

## 조인 (JOIN) - 오라클 조인 구문

조인 (JOIN) - 오라클 조인 구문



## 조인 (JOIN) - 오라클 조인 구문

## 1. 조인(Join)의 개요


## 1.1 여러 테이블에서 데이터 얻기

emp

EMPNO	ENAME	DEPTNO
7839	KING	10
7566	JONES	20
7698	BLAKE	30
...		

dept

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



EMPNO	DEPTNO	DNAME
7934	10	ACCOUNTING
7782	10	ACCOUNTING
7839	10	ACCOUNTING
...		
7698	30	SALES
7499	30	SALES

## 조인 (JOIN) - 오라클 조인 구문

## 1.2 카테시안 프로덕트 (Cartesian Product)

☞ **Cartesian product**는 다음의 경우에 발생

- 조인 조건이 생략된 경우
- 조인 조건이 잘못된 경우
- 첫번째 테이블의 모든 행이 두번째 테이블의 모든 행과 조인되는 경우

☞ **Cartesian product**를 피하기 위해서는 항상 **WHERE** 절에 올바른 조인 조건 문 사용



## 조인 (JOIN) - 오라클 조인 구문

## 1.2 카테시안 프로덕트 (계속)

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

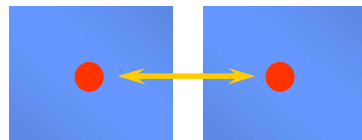
EMPNO	ENAME	DNAME
7839	KING	ACCOUNTING
7566	JONES	ACCOUNTING
7698	BLAKE	ACCOUNTING
7782	CLARK	ACCOUNTING
7902	FORD	ACCOUNTING
7788	SCOTT	ACCOUNTING
7499	ALLEN	ACCOUNTING
7521	WARD	ACCOUNTING
7654	MARTIN	ACCOUNTING
...		
7839	KING	OPERATIONS
7566	JONES	OPERATIONS
7876	ADAMS	OPERATIONS

56 개의 행이 선택되었습니다.

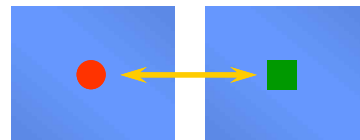
## 조인 (JOIN) - 오라클 조인 구문

## 1.3 조인(Join) 유형

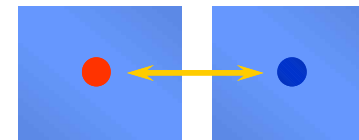
## ☐ 오라클 전용 조인



Equi Join



Non-equi Join



Outer Join



Self Join

## ☐ SQL-1999 표준 조인

- 교차 조인 (Cross Join)
- 자연 조인 (Natural Join)
- USING 절
- 전체 또는 한쪽 포괄 조인
- 포괄 조인에 대한 임의의 조인 조건

## 조인 (JOIN) - 오라클 조인 구문

## 2. 테이블 조인 (오라클 구문)

### 2.1 개요

- ☐ WHERE 절에 조인 조건 작성
- ☐ 하나 이상의 테이블에 똑같은 열 이름이 있을 때 열 이름 앞에 테이블 이름 붙임
- ☐  $n$ 개의 테이블을 함께 조인하려면, 최소  $(n-1)$  개의 조인 조건문이 필요

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



## 조인 (JOIN) - 오라클 조인 구문

## 2.2 Equi Join

EMPNO	ENAME	DEPTNO
7839	KING	10
7566	JONES	20
7698	BLAKE	30
7782	CLARK	10
7902	FORD	20
7788	SCOTT	20
7499	ALLEN	30
7521	WARD	30
7654	MARTIN	30
7844	TURNER	30
7900	JAMES	30
7934	MILLER	10
7369	SMITH	20
7876	ADAMS	20

EMP

Foreign key

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

Primary key

DEPT



## 조인 (JOIN) - 오라클 조인 구문

## 2.2 Equi Join (계속)

```

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

```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7566	JONES	20	20	DALLAS
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7902	FORD	20	20	DALLAS
7788	SCOTT	20	20	DALLAS
7499	ALLEN	30	30	CHICAGO
7521	WARD	30	30	CHICAGO
7654	MARTIN	30	30	CHICAGO
7844	TURNER	30	30	CHICAGO
7900	JAMES	30	30	CHICAGO
7934	MILLER	10	10	NEW YORK
7369	SMITH	20	20	DALLAS
7876	ADAMS	20	20	DALLAS



