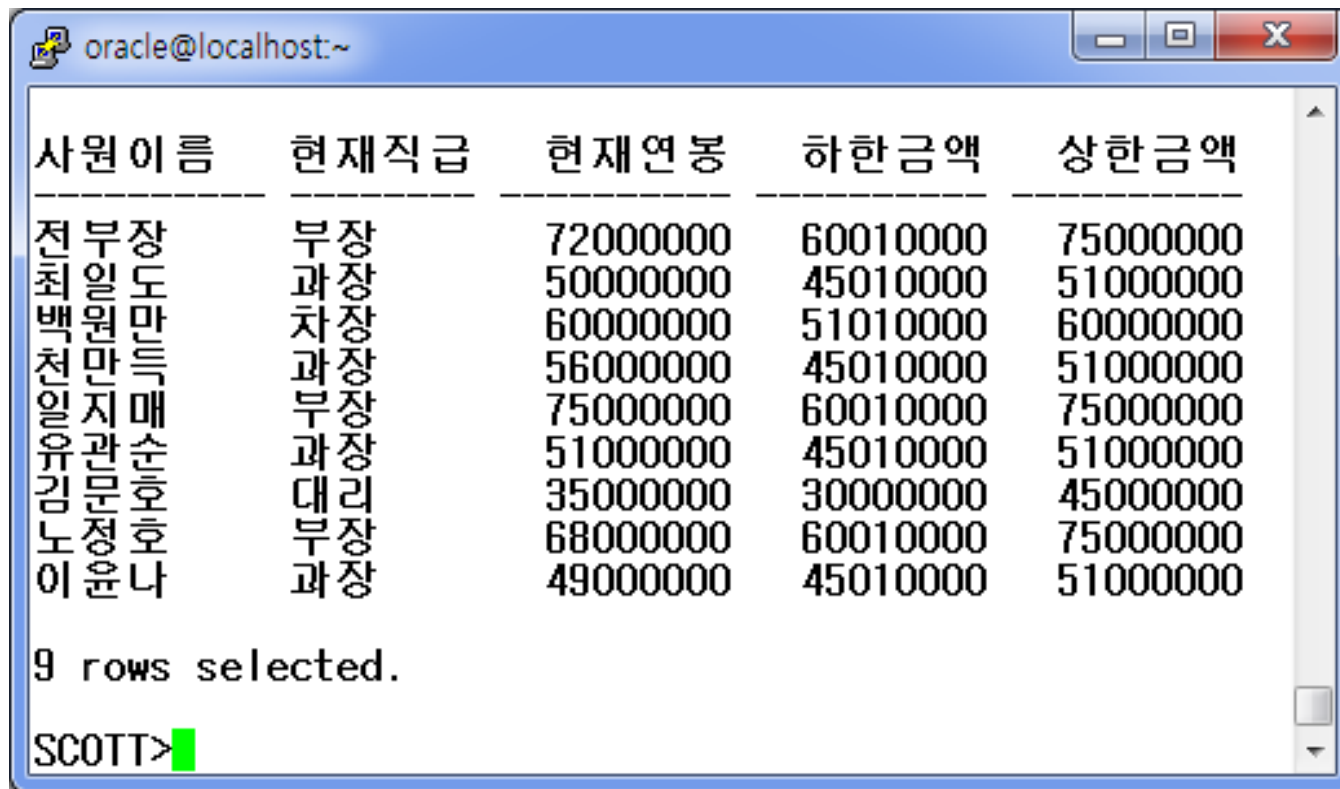
- ANSI Join 문법

```
SCOTT> 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 ;
```

JOIN

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>

JOIN

- Oracle Join 문법

```
SCOTT> SELECT e.name "사원이름" ,  
2          e.position "현재직급" ,  
3          e.pay "현재연봉",  
4          p.s_pay "하한금액" ,  
5          p.e_pay "상한금액"  
6 FROM emp2 e , p_grade p  
7 WHERE e.position = p.position ;
```

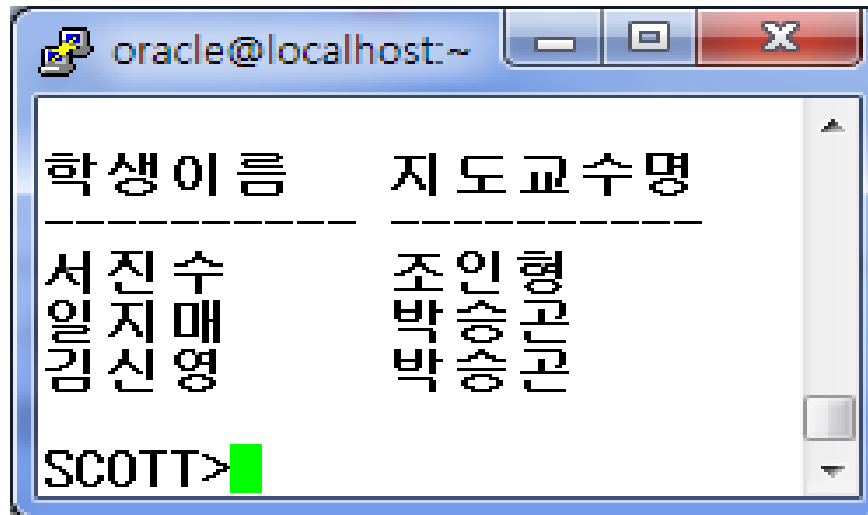
- ANSI Join 문법

```
SCOTT> SELECT e.name "사원이름" ,  
2          e.position "현재직급" ,  
3          e.pay "현재연봉",  
4          p.s_pay "하한금액" ,  
5          p.e_pay "상한금액"  
6 FROM emp2 e JOIN p_grade p  
7 ON e.position = p.position ;
```

