

ORACLE®

07. 조인



목차

1. 조인의 필요성
2. Cross Join
3. Equi Join
4. Non-Equi Join
5. Self Join
6. Outer Join
7. ANSI Join

1. 조인의 필요성(1)

- ◆ 특정 부서 번호에 대한 부서 이름은 무엇인지는 부서(DEPT) 테이블에 있다.
특정 사원에 대한 부서명을 알아내기 위해서는 부서 테이블에서 정보를
얻어 와야 한다.

ENAME	DEPTNO
1 이문세	10
2 이병현	10
3 조인성	10
4 이병현	20
5 안성기	20
6 박충훈	20
7 김사랑	20
8 강혜정	20
9 오지호	30
10 한예슬	30
11 강우성	30
12 조향기	30
13 장농건	30
14 신동협	30

DEPTNO	DNAME	LOC
10	경리부	서울
20	인사부	인천
30	영업부	용인
40	전산부	수원

1. 조인의 필요성(2)

- ❖ 이름이 '김사랑'인 사원이 소속되어 있는 부서의 이름이 무엇인지 알아보려고 한다.
- ❖ '김사랑'의 부서명을 알아내는 일 역시 사원 테이블에서 '김사랑'이 소속된 부서 번호를 알아낸 후에 부서 테이블에서 해당 부서 번호에 대한 부서명을 얻어 와야 한다.

```
SELECT deptno FROM emp  
WHERE ename='김사랑'
```

DEPTNO
20

```
SELECT * FROM dept WHERE  
deptno=20
```

DEP...	DNAME	LOC
1	20 인사부	인천

- ❖ 위 예제처럼 원하는 정보가 두 개 이상의 테이블에 나눠져 있을 때 SQL에서는 한 번의 질의로 원하는 결과를 얻을 수 있는 조인 기능을 제공합니다

2. CROSS Join(1)

❖ Cross Join

- 특별한 키워드 없이 SELECT 문의 FROM 절에 여러 테이블을 연결
- 한 쪽 테이블의 모든 행과 다른 테이블의 모든 행을 조인 시키는 형태

형식

SELECT * FROM emp, dept;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO_1	DNAME	LOC
1	1001	김사랑	사원	1013-07-03 01	300	(null)	20	10	영리부서	서울
2	1002	한예슬	대리	1005-07-04 02	250	80	30	10	영리부서	서울
3	1003	오지호	과장	1005-05-02 10	500	100	30	10	영리부서	서울
4	1004	이병현	부장	1008-03-09 02	600	(null)	20	10	영리부서	서울
5	1005	신동협	과장	1005-05-04 07	450	200	30	10	영리부서	서울
6	1006	장동건	부장	1008-03-10 09	480	(null)	30	10	영리부서	서울
7	1007	이문세	부장	1008-04-01 08	520	(null)	10	10	영리부서	서울
8	1008	강우성	차장	1003-04-03 08	500	0	30	10	영리부서	서울
9	1009	한성기	사장	(null) 96-10-04 1000	(null)	20	10	10	영리부서	서울
10	1010	이병현	과장	1003-05-04 07	500	(null)	10	10	영리부서	서울
11	1011	조향기	사원	1007-07-03 01	280	(null)	30	10	영리부서	서울
12	1012	강혜정	사원	1006-07-08 09	300	(null)	20	10	영리부서	서울
13	1013	박승훈	부장	1003-02-10 09	560	(null)	20	10	영리부서	서울
14	1014	조인성	사원	1006-07-11 09	250	(null)	10	10	영리부서	서울
15	1001	김사랑	사원	1013-07-03 01	300	(null)	20	20	인사부서	인천
16	1002	한예슬	대리	1005-07-04 02	250	80	30	20	인사부서	인천
17	1003	오지호	과장	1005-05-02 10	500	100	30	20	인사부서	인천
18	1004	이병현	부장	1008-03-09 02	600	(null)	20	20	인사부서	인천
19	1005	신동협	과장	1005-05-04 07	450	200	30	20	인사부서	인천
20	1006	장동건	부장	1008-03-10 09	480	(null)	30	20	인사부서	인천
21	1007	이문세	부장	1008-04-01 08	520	(null)	10	20	인사부서	인천
22	1008	강우성	차장	1003-04-03 08	500	0	30	20	인사부서	인천
23	1009	한성기	사장	(null) 96-10-04 1000	(null)	20	20	인사부서	인천	
24	1010	이병현	과장	1003-05-04 07	500	(null)	10	20	인사부서	인천
25	1011	조향기	사원	1007-07-03 01	280	(null)	30	20	인사부서	인천
26	1012	강혜정	사원	1006-07-08 09	300	(null)	20	20	인사부서	인천
27	1013	박승훈	부장	1003-02-10 09	560	(null)	20	20	인사부서	인천
28	1014	조인성	사원	1006-07-11 09	250	(null)	10	20	인사부서	인천
29	1001	김사랑	사원	1013-07-03 01	300	(null)	20	30	영업부	부산
30	1002	한예슬	대리	1005-07-04 02	250	80	30	30	영업부	부산