## 조인 (JOIN) - 오라클 조인 구문

## 2.2 Equi Join (계속)

- ☐ 테이블 접두어를 사용하여 여러 테이블에 있는 열 이름을 자세히 지정
- ☐ 테이블 접두어를 사용하여 성능 개선
- ☐ 열 별칭을 사용하여 다른 테이블에 있는 동일한 이름의 열 구분
- ☐ 테이블 (**Table**) 별칭을 사용하여 질의의 단순화 및 성능 개선
  - 테이블 별칭은 최대 30자까지 가능
  - **FROM** 절에서 테이블 별칭을 사용하는 경우 전체 **SELECT** 문에 적용
  - 테이블 별칭은 현재 **SELECT** 문에 대해서만 유효

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

## 조인 (JOIN) - 오라클 조인 구문

## 2.3 Non-Equi Join

EMPNO	ENAME	SAL
7839	KING	5000
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450
7902	FORD	3000
7788	SCOTT	3000
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500
7900	JAMES	950
7934	MILLER	1300
7369	SMITH	800
7876	ADAMS	1100

EMP

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

SALGRADE

← EMP 테이블의 급여는 SALGRADE 테이블의 최저 급여와 최고 급여 사이에 해당

## 조인 (JOIN) - 오라클 조인 구문

## 2.3 Non-Equi Join (계속)

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

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

## 조인 (JOIN) - 오라클 조인 구문

## 2.4 Outer Join

EMPNO	ENAME	DEPTNO
7839	KING	10
7566	JONES	20
7698	BLAKE	30
7782	CLARK	10
7902	FORD	20
7788	SCOTT	20
7499	ALLEN	30
7521	WARD	30
7654	MARTIN	30
7844	TURNER	30
7900	JAMES	30
7934	MILLER	10
7369	SMITH	20
7876	ADAMS	20

EMP

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

DEPT

↑  
OPERATIONS 부서 (40)에는 사원이  
없음

## 조인 (JOIN) - 오라클 조인 구문

## 2.4 Outer Join (계속)

- ❏ 조인 조건을 만족하지 않는 행들도 보기 위해서 **Outer Join** 사용
- ❏ **Outer Join** 연산자는 더하기 기호 (+)
- ❏ **WHERE** 절 조건의 어느 쪽에도 올 수 있지만, 양쪽 동시에는 사용할 수 없음

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

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

## 조인 (JOIN) - 오라클 조인 구문

## 2.4 Outer Join (계속)

- Outer Join 연산자는 정보가 누락된 쪽에 표시
- Outer Join을 포함하는 조건은 IN 연산자를 사용할 수 없으며, OR 연산자를 사용해 다른 조건에 링크할 수 없음

```
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
KING	10	ACCOUNTING
MILLER	10	ACCOUNTING
...		
	40	OPERATIONS



## 조인 (JOIN) - 오라클 조인 구문

## 2.5 Self Join

EMPNO	ENAME	MGR
7839	KING	
7566	JONES	7839
7698	BLAKE	7839
7782	CLARK	7839
7902	FORD	7566
7788	SCOTT	7566
7499	ALLEN	7698
7521	WARD	7698
...		

EMP (WORKER)

EMPNO	ENAME
7839	KING
7566	JONES
7698	BLAKE
7782	CLARK
7902	FORD
7788	SCOTT
7499	ALLEN
7521	WARD
...	

EMP (MANAGER)

WORKER 테이블의 MGR은  
MANAGER 테이블의 EMPNO와 동일



## 조인 (JOIN) - 오라클 조인 구문

## 2.5 Self Join (계속)

