

조인문(JOIN)





조인문(JOIN)

JOIN이란?

한 개 이상의 테이블에서 데이터를 조회하기 위해 사용하는 것으로, 수행 결과는 하나의 RESULT SET으로 나오게 된다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       DEPT_CODE  
FROM EMPLOYEE;
```

```
SELECT DEPT_ID  
       DEPT_TITLE  
FROM DEPARTMENT;
```

EMP_ID	EMP_NAME	DEPT_CODE
1 200	선동일	D9
2 201	송종기	D9
3 202	노웅철	D9
4 203	송은희	D6
5 204	유재식	D6
6 205	정중하	D6
7 206	박나라	D5
8 207	하미유	D5
9 208	김해솔	D5
10 209	심봉선	D5
11 210	윤은혜	D5
12 211	전형돈	D8
13 212	장프위	D8
14 213	하동운	(null)
15 214	방명수	D1
16 215	대복훈	D5
17 216	차태연	D1
18 217	전지연	D1
19 218	미오리	(null)
20 219	임시환	D2
21 220	이중석	D2
22 221	유하진	D2
23 222	이태림	D8

DEPT_ID	DEPT_TITLE
1 D1	인사관리부
2 D2	회계관리부
3 D3	마케팅부
4 D4	국내영업부
5 D5	해외영업1부
6 D6	해외영업2부
7 D7	해외영업3부
8 D8	기술지원부
9 D9	총무부





조인문(JOIN)

JOIN

1. 오라클 전용 구문

- FROM 절에 ',' 로 구분하여 합치게 될 TABLE명을 기술하고, WHERE절에 합치기에 사용할 컬럼명을 명시한다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       DEPT_CODE,  
       DEPT_TITLE  
FROM EMPLOYEE, DEPARTMENT  
WHERE DEPT_CODE = DEPT_ID;
```

** 연결에 사용할 두 컬럼명이 다른 경우
그냥 사용하면 된다.

EMP_ID	EMP_NAME	DEPT_CODE	DEPT_TITLE
1 200	선동일	D9	총무부
2 201	송중기	D9	총무부
3 202	노웅철	D9	총무부
4 203	송은희	D6	해외영업2부
5 204	유재식	D6	해외영업2부
6 205	정중하	D6	해외영업2부
7 206	박나라	D5	해외영업1부
8 207	하미유	D5	해외영업1부
9 208	김해솔	D5	해외영업1부
10 209	심봉선	D5	해외영업1부
11 210	윤은해	D5	해외영업1부
12 211	전형돈	D8	기술지원부
13 212	장프위	D8	기술지원부
14 214	방명수	D1	인사관리부
15 215	대복훈	D5	해외영업1부
16 216	차태연	D1	인사관리부
17 217	전지연	D1	인사관리부
18 219	임시환	D2	회계관리부
19 220	이중석	D2	회계관리부
20 221	유하진	D2	회계관리부
21 222	이태림	D8	기술지원부





조인문(JOIN)

JOIN

1. 오라클 전용 구문

- FROM 절에 ',' 로 구분하여 합치게 될 TABLE명을 기술하고, WHERE절에 합치기에 사용할 컬럼명을 명시한다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       EMPLOYEE.JOB_CODE,  
       JOB_NAME  
FROM EMPLOYEE, JOB  
WHERE EMPLOYEE.JOB_CODE = JOB.JOB_CODE;
```

- ** 연결에 사용할 두 컬럼명이 같은 경우
테이블명.컬럼명 으로 작성해서 구분해 주어야 한다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       E.JOB_CODE,  
       JOB_NAME  
FROM EMPLOYEE E, JOB J  
WHERE E.JOB_CODE = J.JOB_CODE;
```

- ** FROM절에 사용한 테이블 별칭을 이용해도 된다.

	EMP_ID	EMP_NAME	JOB_CODE	JOB_NAME
1	200	선동일	J1	대표
2	201	송중기	J2	부사장
3	202	노웅철	J2	부사장
4	203	송은희	J4	차장
5	204	유재식	J3	부장
6	205	정중하	J3	부장
7	206	박나라	J7	사원
8	207	하미유	J5	과장
9	208	김해솔	J5	과장
10	209	심봉선	J3	부장
11	210	윤은혜	J7	사원
12	211	전형돈	J6	대리
13	212	장프위	J6	대리
14	213	하동운	J6	대리
15	214	방명수	J7	사원
16	215	대복훈	J5	과장
17	216	차태연	J6	대리
18	217	전지연	J6	대리
19	218	미오리	J7	사원
20	219	임시환	J4	차장
21	220	이중석	J4	차장
22	221	유하진	J4	차장
23	222	이태림	J6	대리





조인문(JOIN)