31	1003	오지호	과장	1005-05-02-10	500	100	30	30	영업부	부산
32	1004	이병현	부장	1008-03-09-02	600	(null)	20	30	영업부	부산
33	1005	신동협	과장	1005-05-04-07	450	200	30	30	영업부	부산
34	1006	장동건	부장	1008-03-10-09	480	(null)	30	30	영업부	부산
35	1007	이문세	부장	1008-04-01-08	520	(null)	10	30	영업부	부산
36	1008	강우성	차장	1003-04-03-08	500	0	30	30	영업부	부산
37	1009	한성기	사장	(null) 96-10-04 1000	(null)	20	30	30	영업부	부산
38	1010	이병현	과장	1003-05-04-07	500	(null)	10	30	영업부	부산
39	1011	조향기	사원	1007-07-03-01	280	(null)	30	30	영업부	부산
40	1012	강혜정	사원	1006-07-08-09	300	(null)	20	30	영업부	부산
41	1013	박승훈	부장	1003-02-10-09	560	(null)	20	30	영업부	부산
42	1014	조인성	사원	1006-07-11-09	250	(null)	10	30	영업부	부산
43	1001	김사랑	사원	1013-07-03-01	300	(null)	20	40	전산부	수원
44	1002	한예슬	대리	1005-07-04-02	250	80	30	40	전산부	수원
45	1003	오지호	과장	1005-05-02-10	500	100	30	40	전산부	수원
46	1004	이병현	부장	1008-03-09-02	600	(null)	20	40	전산부	수원
47	1005	신동협	과장	1005-05-04-07	450	200	30	40	전산부	수원
48	1006	장동건	부장	1008-03-10-09	480	(null)	30	40	전산부	수원
49	1007	이문세	부장	1008-04-01-08	520	(null)	10	40	전산부	수원
50	1008	강우성	차장	1003-04-03-08	500	0	30	40	전산부	수원
51	1009	한성기	사장	(null) 96-10-04 1000	(null)	20	40	전산부	수원	
52	1010	이병현	과장	1003-05-04-07	500	(null)	10	40	전산부	수원
53	1011	조향기	사원	1007-07-03-01	280	(null)	30	40	전산부	수원
54	1012	강혜정	사원	1006-07-08-09	300	(null)	20	40	전산부	수원
55	1013	박승훈	부장	1003-02-10-09	560	(null)	20	40	전산부	수원
56	1014	조인성	사원	1006-07-11-09	250	(null)	10	40	전산부	수원

2. Cross Join(2)

- ❖ Cross Join의 결과 얻어지는 컬럼의 수는 사원 테이블의 컬럼의 수(8)와 부서 테이블의 컬럼의 수를 더한 것이므로 11이 된다. 로우 수는 사원 한 명에 대해서 DEPT 테이블의 4개의 로우와 결합되기에 56개(14×4)가 된다.
- ❖ Cross Join의 결과를 보면 사원 테이블에 부서에 대한 상세정보가 결합되긴 했지만, 조인 될 때 아무런 조건을 제시하지 않았기에 사원 한 명에 대해서 DEPT 테이블의 4개의 로우와 결합된 형태이기에 Cross Join의 결과는 아무런 의미를 갖지 못한다.
- ❖ 조인 결과가 의미를 갖으려면 조인할 때 조건을 지정해야 한다.

2. Cross Join(3)

- ◆ 조인 조건에 따라 조인의 종류가 결정되는데 다음은 조인의 종류를 정리한 표.

종 류	설 명
Equi Join	동일 컬럼을 기준으로 조인
Non-Equi Join	동일 컬럼이 없이 다른 조건을 사용하여 조인
Outer Join	조인 조건에 만족하지 않는 행도 나타낸다.
Self Join	한 테이블 내에서 조인

3. Equi Join(1)

❖ Equi Join

- **가장 많이 사용하는 조인**
- 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치(=)되는 행을 연결하여 결과를 생성하는 조인 방법.