```
SQL> SELECT '사원번호 ' || worker.empno || '인 '
2         || worker.ename || '의 상급자는 사원번호 '
3         || manager.empno || '인 '
4         || manager.ename || '이다.'
5         AS "Employee AND Manager"
6 FROM     emp worker, emp manager
7 WHERE    worker.mgr = manager.empno;
```



## 조인 (JOIN) - 오라클 조인 구문

## 2.5 Self Join (계속)

## Employee AND Manager

-----

사원번호 7566인 JONES의	상급자는 사원번호 7839인 KING이다.
사원번호 7698인 BLAKE의	상급자는 사원번호 7839인 KING이다.
사원번호 7782인 CLARK의	상급자는 사원번호 7839인 KING이다.
사원번호 7902인 FORD의	상급자는 사원번호 7566인 JONES이다.
사원번호 7788인 SCOTT의	상급자는 사원번호 7566인 JONES이다.
사원번호 7499인 ALLEN의	상급자는 사원번호 7698인 BLAKE이다.
사원번호 7521인 WARD의	상급자는 사원번호 7698인 BLAKE이다.
사원번호 7654인 MARTIN의	상급자는 사원번호 7698인 BLAKE이다.
사원번호 7844인 TURNER의	상급자는 사원번호 7698인 BLAKE이다.
사원번호 7900인 JAMES의	상급자는 사원번호 7698인 BLAKE이다.
사원번호 7934인 MILLER의	상급자는 사원번호 7782인 CLARK이다.
사원번호 7369인 SMITH의	상급자는 사원번호 7902인 FORD이다.
사원번호 7876인 ADAMS의	상급자는 사원번호 7788인 SCOTT이다.

## 조인 (JOIN) - SQL-1999 조인 구문

# 조인 (JOIN)-SQL-1999 조인 구문



## 조인 (JOIN) - SQL-1999 조인 구문

## 1. SQL-1999 구문을 사용한 테이블 조인

- ☐ SQL-1999 구문을 사용하여 오라클 구문과 동일한 결과를 얻을 수 있음

```
SELECT    table1.column, table2.column
FROM      table1
[CROSS JOIN table2] |
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2 ON(table1.column_name = table2.column_name)] |
[LEFT | RIGHT | FULL OUTER JOIN table2
                ON(table1.column_name = table2.column_name)];
```



## 조인 (JOIN) - SQL-1999 조인 구문

## 2. 교차 조인 (CROSS JOIN)

- ❑ CROSS JOIN 절은 두 테이블 상호간의 조합을 생성
- ❑ 두 테이블 사이의 카테시안 프러덕트 (Cartesian Product) 와 동일

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

ENAME	DNAME
KING	ACCOUNTING
JONES	ACCOUNTING
BLAKE	ACCOUNTING
CLARK	ACCOUNTING
FORD	ACCOUNTING
SCOTT	ACCOUNTING
ALLEN	ACCOUNTING
WARD	ACCOUNTING
MARTIN	ACCOUNTING
TURNER	ACCOUNTING
...	
ADAMS	OPERATIONS

56 개의 행이 선택되었습니다.

## 조인 (JOIN) - SQL-1999 조인 구문

**3. 자연 조인 (NATURAL JOIN)**

- ☐ NATURAL JOIN 절은 두 테이블에서 동일한 이름을 가진 모든 열을 기준으로 조인
- ☐ 두 테이블의 일치하는 모든 열에서 같은 값을 가진 행을 선택
- ☐ 동일한 이름을 가진 열의 데이터 유형이 서로 다를 경우 오류 반환

```
SQL> SELECT empno, ename, deptno, dname
2 FROM emp NATURAL JOIN dept
3 WHERE deptno = 20;
```

EMPNO	ENAME	DEPTNO	DNAME
7566	JONES	20	RESEARCH
7902	FORD	20	RESEARCH
7788	SCOTT	20	RESEARCH
7369	SMITH	20	RESEARCH
7876	ADAMS	20	RESEARCH



## 조인 (JOIN) - SQL-1999 조인 구문

## 4. USING 절을 포함하는 조인

- ☐ 여러 열이 같은 이름을 가지지만 데이터 유형이 일치하지 않을 경우
  - NATURAL JOIN 절을 수정하여 USING 절을 포함시키면 Equi Join에 사용될 열을 지정할 수 있음
- ☐ USING 절을 사용하면 둘 이상의 열이 일치할 때 한 열만 일치시킬 수 있음
- ☐ 참조되는 열에 테이블 이름이나 별칭을 사용해서는 안됨
- ☐ NATURAL JOIN 및 USING 절은 서로 배타적

