

데이터베이스

- 순서 : Ch4-1 실습(SQL명령, JOIN)
- 학기 : 2018학년도 2학기
- 학과 : 가천대학교 컴퓨터공학과 2학년
- 교수 : 박양재

4.1 JOIN

□ JOIN

지금까지는 하나의 테이블에 대한 SQL 명령어를 사용하였으나 관계형 데이터베이스에서는 테이블 간의 관계가 중요하기 때문에 하나 이상의 테이블을 빈번히 결합하여 사용한다. **한 개 이상의 테이블에서 데이터를 조회하기 위해서 사용되는 것을 조인**이라고 한다.

- 조인의 필요성
- CROSS JOIN
- EQUAL JOIN
- NON-EQUAL JOIN
- SELF JOIN
- OUTER JOIN
- ANSI JOIN

4.1 JOIN

JOIN

scott/tiger

01.조인의 필요성 : 데이터베이스에서는 데이터가 중복되면 여러가지 이상 현상이 발생되기 때문에 데이터가 중복되지 않도록 하기 위해서 2개 이상의 테이블로 나누어 정보를 저장해 놓는다. 하지만 원하는 정보를 얻어 오려면 여러 개의 테이블을 조인해야만 한다.

```
SQL> SELECT * FROM EMP ORDER BY DEPTNO ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7839	KING	PRESIDENT		81/11/17	5000		10
7934	MILLER	CLERK	7782	82/01/23	1300		10
7566	JONES	MANAGER	7839	81/04/02	2975		20
7902	FORD	ANALYST	7566	81/12/03	3000		20
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7369	SMITH	CLERK	7902	80/12/17	800		20
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7900	JAMES	CLERK	7698	81/12/03	950		30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30

```
SQL> SELECT * FROM DEPT ;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

14 개의 행이 선택되었습니다.

*.사원 테이블에서 부서명, 위치는 나오지 않고 DEPT테이블에서 정보를 얻어와야 한다.

4.1 JOIN

□ JOIN

예: scott 사원이 소속된 부서의 이름을 검색하세요.

- 1) 먼저 사원 테이블에서 scott 사원의 부서번호를 찾는 후
- 2) 부서 테이블에서 해당 부서명을 찾아야 한다.

```
SQL> SELECT DEPTNO FROM EMP WHERE ENAME = 'SCOTT';
```

DEPTNO
20

```
SQL> SELECT * FROM DEPT WHERE DEPTNO = 20;
```

DEPTNO	DNAME	LOC
20	RESEARCH	DALLAS

결론 : 특정 사원의 소속 부서명을 알아내기 위해서는 사원 테이블과 부서 테이블을 두 번 검색해야 하는 문제가 있다. 해결방법은 두 개 이상의 테이블을 결합해서 원하는 결과를 얻어내면 한 번의 질의로 해결할 수 있다. 이것이 JOIN 기능이다.

4.1 JOIN

JOIN

02.CROSS JOIN : 2 개 이상의 테이블이 조인 될 때, WHERE절에 의해 공통되는 컬럼에 의한 결합이 발생되지 않는 경우 (특별한 키워드 없이 테이블과 테이블을 연속으로 결합)

