

6. JOIN



- ◆ 본 과정을 마치면 다음을 할 수 있어야 합니다.
 - Natural join(equality join)과 Cross join(non-equality join)을 사용하여 하나 이상의 테이블로부터 데이터를 액세스하는 SELECT 문장을 작성할 수 있어야 합니다.
 - outer join을 사용하여 일반적인 조인 조건을 만족하지 않는 데이터를 출력할 수 있어야 합니다.
 - 자체적으로 테이블을 조인하는 할 수 있어야 합니다.



다중 테이블로부터 데이터 획득

EMP

EMPNO	ENAME	...	DEPTNO
7869	SMITH	...	20
7499	ALLEN	...	30
...			
7782	CLARK	...	10

DEPT

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



EMPNO	DEPTNO	LOC
7369	20	DALLAS
7499	30	CHICAGO
7521	30	CHICAGO
7566	20	DALLAS
7654	30	CHICAGO
7698	30	CHICAGO
...		

14 개의 행이 선택되었습니다.

조인(JOIN)이란?

- ◆ 하나 이상의 테이블로부터 데이터를 질의하기 위해서 조인을 사용합니다.

```
SELECT    table1.column, table2.column  
FROM      table1, table2  
WHERE     table1.column1 = table2.column2;
```

- ◆ WHERE 절에 조인 조건을 작성합니다.
- ◆ 하나 이상의 테이블에 똑같은 열 이름이 있을 때 열 이름 앞에 테이블 이름을 붙입니다.

Cartesian Product

- ◆ Cartesian product 는 다음의 경우에 발생합니다.
 - 조인 조건이 생략된 경우
 - 조인 조건이 잘못된 경우
 - 첫 번째 테이블의 모든 행이 두 번째 테이블의 모든 행과 조인되는 경우
- ◆ Cartesian product를 피하기 위해서는 항상 WHERE 절에 올바른 조인 조건문을 쓰도록 합니다.

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

Cartesian Product 생성

EMP (14 rows)

EMPNO	ENAME	...	DEPTNO
7869	SMITH	...	20
7499	ALLEN	...	30
...			
7782	CLARK	...	10

DEPT (4 rows)

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

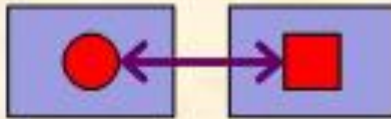


Cartesian product :
 $14 * 4 = 56$ rows

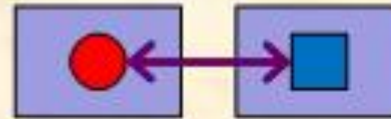
ENAME	DNAME
SMITH	ACCOUNTING
ALLEN	ACCOUNTING
...	
ADAMS	RESEARCH
JAMES	RESEARCH
...	
56 개의 행이 선택되었습니다.	

조인의 유형 (Oracle v6, v7, v8)

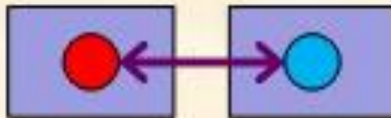
Equi join



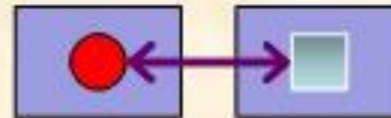
Non-Equi join



Self join

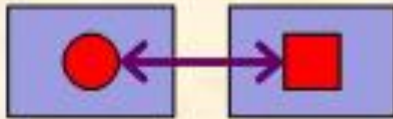


Outer join

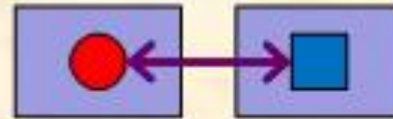


조인의 유형(Oracle v9)

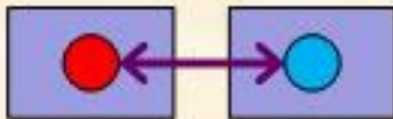
Natural join(Equi join)



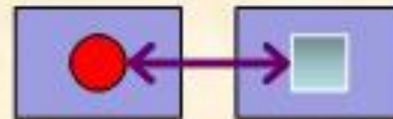
Cross join(Non-Equi join)



Join ~ On절 조인(Self join)



Outer join



Natural Join(Equi Join) 이란?

EMP

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
7782	CLARK	...	10
7566	JONES	...	20
7654	MARTIN	...	30
7499	ALLEN	...	30
7844	TURNER	...	30
7900	JAMES	...	30
7521	WARD	...	20
7902	FORD	...	20
...			
14 개의 행이 선택되었습니다.			