JOIN

5) Equi Join 예 5 :

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



```
oracle@localhost:~  
-----  
학생이름      지도교수명  
-----  
서진수        조인행  
이지매        박승원  
김신영        박승원  
SCOTT>
```

JOIN

- 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 ;
```

JOIN

3. 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.

oracle@localhost:~

고객명	POINT	상 품 명
김재수	65000	참치세트
서정현	73000	참치세트
최순영	99000	참치세트
김박동호	110000	참치세트
박시영	153000	참치세트
김박동호	180000	참치세트
김박동호	265000	세차용품
김박동호	273000	세차용품
김박동호	298000	세차용품
김박동호	315000	세차용품
김박동호	320000	세차용품
김박동호	420000	주방용품
김박동호	470000	산악용품
김박동호	542000	LCD모니터
김박동호	598000	LCD모니터
김박동호	625000	노트북
김박동호	670000	노트북
김박동호	730000	벽걸이TV
김박동호	770000	벽걸이TV
김박동호	980000	양쪽냉장고

20 rows selected.

SCOTT>

JOIN

- Oracle Join 구문

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

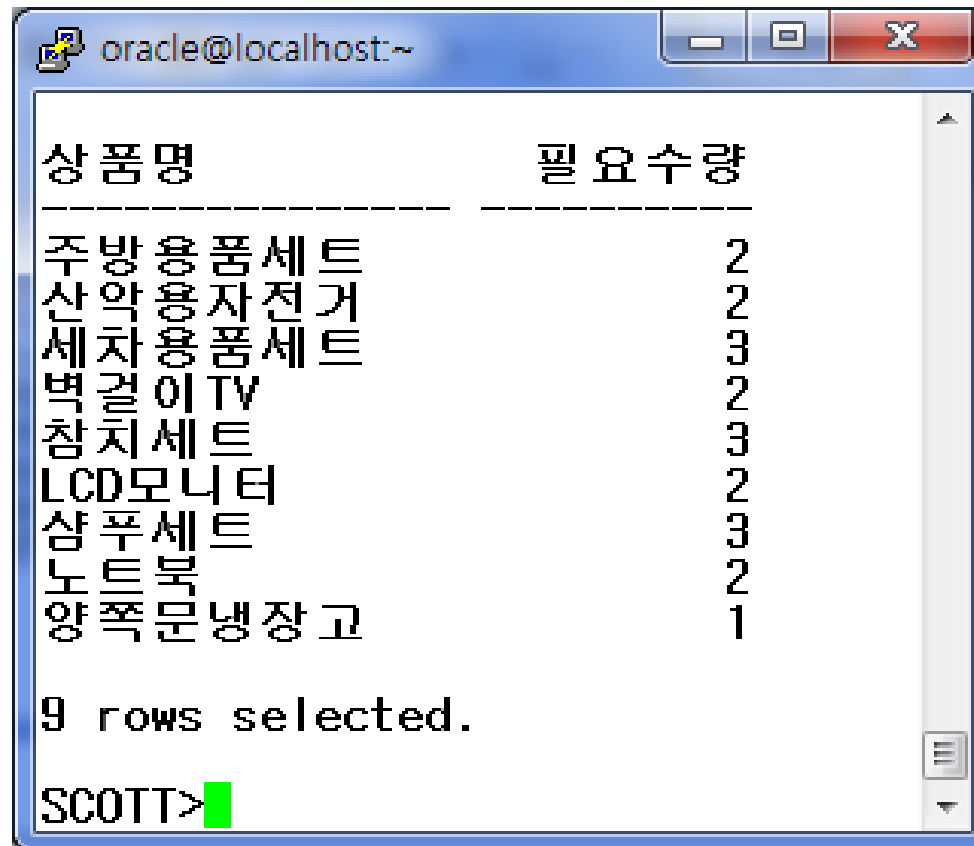
- ANSI Join 구문

```
SCOTT>SELECT go.gname "고객명", go.point "POINT" , gi.gname "상품명"  
2 FROM gogak go JOIN gift gi  
3 ON go.point BETWEEN gi.g_start AND gi.g_end ;
```

JOIN

2) Non-Equi Join 예제 2:

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



```
oracle@localhost:~  
-----  
상 품 명                필 요 수 량  
-----  
주방용품세트                2  
산악용자전거                2  
세차용품세트                3  
벽걸이TV                    2  
참치세트                    3  
LCD모니터                    2  
삼푸세트                    3  
노트북                      2  
양쪽문냉장고                1  
  
9 rows selected.  
  