예: 사원 테이블과 부서 테이블을 CROSS JOIN 하세요.

```
SQL> SELECT * FROM EMP, DEPT;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7369	SMITH	CLERK	7902	80/12/17	800		20	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	81/04/02	2975		20	10	ACCOUNTING	NEW YORK
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	10	ACCOUNTING	NEW YORK
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	10	ACCOUNTING	NEW YORK
7782	CLARK	MANAGER	7839	81/06/09	2450		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	10	ACCOUNTING	NEW YORK
7839	KING	PRESIDENT		81/11/17	5000		10	10	ACCOUNTING	NEW YORK
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	10	ACCOUNTING	NEW YORK
7876	ADAMS	CLERK	7788	87/05/23	1100		20	10	ACCOUNTING	NEW YORK
.....										
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	40	OPERATIONS	BOSTON
7782	CLARK	MANAGER	7839	81/06/09	2450		10	40	OPERATIONS	BOSTON
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	40	OPERATIONS	BOSTON
7839	KING	PRESIDENT		81/11/17	5000		10	40	OPERATIONS	BOSTON
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	40	OPERATIONS	BOSTON
7876	ADAMS	CLERK	7788	87/05/23	1100		20	40	OPERATIONS	BOSTON
7900	JAMES	CLERK	7698	81/12/03	950		30	40	OPERATIONS	BOSTON
7902	FORD	ANALYST	7566	81/12/03	3000		20	40	OPERATIONS	BOSTON
7934	MILLER	CLERK	7782	82/01/23	1300		10	40	OPERATIONS	BOSTON

56 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

02.CROSS JOIN : 2 개 이상의 테이블이 조인 될 때, WHERE절에 의해 공통되는 컬럼에 의한 결합이 발생되지 않는 경우 (특별한 키워드 없이 테이블과 테이블을 연속으로 결합)

CROSS JOIN 결과 :

컬럼 수는 사원 테이블(8) + 부서 테이블(3) = 11컬럼, 로우 수는 사원 1명에 대해 부서 테이블의 4개 로우와 결합되므로 14개 * 4개 = 56개가 되며, 조인 시 아무런 조건이 제시되지 않았기 때문에 의미가 없다. 조인 결과가 의미를 가지려면 조건을 지정해야 한다.

[조인의 종류]

종류	설명
Equi Join	동일 컬럼을 기준으로 조인
Non-Equi Join	동일 컬럼이 없이 다른 조건을 사용하여 조인
Outer Join	조인 조건을 만족하지 않는 로우도 나타냄
Self Join	한 테이블에서 조인

4.1 JOIN

□ JOIN

03.EQUI JOIN : 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치하는 행을 연결하여 결과를 생성하는 방법

예 : 사원 테이블과 부서 테이블의 공통적으로 존재하는 컬럼인 DEPTNO의 값이 일치하는 행들을 연결하여 두 테이블을 결합한다.

```
SELECT *  
FROM EMP, DEPT  
WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7369	SMITH	CLERK	7902	80/12/17	800		20	20	RESEARCH	DALLAS
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	30	SALES	CHICAGO
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	30	SALES	CHICAGO
7566	JONES	MANAGER	7839	81/04/02	2975		20	20	RESEARCH	DALLAS
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30	30	SALES	CHICAGO
7698	BLAKE	MANAGER	7839	81/05/01	2850		30	30	SALES	CHICAGO
7782	CLARK	MANAGER	7839	81/06/09	2450		10	10	ACCOUNTING	NEW YORK
7788	SCOTT	ANALYST	7566	87/04/19	3000		20	20	RESEARCH	DALLAS
7839	KING	PRESIDENT		81/11/17	5000		10	10	ACCOUNTING	NEW YORK
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30	30	SALES	CHICAGO
7876	ADAMS	CLERK	7788	87/05/23	1100		20	20	RESEARCH	DALLAS
7900	JAMES	CLERK	7698	81/12/03	950		30	30	SALES	CHICAGO
7902	FORD	ANALYST	7566	81/12/03	3000		20	20	RESEARCH	DALLAS
7934	MILLER	CLERK	7782	82/01/23	1300		10	10	ACCOUNTING	NEW YORK

14 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

03.EQUI JOIN : 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치하는 행을 연결하여 결과를 생성하는 방법

예 : 조인 결과 사원 테이블의 DEPTNO와 부서 테이블의 DEPTNO가 동일한 행들을 모두 표시하였지만, 사원의 사원명과 부서명만을 알아볼 수는 없다. 사원명과 부서명만을 출력하기 위해서는 SQL명령을 수정해야 한다.

```
SQL> SELECT ENAME, DNAME  
2 FROM EMP, DEPT  
3 WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

14 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

03.EQUI JOIN : 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치하는 행을 연결하여 결과를 생성하는 방법

예 : SCOTT 사원의 이름과 부서명만을 출력하기 위해서는 SQL명령을 수정해야 한다.

```
SQL> SELECT ENAME, DNAME
2  FROM EMP, DEPT
3  WHERE EMP.DEPTNO=DEPT.DEPTNO
4  AND ENAME='SCOTT';
```

ENAME	DNAME
SCOTT	RESEARCH

만일, SELECT ENAME, DNAME, DEPTNO로 기술하면 에러가 발생한다. DEPTNO가 어떤 테이블의 컬럼인지 알 수 없기 때문이다. 두 테이블에서 컬럼명이 동일한 경우는 반드시 테이블명.컬럼명을 지정해야 한다.

4.1 JOIN

□ JOIN

03.EQUI JOIN : 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치하는 행을 연결하여 결과를 생성하는 방법

두 개 이상의 테이블이 조인 할 경우, 컬럼 앞에 테이블명을 기술하는 일이 많아지는데 테이블명이 길면 번거롭기 때문에 **FROM 절 다음에 테이블명을 명시하고 공백을 둔 다음 별칭을 부여하여 사용한다.**

FROM EMP E, DEPT D