❖ 사원 정보를 출력할 때 각 사원들이 소속된 부서의 상세 정보를 출력하기 위해서 두 개의 테이블을 조인한 예제.

예제

```
SELECT * FROM emp, dept  
WHERE emp.deptno=dept.deptno;
```

- ❖ 사원(EMP) 테이블과 부서(DEPT) 테이블의 공통 컬럼인 DEPTNO의 값이 일치(=)되는 조건을 WHERE 절에 기술하여 사용.
- ❖ 테이블을 조인하려면 일치되는 공통 컬럼을 사용해야 하는데, 컬럼의 이름이 같게 되면 혼동이 오기 때문에 컬럼 이름 앞에 테이블 이름을 기술한다.

3. Equi Join(2)

- ❖ 다음은 두 테이블을 조인한 결과로서. 결과를 살펴보면 다음과 같이 부서 번호를 기준으로 같은 값을 가진 사원 테이블의 컬럼과 부서 테이블의 컬럼이 결합된다.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO_1	DNAME	LOC
1	1001	김사랑	사원	1013	07/03/01	300	(null)	20	20	인사부	인천
2	1002	한예슬	대리		1005 07/04/02	250	80	30	30	영업부	용인
3	1003	오지호	과장		1005 05/02/10	500	100	30	30	영업부	용인
4	1004	이병현	부장		1008 03/09/02	600	(null)	20	20	인사부	인천
5	1005	신농협	과장		1005 05/04/07	450	200	30	30	영업부	용인
6	1006	장농건	부장		1008 03/10/09	480	(null)	30	30	영업부	용인
7	1007	이문세	부장		1008 04/01/08	520	(null)	10	10	경리부	서울
8	1008	감우성	차장		1003 04/03/08	500	0	30	30	영업부	용인
9	1009	안성기	사장		(null) 96/10/04	1000	(null)	20	20	인사부	인천
10	1010	이병현	과장		1003 05/04/07	500	(null)	10	10	경리부	서울
11	1011	조향기	사원		1007 07/03/01	280	(null)	30	30	영업부	용인
12	1012	강혜정	사원		1006 07/08/09	300	(null)	20	20	인사부	인천
13	1013	박승훈	부장		1003 02/10/09	560	(null)	20	20	인사부	인천
14	1014	조인성	사원		1006 07/11/09	250	(null)	10	10	경리부	서울

3. Equi Join(3)

- ❖ Equi Join에 AND 연산하기
- ❖ 이름이 '김사랑'인 사람의 부서명 출력

예제

```
SELECT ename, dname FROM emp, dept WHERE  
emp.deptno=dept.deptno AND emp.ename='김사랑';
```

ENAME	DNAME
김사랑	인사부

3. Equi Join(4)

- ❖ 컬럼명의 모호성 해결
- ❖ 두 테이블에 동일한 이름의 컬럼을 사용하면 어느 테이블 소속인지 불분명하기에 애매모호한 상태라는 오류 메시지가 출력된다.

예제

```
SELECT ename, dname, deptno  
FROM emp, dept  
WHERE emp.deptno=dept.deptno  
AND ename='김사랑'
```

```
ORA-00918: column ambiguously defined  
00918, 00000 - "column ambiguously defined"  
*Cause:  
*Action:  
1행, 22열에서 오류 발생
```

3. Equi Join(5)

- ❖ 컬럼명의 모호성 해결
- ❖ 이렇게 동일한 이름의 컬럼은 컬럼 명 앞에 테이블 명을 명시적으로 기술함으로서 컬럼이 어느 테이블 소속인지 구분할 수 있게 된다.

예제

```
SELECT emp.ename, dept.dname, emp.deptno  
FROM emp, dept  
WHERE emp.deptno=dept.deptno  
AND ename='김사랑'
```

ENAME	DNAME	DEPTNO
1 김사랑	인사부	20

3. Equi Join(6)

- ❖ 테이블에 별칭 부여하기
- ❖ 테이블 이름에 별칭을 붙이는 방법은 FROM 절 다음에 테이블 이름을 명시하고 공백을 둔 다음에 별칭을 지정하면 된다.

