

제5장 SQL



5.4 다중테이블 검색

1. 조인

2. 하위 쿼리

3. 유니온

1. 조인

개념

- 두 개 이상의 테이블들을 연결하여 데이터 검색
- 연결하려는 테이블 간에 조인 속성의 이름은 달라도 되지만 도메인은 같아야 함. 일반적으로 외래키가 조인 속성으로 이용됨
- 너무 많은 테이블끼리 조인을 수행할 때는 성능을 떨어뜨리므로 주의해야 함(시스템에 따라 좌우되기는 하지만, 보통 4~5개 정도까지의 테이블 사이의 조인은 크게 문제되지 않는다.)

Join 방법

- Equi Join (등가 조인, 내부조인)
- Outer Join(외부조인)
- Non-Equi Join
- Self Join

1. 조인

Cartesian Product

- ✓ 검색하고자 했던 데이터뿐 아니라 조인에 사용된 테이블들의 모든 데이터가 반환 되는 현상
- ✓ Cartesian Product는 조인 조건을 정의하지 않은 경우 발생한다.
- ✓ **CROSS JOIN**을 이용하면 Cartesian Product 값을 얻을 수 있다.

1. 조인

크로스 조인 예

고객

회사번호	회사명
1001	신영상사
1002	원창

주문

주문번호	회사번호	주문일
10248	1001	2013-07-04
10249	1001	2013-07-05
10250	1002	2013-07-08
10251	1004	2013-07-08

SELECT 주문번호, 회사명, 주문일



FROM 고객 , 주문 ;

주문번호	회사명	주문일
10248	신영상사	2013-07-04
10249	신영상사	2013-07-05
10250	신영상사	2013-07-08
10251	신영상사	2013-07-08
10248	원창	2013-07-04
10249	원창	2013-07-05
10250	원창	2013-07-08
10251	원창	2013-07-08

1. 조인

Equi Join

- 가장 일반적으로 Equality Condition(=)에 의한 조인이다
- Equi join의 성능을 높이기 위해서는 Index 기능을 사용하는 것이 좋다.

Outer Join

- 조인 조건을 만족하지 않는 행까지 결과 집합에 포함시켜 돌려주는 특수한 조인으로, 집계할 때 종종 사용된다.

Self Join

- Equi Join과 같으나 하나의 테이블에서 조인이 일어나는 것이 다르다.
- 같은 테이블에 대해 두 개의 alias를 사용하여 FROM절에 두 개의 테이블을 사용하는 것 처럼 조인한다.

1. 조인

내부조인 예

공급처		
업체번호	회사명	도시명
1	태양식품 (주)	부산광역시
2	신한식품 (주)	서울특별시
3	대양농산 (주)	서울특별시
4	서울무역 (주)	서울특별시
5	현진식품 (주)	서울특별시
6	유미식품 (주)	인천광역시
7	대륙교역 (주)	인천광역시

제품		
업체번호	제품명	단가
1	딸기	10000
2	사과	15000
3	바나나	6000
4	수박	25000
6	참외	10000
7	키위	12000
8	배	6500



```
SELECT 회사명, 제품명, 단가
FROM 공급처
INNER JOIN 제품
ON 공급처.업체번호=제품.업체번호
WHERE 도시명="서울특별시";
```

회사명	제품명	단가
신한식품 (주)	사과	15000
대양농산 (주)	바나나	6000
서울무역 (주)	수박	25000

1. 조인

외부조인 예

공급처		
업체번호	회사명	도시명
1	태양식품 (주)	부산광역시
2	신한식품 (주)	서울특별시
3	대양농산 (주)	서울특별시
4	서울무역 (주)	서울특별시
5	현진식품 (주)	서울특별시
6	유미식품 (주)	인천광역시
7	대륙교역 (주)	인천광역시

제품		
업체번호	제품명	단가
1	딸기	10000
2	사과	15000
3	바나나	6000
4	수박	25000
6	참외	10000
7	키위	12000
8	배	6500

