



조인

# 조인

- 조인

- ✓ Join
- ✓ 2개 이상의 테이블을 결합하여 필요한 데이터를 찾는 것을 의미함
- ✓ 2개 이상의 테이블을 하나의 테이블처럼 묶어서 사용하는 것을 의미함
- ✓ 테이블의 결합 방식에 따라서 다양한 조인 형태를 가짐
- ✓ PK-FK 관계를 맺고 있는 테이블의 경우 조인을 이용해서 처리할 수 있음

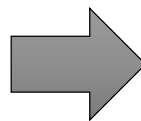
- 조인 종류

- ① 내부 조인(Inner Join)
- ② 외부 조인(Outer Join)
- ③ 크로스 조인(Cross Join)
- ④ 셀프 조인(self Join)
- ⑤ 자연 조인(Natural Join)



실무에서 조인은  
굉장히  
자주 사용됩니다!


조인


조인 결과

# 조인 형식

- ANSI 구문

```
SELECT 테이블1.칼럼, 테이블2.칼럼, ...  
FROM 테이블1 JOIN 테이블2  
ON 조인_조건
```

- 오라클 구문

```
SELECT 테이블1.칼럼, 테이블2.칼럼, ...  
FROM 테이블1, 테이블2  
WHERE 조인_조건
```

# 크로스 조인

- 크로스 조인

- ✓ Cross Join
- ✓ 두 개 테이블에서 연결 가능한 모든 행(Row)과 열(Column)을 결합
- ✓ Cartesian Product(카티전 곱), 즉 곱집합의 개념임
- ✓ A 테이블에 10개 행, B 테이블에 5개 행이 있다고 가정하면 두 테이블의 크로스 조인 결과 행은 50개임( $10 \times 5$ )
- ✓ 시뮬레이션 진행을 위해서 대용량의 테스트 데이터를 생성하는 경우 활용 가능
- ✓ 조인 조건을 생략하거나 잘못 작성하면 CROSS JOIN으로 처리됨

- 사용 방법

- ① SELECT 칼럼 FROM 테이블1 CROSS JOIN 테이블2;
- ② SELECT 칼럼 FROM 테이블1, 테이블2;

# CROSS JOIN

- 조인 조건 없이 두 테이블의 모든 행을 1:1로 모두 결합

DEPARTMENT\_T

DEPT_NO	DEPT_NAME	LOCATION
1	영업부	대구
2	인사부	서울

EMPLOYEE\_T

EMP_NO	NAME	DEPART	POSITION	GENDER	HIRE_DATE	SALARY
1001	구창민	1	과장	M	95-05-01	5000000
1002	김민서	1	사원	M	17-09-01	2500000
1003	이은영	2	부장	F	90-09-01	5500000
1004	한성일	2	과장	M	93-04-01	5000000

ANSI 문법

```
SELECT E.EMP_NO, E.NAME, D.DEPT_NAME  
FROM DEPARTMENT_T D CROSS JOIN EMPLOYEE_T E;
```

Oracle 문법

```
SELECT E.EMP_NO, E.NAME, D.DEPT_NAME  
FROM DEPARTMENT_T D, EMPLOYEE_T E;
```

조인 결과

EMP_NO	NAME	DEPT_NAME
1001	구창민	영업부
1002	김민서	영업부
1003	이은영	영업부
1004	한성일	영업부
1001	구창민	인사부
1002	김민서	인사부
1003	이은영	인사부
1004	한성일	인사부

# 내부 조인

- 내부 조인
  - ✓ Inner Join
  - ✓ 조인 조건으로 사용된 두 칼럼에 공통적으로 존재하는 값만 결합됨
- 2개 테이블 사용 방법
  - ① SELECT 칼럼  
FROM 테이블1 INNER JOIN 테이블2  
ON 테이블1.칼럼 = 테이블2.칼럼;
  - ② SELECT 칼럼  
FROM 테이블1, 테이블2  
WHERE 테이블1.칼럼 = 테이블2.칼럼;
- 3개 테이블 사용 방법
  - ① SELECT 칼럼  
FROM 테이블1 INNER JOIN 테이블2  
ON 테이블1.칼럼 = 테이블2.칼럼 INNER JOIN 테이블3  
ON 테이블2.칼럼 = 테이블3.칼럼;
  - ② SELECT 칼럼  
FROM 테이블1, 테이블2, 테이블3  
WHERE 테이블1.칼럼 = 테이블2.칼럼  
AND 테이블2.칼럼 = 테이블3.칼럼;

# INNER JOIN

- 두 칼럼에 서로 일치하는 값만 포함

DEPARTMENT\_T

DEPT_NO	DEPT_NAME	LOCATION
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

EMPLOYEE\_T

EMP_NO	NAME	DEPART	POSITION	GENDER	HIRE_DATE	SALARY
1001	구창민	1	과장	M	95-05-01	5000000
1002	김민서	1	사원	M	17-09-01	2500000
1003	이은영	2	부장	F	90-09-01	5500000
1004	한성일	2	과장	M	93-04-01	5000000

