

06. 조인(Join)

조인[Join]

- 관계형 데이터베이스 질의(Query)의 대표적인 기능
- 정규화(Normalization) 이론과 연계
 - 정규화 이론: 중복되는 데이터를 별도의 테이블로 분리하는 방법론
 - 정규화를 수행하여 테이블을 분리하면 필연적으로 Join을 하게 됨
 - Join은 두 개 이상의 테이블을 비교/연산하는 작업이 요구되므로 컴퓨팅과 메모리 자원이 요구됨
 - 따라서, 정규화와 Join은 Trade-off 관계
- 조인이 수행될 때는 두 개 이상의 테이블이 사용되는데 이때 둘 중 하나의 테이블을 먼저 읽고 조인 조건 절을 확인하여 나머지 테이블에 가서 데이터를 가져옴
 - 먼저 읽는 테이블을 선행 테이블(driving table 또는 Inner table)이라고 하고, 뒤에 읽는 테이블을 후행 테이블(driven table 또는 Outer table)이라고 함
 - 선행 테이블을 데이터가 적은 테이블로 선택해야 속도 면에서 유리 함

조인 사용법

- 명시적 표현법(Explicit Notation)

- SELECT * FROM EMP

- $\text{INNER JOIN DEPT ON EMP.DEPTNO = DEPT.DEPTNO}$

- 묵시적 표현법(Implicit Notation)

- SELECT * FROM EMP, DEPT

- $\text{WHERE EMP.DEPTNO = DEPT.DEPTNO}$

- * 대부분 DBMS는 두 표현법을 거의 유사한 성능으로 지원

- DBMS를 이해할수록 명시적 표현법 사용을 권장

조인 사용시 주의사항

- SQL의 명확한 의미 파악
 - SQL의 작성 내용에 따라 성능이 크게 좌우됨
- 조인 조건을 명확하게 제공
 - 원치 않는 결과가 쉽게 발생할 수 있음
- 조인 적용 전 반드시 테스트 수행
 - 큰 테이블에 갑자기 적용하는 경우 성능 이슈가 갑자기 발생할 수 있음

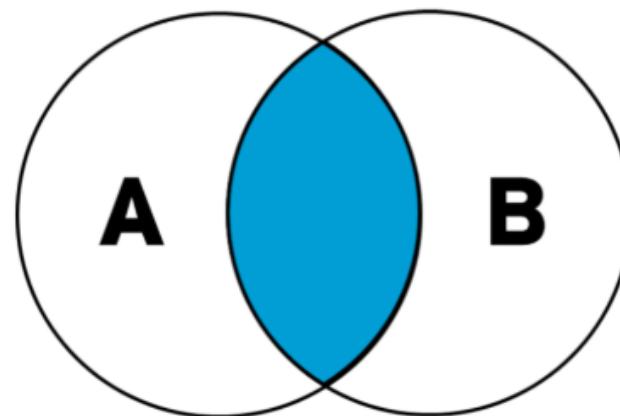
조인의 종류

- INNER JOIN (내부 조인)
 - EQUI JOIN(동등 조인), NATURAL JOIN(자연 조인), CROSS JOIN(교차 조인), SELF JOIN(셀프 조인)
- OUTER JOIN (외부 조인)
 - LEFT OUTER JOIN(왼쪽 외부 조인), RIGHT OUTER JOIN(오른쪽 외부 조인), FULL OUTER JOIN(완전 외부 조인)
- 기타 조인
 - SEMI JOIN(세미 조인), ANTI JOIN(안티/부정형 조인)

INNER JOIN

- 가장 일반적인 형태의 조인 방법
- 공통 속성(컬럼)의 값이 같은 행을 결과로 추출

INNER JOIN



INNER JOIN: EQUI [동등 조인]

- 가장 일반적인 조인 활용 방식으로, =(등호)를 사용하는 경우만 동등 조인으로 명명
- 사용 방법
 - 명시적 표현법
 - SELECT E.EMPNO, E.ENAME, E.JOB, E.MGR, E.HIREDATE, D.DNAME, D.LOC
FROM EMP E

INNER JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
 - 묵시적 표현법
 - SELECT E.EMPNO, E.ENAME, E.JOB, E.MGR, E.HIREDATE, D.DNAME, D.LOC
FROM EMP E, DEPT D

WHERE E.DEPTNO = D.DEPTNO;

INNER JOIN: NATURAL [자연 조인]