SELECT 회사명, 제품명, 단가
FROM 공급처
LEFT JOIN 제품
ON 공급처.업체번호=제품.업체번호
WHERE 도시명="서울특별시";

회사명	제품명	단가
신한식품 (주)	사과	15000
대양농산 (주)	바나나	6000
서울무역 (주)	수박	25000
현진식품 (주)		

1. 조인

내부조인 예

공급처		
업체번호	회사명	도시명
1	태양식품 (주)	부산광역시
2	신한식품 (주)	서울특별시
3	대양농산 (주)	서울특별시
4	서울무역 (주)	서울특별시
5	현진식품 (주)	서울특별시
6	유미식품 (주)	인천광역시
7	대륙교역 (주)	인천광역시

제품		
업체번호	제품명	단가
1	딸기	10000
2	사과	15000
3	바나나	6000
4	수박	25000
6	참외	10000
7	키위	12000
8	배	6500

SELECT 회사명, 제품명, 단가
FROM 공급처, 제품
WHERE 공급처.업체번호=제품.업체번호;

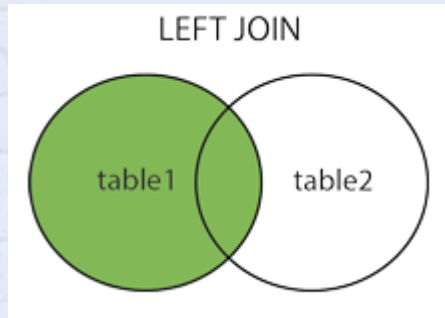
SELECT 회사명, 제품명, 단가
FROM 공급처 **INNER JOIN** 제품
ON 공급처.업체번호=제품.업체번호;

콤마(,) 대신 **INNER JOIN**을
사용 할 수 있으며, **INNER**
는 생략 가능하다. Join 조건
은 **ON** 절에 온다.

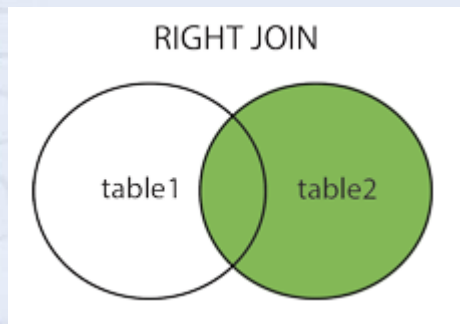
1. 조인

▪ 외부 조인 구문

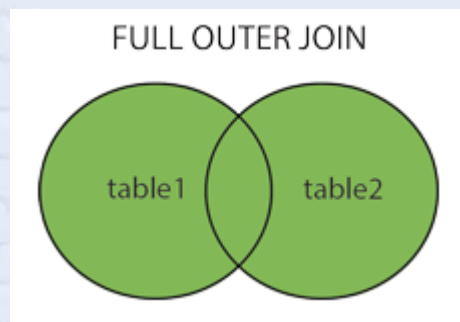
```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name= table2.column_name;
```



```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name= table2.column_name;
```



```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name= table2.column_name;
```



1. 조인

LEFT 외부 조인 예

고객

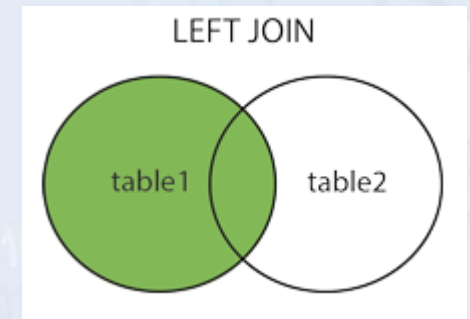
회사번호	회사명
1001	신영상사
1002	원창
1003	동광통상
1004	경성트레이딩
1005	정금상사

주문

주문번호	회사번호	주문일
10248	1001	2013-07-04
10249	1001	2013-07-05
10250	1002	2013-07-08
10251	1004	2013-07-08
10255	1007	2013-07-12