ANSI 문법

```
SELECT D.DEPT_NO, D.DEPT_NAME, E.EMP_NO, E.NAME
FROM DEPARTMENT_T D INNER JOIN EMPLOYEE_T E
ON D.DEPT_NO = E.DEPART;
```

Oracle 문법

```
SELECT D.DEPT_NO, D.DEPT_NAME, E.EMP_NO, E.NAME
FROM DEPARTMENT_T D, EMPLOYEE_T E
WHERE D.DEPT_NO = E.DEPART;
```

조인 결과

DEPT_NO	DEPT_NAME	EMP_NO	NAME
1	영업부	1001	구창민
1	영업부	1002	김민서
2	인사부	1003	이은영
2	인사부	1004	한성일

# 외부 조인

- 외부 조인

- ✓ Outer Join
- ✓ 한 테이블의 내용은 모두 포함되고, 다른 한 테이블의 내용은 일치하는 정보만 포함됨
- ✓ 왼쪽 테이블의 내용을 모두 포함하는 경우 '왼쪽 외부 조인(Left Outer Join)'이라고 함
- ✓ 오른쪽 테이블의 내용을 모두 포함하는 경우 '오른쪽 외부 조인(Right Outer Join)'이라고 함

- 왼쪽 조인 사용 방법

- ① SELECT 칼럼  
FROM 테이블1 LEFT OUTER JOIN 테이블2  
ON 테이블1.칼럼 = 테이블2.칼럼;
- ② SELECT 칼럼  
FROM 테이블1, 테이블2  
WHERE 테이블1.칼럼 = 테이블2.칼럼(+);

- 오른쪽 조인 사용 방법

- ① SELECT 칼럼  
FROM 테이블1 RIGHT OUTER JOIN 테이블2  
ON 테이블1.칼럼 = 테이블2.칼럼;
- ② SELECT 칼럼  
FROM 테이블1, 테이블2  
WHERE 테이블1.칼럼(+) = 테이블2.칼럼;



LEFT OUTER JOIN 대신  
LEFT JOIN이라고 해도 됩니다!  
RIGHT도 마찬가지!



# LEFT OUTER JOIN

- 왼쪽 테이블은 전체 포함하고, 오른쪽 테이블은 일치하는 값만 포함

DEPARTMENT\_T

DEPT_NO	DEPT_NAME	LOCATION
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

EMPLOYEE\_T

EMP_NO	NAME	DEPART	POSITION	GENDER	HIRE_DATE	SALARY
1001	구창민	1	과장	M	95-05-01	5000000
1002	김민서	1	사원	M	17-09-01	2500000
1003	이은영	2	부장	F	90-09-01	5500000
1004	한성일	2	과장	M	93-04-01	5000000

ANSI 문법

```
SELECT D.DEPT_NO, D.DEPT_NAME, E.EMP_NO, E.NAME  
FROM DEPARTMENT_T D LEFT OUTER JOIN EMPLOYEE_T E  
ON D.DEPT_NO = E.DEPART;
```

Oracle 문법

```
SELECT D.DEPT_NO, D.DEPT_NAME, E.EMP_NO, E.NAME  
FROM DEPARTMENT_T D, EMPLOYEE_T E  
WHERE D.DEPT_NO = E.DEPART(+);
```

조인 결과

DEPT_NO	DEPT_NAME	EMP_NO	NAME
1	영업부	1001	구창민
1	영업부	1002	김민서
2	인사부	1003	이은영
2	인사부	1004	한성일
3	총무부	NULL	NULL
4	기획부	NULL	NULL

# RIGHT OUTER JOIN

- 왼쪽 테이블은 일치하는 값만 포함하고, 오른쪽 테이블은 모두 포함

DEPARTMENT\_T

DEPT_NO	DEPT_NAME	LOCATION
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

EMPLOYEE\_T

EMP_NO	NAME	DEPART	POSITION	GENDER	HIRE_DATE	SALARY
1001	구창민	1	과장	M	95-05-01	5000000
1002	김민서	1	사원	M	17-09-01	2500000
1003	이은영	2	부장	F	90-09-01	5500000
1004	한성일	2	과장	M	93-04-01	5000000
1005	김성실	5	대리	F	98-12-01	3500000

ANSI 문법

```
SELECT D.DEPT_NO, D.DEPT_NAME, E.EMP_NO, E.NAME  
FROM DEPARTMENT_T D RIGHT OUTER JOIN EMPLOYEE_T E  
ON D.DEPT_NO = E.DEPART;
```

Oracle 문법

```
SELECT D.DEPT_NO, D.DEPT_NAME, E.EMP_NO, E.NAME  
FROM DEPARTMENT_T D, EMPLOYEE_T E  
WHERE D.DEPT_NO(+) = E.DEPART;
```

