

조인 검색 : 복수의 테이블에 대한 검색

SELECT S.Sname, S.Dept, E.Grade
FROM STUDENT S, ENROL E;

SELECT S.Sname, S.Dept, E.Grade
FROM STUDENT S, ENROL E
WHERE S.Sno = E.Sno;

SELECT S.Sname, S.Dept, E.Grade
FROM STUDENT S, ENROL E
WHERE S.Sno = E.Sno AND E.Cno = 'C413';

- 이론적으로, **FROM** 절에 두개 이상의 테이블이 명세될 때에는, 테이블들의 곱집합의 결과를 검색대상의 테이블로 설정하고 검색문을 수행
- **WHERE** 절에 조인 조건이 나오면 비로소 세타조인이 됨. 예에서는 동일조인.
- 조인 조건은 다른 검색조건과 조합 가능.

조건이 없는 복수테이블 검색

학생
(STUDENT)

학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
100	나 수 영	4	컴퓨터
200	이 찬 수	3	전기
300	정 기 태	1	컴퓨터

등록
(ENROL)

학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	C413	A	90	95
100	E412	A	95	95
200	C123	B	85	80
300	C312	A	90	95
300	C324	C	75	75

SELECT *
FROM STUDENT S, ENROL E;

학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)	학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	나 수 영	4	컴퓨터	100	C413	A	90	95
100	나 수 영	4	컴퓨터	100	E412	A	95	95
100	나 수 영	4	컴퓨터	200	C123	B	85	80
100	나 수 영	4	컴퓨터	300	C312	A	90	95
100	나 수 영	4	컴퓨터	300	C324	C	75	75
200	이 찬 수	3	전기	100	C413	A	90	95
200	이 찬 수	3	전기	100	E412	A	95	95
200	이 찬 수	3	전기	200	C123	B	85	80
200	이 찬 수	3	전기	300	C312	A	90	95
200	이 찬 수	3	전기	300	C324	C	75	75
300	정 기 태	1	컴퓨터	100	C413	A	90	95
300	정 기 태	1	컴퓨터	100	E412	A	95	95
300	정 기 태	1	컴퓨터	200	C123	B	85	80
300	정 기 태	1	컴퓨터	300	C312	A	90	95
300	정 기 태	1	컴퓨터	300	C324	C	75	75

조인 검색

학생
(STUDENT)

학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
100	나 수 영	4	컴퓨터
200	이 찬 수	3	전기
300	정 기 태	1	컴퓨터

등록
(ENROL)

학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	C413	A	90	95
100	E412	A	95	95
200	C123	B	85	80
300	C312	A	90	95
300	C324	C	75	75

SELECT
FROM
WHERE

*
STUDENT S, ENROL E
S.Sno = E.Sno;

학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)	학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	나 수 영	4	컴퓨터	100	C413	A	90	95
100	나 수 영	4	컴퓨터	100	E412	A	95	95
200	이 찬 수	3	전기	200	C123	B	85	80
300	정 기 태	1	컴퓨터	300	C312	A	90	95
300	정 기 태	1	컴퓨터	300	C324	C	75	75

조인 검색

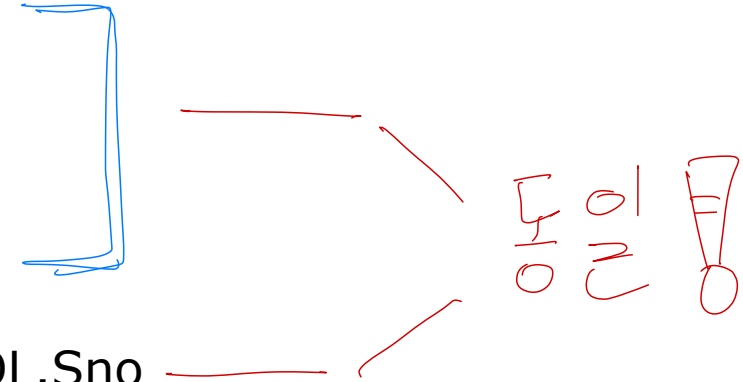
SELECT S1.Sno, S2.Sno
FROM STUDENT S1, STUDENT S2
WHERE S1.Dept = S2.Dept AND S1.Sno < S2.Sno;

- **셀프조인**: FROM 절에 동일한 테이블 2개가 명세됨.
- 서로다른 테이블로 간주.
- (해석) 같은 과 학생들의 학번 쌍을 중복없이 출력하라.

SELECT S.Sname, S.Dept, E.Grade
FROM STUDENT S, ENROL E
WHERE S.Sno = E.Sno AND E.Cno = 'C413'

SELECT Sname, Dept, Grade
FROM STUDENT JOIN ENROL ON STUDENT.Sno=ENROL.Sno
WHERE ENROL.Cno = 'C413';

- 2가지 조인 표기법



조인과 Group by 결합

- 컴퓨터과 학생들이 수강하는 각 과목별, 과목번호, 학생 수를 검색하라.

```
SELECT e.cno, count(*) as student_count  
FROM student s, enrol e  
WHERE s.sno = e.sno AND s.dept = '컴퓨터'  
GROUP BY e.cno
```

LEFT / RIGHT /FULL (OUTER) JOIN

```
SELECT  Sname, Dept, Grade
FROM    STUDENT LEFT OUTER JOIN ENROL ON
STUDENT.Sno=ENROL.Sno
```

- 일반적인 (inner) join은 조인조건을 만족하는 양쪽 튜플의 조합을 새로운 튜플로 반환.
- Left outer join은 기존 조인되는 튜플 쌍 이외에, 왼쪽 테이블에 있는 조인 안되는 튜플들도 결과 테이블에 나옴. 이 경우 오른쪽 테이블의 애트리뷰트 영역은 NULL로 채워져 나옴.
- Right outer join은 기존 조인되는 튜플 쌍 이외에, 오른쪽 테이블에 있는 조인안되는 튜플들도 결과 테이블에 나옴.
- Full outer join 은 조인안되는 양쪽 튜플이 모두 나옴.