SELECT 고객.회사번호, 회사명, 주문일
FROM 고객 **LEFT JOIN** 주문 **ON** 고객.회사번호=주문.회사번호;

회사번호	회사명	주문일
1001	신영상사	2013-07-04
1001	신영상사	2013-07-05
1002	원창	2013-07-08
1003	동광통상	NULL
1004	경성트레이딩	2013-07-08
1005	정금상사	NULL



1. 조인

RIGHT 외부 조인 예

고객

회사번호	회사명
1001	신영상사
1002	원창
1003	동광통상
1004	경성트레이딩
1005	정금상사

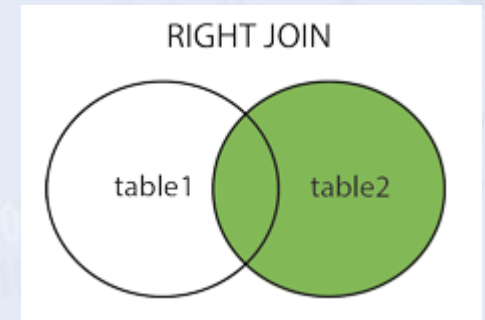
주문

주문번호	회사번호	주문일
10248	1001	2013-07-04
10249	1001	2013-07-05
10250	1002	2013-07-08
10251	1004	2013-07-08
10255	1007	2013-07-12

SELECT 고객.회사번호, 회사명, 주문일

FROM 고객 **RIGHT JOIN** 주문 **ON** 고객.회사번호=주문.회사번호;

회사번호	회사명	주문일
1001	신영상사	2013-07-04
1001	신영상사	2013-07-05
1002	원창	2013-07-08
1004	경성트레이딩	2013-07-08
1007	NULL	2013-07-12



1. 조인

FULL 외부 조인 예

고객

회사번호	회사명
1001	신영상사
1002	원창
1003	동광통상
1004	경성트레이딩
1005	정금상사

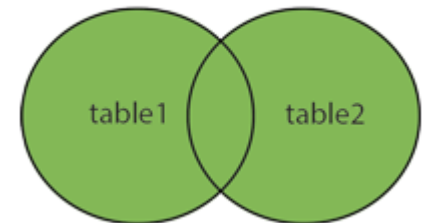
주문

주문번호	회사번호	주문일
10248	1001	2013-07-04
10249	1001	2013-07-05
10250	1002	2013-07-08
10251	1004	2013-07-08
10255	1007	2013-07-12

SELECT 고객.회사번호, 회사명, 주문번호, 주문일
FROM 고객 **FULL OUTER JOIN** 주문 **ON** 고객.회사번호=주문.회사번호;

회사번호	회사명	주문번호	주문일
1001	신영상사	10248	2013-07-04
1001	신영상사	10249	2013-07-05
1002	원창	10250	2013-07-08
1003	동광통상	NUL	NULL
1004	경성트레이딩	10251	2013-07-08
1005	정금상사	NULL	NULL
NULL	NULL	10255	2013-07-12

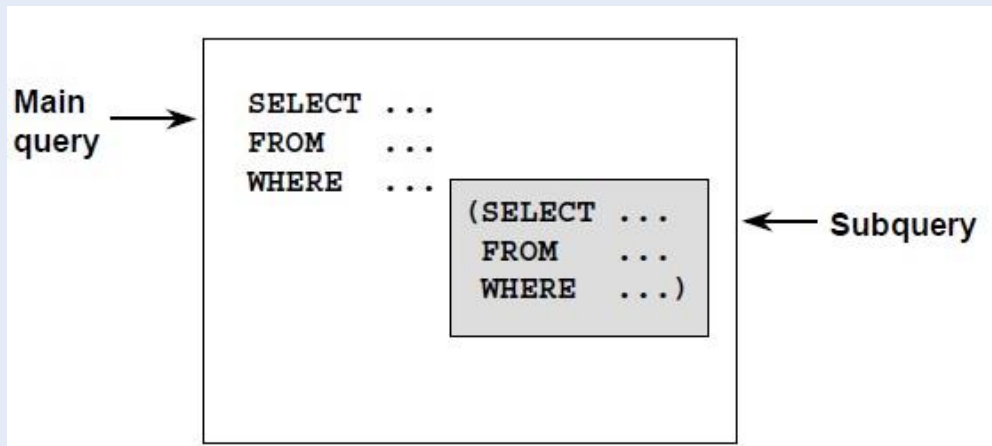
FULL OUTER JOIN



2. 하위 쿼리

하위쿼리란?

