

Memory

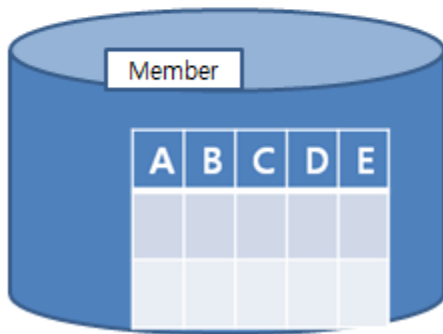



SELECT a.b  
FROM member ;



A B ?

A B C D E ?



HDD

컴퓨터의 원리는  
동일합니다!

EMP

EMPNO	ENAME	.....	DEPTNO
7369	SMITH		20
7499	ALLEN		30
7782	CLARK		10

DEPT

DEPTNO	DNAME
10	ACCOUNTING
20	RESEARCH
30	SALES

ENAME	DNAME
SMITH	RESEARCH
ALLEN	SALES
CLARK	ACCOUNTING

RESULT

## [ Oracle Join 문법 ]

```
SQL> SELECT a.col1 , b.col1  
2 FROM table1 a , table2 b  
3 WHERE a.col2 = b.col2 ;
```

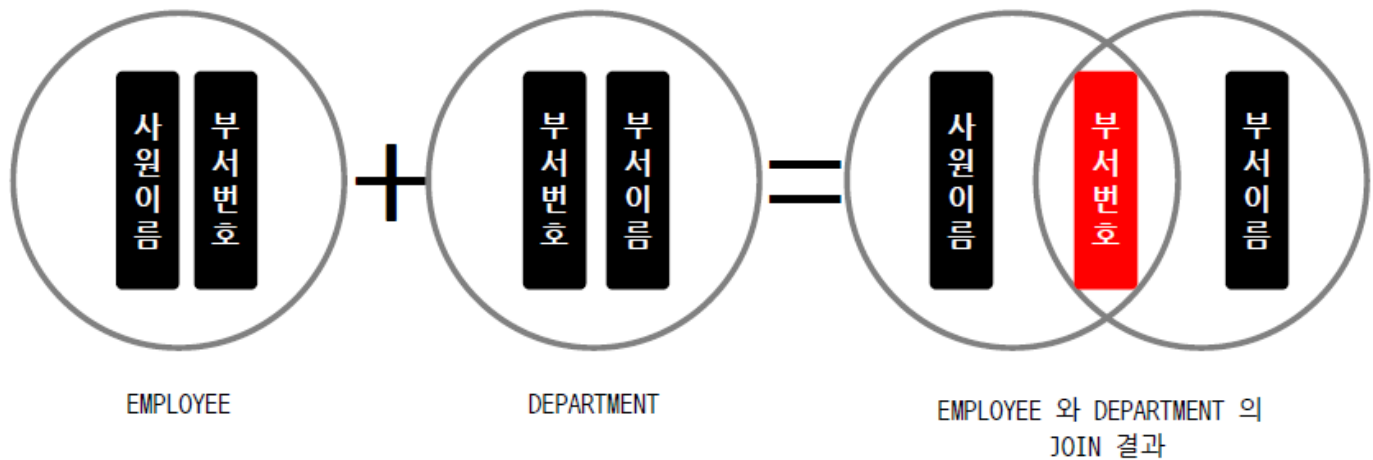
```
SQL> SELECT table1.col1 , table2.col2  
2 FROM table1 alias1 , table2 alias2  
3 WHERE table1.col2 = table2.col2 ;
```

## [ ANSI Join 문법 ]

```
SQL> SELECT a.col1 , b.col1  
2 FROM table1 a [INNER] JOIN table2 b  
3 ON a.col2 = b.col2 ;
```

서로 연관되고 다른 테이블에 존재하는 컬럼들을 한번에 조회하기 위해 사용하는 대표적인 기법

[JOIN 개념 도식 1]



```
SELECT EMP_NAME, DEPT_NAME
FROM   EMPLOYEE E,
       DEPARTMENT D
WHERE  E.DEPT_ID = D.DEPT_ID;
```

- FROM 절에 조회 대상 테이블을 심표로 구분하여 기술
- WHERE 절에 테이블 사이의 관계를 표시하는 조건 기술
- 동일한 이름의 컬럼이 여러 테이블에 존재하는 경우  
→ SELECT 절/WHERE 절에 컬럼 이름을 기술할 때 어떤 테이블에 포함된 컬럼인지 구분해서 표시
- 테이블 이름도 별칭을 사용할 수 있다