- 같은 컬럼명을 가진 컬럼으로 조인을 수행하고, 조인에 해당되는 컬럼은 하나만 출력됨
- 사용 방법
 - 명시적 표현법
 - SELECT * FROM EMP NATURAL JOIN DEPT
 - 묵시적 표현법 없음
 - INNER JOIN과의 비교
 - SELECT * FROM EMP E INNER JOIN DEPT D ON E.DEPTNO = D.DEPTNO;
 - DEPTNO 컬럼이 EMP용과 DEPT용 두 번 출력됨
 - NATURAL JOIN의 경우에는 DEPTNO가 한 번만 출력

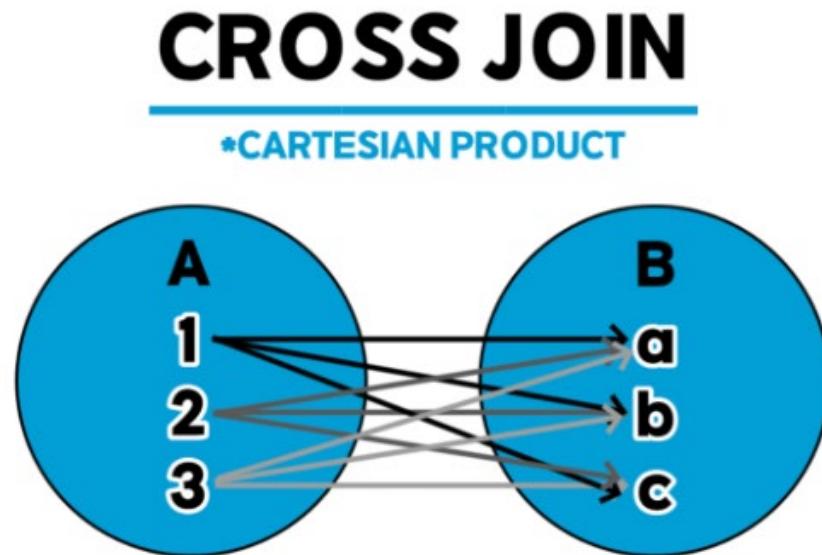
INNER JOIN: CROSS (교차 조인)

- Cartesian Product(카티션 곱)이라고도 함
- 조인에 참여한 모든 테이블이 출력
 - 다음 경우에 의해 발생
 - JOIN 조건을 잘못 기술
 - JOIN 조건을 정의하지 않음
 - JOIN 조건이 조건에 참여하는 테이블의 모든 행에 조인되는 경우
 - 유도하지 않은 교차 조인은 성능상 큰 문제를 발생
 - 반드시 테스트 단계에서 SQL 결과를 확인
 - 팁: ‘조인 테이블이 N개라면, 조인 조건 최소한 $N-1$ 개’를 유념

INNER JOIN: CROSS (교차 조인)

- 사용 방법

- 명시적 표현법
 - SELECT * FROM EMP **CROSS JOIN** DEPT
- 묵시적 표현법
 - SELECT * FROM EMP, DEPT



CROSS JOIN 결과: 전체 행 개수 = 9
3(A 테이블의 행 개수) X 3(B 테이블의 행 개수)

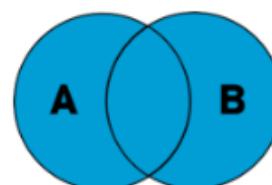
INNER JOIN: SELF [셀프 조인]

- 동일한 테이블 내에서 데이터를 추출하는 방법
 - EMP 테이블에서 사원들의 관리자 정보를 출력하는 경우 등에 사용
 - 같은 테이블을 사용하기 때문에 반드시 alias(별칭) 기능을 사용해야 함
- 사용 방법
 - 명시적 표현법
 - `SELECT E.EMPNO, E.NAME, E.JOB, M.EMPNO, M.ENAME, E.JOB
FROM EMP E
INNER JOIN EMP M ON E.MGR=M.EMPNO;`
 - 묵시적 표현법
 - `SELECT E.EMPNO, E.NAME, E.JOB, M.EMPNO, M.ENAME, E.JOB
FROM EMP E, EMP M
WHERE E.MGR = M.EMPNO;`

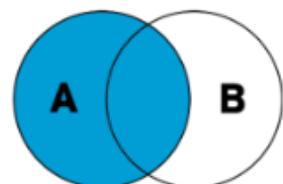
OUTER JOIN [외부 조인]

