

[MSSQL] 조인 방법 쉽게 정리 (INNER JOIN, OUTER JOIN)

🕒 젠트 🏢 데이터베이스 / MSSQL 🕒 2021. 2. 14.



SQL Server에서는 조인을 할 때에는 ANSI JOIN 사용을 권장한다.

아래에는 **조인**(INNER JOIN), **아우터 조인**(LEFT OUTER JOIN, RIGHT OUTER JOIN), **크로스 조인**(CROSS JOIN) 사용법을 정리하였다.

INNER JOIN과 **LEFT OUTER JOIN**은 꼭 사용법을 숙지해야 하며, 나머지 부분은 개념만 이해하고 넘어가면 된다.

조인 (INNER JOIN) : 기준 테이블과 조인 테이블 모두 데이터가 존재해야 조회됨

아우터 조인 (OUTER JOIN) : 기준 테이블에만 데이터가 존재하면 조회됨

조인 (INNER JOIN)

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
```



```
INNER JOIN dept AS b
ON a.deptno = b.deptno
```

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
INNER JOIN dept AS b
ON a.deptno = b.deptno
```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname
1	7698	BLAKE	MANAGER	7839	30	SALES
2	7782	CLARK	MANAGER	7839	10	ACCOUNTING
3	7566	JONES	MANAGER	7839	20	RESEARCH

EMP (직원) 테이블

	empno	ename	job	mgr	deptno
1	7839	KING	PRESIDENT	NULL	90
2	7698	BLAKE	MANAGER	7839	30
3	7782	CLARK	MANAGER	7839	10
4	7566	JONES	MANAGER	7839	20

DEPT (부서) 테이블

	deptno	dname
1	10	ACCOUNTING
2	20	RESEARCH
3	30	SALES
4	40	OPERATIONS



조인은 기준 테이블(emp), 조인 테이블(dept)에 조인 컬럼(deptno)에 해당하는 값이 **모두 존재하는 경우**에만 데이터가 조회된다.

emp 테이블 "KING"의 deptno "90"은 dept 테이블에 존재하지 않기 때문에 조회되지 않는다.

아우터 조인 (LEFT OUTER JOIN)

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
```



```

, b.dname
FROM emp AS a
LEFT OUTER JOIN dept AS b
ON a.deptno = b.deptno

```

```

SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
LEFT OUTER JOIN dept AS b
ON a.deptno = b.deptno

```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname
1	7839	KING	PRESIDENT	NULL	90	NULL
2	7698	BLAKE	MANAGER	7839	30	SALES
3	7782	CLARK	MANAGER	7839	10	ACCOUNTING
4	7566	JONES	MANAGER	7839	20	RESEARCH

EMP (직원) 테이블

	empno	ename	job	mgr	deptno
1	7839	KING	PRESIDENT	NULL	90
2	7698	BLAKE	MANAGER	7839	30
3	7782	CLARK	MANAGER	7839	10
4	7566	JONES	MANAGER	7839	20

DEPT (부서) 테이블

	deptno	dname
1	10	ACCOUNTING
2	20	RESEARCH
3	30	SALES
4	40	OPERATIONS

아우터 조인에서 LEFT, RIGHT는 기준 테이블을 지정하는 것이며, 위의 쿼리에서 LEFT OUTER JOIN의 기준 테이블은 emp 테이블이다.

아우터 조인의 경우 조인 테이블(dept)에 데이터가 없어도 **기준 테이블(emp)의 모든 데이터가 조회**되고 조인 테이블(dept)에 데이터가 존재할 경우 해당 데이터를 참조할 수 있다.

emp 테이블 "KING"의 deptno "90"은 dept 테이블에 존재하지 않아도 조회가 되지만, dname은 참조할 수 없기 때문에 NULL로 표시된다.



아우터 조인 (RIGHT OUTER JOIN)

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
RIGHT OUTER JOIN dept AS b
ON a.deptno = b.deptno
```