조인 결과

DEPT_NO	DEPT_NAME	EMP_NO	NAME
1	영업부	1001	구창민
1	영업부	1002	김민서
2	인사부	1003	이은영
2	인사부	1004	한성일
5	NULL	1005	김성실

# FULL OUTER JOIN

- 왼쪽 외부 조인과 오른쪽 외부 조인을 모두 진행

DEPARTMENT\_T

DEPT_NO	DEPT_NAME	LOCATION
1	영업부	대구
2	인사부	서울
3	총무부	대구
4	기획부	서울

EMPLOYEE\_T

EMP_NO	NAME	DEPART	POSITION	GENDER	HIRE_DATE	SALARY
1001	구창민	1	과장	M	95-05-01	5000000
1002	김민서	1	사원	M	17-09-01	2500000
1003	이은영	2	부장	F	90-09-01	5500000
1004	한성일	2	과장	M	93-04-01	5000000
1005	김성실	5	대리	F	98-12-01	3500000

```
SELECT D.DEPT_NAME, E.NAME  
FROM DEPARTMENT_T D FULL OUTER JOIN EMPLOYEE_T E  
ON D.DEPT_NO = E.DEPART;
```

조인 결과

DEPT_NAME	NAME
영업부	구창민
영업부	김민서
인사부	이은영
인사부	한성일
NULL	김성실
총무부	NULL
기획부	NULL

# 셀프 조인

- 셀프 조인

- ✓ Self Join
- ✓ 하나의 테이블내에 있는 칼럼끼리 연결하는 조인 방식
- ✓ 조인 대상 테이블이 하나뿐이라는 것을 제외하면 일반적인 조인과 다를 바 없음
- ✓ 하나의 테이블에 각각 다른 별명(Alias)을 붙여서 처리

- 사용 방법

- ① SELECT 칼럼  
FROM 테이블 A JOIN 테이블 B  
ON A.칼럼 = B.칼럼;
- ② SELECT 칼럼  
FROM 테이블 A, 테이블 B  
WHERE A.칼럼 = B.칼럼;

# SELF JOIN

- 한 테이블에 존재하는 칼럼들을 이용해 조인 처리

EMPLOYEE\_T

EMP_NO	NAME	MGR_NO	HIRE_DATE	SALARY
1001	구창민	1003	95-05-01	5000000
1002	김민서	1003	17-09-01	2500000
1003	이은영	NULL	90-09-01	5500000
1004	한성일	1003	93-04-01	5000000

EMPLOYEE\_T

EMP_NO	NAME	MGR_NO	HIRE_DATE	SALARY
1001	구창민	1003	95-05-01	5000000
1002	김민서	1003	17-09-01	2500000
1003	이은영	NULL	90-09-01	5500000
1004	한성일	1003	93-04-01	5000000

내부 조인 쿼리와 결과

```
SELECT A.EMP_NO, A.NAME, B.EMP_NO, B.NAME
FROM EMPLOYEE_T A INNER JOIN EMPLOYEE_T B
ON A.EMP_NO = B.MGR_NO;
```

EMP_NO	NAME	EMP_NO_1	NAME_1
1003	이은영	1001	구창민
1003	이은영	1002	김민서
1003	이은영	1004	한성일

외부 조인 쿼리와 결과

```
SELECT A.EMP_NO, A.NAME, B.EMP_NO, B.NAME
FROM EMPLOYEE_T A RIGHT OUTER JOIN EMPLOYEE_T B
ON A.EMP_NO = B.MGR_NO;
```

EMP_NO	NAME	EMP_NO_1	NAME_1
1003	이은영	1001	구창민
1003	이은영	1002	김민서
1003	이은영	1004	한성일
NULL	NULL	1003	이은영

# 조인 순서

- 드라이빙 테이블(Driving Table)
  - ✓ 두 개의 테이블이 조인되는 경우 먼저 처리되는 테이블을 의미함
  - ✓ 인덱스(INDEX)가 설정된 칼럼을 조건으로 활용할 수 있어야 함
- 드리븐 테이블(Driven Table)
  - ✓ 두 개의 테이블이 조인되는 경우 나중에 처리되는 테이블을 의미함
  - ✓ 드라이빙 테이블로부터 상수 값을 받아서 조건이 처리됨
- ✓ 좋은 조인 순서(Good Join Order)
  - ① 일반적으로 처리범위가 적은 쪽에서부터 드라이빙 되는 것이 유리
  - ② 드라이빙 테이블은 FROM절의 가장 왼쪽에 위치
  - ③ 조인 조건을 가장 먼저 작성하되 드라이빙 테이블과 연관된 순서대로 작성  
(조인 조건 이후 일반 조건 작성)
  - ④ 일반 조건 역시 드라이빙 테이블의 조건을 먼저 작성  
(드라이빙 테이블의 Sampling 개수를 줄여서 처리하기 위함)