EMPLOYEE(일부)	
EMP_NAME	DEPT_ID
한선기	90
강중훈	90
최만식	90
정도연	60
안석규	60
조재형	60
정지현	50
김예수	50
나승원	50
김순이	50
성해교	50

DEPARTMENT	
DEPT_ID	DEPT_NAME
20	회계팀
10	본사 인사팀
50	해외영업1팀
60	기술지원팀
80	해외영업2팀
90	해외영업3팀
30	마케팅팀

????(일부)	
EMP_NAME	DEPT_NAME
감우섭	회계팀
이중기	회계팀
김술오	회계팀
임영애	본사 인사팀
권상후	본사 인사팀
고승우	본사 인사팀
박하일	해외영업1팀
성해교	해외영업1팀
김순이	해외영업1팀

## JOIN - OUTER JOIN

조건을 만족시키지 못하는 행까지 Result Set에 포함시키는 조인 유형

EMP_NAME	DEPT_ID
한선기	90
강중훈	90
최만식	90
정도연	60
안석규	60
조재형	60
정지현	50
김예수	50
나승원	50
김순이	50
성해교	50
전우성	80
엄정하	80
심하균	
고승우	10
박하일	50
권상후	10
임영애	10
엄정하	
김솔오	20
이중기	20
감우섭	20

DEPT_ID	DEPT_NAME
20	회계팀
10	본사 인사팀
50	해외영업1팀
60	기술지원팀
80	해외영업2팀
90	해외영업3팀
30	마케팅팀

EMP_NAME	DEPT_NAME
한선기	해외영업3팀
강중훈	해외영업3팀
최만식	해외영업3팀
정도연	기술지원팀
안석규	기술지원팀
조재형	기술지원팀
정지현	해외영업1팀
김예수	해외영업1팀
나승원	해외영업1팀
김순이	해외영업1팀
성해교	해외영업1팀
전우성	해외영업2팀
엄정하	해외영업2팀
고승우	본사 인사팀
박하일	해외영업1팀
권상후	본사 인사팀
임영애	본사 인사팀
김솔오	회계팀
이중기	회계팀
감우섭	회계팀

EMP_NAME	DEPT_NAME
감우섭	회계팀
이중기	회계팀
김솔오	회계팀
임영애	본사 인사팀
권상후	본사 인사팀
고승우	본사 인사팀
박하일	해외영업1팀
성해교	해외영업1팀
김순이	해외영업1팀
나승원	해외영업1팀
김예수	해외영업1팀
정지현	해외영업1팀
조재형	기술지원팀
안석규	기술지원팀
정도연	기술지원팀
엄정하	해외영업2팀
전우성	해외영업2팀
최만식	해외영업3팀
강중훈	해외영업3팀
한선기	해외영업3팀
엄정하	
심하균	

- 연산자 '+' 사용
- 조인 조건을 만족시키는 행이 없는 테이블 기준

```
SELECT EMP_NAME, DEPT_NAME
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.DEPT_ID = D.DEPT_ID(+);
```

소속 부서가 없는 직원까지 포함하는 의미

EMP_NAME	DEPT_NAME
감우섭	회계팀
이중기	회계팀
김솔오	회계팀
임영애	본사 인사팀
권상후	본사 인사팀
고승우	본사 인사팀
박하일	해외영업1팀
성해교	해외영업1팀
김순이	해외영업1팀
나승원	해외영업1팀
김예수	해외영업1팀
정지현	해외영업1팀
조재형	기술지원팀
안석규	기술지원팀
정도연	기술지원팀
엄정하	해외영업2팀
전우성	해외영업2팀
최만식	해외영업3팀
강중훈	해외영업3팀
한선기	해외영업3팀
엄정하	
심하균	

EMP_NAME	DEPT_ID
한선기	90
강중훈	90
최만식	90
정도연	60
안석규	60
조재형	60
정지현	50
김예수	50
나승원	50
김순이	50
성해교	50
전우성	80
엄정하	80
심하균	
고승우	10
박하일	50
권상후	10
임영애	10
엄정하	
김솔오	20
이중기	20
감우섭	20

DEPT_ID	DEPT_NAME
20	회계팀
10	본사 인사팀
50	해외영업1팀
60	기술지원팀
80	해외영업2팀
90	해외영업3팀
30	마케팅팀

--	--

```
SELECT EMP_NAME, DEPT_NAME
FROM EMPLOYEE E, DEPARTMENT D
WHERE D.DEPT_ID = E.DEPT_ID(+);
```

소속 직원이 없는 부서까지 포함하는 의미

EMP_NAME	DEPT_NAME
한선기	해외영업3팀
강중훈	해외영업3팀
최만식	해외영업3팀
정도연	기술지원팀
안석규	기술지원팀
조재형	기술지원팀
정지현	해외영업1팀
김예수	해외영업1팀
나승원	해외영업1팀
김순이	해외영업1팀
성해교	해외영업1팀
전우성	해외영업2팀
엄정하	해외영업2팀
고승우	본사 인사팀
박하일	해외영업1팀
권상후	본사 인사팀
임영애	본사 인사팀
김솔오	회계팀
이중기	회계팀
감우섭	회계팀
	마케팅팀

EMP_NAME	DEPT_ID
한선기	90
강중훈	90
최만식	90
정도연	60
안석규	60
조재형	60
정지현	50
김예수	50
나승원	50
김순이	50
성해교	50
전우성	80
엄정하	80
심하균	10
고승우	50
박하일	10
권상후	10
임영애	10
엄정하	
김솔오	20
이중기	20
감우섭	20
	30

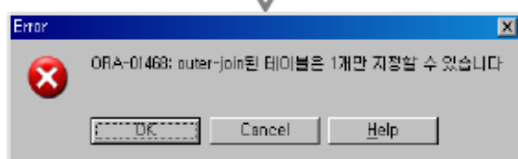
DEPT_ID	DEPT_NAME
20	회계팀
10	본사 인사팀
50	해외영업1팀
60	기술지원팀
80	해외영업2팀
90	해외영업3팀
30	마케팅팀

## JOIN - FULL OUTER JOIN

- 양쪽 테이블을 동시에 OUTER JOIN하는 ANSI 표준 구문
- 오라클 전용 구문은 지원되지 않음

```
SELECT EMP_NAME, DEPT_NAME
FROM EMPLOYEE
FULL JOIN DEPARTMENT USING (DEPT_ID);
```

```
SELECT EMP_NAME, DEPT_NAME
FROM EMPLOYEE E, DEPARTMENT D
WHERE E.DEPT_ID(+) = D.DEPT_ID(+);
```



EMP_NAME	DEPT_NAME
감우섭	회계팀
이중기	회계팀
김솔오	회계팀
임영애	본사 인사팀
권상후	본사 인사팀
고승우	본사 인사팀
박하일	해외영업1팀
성해교	해외영업1팀
김순이	해외영업1팀
나승원	해외영업1팀
김예수	해외영업1팀
정지현	해외영업1팀
조재형	기술지원팀
안석규	기술지원팀
정도연	기술지원팀
엄정하	해외영업2팀
전우성	해외영업2팀
최만식	해외영업3팀
강중훈	해외영업3팀
한선기	해외영업3팀
엄정하	
심하균	
	마케팅팀

## SET Operator - 개념

- 두 개 이상의 쿼리 결과를 하나로 결합시키는 연산자
- SELECT 절에 기술하는 컬럼 개수와 데이터 타입은 모든 쿼리에서 동일해야

유형	설명	도식
UNION	양쪽 쿼리 결과를 모두 포함 (중복 결과는 1번만 표현)	
UNION ALL	양쪽 쿼리 결과를 모두 포함 (중복 결과도 모두 표현)	
INTERSECT	양쪽 쿼리 결과에 모두 포함되는 행만 표현	
MINUS	쿼리1 결과에만 포함되고 쿼리2 결과에는 포함되지 않는 행만 표현	

SELECT EMP\_ID,  
       ROLE\_NAME  
 FROM EMPLOYEE\_ROLE

쿼리1

**UNION**  
 SELECT EMP\_ID,  
       ROLE\_NAME  
 FROM ROLE\_HISTORY;

쿼리2

SELECT EMP\_ID,  
       ROLE\_NAME  
 FROM EMPLOYEE\_ROLE

쿼리1

**INTERSECT**  
 SELECT EMP\_ID,  
       ROLE\_NAME  
 FROM ROLE\_HISTORY;

쿼리2

JOIN → 여러 개의 테이블을 행 단위로 병합하여 데이터를 추출하는 작업

SELECT 컬럼명,...

FROM 테이블명1, 테이블명2

WHERE **조인조건**

AND 행에 대한 조건

SELECT 컬럼명,...

FROM 테이블명1

JOIN 테이블명2 ON **조인조건**

WHERE 행에 대한 조건

EMP			
EMPNO	NAME	SAL	DEPTNO
1	XX	3500	D20
2	YY	6300	D30
3	ZZ	7800	D20
4	AA	2500	D10
5	BB	9300	D30

DEPT			
DEPTNO	DNAME	LOC	XID
D10	SALES	DEL	X2
D20	PURCHASE	MUM	X4
D30	FINANCE	PUNE	X2
D40	ADMIN	KOL	X3

//Display EMPNO, NAME, SALARY, DEPTNO and LOCATION of all employee

```
SELECT EMP.EMPNO, EMP.NAME, EMP.SAL, DEPT.DNAME, DNAME.LOC
FROM EMP, DEPT
WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

//Prefixing column name is mandatory for column 'DEPTNO', since both 'EMP' and 'DEPT' //have a column named 'DEPTNO'.

```
SELECT EMPNO, NAME, SAL, EMP.DEPTNO, DNAME, LOC
FROM EMP, DEPT
WHERE EMP.DEPTNO=DEPT.DEPTNO;
```

EMP				DEPT			
EMPNO	NAME	SAL	DEPTNO	DEPTNO	DNAME	LOC	XID
1	XX	3500	D20	D10	SALES	DEL	X2
2	YY	6300	D30	D20	PURCHASE	MUM	X4
3	ZZ	7800	D20	D30	FINANCE	PUNE	X2
4	AA	2500	D10	D40	ADMIN	KOL	X3
5	BB	9300	D30				

Result of 1st and 2nd Query

EMPNO	NAME	SAL	DEPTNO	DNAME	LOC
1	XX	3500	D20	PURCHASE	MUM
2	YY	6300	D30	FINANCE	PUNE
3	ZZ	7800	D20	PURCHASE	MUM
4	AA	2500	D10	SALES	DEL
5	BB	9300	D30	FINANCE	PUNE

EMP

EMPNO	NAME	SAL	DEPTNO
1	XX	7800	D20
2	YY	3500	D10
3	ZZ	6200	D20
4	AA	2200	D30
5	BB	8300	D10

SAL\_GRADE

GRADE	LOW_SAL	HIGH_SAL
A	700	1200
B	1201	3000
C	3001	5000
D	5001	8000
E	8001	9999

```
//Display EMPNO, NAME, SAL and GRADE of all employee
SELECT EMPNO, NAME, SAL, GRADE
FROM EMP, SAL_GRADE
WHERE SAL BETWEEN LOW_SAL AND HIGH_SAL;
```

EMP

EMPNO	NAME	SAL	DEPTNO
1	XX	7800	D20
2	YY	3500	D10
3	ZZ	6200	D20
4	AA	2200	D30
5	BB	8300	D10

SAL\_GRADE

GRADE	LOW_SAL	HIGH_SAL
A	700	1200
B	1201	3000
C	3001	5000
D	5001	8000
E	8001	9999

EMP

EMPNO	NAME	MGR	SAL
1	XX	2	3500
2	YY	3	6900
3	ZZ		9500
5	AA	3	7600
4	BB	4	3600



```
SELECT W.NAME || ' IS WORKING UNDER ' || M.NAME
FROM EMP W, EMP M
WHERE W.MGR=M.EMPNO;
```

EMPNO	NAME	MGR	SAL
1	XX	2	3500
2	YY	3	6900
3	ZZ	3	9500
4	AA	3	7600
5	BB	4	3600

EMPNO	NAME	MGR	SAL
1	XX	2	3500
2	YY	3	6900
3	ZZ	3	9500
4	AA	3	7600
5	BB	4	3600

## [ ANSI JOIN 구문 ]

```
SELECT ...
FROM   table1
{[INNER] JOIN table2 ON (condition1 [AND condition2 ...]) |
 [INNER] JOIN table2 USING (column1 [, ...]) |
 NATURAL [INNER] JOIN table2 |
 LEFT|RIGHT|FULL [OUTER] JOIN table2 ON (condition1 [AND condition2 ...]) |
 LEFT|RIGHT|FULL [OUTER] JOIN table2 USING (column1 [, ...]) |
 CROSS JOIN table2 }
WHERE ...
GROUP BY column_name | expr
HAVING condition
ORDER BY 기준1 [ ASC | DESC] [, 기준2 [ASC | DESC], ... ];
```