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
RIGHT OUTER JOIN dept AS b
ON a.deptno = b.deptno
```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname
1	7782	CLARK	MANAGER	7839	10	ACCOUNTING
2	7566	JONES	MANAGER	7839	20	RESEARCH
3	7698	BLAKE	MANAGER	7839	30	SALES
4	NULL	NULL	NULL	NULL	NULL	OPERATIONS

EMP (직원) 테이블

	empno	ename	job	mgr	deptno
1	7839	KING	PRESIDENT	NULL	90
2	7698	BLAKE	MANAGER	7839	30
3	7782	CLARK	MANAGER	7839	10
4	7566	JONES	MANAGER	7839	20

DEPT (부서) 테이블

	deptno	dname
1	10	ACCOUNTING
2	20	RESEARCH
3	30	SALES
4	40	OPERATIONS

LEFT OUTER JOIN과 개념을 동일하지만 RIGHT에 해당하는 dept 테이블이 기준 테이블이 된다. LEFT OUTER JOIN의 결과와 다르게 dept 테이블의 모든 데이터가 조회되고 조인 테이블(emp)의 테이블에 데이터가 있을 경우 해당 값을 표시한다.



대부분 LEFT OUTER JOIN을 많이 사용하지만 상황에 따라서 RIGHT OUTER JOIN을 사용할 수 있으니 개념을 꼭 이해하고 있어야 한다.

아우터 조인을 사용하는 이유는 기준 테이블의 데이터를 누락 없이 모두 조회하고 참조 테이블의 값이 있을 경우 해당 값을 사용하기 위해서이다.

아우터 조인 (FULL OUTER JOIN)

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
FULL OUTER JOIN dept AS b
  ON a.deptno = b.deptno
```

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
FULL OUTER JOIN dept AS b
  ON a.deptno = b.deptno
```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname
1	7839	KING	PRESIDENT	NULL	90	NULL
2	7698	BLAKE	MANAGER	7839	30	SALES
3	7782	CLARK	MANAGER	7839	10	ACCOUNTING
4	7566	JONES	MANAGER	7839	20	RESEARCH
5	NULL	NULL	NULL	NULL	NULL	OPERATIONS

FULL OUTER JOIN은 두개의 테이블을 합쳐서 조회한다고 생각하면 된다. JOIN이 될 경우 해당값을 표시하고 JOIN이 안되면 NULL로 표시하되 두개의 테이블 모든 데이터가 조회된다.



자주 사용하지 않기 때문에 개념만 이해하면 된다.

크로스 조인 (CROSS JOIN)

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
CROSS JOIN dept AS b
```

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
CROSS JOIN dept AS b
```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname
1	7839	KING	PRESIDENT	NULL	90	ACCOUNTING
2	7839	KING	PRESIDENT	NULL	90	RESEARCH
3	7839	KING	PRESIDENT	NULL	90	SALES
4	7839	KING	PRESIDENT	NULL	90	OPERATIONS
5	7698	BLAKE	MANAGER	7839	30	ACCOUNTING
6	7698	BLAKE	MANAGER	7839	30	RESEARCH
7	7698	BLAKE	MANAGER	7839	30	SALES
8	7698	BLAKE	MANAGER	7839	30	OPERATIONS
9	7782	CLARK	MANAGER	7839	10	ACCOUNTING
10	7782	CLARK	MANAGER	7839	10	RESEARCH
11	7782	CLARK	MANAGER	7839	10	SALES
12	7782	CLARK	MANAGER	7839	10	OPERATIONS
13	7566	JONES	MANAGER	7839	20	ACCOUNTING
14	7566	JONES	MANAGER	7839	20	RESEARCH
15	7566	JONES	MANAGER	7839	20	SALES
16	7566	JONES	MANAGER	7839	20	OPERATIONS



크로스 조인은 기준 테이블(emp)의 행을 조인 테이블(dept) 행만큼 증가를 시킨다.
아주 가끔씩 사용할 일이 있으니 개념만 이해하면 된다.

활용 예제

```
SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
      , c.lname
FROM emp AS a
LEFT OUTER JOIN dept AS b
  ON a.deptno = b.deptno
INNER JOIN loc AS c
  ON b.locno = c.locno
WHERE job = 'MANAGER'
```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname	lname
1	7566	JONES	MANAGER	7839	20	RESEARCH	TEXAS
2	7698	BLAKE	MANAGER	7839	30	SALES	TEXAS
3	7782	CLARK	MANAGER	7839	10	ACCOUNTING	WASHINGTON

emp 테이블과 dept 테이블 아우터 조인

dept 테이블과 loc 테이블 조인



```

SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
      , c.ename
FROM emp AS a
LEFT OUTER JOIN dept AS b
  ON a.deptno = b.deptno
INNER JOIN emp AS c
  ON a.mgr = c.empno

```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname	ename
1	7698	BLAKE	MANAGER	7839	30	SALES	KING
2	7782	CLARK	MANAGER	7839	10	ACCOUNTING	KING
3	7566	JONES	MANAGER	7839	20	RESEARCH	KING

emp 테이블과 dept 테이블 아우터 조인

emp 테이블과 emp 테이블 조인

조인 구문 작성 위치와는 상관없이 ON 절에 해당 테이블의 조인 조건이 있으면 조인 된다.