- 조인의 경우 특정 테이블의 데이터가 모두 필요한 경우
 - 두 테이블 사이에 일치 하지 않는 행도 포함하여 기준이 되는 테이블을 반환하는 조인
- 묵시적 표현법이 없음

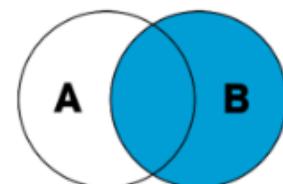
OUTER JOIN



FULL OUTER JOIN



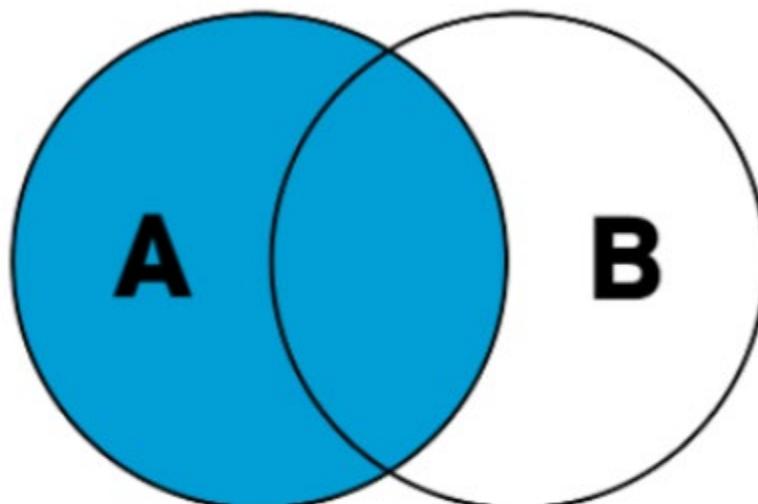
LEFT OUTER JOIN



RIGHT OUTER JOIN

LEFT OUTER JOIN (왼쪽 외부 조인)

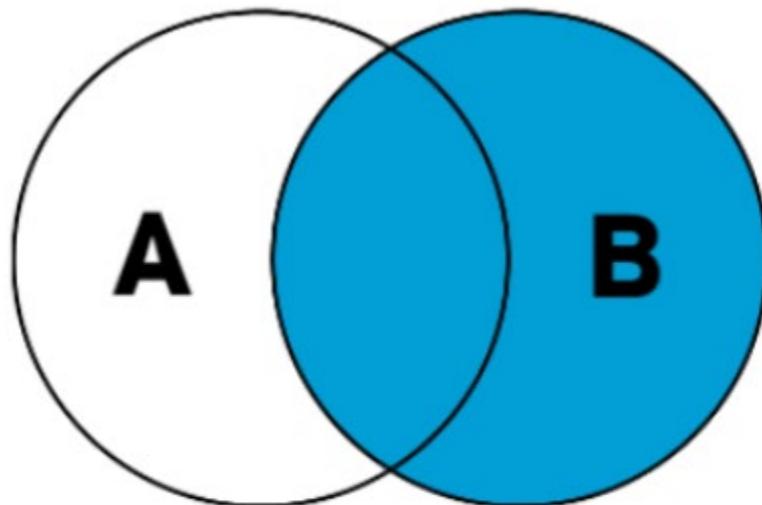
- 왼쪽 테이블의 모든 행과 오른쪽 테이블에서 일치하는 행을 반환
- 오른쪽 테이블에서 일치하는 행이 없는 경우 NULL 값으로 채워짐
- 사용 방법(명시적 표현법)
 - SELECT * FROM EMP
LEFT (OUTER) JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;



LEFT OUTER JOIN

RIGHT OUTER JOIN (오른쪽 외부 조인)