JOIN

2. ANSI 표준 구문

- 연결에 사용하려는 컬럼명이 같은 경우 USING()을 사용하고, 다른 경우 ON()을 사용한다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       JOB_CODE,  
       JOB_NAME  
FROM EMPLOYEE  
JOIN JOB USING(JOB_CODE);
```

	EMP_ID	EMP_NAME	JOB_CODE	JOB_NAME
1	200	선동일	J1	대표
2	201	송종기	J2	부사장
3	202	노용철	J2	부사장
4	203	송은희	J4	차장
5	204	유재식	J3	부장
6	205	정중하	J3	부장
7	206	박나라	J7	사원
8	207	하미유	J5	과장
9	208	김해술	J5	과장
10	209	심봉선	J3	부장
11	210	윤은해	J7	사원
12	211	전형돈	J6	대리
13	212	장프위	J6	대리
14	213	하동운	J6	대리
15	214	방명수	J7	사원
16	215	대복훈	J5	과장
17	216	차태연	J6	대리
18	217	전지연	J6	대리
19	218	미오리	J7	사원
20	219	임시환	J4	차장
21	220	이종석	J4	차장
22	221	유하진	J4	차장
23	222	이태림	J6	대리





조인문(JOIN)

JOIN

2. ANSI 표준 구문

- 연결에 사용하려는 컬럼명이 같은 경우 USING()을 사용하고, 다른 경우 ON()을 사용한다.

```
SELECT EMP_ID, EMP_NAME, DEPT_CODE, DEPT_TITLE  
FROM EMPLOYEE  
JOIN DEPARTMENT ON(DEPT_CODE = DEPT_ID);
```

EMP_ID	EMP_NAME	DEPT_CODE	DEPT_TITLE
1 200	선동일	D9	총무부
2 201	송중기	D9	총무부
3 202	노용철	D9	총무부
4 203	송은희	D6	해외영업2부
5 204	유재석	D6	해외영업2부
6 205	정중하	D6	해외영업2부
7 206	박나라	D5	해외영업1부
8 207	하미유	D5	해외영업1부
9 208	김해솔	D5	해외영업1부
10 209	심봉선	D5	해외영업1부
11 210	윤은해	D5	해외영업1부
12 211	전형돈	D8	기술지원부
13 212	장쯔위	D8	기술지원부
14 214	방명수	D1	인사관리부
15 215	대북촌	D5	해외영업1부
16 216	차태연	D1	인사관리부
17 217	전지연	D1	인사관리부
18 219	임시환	D2	회계관리부
19 220	이중석	D2	회계관리부
20 221	유하진	D2	회계관리부
21 222	이태림	D8	기술지원부





조인문(JOIN)

INNER JOIN과 OUTER JOIN

두 개 이상의 테이블을 조인할 때, 일치하는 값이 없는 행은 조인에서 제외된다. 이것을 INNER JOIN이라고 하며, 명시적으로 사용하지 않을 시에는 기본적으로 INNER JOIN이다.

하지만 일치하지 않은 값 또한 JOIN에 포함시킬 수도 있다. 이것을 OUTER JOIN이라고 하며, 반드시 OUTER JOIN임을 명시해야 한다.





조인문(JOIN)

INNER JOIN과 OUTER JOIN

```
SELECT EMP_NAME, DEPT_TITLE  
FROM EMPLOYEE  
JOIN DEPARTMENT ON (DEPT_CODE = DEPT_ID);
```

EMP_NAME	DEPT_TITLE
1 선동일	총무부
2 송종기	총무부
3 노웅철	총무부
4 송은희	해외영업2부
5 유재식	해외영업2부
6 정중하	해외영업2부
7 박나라	해외영업1부
8 하미유	해외영업1부
9 김해솔	해외영업1부
10 심봉선	해외영업1부
11 윤은해	해외영업1부
12 전형돈	기술지원부
13 장프위	기술지원부
14 방명수	인사관리부
15 대북혼	해외영업1부
16 차태연	인사관리부
17 전지연	인사관리부
18 임시환	회계관리부
19 이중석	회계관리부
20 유하진	회계관리부
21 이태림	기술지원부