```
SQL> SELECT E.ENAME, D.DNAME, D.DEPTNO  
2 FROM EMP E, DEPT D  
3 WHERE E.DEPTNO=D.DEPTNO  
4 AND E.ENAME='SCOTT';
```

ENAME	DNAME	DEPTNO
SCOTT	RESEARCH	20

4.1 JOIN

□ JOIN

03.EQUI JOIN : 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치하는 행을 연결하여 결과를 생성하는 방법

예1: 조인을 사용하여 뉴욕에 근무하는 사원의 이름과 급여를 출력하세요.

```
SELECT E.ENAME, E.SAL, D.LOC  
FROM EMP E, DEPT D  
WHERE E.DEPTNO=D.DEPTNO  
AND D.LOC='NEW YORK';
```

ENAME	SAL	LOC
CLARK	2450	NEW YORK
KING	5000	NEW YORK
MILLER	1300	NEW YORK

4.1 JOIN

□ JOIN

03.EQUI JOIN : 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치하는 행을 연결하여 결과를 생성하는 방법

예2: 조인을 사용하여 ACCOUNTING 부서 소속 사원의 이름과 입사일을 출력하세요.

```
SELECT E.ENAME, E.HIREDATE, D.DNAME  
FROM EMP E, DEPT D  
WHERE E.DEPTNO=D.DEPTNO  
AND D.DNAME= 'ACCOUNTING' ;
```

ENAME	HIREDATE	DNAME
CLARK	81/06/09	ACCOUNTING
KING	81/11/17	ACCOUNTING
MILLER	82/01/23	ACCOUNTING

4.1 JOIN

□ JOIN

03.EQUI JOIN : 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치하는 행을 연결하여 결과를 생성하는 방법

예3: 조인을 사용하여 직급이 MANAGER인 사원의 이름과 부서명을 출력하세요.

```
SELECT E.ENAME, D.DNAME, E.JOB  
FROM EMP E, DEPT D  
WHERE E.DEPTNO=D.DEPTNO  
AND E.JOB= 'MANAGER';
```

ENAME	DNAME	JOB
JONES	RESEARCH	MANAGER
BLAKE	SALES	MANAGER
CLARK	ACCOUNTING	MANAGER

4.1 JOIN

□ JOIN

04.Non-EQUI JOIN : 조인 조건의 특정범위 내에 있는지 조사하기 위해 WHERE절 의 조인조건으로 = 연산자와 비교 연산자를 사용한다.

급여 등급 테이블의 내용은 다음과 같다.

```
SQL> SELECT * FROM SALGRADE;
```

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

급여등급 테이블에 대한 등급은 총 5등급으로, 1등급(700~1200), 2등급(1201~1400), 3등급(1401~2000), 4등급(2000~3000), 5등급(3001~9999)이다.

급여 등급을 5개로 나누어 놓은 SALGRADE테이블에서 정보를 얻어와서 각 사원의 급여등급을 부여해 보자. 그러기 위해서는 사원 테이블과 급여등급 테이블을 조인한다.

4.1 JOIN

□ JOIN

04.Non-EQUI JOIN : 조인 조건의 특정범위 내에 있는지 조사하기 위해 WHERE절 의 조인조건으로 = 연산자와 비교 연산자를 사용한다.

사원의 급여가 몇 등급인지 확인

```
SELECT ENAME, SAL, GRADE  
FROM EMP, SALGRADE  
WHERE SAL BETWEEN LOSAL AND HISAL;
```

ENAME	SAL	GRADE
SMITH	800	1
JAMES	950	1
ADAMS	1100	1
WARD	1250	2
MARTIN	1250	2
MILLER	1300	2
TURNER	1500	3
ALLEN	1600	3
CLARK	2450	4
BLAKE	2850	4
JONES	2975	4
SCOTT	3000	4
FORD	3000	4
KING	5000	5

14 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

04.Non-EQUI JOIN : 조인 조건의 특정범위 내에 있는지 조사하기 위해 WHERE절 의 조인조건으로 = 연산자와 비교 연산자를 사용한다.

사원의 급여가 몇 등급인지 확인