- ✓ 다른 하나의 SQL 문장의 절에 NESTED된 SELECT 문장 이다.
- ✓ SELECT, UPDATE, DELETE, INSERT와 같은 DML문과 CREATE TABLE 또는 VIEW의 WHERE절이나 HAVING절에서 사용 된다.
- ✓ 보통 Subquery는 Main Query 이전에 한 번 실행된다.
- ✓ Subquery는 괄호로 묶어야 한다.
- ✓ 단일 행 연산자(=, >, >=, <, <=, <>, !=)와 다중 행 연산자(IN, NOT IN, ANY, ALL, EXISTS)들이 서브쿼리에 사용 된다.
- ✓ Subquery는 연산자의 오른쪽에 위치해야 한다.



2. 하위 쿼리

하위쿼리의 유형

- ✓ 단일 행(Sing-Row) 서브쿼리 : SELECT 문장으로 부터 오직 하나의 행 만을 검색하는 질의이다.
- ✓ 다중 행(Multiple-Row) 서브쿼리 : SELECT 문장으로부터 하나 이상의 행을 검색하는 질의이다.
- ✓ 다중 열(Multiple-Column) 서브쿼리 : SELECT 문장으로부터 하나 이상의 컬럼을 검색하는 질의이다.
- ✓ FROM절상의 서브쿼리(INLINE VIEW) : FROM절상에 오는 서브쿼리로 VIEW처럼 작용 한다.
- ✓ 상관관계 서브 쿼리 : 바깥쪽 쿼리의 컬럼 중의 하나가 안쪽 서브쿼리의 조건에 이용되는 처리 방식 이다.

2. 하위 쿼리

하위쿼리의 유형

Single-Row Subquery

- SELECT 문장에서 오직 하나의 행(값)만 반환하는 Query이다
- 단일 행 연산자(=, >, >=, <, <=, <>, !=) 만 사용된다.

예제 : 사번이 7369인 사원의 직업과 같은 직업을 갖는 사원의 이름과 직업을 검색

```
SELECT 이름, 직업
FROM 사원
WHERE 직업 = (SELECT 직업
               FROM 사원
               WHERE 사번 = 7369);
```


2. 하위 쿼리

하위쿼리의 유형

Multiple-Row Subquery란?

- 하나 이상의 행을 반환하는 Subquery이다
- 단일 행 연산자를 사용하지 못하며, 다중 행 연산자(IN, NOT IN, ANY, ALL, EXISTS)만 사용이 가능하다.

예제: 부서별로 가장 급여를 많이 받는 사원의 정보를 검색

```
SELECT 사번, 성명, 급여, 부서번호  
FROM 사원  
WHERE 급여 IN (SELECT MAX(급여)  
                FROM 사원  
                GROUP BY 부서번호);
```

사번	성명	급여	부서번호
7698	BLAKE	2850	30
7788	SCOTT	3000	20
7902	FORD	3000	20

2. 하위 쿼리

하위쿼리의 유형

다중 행 하위쿼리에 사용 가능한 연산자

연산자	설명
IN	부속 질의문의 결과 값 중 일치하는 것이 있으면 검색 조건이 참
NOT IN	부속 질의문의 결과 값 중 일치하는 것이 없으면 검색 조건이 참
EXISTS	부속 질의문의 결과 값이 하나라도 존재하면 검색 조건이 참
NOT EXISTS	부속 질의문의 결과 값이 하나도 존재하지 않으면 검색 조건이 참
ALL	부속 질의문의 결과 값 모두와 비교한 결과가 참이면 검색 조건을 만족(비교 연산자와 함께 사용)
ANY 또는 SOME	부속 질의문의 결과 값 중 하나라도 비교한 결과가 참이면 검색 조건을 만족(비교 연산자와 함께 사용)

2. 하위 쿼리

하위쿼리의 유형

ANY 연산자

- ANY 연산자는 Subquery의 여러 결과값 중 어느 하나의 값만 만족이 되면 행을 반환 한다.