```
SELECT EMP_NAME, DEPT_TITLE  
FROM EMPLOYEE  
LEFT JOIN DEPARTMENT ON (DEPT_CODE = DEPT_ID);
```

EMP_NAME	DEPT_TITLE
1 전지연	인사관리부
2 차태연	인사관리부
3 방명수	인사관리부
4 유하진	회계관리부
5 이중석	회계관리부
6 임시환	회계관리부
7 대북혼	해외영업1부
8 윤은해	해외영업1부
9 심봉선	해외영업1부
10 김해솔	해외영업1부
11 하미유	해외영업1부
12 박나라	해외영업1부
13 정중하	해외영업2부
14 유재식	해외영업2부
15 송은희	해외영업2부
16 이태림	기술지원부
17 장프위	기술지원부
18 전형돈	기술지원부
19 노웅철	총무부
20 송종기	총무부
21 선동일	총무부
22 이오리	(null)
23 하동운	(null)





조인문(JOIN)

OUTER JOIN의 종류

1. LEFT OUTER JOIN

- 합치기에 사용한 두 테이블 중 왼쪽 편에 기술된 테이블의 컬럼 수를 기준으로 JOIN을 할 때 사용한다.

2. RIGHT OUTER JOIN

- 합치기에 사용한 두 테이블 중 오른쪽 편에 기술된 테이블의 컬럼 수를 기준으로 JOIN을 할 때 사용한다.

3. FULL OUTER JOIN

- 합치기에 사용한 두 테이블이 가진 모든 행을 결과에 포함 시킬 경우 사용한다.





조인문(JOIN)

OUTER JOIN

```
SELECT EMP_NAME, DEPT_TITLE  
FROM EMPLOYEE  
RIGHT JOIN DEPARTMENT ON (DEPT_CODE = DEPT_ID);
```

EMP_NAME	DEPT_TITLE
1 선동일	총무부
2 송종기	총무부
3 노웅철	총무부
4 송은희	해외영업2부
5 유재식	해외영업2부
6 정중하	해외영업2부
7 박나라	해외영업1부
8 하미유	해외영업1부
9 김해술	해외영업1부
10 심봉선	해외영업1부
11 윤은해	해외영업1부
12 전형돈	기술지원부
13 장프위	기술지원부
14 방명수	인사관리부
15 대북훈	해외영업1부
16 차태연	인사관리부
17 전지연	인사관리부
18 임시환	회계관리부
19 이종석	회계관리부
20 유하진	회계관리부
21 이태림	기술지원부
22 (null)	해외영업3부
23 (null)	마케팅부
24 (null)	국내영업부

```
SELECT EMP_NAME, DEPT_TITLE  
FROM EMPLOYEE  
FULL JOIN DEPARTMENT ON (DEPT_CODE = DEPT_ID);
```

EMP_NAME	DEPT_TITLE
1 선동일	총무부
2 송종기	총무부
3 노웅철	총무부
4 송은희	해외영업2부
5 유재식	해외영업2부
6 정중하	해외영업2부
7 박나라	해외영업1부
8 하미유	해외영업1부
9 김해술	해외영업1부
10 심봉선	해외영업1부
11 윤은해	해외영업1부
12 전형돈	기술지원부
13 장프위	기술지원부
14 하종문	(null)
15 방명수	인사관리부
16 대북훈	해외영업1부
17 차태연	인사관리부
18 전지연	인사관리부
19 미오리	(null)
20 임시환	회계관리부
21 이종석	회계관리부
22 유하진	회계관리부
23 이태림	기술지원부
24 (null)	해외영업3부
25 (null)	마케팅부
26 (null)	국내영업부





조인문(JOIN)

오라클 전용 OUTER JOIN

```
SELECT EMP_NAME, DEPT_TITLE  
FROM EMPLOYEE, DEPARTMENT  
WHERE DEPT_CODE(+) = DEPT_ID;
```

```
SELECT EMP_NAME, DEPT_TITLE  
FROM EMPLOYEE, DEPARTMENT  
WHERE DEPT_CODE(+) = DEPT_ID(+);
```

** 에러남, (+)는 한 쪽에만 사용할 수 있음

EMP_NAME	DEPT_TITLE
1 선동일	총무부
2 송종기	총무부
3 노웅철	총무부
4 송은희	해외영업2부
5 유재식	해외영업2부
6 정중하	해외영업2부
7 박나라	해외영업1부
8 하이유	해외영업1부
9 김해술	해외영업1부
10 심봉선	해외영업1부
11 윤은혜	해외영업1부
12 전형돈	기술지원부
13 장프위	기술지원부
14 방명수	인사관리부
15 대북훈	해외영업1부
16 차태연	인사관리부
17 전지연	인사관리부
18 임시환	회계관리부
19 이종석	회계관리부
20 유하진	회계관리부
21 이태림	기술지원부
22 (null)	해외영업3부
23 (null)	마케팅부
24 (null)	국내영업부





조인문(JOIN)

CROSS JOIN

카테이션곱(Cartesian product)라고도 한다. 조인 되는 테이블의 각 행들이 모두 매핑된 데이터가 검색 되는 조인 방법이다.