```
SELECT E.ENAME, E.SAL, S.GRADE
FROM   EMP E, SALGRADE S
WHERE  E.SAL >= S.LOSAL AND E.SAL <= S.HISAL;
```

ENAME	SAL	GRADE
SMITH	800	1
JAMES	950	1
ADAMS	1100	1
WARD	1250	2
MARTIN	1250	2
MILLER	1300	2
TURNER	1500	3
ALLEN	1600	3
CLARK	2450	4
BLAKE	2850	4
JONES	2975	4
SCOTT	3000	4
FORD	3000	4
KING	5000	5

14 개의 행이 선택되었습니다.

4.1 JOIN

❑ JOIN

04.Non-EQUI JOIN : 조인 조건의 특정범위 내에 있는지 조사하기 위해 WHERE절 의 조인조건으로 = 연산자와 비교 연산자를 사용한다.

한 명사원 이름과 소속부서, 급여의 등급을 출력한다. 사원 테이블에는 부서이름과 급여등급이 없고, 부서번호와 급여만 있다. 사원 테이블의 부서번호로 부서 테이블을 참조하여 부서명을 얻어와야하고, 사원 테이블의 급여로 급여등급 테이블을 참조하여 등급을 얻어온다. 즉, 3개 테이블(EMP, DEPT, SALGRADE)를 조인해야 한다.

```
SELECT E.ENAME, D.DNAME, S.GRADE  
FROM   EMP E, SALGRADE S, DEPT D  
WHERE  E.DEPTNO=D.DEPTNO  
AND    E.SAL BETWEEN S.LOSAL AND S.HISAL;
```

4.1 JOIN

□ JOIN

05.SELF JOIN : 하나의 테이블 내에서 조인(자기 자신과 조인)

예: 이름이 SMITH인 사원의 매니저 이름을 알아내려면 어떻게 해야하나?

```
SQL> SELECT ENAME, MGR FROM EMP;
```

ENAME	MGR
SMITH	7902
ALLEN	7698
WARD	7698
JONES	7839
MARTIN	7698
BLAKE	7839
CLARK	7839
SCOTT	7566
KING	
TURNER	7698
ADAMS	7788
JAMES	7698
FORD	7566
MILLER	7782

14 개의 행이 선택되었습니다.

```
SQL> SELECT EMPNO, ENAME FROM EMP;
```

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

14 개의 행이 선택되었습니다.

매니저도 사원이기 때문에 매니저를 위한 테이블이 있으면 데이터 중복이 발생되므로 사원 테이블을 참조해서 EMP테이블이 EMP테이블을 조인한다.

4.1 JOIN

□ JOIN

05.SELF JOIN : 하나의 테이블 내에서 조인(자기 자신과 조인)

예: 이름이 SMITH인 사원의 매니저 이름을 알아내려면 어떻게 해야하나?

하나의 테이블로 조인하려면 FROM절 다음에 EMP 테이블을 두번 기술할 수 없다.
그러므로 같은 테이블이 하나 더 존재하는 것 처럼 생각할 수 있도록 테이블 별칭을 사용한다.

```
SQL> SELECT ENAME, MGR FROM EMP;
```

ENAME	MGR
SMITH	7902
ALLEN	7698
WARD	7698
JONES	7839
MARTIN	7698
BLAKE	7839
CLARK	7839
SCOTT	7566
KING	
TURNER	7698
ADAMS	7788
JAMES	7698
FORD	7566
MILLER	7782

14 개의 행이 선택되었습니다.

EMPLOYEE 테이블

```
SQL> SELECT EMPNO, ENAME FROM EMP;
```

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7521	WARD
7566	JONES
7654	MARTIN
7698	BLAKE
7782	CLARK
7788	SCOTT
7839	KING
7844	TURNER
7876	ADAMS
7900	JAMES
7902	FORD
7934	MILLER

14 개의 행이 선택되었습니다.

MANAGER 테이블

4.1 JOIN

□ JOIN

05.SELF JOIN : 하나의 테이블 내에서 조인(자기 자신과 조인)

예: 이름이 SMITH인 사원의 매니저 이름을 알아내려면 어떻게 해야하나?