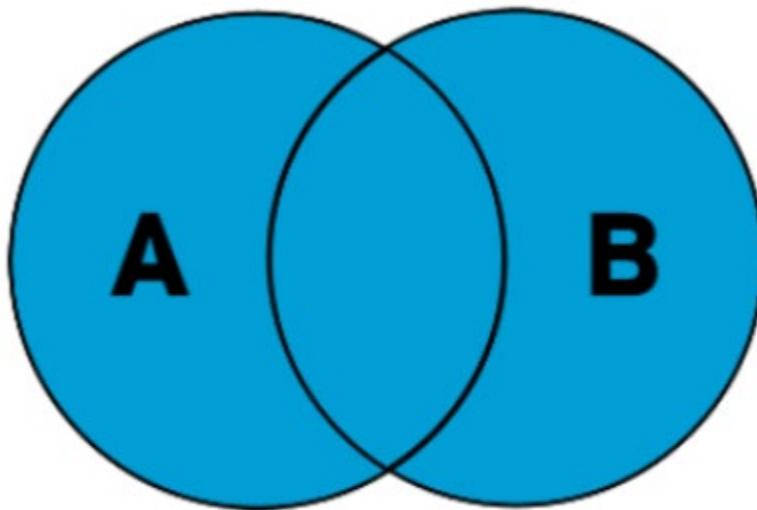
- 오른쪽 테이블의 모든 행과 왼쪽 테이블에서 일치하는 행을 반환
- 왼쪽 테이블에서 일치하는 행이 없는 경우 NULL 값으로 채워짐
- 사용 방법(명시적 표현법)
 - SELECT * FROM EMP
RIGHT (OUTER) JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;



RIGHT OUTER JOIN

FULL OUTER JOIN (완전 외부 조인)

- 양쪽 테이블의 모든 행을 반환
- 양쪽 테이블에서 일치하는 행이 없는 경우 NULL 값으로 채워짐
- 사용 방법(명시적 표현법)
 - `SELECT * FROM EMP`
 - `FULL OUTER JOIN DEPT ON EMP.DEPTNO=DEPT.DEPTNO;`

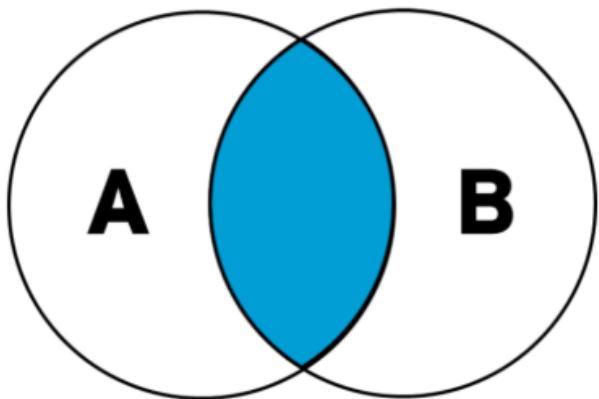


FULL OUTER JOIN

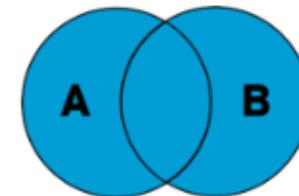
정리

- 조인은 두 개의 테이블을 서로 묶어서 하나의 결과를 만들어 내는 것을 말함
- INNER JOIN(내부 조인)은 두 테이블을 조인할 때, 두 테이블에 모두 지정한 열의 데이터가 있어야 함
- OUTER JOIN(외부 조인)은 두 테이블을 조인할 때, 1개의 테이블에만 데이터가 있어도 결과가 나옴
- CROSS JOIN(교차 조인)은 한쪽 테이블의 모든 행과 다른 쪽 테이블의 모든 행을 조인하는 기능
- SELF JOIN(셀프 조인)은 자신이 자신과 조인한다는 의미로, 1개의 테이블을 사용
- 성능적으로 보면 **INNER JOIN**이 **OUTER JOIN**보다 교집합만 조회함으로써 더 빠름. 주로 INNTER JOIN을 많이 사용하며 그렇다고 OUTER JOIN이 좋지 않다는 것은 아니니 상황에 맞게 잘 사용하도록 해야함

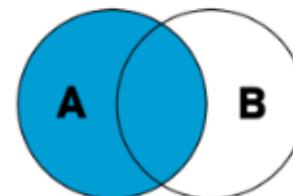
INNER JOIN



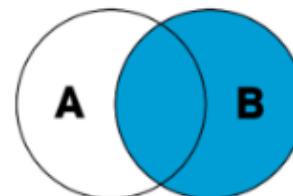
OUTER JOIN



FULL OUTER JOIN

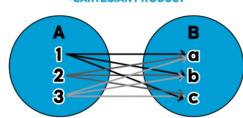


LEFT OUTER JOIN



RIGHT OUTER JOIN

CROSS JOIN +CARTESIAN PRODUCT



CROSS JOIN 결과: 전체 행 개수 = 9
3(A 테이블의 행 개수) X 3(B 테이블의 행 개수)

SELF JOIN