LEFT OUTER JOIN

학생
(STUDENT)

학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
100	나 수 영	4	컴퓨터
200	이 찬 수	3	전기
300	정 기 태	1	컴퓨터

등록
(ENROL)

학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	C413	A	90	95
100	E412	A	95	95
300	C312	A	90	95
300	C324	C	75	75

SELECT
FROM

*
STUDENT S **JOIN** ENROL E
ON S.Sno = E.Sno;

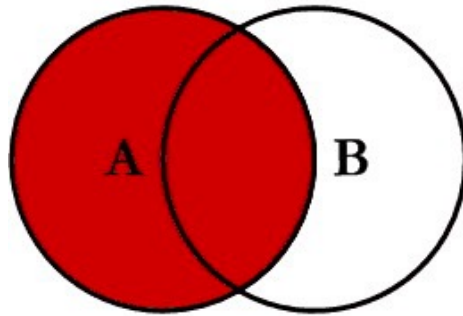
학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)	학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	나 수 영	4	컴퓨터	100	C413	A	90	95
100	나 수 영	4	컴퓨터	100	E412	A	95	95
300	정 기 태	1	컴퓨터	300	C312	A	90	95
300	정 기 태	1	컴퓨터	300	C324	C	75	75

SELECT
FROM

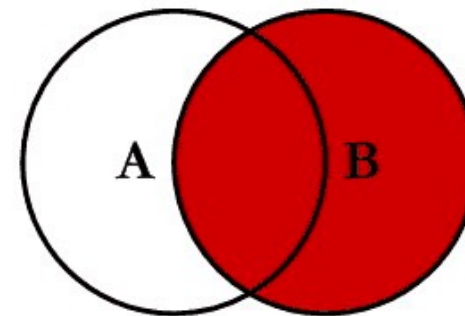
*
STUDENT S **LEFT OUTER JOIN** ENROL E
ON S.Sno = E.Sno;

학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)	학번 (Sno)	과목번호 (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	나 수 영	4	컴퓨터	100	C413	A	90	95
100	나 수 영	4	컴퓨터	100	E412	A	95	95
200	이 찬 수	3	전기	NULL	NULL3	NULL	NULL	NULL
300	정 기 태	1	컴퓨터	300	C312	A	90	95
300	정 기 태	1	컴퓨터	300	C324	C	75	75

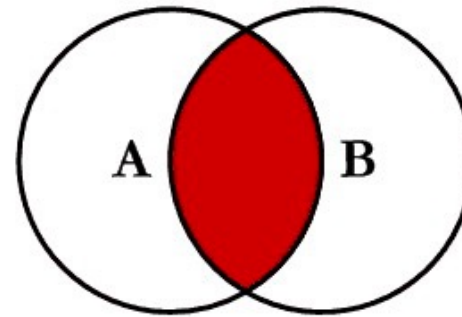
SQL JOINS



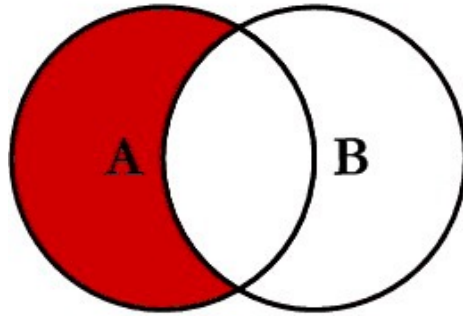
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



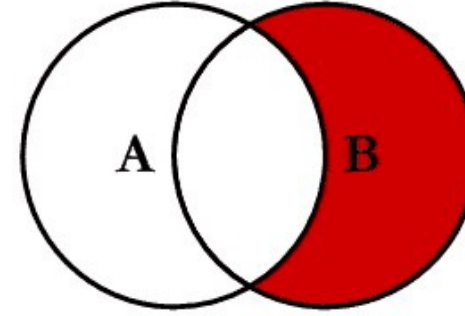
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



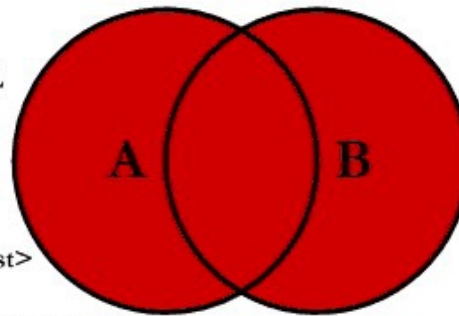
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



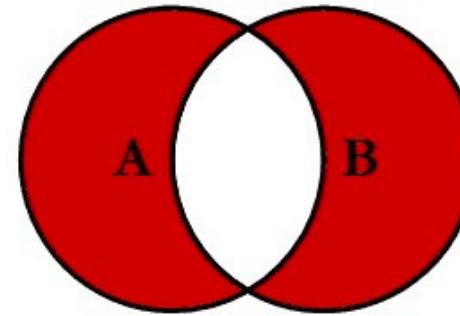
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```