```
SELECT EMPLOYEE.ENAME || '의 매니저는 ' || MANAGER.ENAME || '입니다'
FROM EMPLOYEE, EMP_MANAGER
WHERE EMPLOYEE.MGR=MANAGER.EMPNO ;
```

EMPLOYEE.ENAME || '의 매니저는' || MANAGER.

FORD의 매니저는 JONES입니다
SCOTT의 매니저는 JONES입니다
JAMES의 매니저는 BLAKE입니다
TURNER의 매니저는 BLAKE입니다
MARTIN의 매니저는 BLAKE입니다
WARD의 매니저는 BLAKE입니다
ALLEN의 매니저는 BLAKE입니다
MILLER의 매니저는 CLARK입니다
ADAMS의 매니저는 SCOTT입니다
CLARK의 매니저는 KING입니다
BLAKE의 매니저는 KING입니다
JONES의 매니저는 KING입니다
SMITH의 매니저는 FORD입니다

13 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

05.SELF JOIN : 하나의 테이블 내에서 조인(자기 자신과 조인)

예제1: 매니저가 KING인 직원들의 직급을 출력하세요.

```
SELECT WORK.ENAME, WORK.JOB  
FROM EMP WORK, EMP MANAGER  
WHERE WORK.MGR=MANAGER.EMPNO  
AND MANAGER.ENAME='KING';
```

ENAME	JOB
JONES	MANAGER
BLAKE	MANAGER
CLARK	MANAGER

4.1 JOIN

❑ JOIN

05.SELF JOIN : 하나의 테이블 내에서 조인(자기 자신과 조인)

예제2: SCOTT과 동일한 근무지에 근무하는 사원의 이름을 출력 하세요.

```
SELECT WORK.ENAME, FRIEND.ENAME  
FROM EMP WORK, EMP FRIEND  
WHERE WORK.DEPTNO=FRIEND.DEPTNO  
AND WORK.ENAME='SCOTT'  
AND FRIEND.ENAME<>'SCOTT' ;
```

ENAME	ENAME
SCOTT	SMITH
SCOTT	JONES
SCOTT	ADAMS
SCOTT	FORD

```
SELECT WORK.ENAME, FRIEND.ENAME  
FROM EMP WORK, EMP FRIEND  
WHERE WORK.DEPTNO=FRIEND.DEPTNO  
AND WORK.ENAME='SCOTT'  
--AND FRIEND.ENAME<>'SCOTT' ;
```

ENAME	ENAME
SCOTT	SMITH
SCOTT	JONES
SCOTT	SCOTT
SCOTT	ADAMS
SCOTT	FORD

4.1 JOIN

□ JOIN

06.OUTER JOIN : 2개 이상의 테이블이 조인될 때, 어느 한쪽의 테이블에는 해당하는 데이터가 존재하는데 다른 한쪽 테이블에는 데이터가 존재하지 않는 경우 그 데이터를 출력되지 않는 문제를 해결하기 위해서 사용되는 조인 기법이다.

SELF 조인 결과 13명만 출력되고, 이름이 KING인 사원은 빠져 있다. KING은 이 회사의 사장이므로 매니저가 존재하지 않으므로 MGR값이 NULL이다. **사원번호(EMPNO)가 NULL인 사원은 없으므로 조인 조건에 만족하지 않기 때문에 KING은 SELF JOIN의 결과 배제** 되었다.

이와 같이 조인 조건에 만족하지 못하였더라도 해당 행을 나타내고 싶을 때 사용하는 조인이 외부조인(Outer Join)이다. OUTER JOIN은 NULL값이기 때문에 배제된 행을 결과에 포함 시킬 수 있으며, **(+)기호를 조인조건에서 정보가 부족한 컬럼 뒤에 붙여 사용**한다.

4.1 JOIN

JOIN

06.OUTER JOIN : 2개 이상의 테이블이 조인될 때, 어느 한쪽의 테이블에는 해당하는 데이터가 존재하는데 다른 한쪽 테이블에는 데이터가 존재하지 않는 경우 그 데이터를 출력되지 않는 문제를 해결하기 위해서 사용되는 조인 기법이다.

```
SELECT EMPLOYEE.ENAME || '의 매니저는 ' || MANAGER.ENAME || '입니다'
FROM EMPLOYEE, EMP_MANAGER
WHERE EMPLOYEE.MGR=MANAGER.EMPNO ;
```

EMPLOYEE.ENAME || '의 매니저는' || MANAGER.

FORD의 매니저는 JONES입니다
SCOTT의 매니저는 JONES입니다
JAMES의 매니저는 BLAKE입니다
TURNER의 매니저는 BLAKE입니다
MARTIN의 매니저는 BLAKE입니다
WARD의 매니저는 BLAKE입니다
ALLEN의 매니저는 BLAKE입니다
MILLER의 매니저는 CLARK입니다
ADAMS의 매니저는 SCOTT입니다
CLARK의 매니저는 KING입니다
BLAKE의 매니저는 KING입니다
JONES의 매니저는 KING입니다
SMITH의 매니저는 FORD입니다

13 개의 행이 선택되었습니다.