Foreign Key

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	ERSEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	ERSEARCH	DALLAS
...		
14 개의 행이 선택되었습니다.		

Primary Key

8i의 Equi join 으로 레코드 검색

```
SQL> SELECT  emp.empno, emp.ename, emp.deptno,  
2           dept.deptno, dept.loc  
3 FROM      emp, dept  
4 WHERE     emp.deptno = dept.deptno;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7369	SMITH	20	20	DALLAS
7499	ALLEN	30	30	CHICAGO
7521	WARD	30	30	CHICAGO
7566	JONES	20	20	DALLAS

...

14 개의 행이 선택되었습니다.

9i의 Natural Join으로 레코드 검색

```
SQL> SELECT empno, ename, deptno, loc  
2 FROM emp  
3 NATURAL JOIN dept;
```

EMPNO	ENAME	DEPTNO	LOC
7369	SMITH	20	DALLAS
7499	ALLEN	30	CHICAGO
7521	WARD	30	CHICAGO
7566	JONES	20	DALLAS

...

14 개의 행이 선택되었습니다.

AND 연산자를 이용한 추가적인 검색 조건

EMP

EMPNO	ENAME	...	DEPTNO
-----	-----	...	-----
7839	KING	...	10
7698	BLAKE	...	30
7782	CLARK	...	10
7566	JONES	...	20
7654	MARTIN	...	30
7499	ALLEN	...	30
7844	TURNER	...	30
7900	JAMES	...	30
7521	WARD	...	20
7902	FORD	...	20
...			
14 개의 행이 선택되었습니다.			

DEPT

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	ERSEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	ERSEARCH	DALLAS
...		
14 개의 행이 선택되었습니다.		

테이블 별칭(Alias) 사용



테이블 별칭을 사용하여 질의를 간단하게 합니다.



```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
2          dept.deptno, dept.loc  
3 FROM emp, dept  
4 WHERE emp.deptno = dept.deptno;
```

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
2          d.deptno, d.loc  
3 FROM emp, dept d  
4 WHERE emp.deptno = d.deptno;
```

두 개 이상의 테이블 조인

CUSTOMER

NAME	CUSTID
HW SPORTS	100
JK GOLF	200
WOMENS SPORTS	105
INNER SPORTS	301
JUST TENNIS	205
HYUN SPORTS	102
FOOT BALL	208
...	
9 개의 행이 선택...	

ORDER

CUSTID	ORDID
101	611
102	612
104	601
106	602
102	
104	
106	
...	
21 개의 행이 선택...	

ORDID	ITEMID
611	3
612	1
601	1
602	1
605	2
...	
21 개의 행이 선택...	

Cross Join(Non-Equi Join)

EMP

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7566	JONES	2975
7654	MARTIN	1250
7698	BLAKE	2850
7782	CLARK	2450
7788	SCOTT	3000
7839	KING	5000
7844	TURNER	1500
...		
14 개의 행이 선택되었습니다.		

SALGRADE

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

EMP 테이블의 salary는
SALGRADE 테이블에서 어떤
low salary 와 high salary 사
이에 존재하는가에 따라서 급
여 등급이 결정됩니다.

Cross Join(Non-Equi Join)으로 레코드 검색

```
SQL> SELECT e.ename, e.sal, s.grade  
2 FROM emp e  
3 CROSS JOIN salgrade s  
4 WHERE e.sal  
5 BETWEEN s.losal AND s.hisal;
```

ENAME	SAL	GRADE
SMITH	800	1
JAMES	950	1
ADAMS	1100	1
...		

14 개의 행이 선택되었습니다.

Self Join

EMP(WORKER)

EMPNO	ENAME	MGR
7839	KING	
7698	BLAKE	7839
7782	CLARK	7839
7566	JONES	7839
7654	MARTIN	7698
7499	ALLEN	7698

EMP(MANAGER)

EMPNO	ENAME
7839	KING
7839	KING
7839	KING
7698	BLAKE
7698	BLAKE

WORKER 테이블에 있는 MGR은 MANAGER
테이블에 있는 EMPNO와 동일합니다.

Self Join ^ㅅㅇ(8i)

```
SQL> SELECT worker.ename || ' works for ' || manager.ename  
2 FROM emp worker, emp manager  
3 WHERE worker.mgr = manager.empno;
```

```
WORKER.ENAME || 'WORKSFOR' || MANAG  
-----
```

```
FORD works for JONES  
SCOTT works for JONES  
JAMES works for BLAKE  
TURNER works for BLAKE  
...
```

13 개의 행이 선택되었습니다.

