

# Join

## Join exam

M		S		X	
M1	M2	S1	S2	X1	X2
A	1	A	X	A	DATA
B	1	B	Y		
C	3	NULL	Z		
NULL	3				

# 테이블간의 관계에 따른 분류

- INNER JOIN
- CROSS JOIN
- OUTER JOIN
- SELF JOIN
- NONEQUI JOIN

# INNER JOIN

- 각 테이블의 조인 컬럼 (공통 컬럼)을 비교하여 **조인조건을 만족하는 레코드만 선택하는 조인**
- 테이블 M, S에서  $M1=S1$  조건을 만족하는 레코드만 가져오는 JOIN문을 작성해 보자

M		S	
M1	M2	S1	S2
A	1	A	X
B	1	B	Y
C	3	NULL	Z
NULL	3		

비교컬럼

INNER JOIN 조건을 만족하는 컬럼

# INNER JOIN

--Ansi JOIN

```
SELECT * FROM M INNER JOIN S ON M1=S1
```

-- SQL SERVER JOIN –

```
SELECT * FROM M,S WHERE M1=S1
```

[결과]

M1	M2	S1	S2
A	1	A	X
B	1	B	Y

# INNER JOIN

**Q1.** INNER JOIN을 이용해서 사원의 이름과 그 사원이 속해있는 부서이름을 출력해 보라

```
SELECT ENAME, DNAME
```

```
FROM EMP, DEPT
```

```
WHERE EMP.DEPTNO=DEPT.DEPTNO
```

**Q2.** 사원의 이름과 그 사원이 속해있는 부서이름과 부서번호를 출력하라.

```
SELECT ENAME, DNAME, DEPTNO
```

```
FROM EMP, DEPT
```

```
WHERE EMP.DEPTNO=DEPT.DEPTNO
```

# CROSS JOIN

각 테이블의 모든 로우에 대해서 가능한 모든 조합을 가지는 쿼리 결과를 만들어 내는조인

-- ANSI JOIN--

```
SELECT *  
FROM M CROSS JOIN S
```

-- SQL SERVER JOIN --

```
SELECT *  
FROM M,S
```

M			S	
M1	M2		S1	S2
A	1		A	X
E	1		B	Y
C	3		NULL	Z
NULL	3			

# OUTER JOIN

- 두 테이블간에 주, 종관계를 두고 주 테이블의 모든 레코드와 종 테이블에서 조인 조건을 만족하는 레코드만 가져올 때 사용
- 주 테이블의 위치에 따라서 LEFT OUTER JOIN, RIGHT OUTER JOIN, 그리고 두 개의 결과를 합한 FULL OUTER JOIN으로 구분된다.



# OUTER JOIN

- **LEFT OUTER JOIN**

- M 테이블을 주 테이블로 놓고, S 테이블을 종 테이블로 하여 조인을 걸어서  $M1=S1$  조건을 만족하는 레코드를 가져오는 LEFT OUTER JOIN을 작성

```
SELECT *  
FROM M LEFT OUTER JOIN S  
ON M1=S1
```

```
-- SQL SERVER JOIN --  
SELECT *  
FROM M,S  
WHERE M1=S1(+);
```

M		S	
M1	M2	S1	S2
A	1	A	X
B	1	B	Y
C	3	NULL	Z
NULL	3		

# OUTER JOIN

- **RIGHT OUTER JOIN**

- S 테이블을 주 테이블에 놓고, M 테이블을 종 테이블로 한 RIGHT OUTER JOIN을 작성

```
--ANSI  
SELECT *  
FROM M RIGHT OUTER JOIN S  
ON M1=S1
```

```
-- SQL SERVER JOIN --  
SELECT *  
FROM M,S  
WHERE M1(+)=S1;
```

[결과]

M1	M2	S1	S2
A	1	A	X
B	1	B	Y
NULL	NULL	NULL	Z

조건  
WHERE M1=S1

종 테이블의 조건에 맞지  
않은 데이터들은 NULL값  
을 반환한다.

# FULL OUTER JOIN

- 두 테이블을 LEFT OUTER JOIN, RIGHT OUTER JOIN을 한 결과를 합한 것과 동일한 결과를 가져온다.

```
-- ANSI JOIN --  
SELECT *  
FROM M FULL OUTER JOIN S  
ON M1=S1
```

- LEFT, RIGHT OUTER JOIN을 사용한 쿼리를 UNION으로 합했을 경우

```
SELECT *  
FROM M,S  
WHERE M1(+)=S1  
UNION  
SELECT *  
FROM M,S  
WHERE M1=S1(+)
```

# 세 개 이상의 테이블에서 조인 걸기

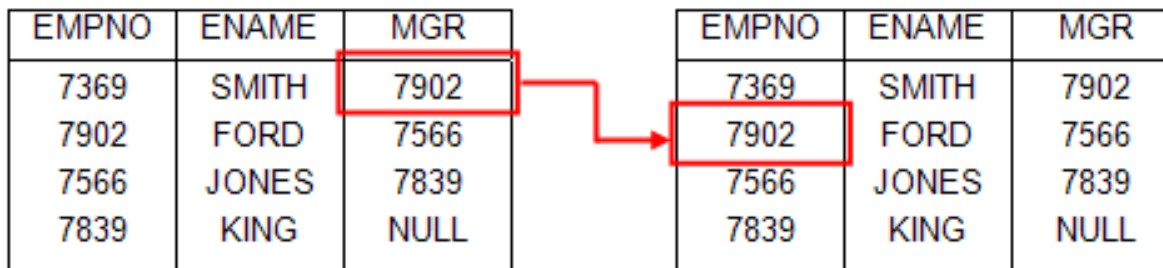
M		S		X	
M1	M2	S1	S2	X1	X2
A	1	A	X	A	DATA
B	1	B	Y		
C	3	NULL	Z		
NULL	3				

```
-- ANSI JOIN --  
SELECT *  
FROM M INNER JOIN S  
ON M1=S1  
      INNER JOIN X  
      ON S1=X1
```

```
-- SQL SERVER JOIN --  
SELECT *  
FROM M,S,X  
WHERE M1=S1 AND S1=X1
```

# SELF JOIN

- 하나의 테이블 내에서 서로 다른 컬럼간에 참조 관계가 있을 때 걸리는 JOIN.



EMPNO	ENAME	MGR
7369	SMITH	7902
7902	FORD	7566
7566	JONES	7839
7839	KING	NULL

EMPNO	ENAME	MGR
7369	SMITH	7902
7902	FORD	7566
7566	JONES	7839
7839	KING	NULL

```
SELECT 사원.EMPNO,사원.ENAME,관리자.EMPNO,관리자.ENAME  
FROM EMP 사원, EMP 관리자  
WHERE 사원.MGR=관리자.EMPNO(+)
```

# NONEQUI JOIN

**Q1.** 각 사원의 이름과 월급, 그리고 그 사원의 급여등급을 출력하라.

```
SELECT E.ENAME, E.SAL, S.GRADE  
FROM EMP E, SALGRADE S  
WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL
```

**Q2.** 각 사원의 이름, 월급, 급여등급, 그가 속한 부서이름을 출력하라.

```
SELECT E.ENAME, E.SAL, D.DNAME, S.GRADE  
FROM EMP E, DEPT D, SALGRADE S  
WHERE E.DEPTNO=D.DEPTNO  
AND E.SAL BETWEEN S.LOSAL AND S.HISAL
```