SCOTT>
```


JOIN

- Oracle Join 구문

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

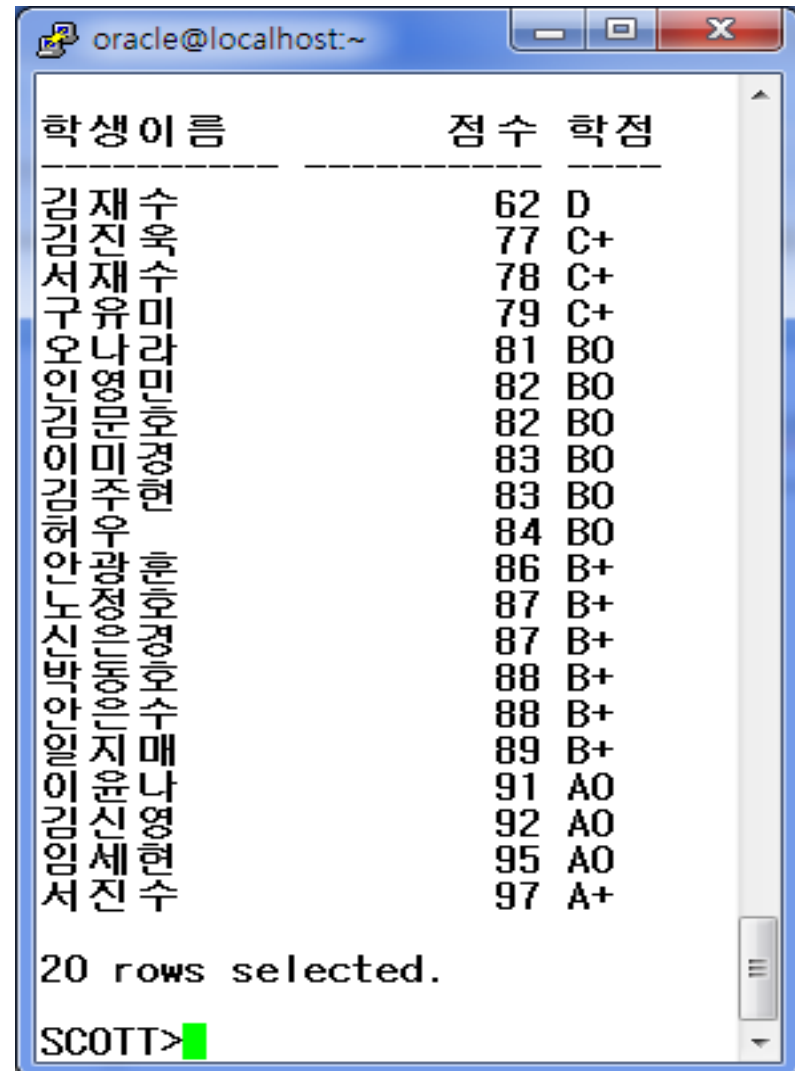
- ANSI Join 구문

```
SCOTT> SELECT gi.gname "상품명" , COUNT(*) "필요수량"  
2  FROM gogak go JOIN gift gi  
3  ON go.point BETWEEN gi.g_start AND gi.g_end  
4  GROUP BY gi.gname ;
```

JOIN

3) Non-Equi Join 예제 3:

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



학생이름	점수	학점
김재수	62	D
김진욱	77	C+
서재수	78	C+
구유미	79	C+
오나라	81	B0
인영민	82	B0
김영문	82	B0
이미호	83	B0
김주현	83	B0
김하우	84	B0
허안광	86	B+
노호정	87	B+
신영호	87	B+
박동호	88	B+
안동수	88	B+
이재매	89	B+
이윤나	91	A0
김신영	92	A0
임세현	95	A0
서진수	97	A+

20 rows selected.

SCOTT>

JOIN

- Oracle Join 문법

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

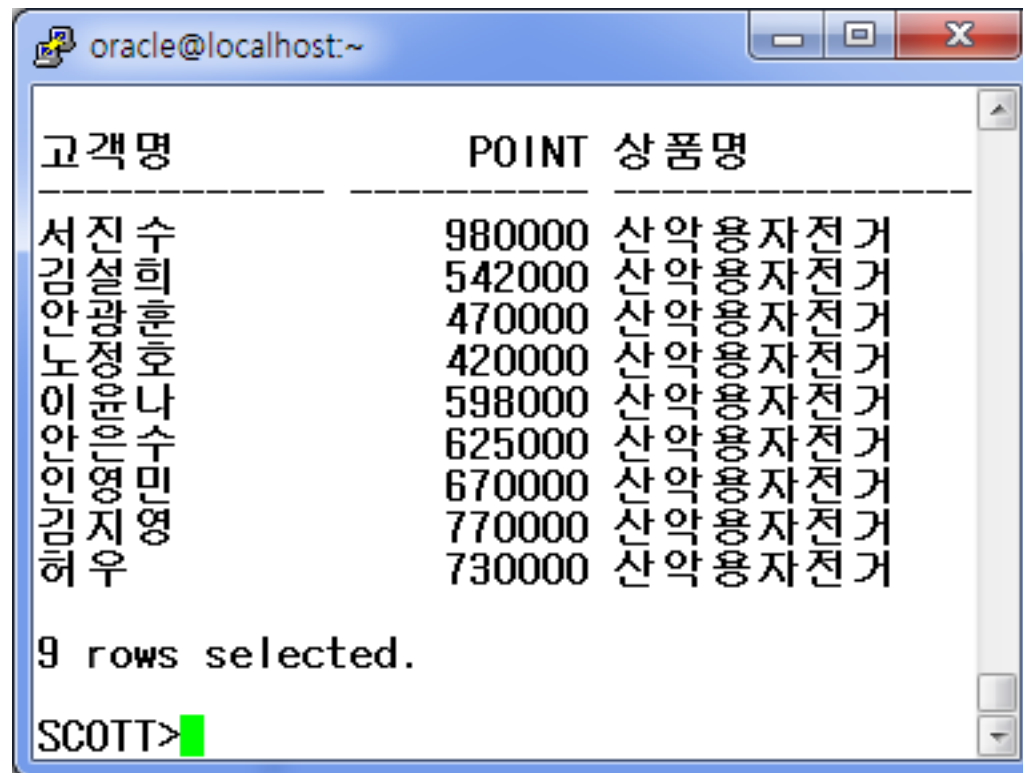
- ANSI Join 문법

```
SCOTT>SELECT s.name "학생이름" ,e.total "점수" , h.grade "학점"  
2 FROM student s JOIN exam_01 e  
3 ON s.studno = e.studno  
4 JOIN hakjum h  
5 ON e.total BETWEEN h.min_point AND h.max_point ;
```

JOIN

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.



The screenshot shows a terminal window titled 'oracle@localhost:~'. It displays the result of a SQL query. The output is a table with three columns: '고객명' (Customer Name), 'POINT' (Points), and '상품명' (Product Name). There are 9 rows of data, all showing '산악용 자전거' (Mountain Bike) as the product. Below the table, it says '9 rows selected.' and the prompt 'SCOTT>' is visible.

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

9 rows selected.

SCOTT>

JOIN

- Oracle Join 문법

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

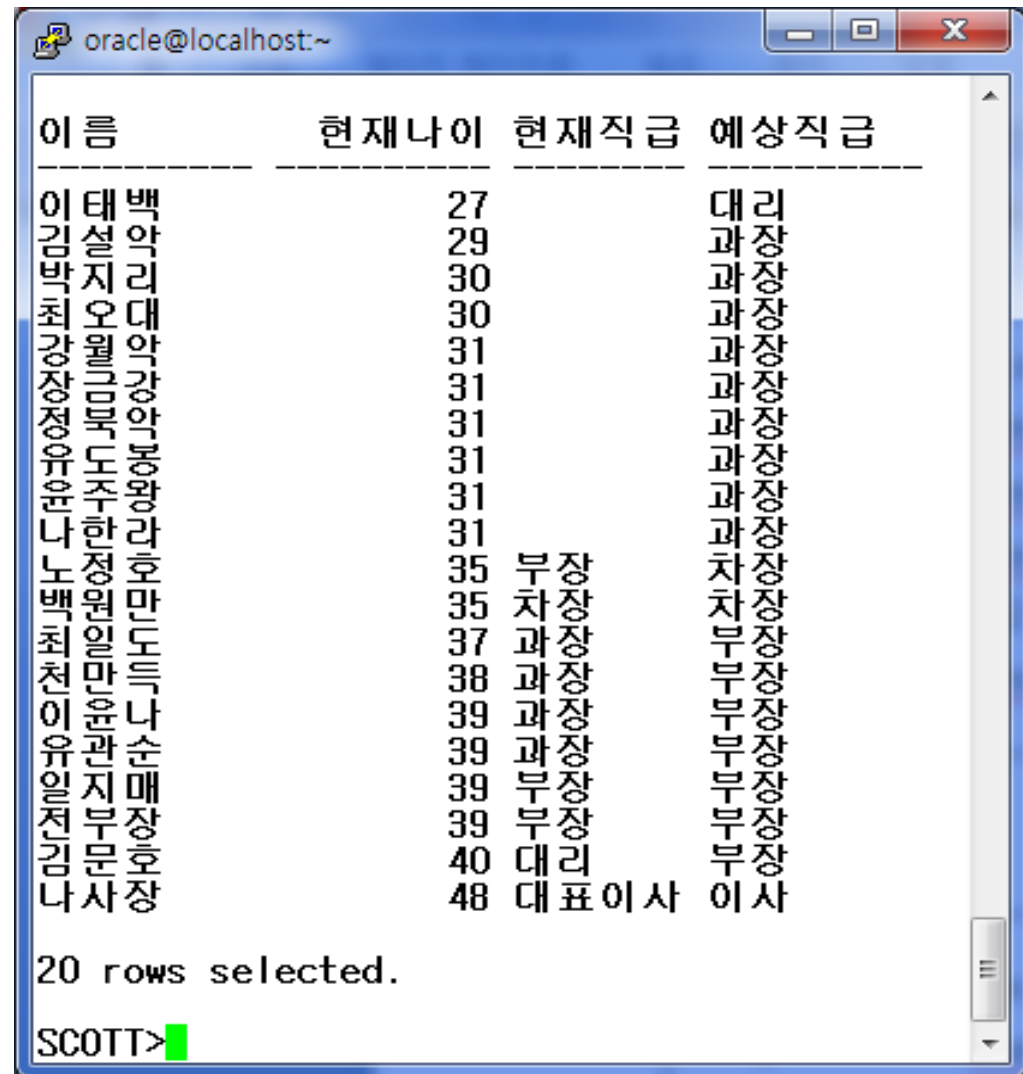
- ANSI Join 문법

```
SCOTT>SELECT go.gname "고객명" , go.point "POINT" , gi.gname "상품명"  
2  FROM gogak go JOIN gift gi  
3  ON gi.g_start <= go.point  
4  AND gi.gname ='산악용자전거';
```

JOIN

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.



이름	현재나이	현재직급	예상직급
이태백	27		대리
김설리	29		과장
박지오	30		과장
최강오	30		과장
강민호	31		과장
정복도	31		과장
유윤주	31		과장
나한라	31		과장
노정호	31		과장
백일만	35	부장	차장
최원노	35	차장	차장
천일만	37	과장	차장
이윤관	38	과장	부장
유관지	39	과장	부장
일지문	39	과장	부장
전지문	39	과장	부장
김문사	40	대리	부장
나사	48	대표이사	이사

20 rows selected.

SCOTT>

JOIN

- Oracle Join 문법

```
SCOTT>SELECT e.name "이름",  
2      trunc((sysdate-e.birthday)/365,0) "현재나이" ,  
3      e.position "현재직급" ,  
4      g.position "예상직급"  
5 FROM emp2 e , p_grade g  
6 WHERE trunc((sysdate-e.birthday)/365,0) BETWEEN g.s_age AND g.e_age ;
```

- ANSI Join 문법

```
SCOTT>SELECT e.name "이름",  
2      trunc((sysdate-e.birthday)/365,0) "현재나이" ,  
3      e.position "현재직급" ,  
4      g.position "예상직급"  
5 FROM emp2 e JOIN p_grade g  
6 ON trunc((sysdate-e.birthday)/365,0) BETWEEN g.s_age AND g.e_age ;
```

JOIN

4. 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

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

- Oracle Outer Join 문법

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


JOIN

- 결과 화면

학생이름	교수이름
서일	조박
김서	박양
신서	김나
이신	심최
임김	박박
아김	박하
오박	하하
노구	하하
구하	하하
하김	하하
이아	하하
이판	하하
현민	하하
수나	하하
수영	하하
영호	하하
호미	하하
미하	하하
하하	하하
하하	하하
하하	하하
하하	하하
하하	하하
하하	하하
하하	하하

20 rows selected.

SCOTT>

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

JOIN

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	심성원
김정호	양희정
김동수	조기범
노재진	영슬기
서광나	원한승
서미지	박신도
한일세	김재민
임영현	전승배
*****	김종하
*****	김승자
*****	정권철
*****	도민준
*****	비현도
*****	범
*****	자

22 rows selected.

SCOTT>

지도학생이 결정
안 된 교수들입니다.

JOIN

- 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 ORDER BY 1 ;
```

- ANSI Outer Join 문법

```
SCOTT>SELECT s.name "학생이름", p.name "교수이름"  
2 FROM student s RIGHT OUTER JOIN professor p  
3 ON s.profno = p.profno  
4 ORDER BY 1 ;
```

JOIN

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

oracle@localhost... X

학생이름	교수이름
서서이김박김신오구임일김아기노이아이기림혜	형희열 범고조범 고희기범 인선한스원승영원의스스승선슬원은 조양나심박박김박허심박양최박혜
수수경수호영정라미현매욱훈호하나수민현	***** ***** ***** ***** *****
진재미재동신이나유세지광문정윤영주우	***** 김재주바김송차도민승비현도범
***** ***** ***** ***** ***** *****	형 재 정권철

27 rows selected.

SCOTT>

JOIN

- 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 ;
```

JOIN

5. SELF Join

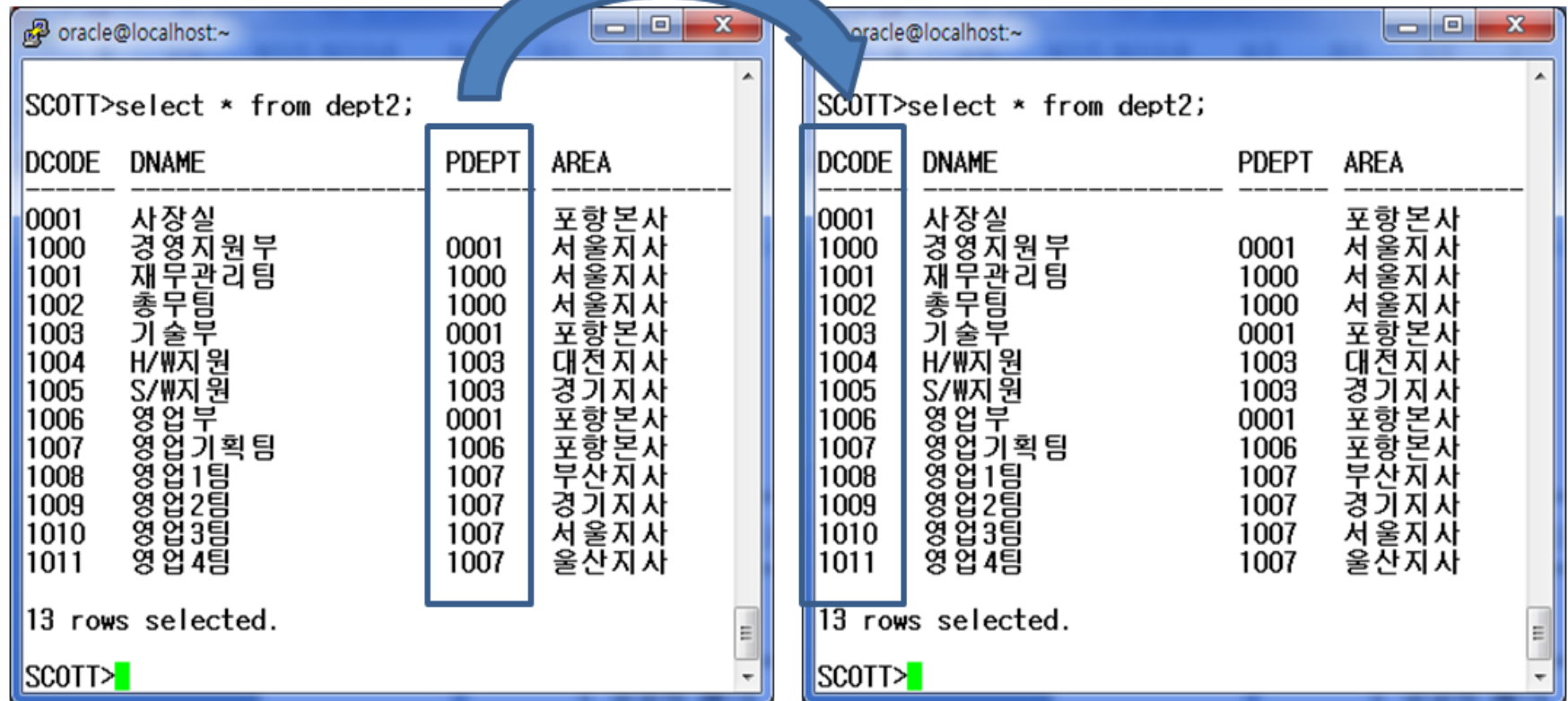
```
oracle@localhost:~  
SCOTT>select * from dept2;
```

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”

JOIN



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/W지원원	1003	대전지사
1005	S/W지원원	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>

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/W지원원	1003	대전지사
1005	S/W지원원	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>

JOIN