```
SELECT EMP_NAME, DEPT_TITLE  
FROM EMPLOYEE  
CROSS JOIN DEPARTMENT;
```

SQL | 인출된 모든 행: 207(0.016초)

EMP_N...	DEPT_TITLE
1 선동일	인사관리부
2 송종기	인사관리부
3 노웅철	인사관리부
4 송은희	인사관리부
5 유재식	인사관리부
6 정중하	인사관리부
7 박나라	인사관리부
8 하이유	인사관리부
9 김해술	인사관리부
10 심봉선	인사관리부
11 윤은혜	인사관리부
...	
205 비봉석	총무부
206 유하진	총무부
207 이태림	총무부





조인문(JOIN)

NON_EQU JOIN

지정한 컬럼 값이 일치하는 경우가 아닌, **값의 범위에 포함되는 행들을** 연결하는 방식이다.

```
SELECT EMP_ID,  
       EMP_NAME,  
       SALARY,  
       SAL_LEVEL  
FROM EMPLOYEE  
JOIN SAL_GRADE ON (SALARY BETWEEN MIN_SAL AND MAX_SAL)  
ORDER BY 3 DESC;
```

SQL 인출된 모든 행: 23(0초)			
EMP_ID	EMP_NAME	근속년수	YEAR_LEVEL
1 200	선동일	27	A
2 221	유하진	23	A
3 207	하이유	23	A
4 203	송은희	21	A
5 222	이태림	20	A
6 219	임시환	18	A
7 205	정중하	18	A
22 215	대복훈	0	D
23 218	이오리	0	D





조인문(JOIN)

SELF JOIN

조인은 두 개 이상의 서로 다른 테이블을 연결하기도 하지만, 같은 테이블을 조인 하는 경우도 있다. 이러한 경우 자기 자신과 조인을 맺는 것이라 하여 SELF JOIN이라 한다.

```
SELECT E.EMP_ID,  
       E.EMP_NAME 사원이름,  
       E.DEPT_CODE  
       E.MANAGER_ID,  
       M.EMP_NAME 관리자이름  
FROM EMPLOYEE E, EMPLOYEE M  
WHERE E.MANAGER_ID = M.EMP_ID;
```

EMP_ID	사원이름	DEPT_CODE	MANAGER_ID	관리자이름
1 214	방명수	D1	200	선동일
2 211	전형돈	D8	200	선동일
3 207	하미유	D5	200	선동일
4 204	유재식	D6	200	선동일
5 201	송중기	D9	200	선동일
6 202	노웅철	D9	201	송중기
7 205	정중하	D6	204	유재식
8 203	송은희	D6	204	유재식
9 210	윤은해	D5	207	하미유
10 209	심봉선	D5	207	하미유
11 208	김해술	D5	207	하미유
12 206	박나라	D5	207	하미유
13 212	장프위	D8	211	전형돈
14 217	전지연	D1	214	방명수
15 216	차태연	D1	214	방명수





조인문(JOIN)

다중 JOIN

```
SELECT EMP_ID,  
       EMP_NAME,  
       DEPT_CODE,  
       DEPT_TITLE,  
       LOCAL_NAME  
FROM EMPLOYEE  
JOIN DEPARTMENT ON(DEPT_CODE = DEPT_ID)  
JOIN LOCATION ON(LOCATION_ID = LOCAL_CODE);
```

```
SELECT EMP_ID,  
       EMP_NAME,  
       DEPT_CODE,  
       DEPT_TITLE,  
       LOCAL_NAME  
FROM EMPLOYEE  
JOIN LOCATION ON(LOCATION_ID = LOCAL_CODE)  
JOIN DEPARTMENT ON(DEPT_CODE = DEPT_ID);
```

EMP_ID	EMP_NAME	DEPT_CODE	DEPT_TITLE	LOCAL_NAME
1 200	선동일	D9	총무부	ASIA1
2 201	송종기	D9	총무부	ASIA1
3 202	노용철	D9	총무부	ASIA1
4 203	송은희	D6	해외영업2부	ASIA3
5 204	유재식	D6	해외영업2부	ASIA3
6 205	정중하	D6	해외영업2부	ASIA3
7 206	박나라	D5	해외영업1부	ASIA2
8 207	하미유	D5	해외영업1부	ASIA2
9 208	김해솔	D5	해외영업1부	ASIA2
10 209	심봉선	D5	해외영업1부	ASIA2
11 210	윤은해	D5	해외영업1부	ASIA2
12 211	전형돈	D8	기술지원부	EU
13 212	장프위	D8	기술지원부	EU
14 214	방명수	D1	인사관리부	ASIA1
15 215	대북훈	D5	해외영업1부	ASIA2
16 216	차태연	D1	인사관리부	ASIA1
17 217	전지연	D1	인사관리부	ASIA1
18 219	임시환	D2	회계관리부	ASIA1
19 220	이중석	D2	회계관리부	ASIA1
20 221	유하진	D2	회계관리부	ASIA1
21 222	이태림	D8	기술지원부	EU

** 에러 발생

다중 조인의 경우 조인의 순서가 중요하다.