예제

```
SELECT e.ename, d.dname, e.deptno  
FROM emp e, dept d  
WHERE e.deptno=d.deptno  
AND e.ename='김사랑'
```

4. Non-Equi Join(1)

- ❖ Non-Equi Join
 - 조인 조건에 특정 범위 내에 있는지를 조사하기 위해서 WHERE 절에 조인 조건을 = 연산자 이외의 비교 연산자를 사용.
- ❖ Non-Equi Join을 학습하기 전에 급여 등급 테이블(SALGRADE) 을 살펴보자.

예제

SELECT * FROM salgrade;

GRADE	LOSAL	HISAL
1	100	200
2	201	300
3	301	400
4	401	500
5	501	9999

4. Non-Equi Join(2)

- ❖ 급여 등급 테이블(salgrade)에는 급여에 대한 등급을 다음과 같이 나누어 놓았다.
- ❖ 급여의 등급은 총 5등급으로 나누어져 있으며,
1등급은 급여가 100부터 200 사이이고,
2등급은 201부터 300 사이이고,
3등급은 301부터 400 사이이고,
4등급은 401부터 500사이이고,
5등급이면 501부터 9999사이 이다.
- ❖ 급여 등급을 5개로 나누어 놓은 salgrade에서 정보를 얻어 와서 각 사원의 급여 등급을 지정해보도록 하자. 이를 위해서 사원(emp) 테이블과 급여 등급(salgrade) 테이블을 조인한다.

4. Non-Equi Join(3)

- ❖ ex) 사원의 급여가 몇 등급인지 살펴보는 예제.

예제

```
SELECT ename, sal, grade  
FROM emp, salgrade  
WHERE sal BETWEEN losal AND hisal;
```

	ENAME	SAL	GRADE
1	김사랑	300	2
2	한예슬	250	2
3	조향기	280	2
4	강혜정	300	2
5	조인성	250	2
6	오지호	500	4
7	신농협	450	4
8	장농건	480	4
9	감우성	500	4
10	이병현	500	4
11	이병현	600	5
12	이문세	520	5
13	안성기	1000	5
14	박충훈	560	5

5. Self Join(1)

❖ Self Join

- 자기 자신과 조인 맺는 형태
- 두 개 이상의 서로 다른 테이블을 서로 연결하는 것뿐만 아니라, **하나의 테이블 내에서 조인**을 해서 원하는 결과를 얻어야 하는 경우에 사용.
- FROM절에 동일 테이블명이 두 번 이상 선언
- 같은 테이블이라도 각각 별칭을 사용해야 한다.

❖ 사원들의 관리자 코드를 이용하여 관리자 정보를 출력하는 예제.

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	1001	김사랑	사원	1013	07/03/01	300	(null)	20
2	1002	한예슬	대리	1005	07/04/02	250	80	30
3	1003	오지호	과장	1005	05/02/10	500	100	30
4	1004	이병현	부장	1008	03/09/02	600	(null)	20
5	1005	신동협	과장	1005	05/04/07	450	200	30
6	1006	장농건	부장	1008	03/10/09	480	(null)	30
7	1007	이문세	부장	1008	04/01/08	520	(null)	10
8	1008	감우성	차장	1003	04/03/08	500	0	30
9	1009	안성기	사장	(null)	96/10/04	1000	(null)	20
10	1010	이병현	과장	1003	05/04/07	500	(null)	10
11	1011	조향기	사원	1007	07/03/01	280	(null)	30
12	1012	강혜정	사원	1006	07/08/09	300	(null)	20
13	1013	박승훈	부장	1003	02/10/09	560	(null)	20
14	1014	조인성	사원	1006	07/11/09	250	(null)	10

5. Self Join(2)

예제

```
SELECT E1.EMPNO AS "사원 코드", E1.ENAME AS "사원 이름",
E1.MGR AS "관리자 코드", E2.ENAME AS "관리자 이름"
FROM EMP E1, EMP E2
WHERE E1.MGR = E2.EMPNO
ORDER BY E2.MGR DESC, E1.MGR, E1.EMPNO;
```

	◆ 사원 코드	◆ 사원 이름	◆ 관리자 코드	◆ 관리자 이름
1	1012	강혜정	1006	장동건
2	1014	조인성	1006	장동건
3	1011	조향기	1007	이문세
4	1008	감우성	1003	오지호
5	1010	이병현	1003	오지호
6	1013	박충훈	1003	오지호
7	1002	한예슬	1005	신농협
8	1003	오지호	1005	신농협
9	1005	신농협	1005	신농협
10	1004	이병현	1008	감우성
11	1006	장동건	1008	감우성
12	1007	이문세	1008	감우성
13	1001	김사랑	1013	박충훈