```
SELECT EMPLOYEE.ENAME || '의 매니저는 ' || MANAGER.ENAME || '입니다'
FROM EMPLOYEE, EMP_MANAGER
WHERE EMPLOYEE.MGR=MANAGER.EMPNO(+);
```

EMPLOYEE.ENAME || '의 매니저는' || MANAGER.

FORD의 매니저는 JONES입니다
SCOTT의 매니저는 JONES입니다
JAMES의 매니저는 BLAKE입니다
TURNER의 매니저는 BLAKE입니다
MARTIN의 매니저는 BLAKE입니다
WARD의 매니저는 BLAKE입니다
ALLEN의 매니저는 BLAKE입니다
MILLER의 매니저는 CLARK입니다
ADAMS의 매니저는 SCOTT입니다
CLARK의 매니저는 KING입니다
BLAKE의 매니저는 KING입니다
JONES의 매니저는 KING입니다
SMITH의 매니저는 FORD입니다
KING의 매니저는 입니다

14 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

06.OUTER JOIN : 2개 이상의 테이블이 조인될 때, 어느 한쪽의 테이블에는 해당하는 데이터가 존재하는데 다른 한쪽 테이블에는 데이터가 존재하지 않는 경우 그 데이터를 출력되지 않는 문제를 해결하기 위해서 사용되는 조인 기법이다.

예제 1: 사원 테이블과 부서 테이블을 조인하여 사원명과 부서번호와 부서명을 출력하세요.

```
SELECT E.ENAME, D.DEPTNO, D.DNAME  
FROM EMP E, DEPT D  
WHERE E.DEPTNO=D.DEPTNO  
ORDER BY D.DEPTNO ;
```

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
KING	10	ACCOUNTING
MILLER	10	ACCOUNTING
JONES	20	RESEARCH
FORD	20	RESEARCH
ADAMS	20	RESEARCH
SMITH	20	RESEARCH
SCOTT	20	RESEARCH
WARD	30	SALES
TURNER	30	SALES
ALLEN	30	SALES
JAMES	30	SALES
BLAKE	30	SALES
MARTIN	30	SALES

14 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

06.OUTER JOIN : 2개 이상의 테이블이 조인될 때, 어느 한쪽의 테이블에는 해당하는 데이터가 존재하는데 다른 한쪽 테이블에는 데이터가 존재하지 않는 경우 그 데이터를 출력되지 않는 문제를 해결하기 위해서 사용되는 조인 기법이다.

예제2: 부서 테이블의 40번 부서와 조인 할 사원 테이블의 부서 번호는 없지만, 그림과 같이 40번 부서의 부서 이름도 출력되도록 쿼리문을 작성 하세요.

```
SELECT E.ENAME, D.DEPTNO, D.DNAME  
FROM EMP E, DEPT D  
WHERE E.DEPTNO(+) = D.DEPTNO  
ORDER BY D.DEPTNO ;
```

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
KING	10	ACCOUNTING
JONES	20	RESEARCH
SMITH	20	RESEARCH
SCOTT	20	RESEARCH
FORD	20	RESEARCH
ADAMS	20	RESEARCH
WARD	30	SALES
TURNER	30	SALES
ALLEN	30	SALES
JAMES	30	SALES
MARTIN	30	SALES
BLAKE	30	SALES
	40	OPERATIONS

15 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

07.ANSI JOIN : 앞에서 설명한 JOIN은 SQL/86에 정의된 조인 구문이며, 대부분의 상용 DBMS에서 표준언어로 미국표준협회(ANSI)SQL에서 제시한 표준 기능을 대부분 준수하고 있다. 다른 DBMS와 호환성을 위해서 ANSI 조인을 사용하는 것이 좋다.

01)ANSI CROSS JOIN

이전 CROSS JOIN 구문에서는 WHERE절을 생략하고 FROM 다음에 테이블명을 구별하기 위해 쉼표 기술했으나 ANSI CROSS JOIN에서는 **쉼표없이 CROSS JOIN** 지정한다.(조인 타입을 명확하게 구분해서 지정 가능하다.)