```
SQL> SELECT e.empno, e.ename, d.dname
       2 FROM emp e JOIN dept d USING (deptno);
```

EMPNO	ENAME	DNAME
7839	KING	ACCOUNTING
7566	JONES	RESEARCH
7698	BLAKE	SALES
7782	CLARK	ACCOUNTING
7902	FORD	RESEARCH
7788	SCOTT	RESEARCH

...



## 조인 (JOIN) - SQL-1999 조인 구문

## 5. ON 절을 포함하는 조인

- ☐ 자연 조인 (NATURAL JOIN)의 조인 조건은 기본적으로 같은 이름을 가진 모든 열의 **Equi Join**
- ☐ 임의의 조건을 지정하거나 조인할 열을 지정하려면 **ON 절**을 사용
- ☐ 조인 조건이 다른 검색조건과 분리
- ☐ **ON 절**을 사용하면 코드가 이해하기 쉬워짐

```
SQL> SELECT e.empno, e.ename, e.deptno, d.deptno, d.loc
2 FROM emp e JOIN dept d ON e.deptno = d.deptno;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7566	JONES	20	20	DALLAS
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7902	FORD	20	20	DALLAS
7788	SCOTT	20	20	DALLAS
7499	ALLEN	30	30	CHICAGO
7521	WARD	30	30	CHICAGO

...

## 조인 (JOIN) - SQL-1999 조인 구문

## 5. ON 절을 포함하는 조인 (계속)

## ☐ ON 절로 3-way 조인

- 세 테이블 조인
- 두번째 조인 조건은 세 테이블 모두의 열을 참조할 수 있음

```
SQL> SELECT empno, grade, dname
2   FROM   emp e
3   JOIN   dept d ON d.deptno = e.deptno
4   JOIN   salgrade s ON e.sal BETWEEN losal AND hisal;
```

EMPNO	GRADE	DNAME
7900	1	SALES
7369	1	RESEARCH
7876	1	RESEARCH
7521	2	SALES
7654	2	SALES
7934	2	ACCOUNTING
7499	3	SALES
7844	3	SALES
7566	4	RESEARCH

...

## 6. INNER JOIN과 OUTER JOIN

### 6.1 개요

- ☐ SQL-1999 에서 두 테이블을 조인해서 일치하는 열만 반환하는 조인이 **내부 조인 (INNER JOIN)**
- ☐ 두 테이블을 조인해서 내부 조인의 결과와 함께 일치하지 않은 왼쪽 (또는 오른쪽) 테이블의 행을 반환하는 조인이 **왼쪽 (또는 오른쪽) 포괄 조인 (OUTER JOIN)**
- ☐ 두 테이블을 조인해서 내부 조인의 결과와 함께 왼쪽 및 오른쪽 조인의 결과를 반환하는 조인이 **전체 포괄 조인 (FULL OUTER JOIN)**
- ☐ SQL-1999와 오라클 구문 비교

오라클	SQL-1999
Equi Join	Natural   Inner Join
Outer Join	Left   Right Outer Join
Self Join	Join ON
Non-Equi Join	Join USING
Cartesian Prodict	Cross Join

## 조인 (JOIN) - SQL-1999 조인 구문

## 6.2 LEFT OUTER JOIN

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

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

## 조인 (JOIN) - SQL-1999 조인 구문

## 6.3 RIGHT OUTER JOIN

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

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

## 조인 (JOIN) - SQL-1999 조인 구문

## 6.4 FULL OUTER JOIN

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

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



## 조인 (JOIN) - SQL-1999 조인 구문

## 6.5 추가 조건

```
SQL> SELECT e.empno, e.ename, e.deptno, d.deptno, d.loc
2 FROM emp e JOIN dept d ON (e.deptno = d.deptno)
3 AND e.mgr = 7839;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7566	JONES	20	20	DALLAS
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK

