

2. JOIN

1) JOIN

JOIN이란 FROM절에서 두개 이상의 테이블을 연결하여 사용자에게 필요한 데이터를 검색, 출력하는 방법이다.

보통 둘 이상의 행들의 공통된 값을 사용하여 조인하므로 조인할 행들에 같은 칼럼이 존재해야 한다.

JOIN의 종류는 크게 4가지로 구분할 수 있다.

- EQUI JOIN(등가조인) - 동일한 칼럼을 기준으로 조인
- Non-EQUI JOIN(비등가조인) - 동일한 칼럼 없이 다른 조건을 사용하여 조인
- OUTER JOIN - 조인조건에 만족하지 않는 행도 나타내는 조인
- SELF JOIN - 하나의 같은 테이블로 한 테이블 내에서 조인
-

● SQL문 로직 순서

"SQL문 실행순서 (오라클 SQL 기준)"	
5. SELECT	{*, column(s) [alias],}
1. FROM	table(s) [alias]
Ex: SELECT 1 FROM DUAL;	
2. [WHERE	condition(s)]
3. [GROUP BY	column(s)
4. [HAVING	condition(s)]
6. [ORDER BY	column(s) [alias][DESC], ...];

1. FROM : SQL은 FROM절에 명시된 테이블들을 먼저 가져온다.
2. WHERE : 가져온 테이블에서 주어진 조건에 맞는 데이터들을 추출한다..
3. GROUP BY : 조건에 맞는 데이터가 추출 후, GROUP BY절의 칼럼명 대로 정렬한다.
4. HAVING : 정렬된 데이터 그룹 중, 주어진 조건에 맞는 그룹들을 추출한다.
5. SELECT : 최종적으로 추출된 데이터들을 SELECT절의 명시된 칼럼명 및 조건에 맞는 데이터를 추출한다.
6. ORDER BY : 출력 전 ORDER BY절의 조건대로 데이터들을 정렬해 출력한다.

SELECT절에서 선언한 별칭을 사용할 수 있다..

(1) EQUI JOIN(등가조인)

EQUI JOIN은 가장 많이 사용하는 조인 방법으로 조인 대상이 되는 두 테이블에서 공통적으로 존재하는 컬럼의 값이 일치되는 행을 연결하여 결과를 생성하는 조인 방법이다. 일치하는 같은 컬럼을 결합하여 출력해야 하기 때문에 SQL문에서 테이블 별로 별칭을 주어 같은 컬럼을 구분하고 명시해줘야 오류를 발생하지 않고 원하는 결과를 얻을수 있다.

Ex) 기본 문법

```
SELECT 테이블1.컬럼명, 테이블2.컬럼명, ...
FROM 테이블1, 테이블2
WHERE 테이블1.컬럼명1 = 테이블2.컬럼명2;
```

Ex) 예제

두 개의 테이블(EMP, DEPT)의 모든 정보를 한 행에 출력, 같은 DEPTNO를 이용하여 조인한다.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369 SMITH	CLERK	7902	17-DEC-80	800	(null)	20
2	7499 ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
3	7521 WARD	SALESMAN	7698	22-FEB-81	1250	500	30
4	7566 JONES	MANAGER	7839	02-APR-81	2975	(null)	20
5	7654 MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
6	7698 BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30
7	7782 CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10
8	7788 SCOTT	ANALYST	7566	09-DEC-82	3000	(null)	20
9	7839 KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10
10	7844 TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
11	7876 ADAMS	CLERK	7788	12-JAN-83	1100	(null)	20
12	7900 JAMES	CLERK	7698	03-DEC-81	950	(null)	30
13	7902 FORD	ANALYST	7566	03-DEC-81	3000	(null)	20
14	7934 MILLER	CLERK	7782	23-JAN-82	1300	(null)	10

DEPTNO	DNAME	LOC
1	10 ACCOUNTING	NEW YORK
2	20 RESEARCH	DALLAS
3	30 SALES	CHICAGO
4	40 OPERATIONS	BOSTON

-- 1. EQUI JOIN (동등조인 : 공통필드엔 DEPTNO값이 일치되는 조건 JOIN)

```
SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, EMP.DEPTNO, DNAME, LOC
FROM EMP, DEPT
WHERE EMP.DEPTNO=DEPT.DEPTNO; -- 두개의 테이블에 같은 컬럼인 DEPTNO를 이용, 조인한다.
-- 한 행의 정보에 같은 컬럼의 다른 테이블 정보를 붙이는 느낌이다.
```

-- FROM절에 테이블의 별칭을 정의하고 WHERE절에 별칭으로 컬럼 접근 가능.

```
SELECT EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, E.DEPTNO, DNAME, LOC
FROM EMP E, DEPT D -- EMP테이블에 E, DEPT테이블에 D
WHERE E.DEPTNO=D.DEPTNO;
```

결과)

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	DNAME	LOC
1	7369 SMITH	CLERK	7902	17-DEC-80	800	(null)	20	RESEARCH	DALLAS
2	7499 ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	SALES	CHICAGO
3	7521 WARD	SALESMAN	7698	22-FEB-81	1250	500	30	SALES	CHICAGO
4	7566 JONES	MANAGER	7839	02-APR-81	2975	(null)	20	RESEARCH	DALLAS
5	7654 MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	SALES	CHICAGO
6	7698 BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30	SALES	CHICAGO
7	7782 CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10	ACCOUNTING	NEW YORK
8	7788 SCOTT	ANALYST	7566	09-DEC-82	3000	(null)	20	RESEARCH	DALLAS
9	7839 KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10	ACCOUNTING	NEW YORK
10	7844 TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	SALES	CHICAGO
11	7876 ADAMS	CLERK	7788	12-JAN-83	1100	(null)	20	RESEARCH	DALLAS
12	7900 JAMES	CLERK	7698	03-DEC-81	950	(null)	30	SALES	CHICAGO
13	7902 FORD	ANALYST	7566	03-DEC-81	3000	(null)	20	RESEARCH	DALLAS
14	7934 MILLER	CLERK	7782	23-JAN-82	1300	(null)	10	ACCOUNTING	NEW YORK

(2) Non-EQUI JOIN(비등가조인)

Non-EQUI JOIN 동일한 칼럼 없이 다른 조건을 사용하여 조인하는 방법인데 조건이 한 테이블의 어떠한 칼럼도 조인 할 테이블의 칼럼에 직접적으로 일치하지 않는 경우에 사용 한다.

Equal 연산자(=) 이외의 연산자들(>, >=, <, <=, <>, BETWEEN ... AND)을 이용하여 조인한다.

조인할 칼럼의 특정 범위 내에 데이터 값을 조회할 때 주로 사용된다.

Ex) 기본 문법

```
SELECT 테이블1.칼럼명, 테이블2.칼럼명, ...
FROM   테이블1, 테이블2
WHERE  테이블1.칼럼명1 BETWEEN 테이블2.칼럼명1 AND 테이블2.칼럼명2;
```

Ex) 예제

두 개의 테이블(EMP, SALGRADE)의 모든 정보를 한 행에 출력, 해당 행의 SAL의 등급을 조회한다..

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369 SMITH	CLERK	7902	17-DEC-80	800	(null)	20
2	7499 ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
3	7521 WARD	SALESMAN	7698	22-FEB-81	1250	500	30
4	7566 JONES	MANAGER	7839	02-APR-81	2975	(null)	20
5	7654 MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
6	7698 BLAKE	MANAGER	7839	01-MAY-81	2850	(null)	30
7	7782 CLARK	MANAGER	7839	09-JUN-81	2450	(null)	10
8	7788 SCOTT	ANALYST	7566	09-DEC-82	3000	(null)	20
9	7839 KING	PRESIDENT	(null)	17-NOV-81	5000	(null)	10
10	7844 TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
11	7876 ADAMS	CLERK	7788	12-JAN-83	1100	(null)	20
12	7900 JAMES	CLERK	7698	03-DEC-81	950	(null)	30
13	7902 FORD	ANALYST	7566	03-DEC-81	3000	(null)	20
14	7934 MILLER	CLERK	7782	23-JAN-82	1300	(null)	10

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

-- 2. NON-EQUI JOIN

-- ex. 모든 사원의 사번, 이름, job, 상사사번, 급여, 급여등급(1등급, 2등급, ..)

```
SELECT EMPNO, ENAME, JOB, MGR, SAL, GRADE||'등급' GRADE
FROM EMP, SALGRADE
WHERE SAL BETWEEN LOSAL AND HISAL; -- 비교 칼럼사이의 해당 데이터(GRADE)출력
-- 낮은 값이 먼저 와야한다.
```

결과)

	EMPNO	ENAME	JOB	MGR	SAL	GRADE '등급'
1	7369	SMITH	CLERK	7902	800	1등급
2	7876	ADAMS	CLERK	7788	1100	1등급
3	7900	JAMES	CLERK	7698	950	1등급
4	7521	WARD	SALESMAN	7698	1250	2등급
5	7654	MARTIN	SALESMAN	7698	1250	2등급
6	7934	MILLER	CLERK	7782	1300	2등급
7	7499	ALLEN	SALESMAN	7698	1600	3등급
8	7844	TURNER	SALESMAN	7698	1500	3등급
9	7566	JONES	MANAGER	7839	2975	4등급
10	7698	BLAKE	MANAGER	7839	2850	4등급
11	7782	CLARK	MANAGER	7839	2450	4등급
12	7788	SCOTT	ANALYST	7566	3000	4등급
13	7902	FORD	ANALYST	7566	3000	4등급
14	7839	KING	PRESIDENT	(null)	5000	5등급

(3) OUTER JOIN

조인 조건에 만족하지 못하였더라도 해당 데이터를 나타내고 싶을 때에 사용하는 조인이다.

Outer Join은 NULL 값이기에 배제된 행을 결과에 포함시킬 수 있으며 WHERE절에 조인시킬 값이 없는 칼럼쪽에 (+) 기호를 붙여 사용한다. 쉽게 말해 조건에 만족하지 않는 null인 값을 출력하는 구문이다.

EX) 기본문법

```
SELECT 테이블1.칼럼명, 테이블2.칼럼명, ...  
FROM   테이블1, 테이블2  
WHERE  테이블1.칼럼명1(+) = 테이블2.칼럼명2; -- 해당 값이 없는 칼럼에 (+)
```

EX)예제

-- 4. OUTER JOIN : 조인시 조건에 만족하지 않은 행까지 출력한다.

```
SELECT W.EMPNO, W.ENAME, W.MGR, M.ENAME  
FROM   EMP W, EMP M  
WHERE  W.MGR=M.EMPNO(+); -- M.EMPNO가 없는 W 테이블의 값 출력  
-- KING에게는 MGR데이터가 없음
```

```
SELECT ENAME, D.DEPTNO, DNAME  
FROM   EMP E, DEPT D  
WHERE  E.DEPTNO(+) = D.DEPTNO; -- 40번 부서 존재  
-- E.DEPTNO에는 40번 부서 없음
```

결과)

	EMPNO	ENAME	MGR	ENAME_1
1	7902	FORD	7566	JONES
2	7788	SCOTT	7566	JONES
3	7900	JAMES	7698	BLAKE
4	7844	TURNER	7698	BLAKE
5	7654	MARTIN	7698	BLAKE
6	7521	WARD	7698	BLAKE
7	7499	ALLEN	7698	BLAKE
8	7934	MILLER	7782	CLARK
9	7876	ADAMS	7788	SCOTT
10	7782	CLARK	7839	KING
11	7698	BLAKE	7839	KING
12	7566	JONES	7839	KING
13	7369	SMITH	7902	FORD
14	7839	KING	(null)	(null)

	ENAME	DEPTNO	DNAME
1	SMITH	20	RESEARCH
2	ALLEN	30	SALES
3	WARD	30	SALES
4	JONES	20	RESEARCH
5	MARTIN	30	SALES
6	BLAKE	30	SALES
7	CLARK	10	ACCOUNTING
8	SCOTT	20	RESEARCH
9	KING	10	ACCOUNTING
10	TURNER	30	SALES
11	ADAMS	20	RESEARCH
12	JAMES	30	SALES
13	FORD	20	RESEARCH
14	MILLER	10	ACCOUNTING
15	(null)	40	OPERATIONS

(4) SELF JOIN

SELF JOIN은 EQUI JOIN과 같으나 하나의 테이블에 조인이 일어나는 것이 다르게 같은 테이블에 대해 두 개의 Alias(별칭)로 구분해 작성 함으로서 FROM절에 다른 두 개의 테이블처럼 사용하는 문법이다. 같은 테이블 두 개를 사용하기 때문에 반드시 테이블 별칭으로 칼럼을 구분해 주어야한다.

EX)기본문법

```
SELECT "별칭1".칼럼명, "별칭1".칼럼명, ...
FROM   테이블1 "별칭1", 테이블1 "별칭2"
WHERE  "별칭1".칼럼명1 = "별칭2".칼럼명2;
```

EX)예제

```
-- ex. 모든 사원의 사번, 이름, 상사사번, 상사이름
SELECT W.EMPNO "사번", W.ENAME "이름", W.MGR "상사사번", M.ENAME "상사이름"
FROM   EMP "W", EMP "M"
WHERE  W.MGR=M.EMPNO;
```

결과)

	사번	이름	상사사번	상사이름
1	7902	FORD	7566	JONES
2	7788	SCOTT	7566	JONES
3	7900	JAMES	7698	BLAKE
4	7844	TURNER	7698	BLAKE
5	7654	MARTIN	7698	BLAKE
6	7521	WARD	7698	BLAKE
7	7499	ALLEN	7698	BLAKE
8	7934	MILLER	7782	CLARK
9	7876	ADAMS	7788	SCOTT
10	7782	CLARK	7839	KING
11	7698	BLAKE	7839	KING
12	7566	JONES	7839	KING
13	7369	SMITH	7902	FORD

위의 예제를 그림으로 표현하면 아래와 같다.

즉 동일한 테이블에서 해당하는 값에 해당하는 다른 칼럼의 값을 붙인다고 생각하면 된다.