```
SQL> SELECT * FROM EMP, DEPT ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7369	SMITH	CLERK	7902	80/12/17	800		20	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	81/04/02	2975		20	10	ACCOUNTING	NEW YORK

```
SQL> SELECT * FROM EMP CROSS JOIN DEPT ;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DEPTNO	DNAME	LOC
7369	SMITH	CLERK	7902	80/12/17	800		20	10	ACCOUNTING	NEW YORK
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30	10	ACCOUNTING	NEW YORK
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30	10	ACCOUNTING	NEW YORK
7566	JONES	MANAGER	7839	81/04/02	2975		20	10	ACCOUNTING	NEW YORK

4.1 JOIN

❑ JOIN

02)ANSI INNER JOIN

동일 컬럼을 기준으로 조인하기 위한 EQUI JOIN을 ANSI JOIN에서는 **FROM절** 다음에 **INNER JOIN**이란 단어를 사용하여 조인 할 테이블명을 명시하고, **ON절**을 사용하여 조인 조건을 명시한다.

```
SELECT ENAME, DNAME  
FROM EMP, DEPT  
WHERE EMP.DEPTNO=DEPT.DEPTNO ;
```

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

14 개의 행이 선택되었습니다.

```
SELECT ENAME, DNAME  
FROM EMP INNER JOIN DEPT  
ON EMP.DEPTNO=DEPT.DEPTNO ;
```

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

14 개의 행이 선택되었습니다.

4.1 JOIN

❑ JOIN

02)ANSI INNER JOIN

예:SCOTT의 부서명을 알아내기 위한 쿼리문을 기존 조인구문으로 작성하면

```
SELECT ENAME, DNAME  
FROM EMP,DEPT  
WHERE EMP.DEPTNO=DEPT.DEPTNO  
AND ENAME='SCOTT' ;
```

ENAME	DNAME
SCOTT	RESEARCH

INNER JOIN 구문으로 표시하면 조인정보를 ON절에 기술하여 조건을 명확히 지정하고 다른 조건은 WHERE 구문에서 지정하여 간결하다.

```
SELECT ENAME, DNAME  
FROM EMP INNER JOIN DEPT  
ON EMP.DEPTNO=DEPT.DEPTNO  
WHERE ENAME='SCOTT' ;
```

ENAME	DNAME
SCOTT	RESEARCH

4.1 JOIN

□ JOIN

02)ANSI INNER JOIN

(1)USING을 이용한 조인 조건 지정하기

앞에서는 ON절을 사용하여 조인조건을 지정했으나, 대부분의 조인은 동등조인이다. 동등조인에서는 두 테이블의 관련 컬럼을 비교하여 값이 같은지 알아볼 수 있다. 컬럼명이 동일하면 USING절에서 지정하여 간단히 표현이 가능하다.

```
SELECT ENAME, DNAME  
FROM EMP INNER JOIN DEPT  
USING (DEPTNO) ;
```

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

14 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

02)ANSI INNER JOIN

(2)NATURAL JOIN

USING절은 동등조인을 간편하고 쉽게 표현할 수 있는 구문이며, ON절보다 이해하기 쉬운 질의 결과를 얻을 수 있지만, NATURAL JOIN이 더욱 간결하여 쉽고 많이 사용한다.

이전 조인에서 두 개 이상의 테이블을 SELECT하려면 WHERE절에 테이블의 공통된 컬럼을 사용하였으나 ANSI 조인 부터는 NATURAL JOIN을 사용하여 조건없이 사용하는 것이 가능해 졌다.

```
SELECT ENAME, DNAME  
FROM EMP NATURAL JOIN DEPT ;
```

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
WARD	SALES
JONES	RESEARCH
MARTIN	SALES
BLAKE	SALES
CLARK	ACCOUNTING
SCOTT	RESEARCH
KING	ACCOUNTING
TURNER	SALES
ADAMS	RESEARCH
JAMES	SALES
FORD	RESEARCH
MILLER	ACCOUNTING

14 개의 행이 선택되었습니다.

4.1 JOIN

□ JOIN

03)ANSI OUTER JOIN : 기존 조인에서는 모든 레코드가 반드시 출력되어야 할 때는 (+)표시를 했다. 하지만 ANSI조인에서는 OUTER JOIN을 그 전까지 지원하지 않았던 FULL까지 지원하여 LEFT, RIGHT, FULL 3가지를 지원한다.

[형식] SELECT * FROM table1[LEFT|RIGHT|FULL] OUTER JOIN table2

OUTER JOIN은 어느 한쪽 테이블에는 해당하는 데이터가 존재하는데 다른 테이블에는 데이터가 존재하지 않을 경우, 그 데이터가 출력되지 않는 문제점을 해결하기 위해서 사용하는 조인 기법이다.

4.1 JOIN

□ JOIN

03)ANSI OUTER JOIN

예제: OUTER JOIN은 어느 한쪽 테이블에는 해당하는 데이터가 존재하는데 다른 테이블에는 데이터가 존재하지 않을 경우, 그 데이터가 출력되지 않는 문제점을 해결하기 위해서 사용하는 조인 기법이다.

실습1. 예제 테이블 존재 확인