6. Outer Join(1)

❖ Outer Join

- 조인 조건에 만족하지 못하였더라도 해당 행을 나타내고 싶을 때에 사용하는 것이 외부 조인(Outer Join)이다.
- NULL 값이기에 배제된 행을 결과에 포함시킬 수 있으며 다음과 같이 '(+)' 기호를 조인 조건에서 정보가 부족한 컬럼 이름 뒤에 덧붙인다.

❖ Self Join을 학습하며 특정 사원의 매니저 정보를 구해봤다.

결과를 꼼꼼히 살펴보면 이름이 '안성기'인 사원 한사람의 정보가 빠져 있음을 확인할 수 있다.

'안성기'는 이 회사의 사장(PRESIDENT)으로 매니저가 존재하지 않으므로 MGR 컬럼 값이 NULL이다. 사원 번호(EMPNO)가 NULL인 사원은 없으므로 조인 조건에 만족하지 않아서 Self Join의 결과에서 배제가 되었다.

❖ 사원 번호(EMPNO)가 NULL인 사원은 없으므로 manager.empno 뒤에 '(+)' 기호를 덧붙인다.

6. Outer Join(2)

예제

```
SELECT employee.ename || '의 매니저는 ' || manager.ename || '입니다.'  
FROM emp employee, emp manager  
WHERE employee.mgr = manager.empno(+);
```

EMPLOYEE.ENAME '의 매니저는' MANAGER.ENAME '입니다.'	
1	박승훈의 매니저는 오지호입니다.
2	이병헌의 매니저는 오지호입니다.
3	감우성의 매니저는 오지호입니다.
4	신농협의 매니저는 신농협입니다.
5	오지호의 매니저는 신농협입니다.
6	한예슬의 매니저는 신농협입니다.
7	조인성의 매니저는 장농건입니다.
8	강혜정의 매니저는 장농건입니다.
9	조향기의 매니저는 이문세입니다.
10	이문세의 매니저는 감우성입니다.
11	장농건의 매니저는 감우성입니다.
12	이병헌의 매니저는 감우성입니다.
13	김사랑의 매니저는 박승훈입니다.
14	안성기의 매니저는입니다.

7. ANSI Join

❖ ANSI Join

- ANSI(미국표준협회)에서 표준화한 Join 방법으로 Oracle, MySQL, MS-SQL등에서 공통적으로 사용된다.

❖ ANSI Cross Join

예제

SELECT * FROM emp Cross Join dept;

#	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO_L	DNAME	LOC
1	1001	김사랑	사원	1013	07/03/01	300	(null)	20	10	정밀리	부서
2	1002	한예슬	대리	1005	07/04/02	250	80	30	10	정밀리	부서
3	1003	오지호	과장	1005	05/02/10	500	100	30	10	정밀리	부서
4	1004	이병현	부장	1008	03/09/02	600	(null)	20	10	정밀리	부서
5	1005	신동혁	과장	1005	05/04/07	450	200	30	10	정밀리	부서
6	1006	장동건	부장	1008	03/10/09	480	(null)	30	10	정밀리	부서
7	1007	이문세	부장	1008	04/01/08	520	(null)	10	10	정밀리	부서
8	1008	김우성	차장	1003	04/03/08	500	0	30	10	정밀리	부서
9	1009	한성기	사장	(null)	96/10/04	1000	(null)	20	10	정밀리	부서
10	1010	이병현	과장	1003	05/04/07	500	(null)	10	10	정밀리	부서
11	1011	조향기	사원	1007	07/03/01	280	(null)	30	10	정밀리	부서
12	1012	곽혜정	사원	1006	07/08/09	300	(null)	20	10	정밀리	부서
13	1013	박충훈	부장	1003	02/10/09	560	(null)	20	10	정밀리	부서
14	1014	조인성	사원	1006	07/11/09	250	(null)	10	10	정밀리	부서
15	1001	김사랑	사원	1013	07/03/01	300	(null)	20	20	인사부	인천전
16	1002	한예슬	대리	1005	07/04/02	250	80	30	20	인사부	인천전
17	1003	오지호	과장	1005	05/02/10	500	100	30	20	인사부	인천전
18	1004	이병현	부장	1008	03/09/02	600	(null)	20	20	인사부	인천전
19	1005	신동혁	과장	1005	05/04/07	450	200	30	20	인사부	인천전
20	1006	장동건	부장	1008	03/10/09	480	(null)	30	20	인사부	인천전
21	1007	이문세	부장	1008	04/01/08	520	(null)	10	20	인사부	인천전
22	1008	김우성	차장	1003	04/03/08	500	0	30	20	인사부	인천전
23	1009	한성기	사장	(null)	96/10/04	1000	(null)	20	20	인사부	인천전
24	1010	이병현	과장	1003	05/04/07	500	(null)	10	20	인사부	인천전
25	1011	조향기	사원	1007	07/03/01	280	(null)	30	20	인사부	인천전
26	1012	곽혜정	사원	1006	07/08/09	300	(null)	20	20	인사부	인천전
27	1013	박충훈	부장	1003	02/10/09	560	(null)	20	20	인사부	인천전
28	1014	조인성	사원	1006	07/11/09	250	(null)	10	30	영업부	용인
29	1001	김사랑	사원	1013	07/03/01	300	(null)	20	30	영업부	용인
30	1002	한예슬	대리	1005	07/04/02	250	80	30	30	영업부	용인