- 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

- 출력 결과

[illegible]

JOIN

** 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	허은	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>

Select a.profno, a.name, a.hiredate,
count(b.hiredate)

From professor a, professor b

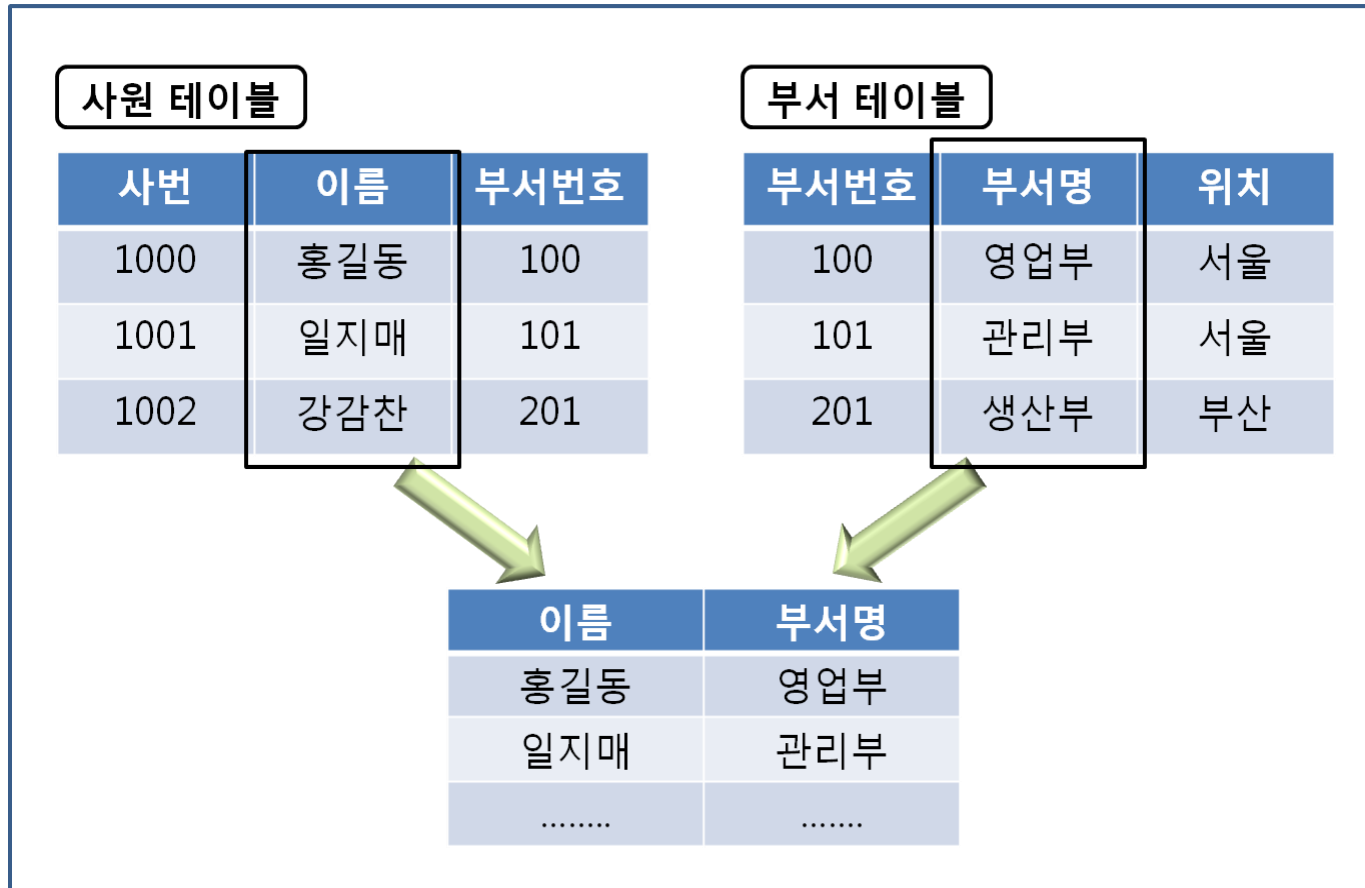
Where b.hiredate(+) < a.hiredate

Group by a.profno, a.name, a.hiredate

Order by 4;

JOIN

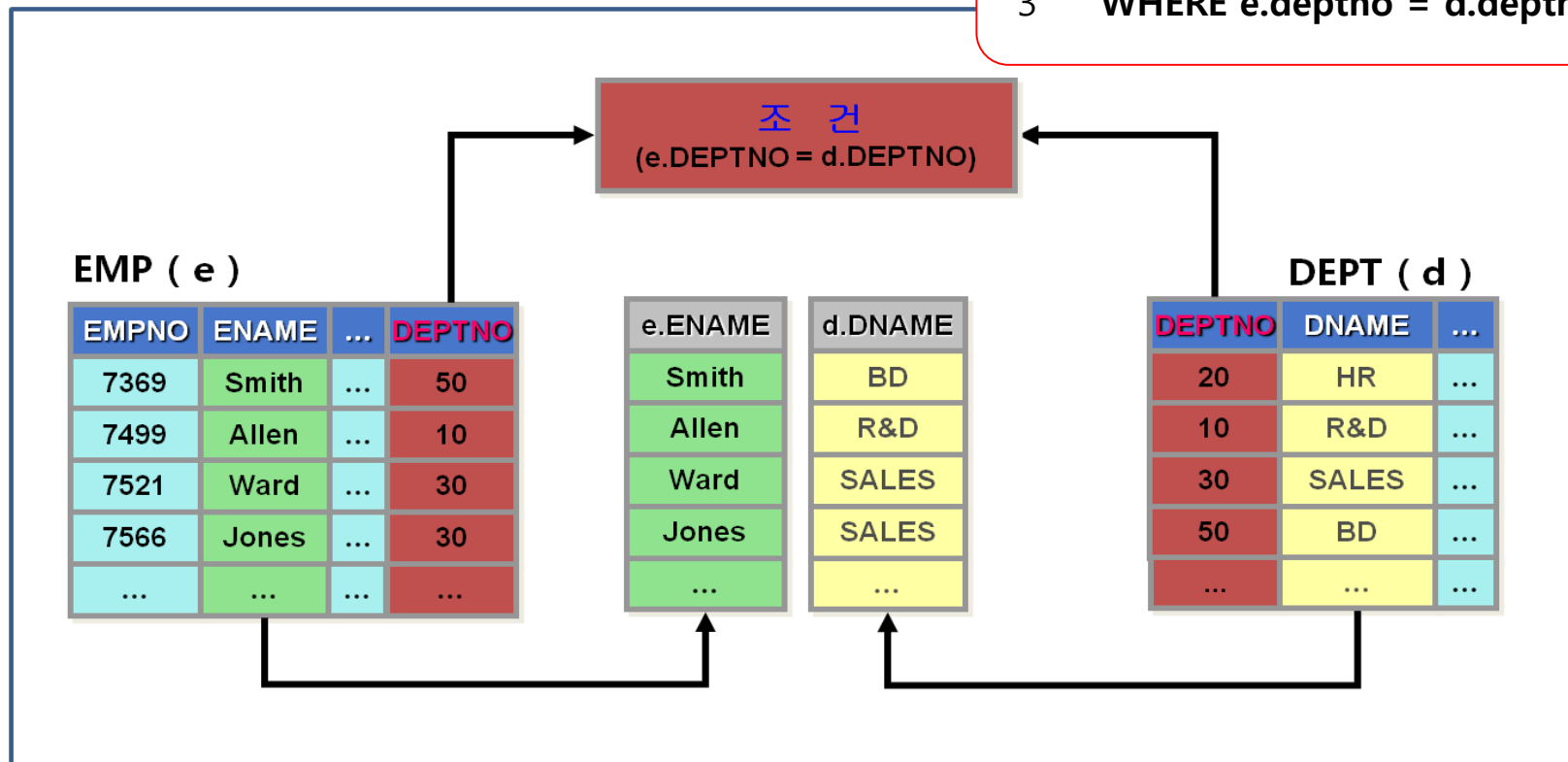
6. JOIN 원리



JOIN

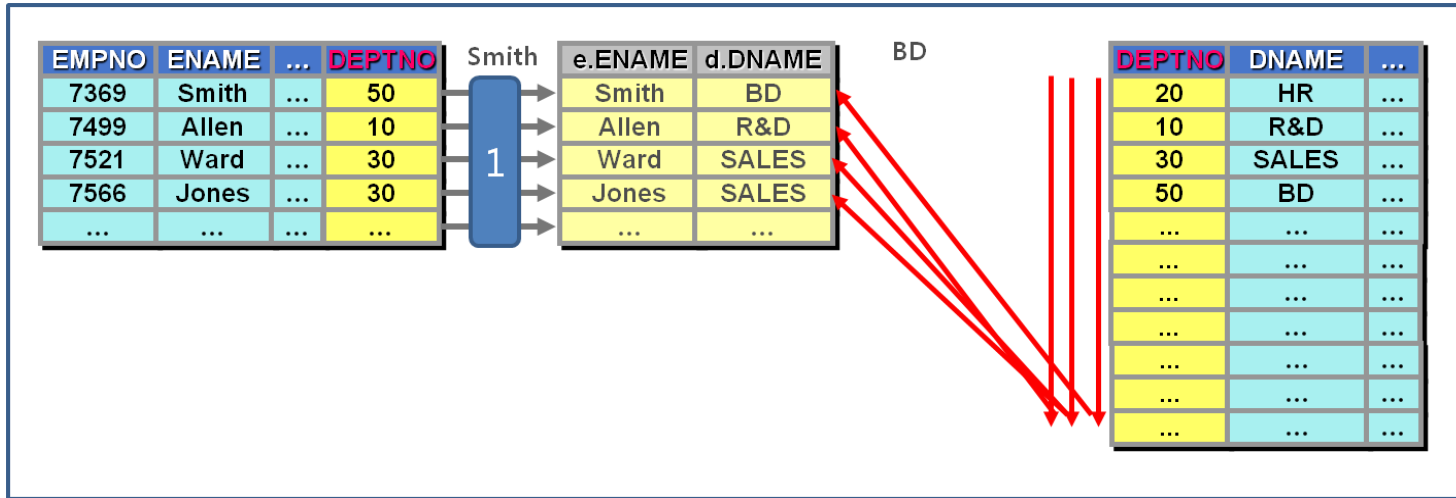
1) Nested Loop Join (가장 기본적인 Join 기법입니다)