```
SQL> DROP TABLE DEPT01;  
DROP TABLE DEPT01  
*  
1행에 오류:  
ORA-00942: 테이블 또는 뷰가 존재하지 않습니다
```

실습2. DEPT01 테이블 생성 및 확인

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

```
SQL> SELECT * FROM DEPT01 ;
```

선택된 레코드가 없습니다.

4.1 JOIN

□ JOIN

03)ANSI OUTER JOIN

예제: OUTER JOIN은 어느 한쪽 테이블에는 해당하는 데이터가 존재하는데 다른 테이블에는 데이터가 존재하지 않을 경우, 그 데이터가 출력되지 않는 문제점을 해결하기 위해서 사용하는 조인 기법이다.

실습3. 예제 테이블에 데이터를 추가 및 확인

```
SQL> INSERT INTO DEPT01 VALUES(10, 'ACCOUNT');
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO DEPT01 VALUES(20, 'RESEARCH');
```

1 개의 행이 만들어졌습니다.

```
SQL> SELECT * FROM DEPT01 ;
```

DEPTNO	DNAME
--------	-------

10	ACCOUNT
----	---------

20	RESEARCH
----	----------

4.1 JOIN

□ JOIN

03)ANSI OUTER JOIN

실습4. 동일한 방법으로 DEPT2 테이블 생성 및 행 추가 확인

```
SQL> DROP TABLE DEPT02;  
DROP TABLE DEPT02
```

```
*  
1행에 오류:  
ORA-00942: 테이블 또는 뷰가 존재하지 않습니다
```

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

테이블이 생성되었습니다.

```
SQL> INSERT INTO DEPT02 VALUES(10, 'ACCOUNT');
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO DEPT02 VALUES(30, 'SALES');
```

1 개의 행이 만들어졌습니다.

```
SQL> SELECT * FROM DEPT02;
```

DEPTNO	DNAME
10	ACCOUNT
30	SALES

4.1 JOIN

□ JOIN

03)ANSI OUTER JOIN

실습5. DEPT01 테이블의 20번 부서와 DEPT02 테이블에는 조인할 부서번호가 없지만 20번 부서도 출력하기 위해서 DEPT02.DEPTNO(+)기호를 붙였다.

```
SELECT *  
FROM DEPT01, DEPT02  
WHERE DEPT01.DEPTNO=DEPT02.DEPTNO(+) ;|
```

DEPTNO	DNAME	DEPTNO	DNAME
10	ACCOUNT	10	ACCOUNT
20	RESEARCH		

(1)LEFT OUTER JOIN

```
SELECT *  
FROM DEPT01 LEFT OUTER JOIN DEPT02  
ON DEPT01.DEPTNO=DEPT02.DEPTNO ;|
```

DEPTNO	DNAME	DEPTNO	DNAME
10	ACCOUNT	10	ACCOUNT
20	RESEARCH		

4.1 JOIN

❑ JOIN

03) ANSI OUTER JOIN

(2) RIGHT OUTER JOIN – DEPT02 테이블에만 있는 30번 부서까지 출력되도록 하기 위해서 사용

```
SELECT *  
FROM DEPT01 RIGHT OUTER JOIN DEPT02  
ON DEPT01.DEPTNO=DEPT02.DEPTNO ;
```

DEPTNO	DNAME
--------	-------

10	ACCOUNT
----	---------

DEPTNO	DNAME
--------	-------

10	ACCOUNT
30	SALES

(3) FULL OUTER JOIN – LEFT OUTER JOIN과 RIGHT OUTER JOIN을 합한 형태

```
SELECT *  
FROM DEPT01 FULL OUTER JOIN DEPT02  
ON DEPT01.DEPTNO=DEPT02.DEPTNO ;
```

DEPTNO	DNAME
--------	-------

10	ACCOUNT
----	---------

20	RESEARCH
----	----------

DEPTNO	DNAME
--------	-------

10	ACCOUNT
30	SALES