```

SELECT a.empno
      , a.ename
      , a.job
      , a.mgr
      , a.deptno
      , b.dname
FROM emp AS a
INNER JOIN dept AS b
  ON (a.deptno = b.deptno AND b.deptno IN (10, 20, 30))

```

97 %

결과 메시지

	empno	ename	job	mgr	deptno	dname
1	7698	BLAKE	MANAGER	7839	30	SALES
2	7782	CLARK	MANAGER	7839	10	ACCOUNTING
3	7566	JONES	MANAGER	7839	20	RESEARCH

조인 조건이 여러개인 경우 ON 절에 AND를 사용하여 작성하면 된다.



아래의 예제 파일에 테이블뷰가 포함되어 있으니 다운 받아서 참고 하세요.

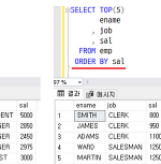
예제쿼리(테이블뷰포함).txt

2

구독하기

'데이터베이스 / MSSQL' 관련 글

더 보기

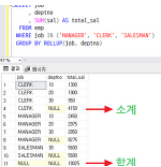


```
SELECT TOP(5)
  empno,
  job,
  sal
FROM emp
ORDER BY sal
```

empno	job	sal
1	CLERK	800
2	CLERK	900
3	CLERK	1100
4	SALESMAN	1200
5	SALESMAN	1300

[MSSQL] ROWNUM 사용법 정리 (조회 순번, 조회 개수)

🕒 2021.02.17



```
SELECT empno, job, sal,
  SUM(sal) OVER (PARTITION BY job ORDER BY empno) AS total_sal
FROM emp
WHERE job IN ('MANAGER', 'CLERK', 'SALESMAN')
GROUP BY ROLLUP(job, empno)
```

empno	job	sal	total_sal
1	MANAGER	1300	1300
2	MANAGER	1200	2500
3	MANAGER	1100	3600
4	MANAGER	900	4500
5	MANAGER	800	5300
6	CLERK	1100	1100
7	CLERK	900	2000
8	CLERK	800	2800
9	SALESMAN	1200	1200
10	SALESMAN	1300	2500
11	SALESMAN	1400	3900

[MSSQL] ROLLUP 사용법 (소계, 합계)

🕒 2021.02.16

[MSSQL] IIF 함수 사용법 (CASE, DECODE)

🕒 2021.02.10




```

ELECT ename
      , salary
      , ISNULL(salary, 0) AS sal
FROM emp

```

ename	salary	sal
JONES	3000	3000
SCOTT	NULL	0

© 2021.02.09



2021.07.07 11:39

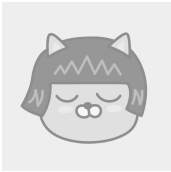
← 답글 ⚙ 수정

2021.07.07 12:16

신고



수정



indev

2021.09.09 08:20

고맙습니다! 이해가 잘 됩니다

답글 수정



젠트

2021.09.09 11:56

신고

감사합니다~^^

수정

	<input type="text" value="Name"/>		<input type="password" value="Password"/>
--	-----------------------------------	--	---

비밀 글

▶ 등록

카테고리

프로그래밍	(43)
— 자바스크립트	(19)
— 씨샵(C#)	(14)
— 아두이노	(10)
데이터베이스	(1)

— 오라클	(127)
— MSSQL	(30)
IT정보	(48)
— IT인물	(3)
— IT영화	(8)
— IT세미나	(37)
유용한팁	(111)
— 컴퓨터(PC)	(27)
— 맥북(Mac)	(9)
— 아이폰	(62)
— 애플워치	(4)
— 구글홈	(6)
— 오쿨러스퀘스트2	(3)
일상생활	(51)
— 소소한지름	(15)
— 금융&재테크	(4)
— 여행&맛집	(13)
— 기타	(19)
기타	(13)



[Oracle] 오라클 정렬 순서 지정 방법 (...)

```
YEAR(GETDATE()) AS  
MONTH(GETDATE()) AS  
DAY(GETDATE()) AS
```

[MSSQL] 년, 월, 일 추출 방법 (YEAR, M...

메시지	
MONTH	DAY
9	7



[MSSQL] 중복 제거 방법 (DISTINCT)



[아이폰] 타이머 진동 만드는 방법 (무...

```
'SQL', 'Server', '  
2021, 12, 31, 23, 59
```

[MSSQL] 문자열 합치기 (CONCAT, +)

str2
20211231235959

```
~ver 2019', 5, 6  
→ 시작위치  
2019', 3)  
왼쪽에서 자를 길이  
2019', 4)  
→ 오른쪽에서 자를 길이
```

[MSSQL] 문자열 자르기 (SUBSTRING, LEF...