Self Join ^ㅅㅇ(9i)

```
SQL> SELECT  worker.ename || ' works for ' || manager.ename  
2  FROM      emp worker JOIN emp manager  
3           ON worker.mgr = manager.empno;
```

```
WORKER. ENAME || 'WORKSFOR' || MANAG  
-----
```

```
FORD works for JONES
```

```
SCOTT works for JONES
```

```
JAMES works for BLAKE
```

```
TURNER works for BLAKE
```

```
...
```

```
13 개의 행이 선택되었습니다.
```

Outer Join

EMP

ENAME	DEPTNO
-----	-----
KING	10
BLAKE	30
CLARK	10
JONES	20
...	

DEPT

DEPTNO	DNAME
-----	-----
10	ACCOUNTING
30	SALES
10	ACCOUNTING
20	RESEARCH
...	

40 OPERATIONS



OPERATIONS 부서에는 종업원이
존재하지 않습니다.

Outer Join(8i)

- ◆ 조인 조건을 만족하지 않는 행들도 보기 위해서 outer join을 사용합니다.
- ◆ Outer join 연산자는 더하기 기호(+)입니다.

```
SELECT    table.column, table.column  
FROM      table1, table2  
WHERE     table1.column(+) = table2.column;
```

```
SELECT    table.column, table.column  
FROM      table1, table2  
WHERE     table1.column = table2.column(+);
```

Outer Join(8i) 사용

```
SQL> SELECT    e.ename, d.deptno, d.dname  
2 FROM      emp e, dept d  
3 WHERE      e.deptno(+) = d.deptno  
4 ORDER BY  e.deptno;
```

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
...		

40 OPERATIONS

15 개의 행이 선택되었습니다.

Outer Join(9i)



오라클 9i 버전에서 제공하는 Outer join은 3 가지 유형이 있습니다.

```
SELECT    table.column, table.column  
FROM      table1, RIGHT OUTER JOIN table2  
ON        table1.column = table2.column;
```

```
SELECT    table.column, table.column  
FROM      table1, LEFT  OUTER JOIN table2  
ON        table1.column = table2.column;
```

```
SELECT    table.column, table.column  
FROM      table1, FULL OUTER JOIN table2  
ON        table1.column = table2.column;
```

Outer Join(9i) 사용

```
SQL> SELECT e.ename, d.deptno, d.dname  
2 FROM emp e  
3 RIGHT OUTER JOIN dept d  
4 ON e.deptno = d.deptno;
```

ENAME	DEPTNO	DNAME
CLARK	10	ACCOUNTING
MILLER	10	ACCOUNTING
...		

40 OPERATIONS

15 개의 행이 선택되었습니다.

Quiz 1

- 다음 중 cartesian product를 발생시키는 문장은 어느 것인가?
- A. `SELECT ename, e.dept_id, d.name FROM s_emp e, s_dept d WHERE d.name = '영역부';`
 - B. `SELECT e.name, e.salary, g.grade FROM s_emp e, salgrade g WHERE e.salary BETWEEN g.losal AND g.hisal;`
 - C. `SELECT e.name, e.id, c.name FROM s_emp e, s_customer c WHERE e.id(+) = c.sales_rep_id;`
 - D. `SELECE w.id, w.name, m.id, m.name FROM s_emp w, s_emp m WHERE w.mansger_id = m.id;`

Quiz 2

□ 다음 테이블 간에 self join을 하는 SELECT 문은?
A. SELECT e.ename, e.sal, d.dname FROM emp e, dept d
WHERE e.deptno = d.deptno;
B. SELECT e.ename, e.job, e.sal, s.grade FROM emp e,
salgrade s WHERE e.sal BETWEEN s.losal AND s.hisal;
C. SELECT e.ename, e.sal, d.dname FROM emp e, dept d
WHERE e.deptno = d.deptno(+);
D. SELECT w.ename, m.ename FROM emp w, emp m WHERE
w.mgr = m.empno;

- A. SELECT e.ename, e.sal, d.dname FROM emp e, dept d
WHERE e.deptno = d.deptno;
- B. SELECT e.ename, e.job, e.sal, s.grade FROM emp e,
salgrade s WHERE e.sal BETWEEN s.losal AND s.hisal;
- C. SELECT e.ename, e.sal, d.dname FROM emp e, dept d
WHERE e.deptno = d.deptno(+);
- D. SELECT w.ename, m.ename FROM emp w, emp m WHERE
w.mgr = m.empno;

Quiz 3

□ 오류가 있는 변호는?

A. SELECT e.empno,

B. emp.ename,

C. e.deptno,

D. d.deptno

E. FROM emp e, dept d WHERE e.deptno = d.deptno ;