```
SQL> SELECT e.ename , d.dname  
2   FROM emp e , dept d  
3   WHERE e.deptno = d.deptno ;
```

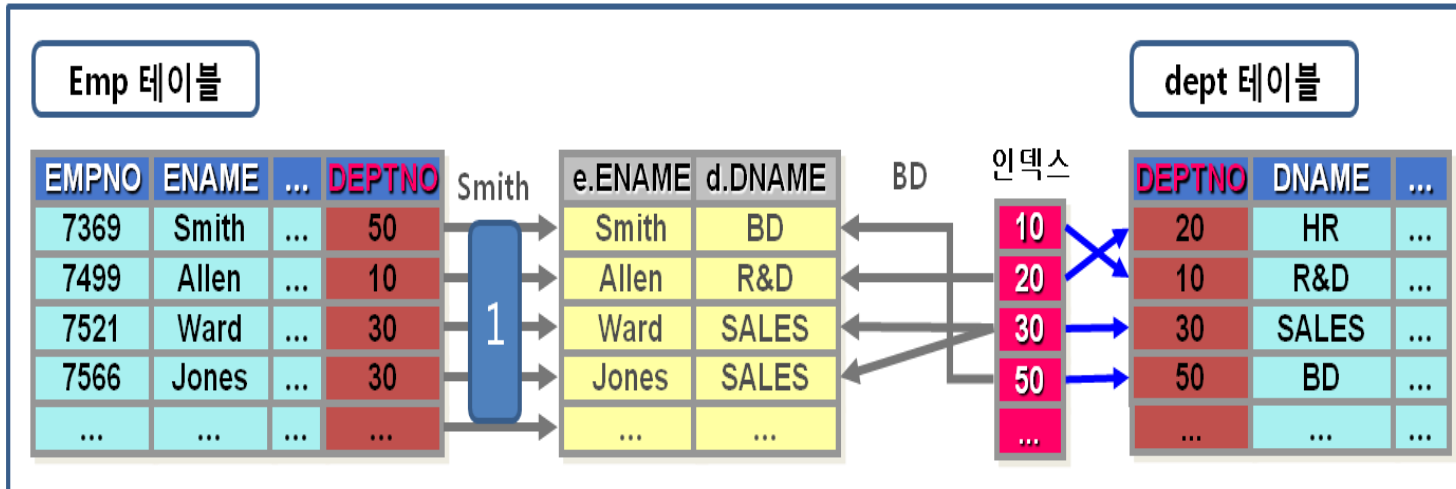


JOIN

2) Sort-Merge join



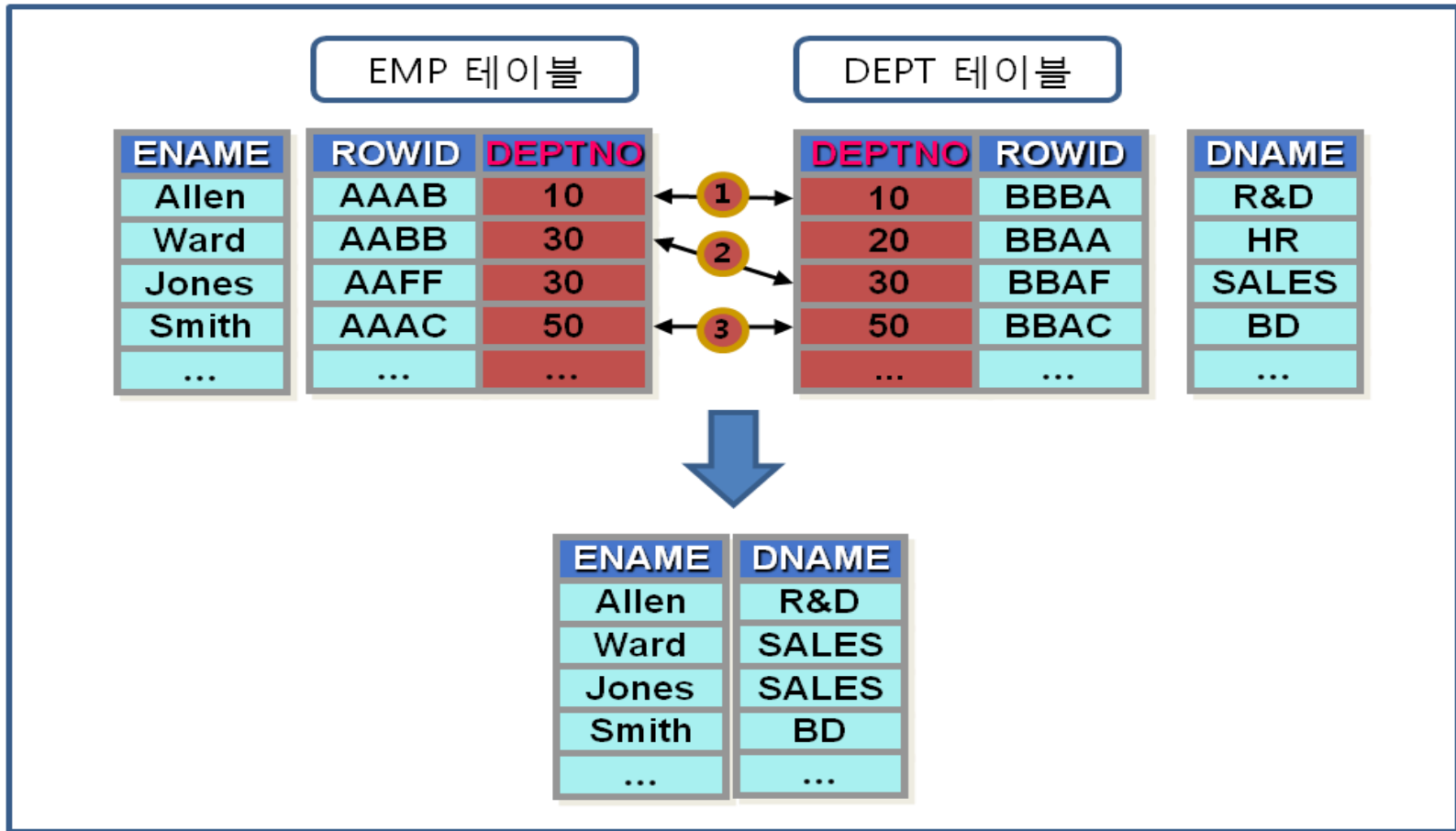
인덱스가 없을 경우



인덱스가 있을 경우

JOIN

- Sort 후 Merge 작업



JOIN

3) Hash Join (CBO 에서만 가능합니다)