31	1003	오지호	과장	1005	05/02/10	500	100	30	30	영업부	용인
32	1004	이병현	부장	1008	03/09/02	600	(null)	20	30	영업부	용인
33	1005	신동혁	과장	1005	05/04/07	450	200	30	30	영업부	용인
34	1006	장동건	부장	1008	03/10/09	480	(null)	30	30	영업부	용인
35	1007	이문세	부장	1008	04/01/08	520	(null)	10	30	영업부	용인
36	1008	김우성	차장	1003	04/03/08	500	0	30	30	영업부	용인
37	1009	한성기	사장	(null)	96/10/04	1000	(null)	20	30	영업부	용인
38	1010	이병현	과장	1003	05/04/07	500	(null)	10	30	영업부	용인
39	1011	조향기	사원	1007	07/03/01	280	(null)	30	30	영업부	용인
40	1012	곽혜정	사원	1006	07/08/09	300	(null)	20	30	영업부	용인
41	1013	박충훈	부장	1003	02/10/09	560	(null)	20	30	영업부	용인
42	1014	조인성	사원	1006	07/11/09	250	(null)	10	30	영업부	용인
43	1001	김사랑	사원	1013	07/03/01	300	(null)	20	40	전산부	수원
44	1002	한예슬	대리	1005	07/04/02	250	80	30	40	전산부	수원
45	1003	오지호	과장	1005	05/02/10	500	100	30	40	전산부	수원
46	1004	이병현	부장	1008	03/09/02	600	(null)	20	40	전산부	수원
47	1005	신동혁	과장	1005	05/04/07	450	200	30	40	전산부	수원
48	1006	장동건	부장	1008	03/10/09	480	(null)	30	40	전산부	수원
49	1007	이문세	부장	1008	04/01/08	520	(null)	10	40	전산부	수원
50	1008	김우성	차장	1003	04/03/08	500	0	30	40	전산부	수원
51	1009	한성기	사장	(null)	96/10/04	1000	(null)	20	40	전산부	수원
52	1010	이병현	과장	1003	05/04/07	500	(null)	10	40	전산부	수원
53	1011	조향기	사원	1007	07/03/01	280	(null)	30	40	전산부	수원
54	1012	곽혜정	사원	1006	07/08/09	300	(null)	20	40	전산부	수원
55	1013	박충훈	부장	1003	02/10/09	560	(null)	20	40	전산부	수원
56	1014	조인성	사원	1006	07/11/09	250	(null)	10	40	전산부	수원

7.1 Inner Join(1)

❖ Inner Join

- Equi join과 같은 결과값을 출력하지만 사용하는 방법이 조금 다르다.
- FROM절에 조인할 테이블을 명시하고 ON 절을 이용하여 조인 조건을 명시한다.

형식

```
SELECT * FROM table1 Inner Join table2  
ON table1.column1 = table2.column2;
```

❖ ON 절에 조인 조건을 명시하고, 다른 조건은 WHERE절에 지정한다.

예제

```
SELECT ename, dname  
FROM emp Inner Join dept  
ON emp.deptno=dept.deptno  
WHERE ename='김사랑';
```

7.1 Inner Join(2)

❖ USING

- 두 테이블에 각각 조인을 정의한 컬럼의 이름이 동일하다면 USING 절에서 조인할 컬럼을 지정하여 구문을 더 간단하게 표현할 수 있다.

형식

```
SELECT * FROM table1 Join table2  
USING(공통컬럼)
```

- ## ❖ emp 테이블과 dept 테이블에 공통된 컬럼인 deptno로 조인문을 기술할 수 있다.

예제

```
SELECT ename, dname  
FROM emp Inner Join dept  
USING(deptno);
```

7.1 Inner Join(3)

❖ Natural Join

- 두 테이블에 각각 동일한 타입과 이름을 가진 컬럼을 자동 조인.
- 컬럼을 명시하지 않아도 자동으로 조인된다.
- 테이블간에 동일한 형식을 갖는 컬럼이 반드시 하나만 존재해야 한다.

형식

SELECT * FROM table1 **Natural Join table2**

- ## ❖ emp 테이블과 dept 테이블에 공통된 컬럼인 deptno를 따로 명시할 필요가 없다.

예제

**SELECT ename, dname
FROM emp **Natural Join** dept;**

7.2 Outer Join(1)

❖ Outer Join

- Left Outer Join, Right Outer Join, Full Outer Join 세 가지 타입의 조인이 있다.
- Outer Join은 이미 설명했듯이 어느 한쪽 테이블에는 해당하는 데이터가 존재하는데 다른 쪽 테이블에는 데이터가 존재하지 않을 경우 그 데이터가 출력되지 않는 문제점을 해결하기 위해 사용하는 조인 기법이다.
- Outer 문구는 생략 가능.

형식

```
SELECT * FROM table1  
[Left | Right | Full] Outer Join table2
```

7.2 Outer Join(2)

- ❖ 확인을 위해 동일한 컬럼을 가지는 테이블 2개를 만들고 데이터를 추가한다.

예제

```
CREATE TABLE DEPT01(  
    DEPTNO NUMBER(2),  
    DNAME VARCHAR2(14));
```

```
INSERT INTO DEPT01 VALUES(10, 'ACCOUNTING');  
INSERT INTO DEPT01 VALUES(20, 'RESEARCH');
```

```
CREATE TABLE DEPT02(  
    DEPTNO NUMBER(2),  
    DNAME VARCHAR2(14));
```

```
INSERT INTO DEPT02 VALUES(10, 'ACCOUNTING');  
INSERT INTO DEPT02 VALUES(30, 'SALES');
```

7.2 Outer Join(3)

- ❖ DEPT01 테이블의 20번 부서와 조인할 부서번호가 DEPT02테이블에는 없지만, 20번 부서도 출력되도록 하기 위해서 DEPT01 테이블이 왼쪽에 존재하기 때문에 **LEFT OUTER JOIN**을 사용한다.

예제

```
SELECT *
FROM DEPT01 LEFT OUTER JOIN DEPT02
USING(DEPTNO);
```

DEPTNO	DNAME	DNAME_1
10	ACCOUNTING	ACCOUNTING
20	RESEARCH	(null)

7.2 Outer Join(4)

- ❖ DEPT02 테이블을 기준으로 출력되도록 하기 위해서 **RIGHT OUTER JOIN** 을 사용한다.

예제

```
SELECT *
FROM DEPT01 RIGHT OUTER JOIN DEPT02
USING(DEPTNO);
```

DEPTNO	DNAME	DNAME_1
10	ACCOUNTING	ACCOUNTING
30	(null)	SALES

7.2 Outer Join(5)

- ❖ FULL OUTER JOIN은 LEFT JOIN과 RIGHT JOIN을 합한 형태라고 볼 수 있다.

예제

```
SELECT *
FROM DEPT01 FULL OUTER JOIN DEPT02
USING(DEPTNO);
```

DEPTNO	DNAME	DNAME_1
10	ACCOUNTING	ACCOUNTING
30	(null)	SALES
20	RESEARCH	(null)

문제

1. 경리부에서 근무하는 사원의 이름과 입사일을 출력하시오.

ENAME	HIREDATE
이문세	04/01/08
이병현	05/04/07
조인성	07/11/09

문제

2. 인천에서 근무하는 사원의 이름과 급여를 출력하시오.

ENAME	SAL
김사랑	300
이병현	600
안성기	1000
강혜정	300
박승훈	560