예제: 직업이 "회계사"인 사원의 급여보다 많이 받는 사원의 사원의 성명과 직업, 급여를 검색

```
SELECT 성명, 직업, 급여
FROM 사원
WHERE 급여 > ANY (SELECT 급여
                     FROM 사원
                     WHERE 직업 = ' 회계사');
```

2. 하위 쿼리

하위쿼리의 유형

ALL 연산자

- ALL 연산자는 Subquery의 여러 결과값 중 모든 결과 값을 만족해야 행을 반환 한다.

예제: 직업이 "회계사"인 모든 사원의 급여보다 많이 받는 사원의 사원의 성명과 직업, 급여를 검색

```
SELECT 성명, 직업, 급여
FROM 사원
WHERE 급여 > ALL (SELECT 급여
                     FROM 사원
                     WHERE 직업 = ' 회계사');
```

ENAME	SAL
-----	-----
CLARK	2450
BLAKE	2850
KING	5000

2. 하위 쿼리

하위쿼리의 유형

EXISTS 연산자

- EXISTS 연산자는 Subquery 데이터가 존재하는가를 체크해 존재 여부 (TRUE,FALSE)를 결과로 반환한다.
- EXISTS절에는 반드시 메인 쿼리와 연결이 되는 조인 조건을 가지고 있어야 한다.
- subquery에서 결과 행을 찾으면, inner query 수행을 중단하고 TRUE를 반환한다.

2. 하위 쿼리

하위쿼리의 유형

```
SELECT DISTINCT 부서.부서번호, 부서.부서명  
FROM 부서, 사원  
WHERE 부서.부서번호 = 사원.부서번호;
```



EXISTS를 사용하는 Subquery로 변경
-- 추출하고자 하는 대상만을 FROM절에 놓고 사원 테이블은
체크만 하기위해 EXISTS절에 위치시켰으며 이로 인해
수행속도가 대폭 감소하게 된다.

```
SELECT 부서.부서번호, 부서.부서명  
FROM 부서  
WHERE EXISTS (SELECT *  
                FROM 사원  
                WHERE 사원.부서번호 = 부서.부서번호 );
```

2. 하위 쿼리

하위쿼리의 유형

Multiple-Column Subquery란?

결과 값이 두 개 이상의 컬럼을 반환하는 Subquery이다

Pairwise(쌍비교) Subquery

Subquery가 한 번 실행되면서 두 개 이상의 컬럼을 검색해서 주 쿼리로 넘겨 준다.

```
SELECT 사번, 급여, 부서번호  
FROM 사원  
WHERE (급여, 부서번호) IN (SELECT 급여, 부서번호  
                           FROM 사원  
                           WHERE 부서번호=30) ;
```

3. 유니온

개념

- ✓ UNION과 UNION ALL의 차이는 정렬작업의 수행 여부이다.
- ✓ UNION은 중복을 제거해야 하기 때문에 정렬작업을 수행하므로 성능이 저하된다.
- ✓ UNION, MINUS, INTERSECT는 전체범위를 모두 액세스 하는 정렬작업을 수행하기 때문에 부분범위처리가 불가능하다.

3. 유니온

UNION : 합집합

UNION은 두 테이블의 결합을 나타내며, 결합시키는 두 테이블의 중복되지 않은 값들을 반환 한다.

공급처		
업체번호	회사명	도시명
1	태양식품 (주)	부산광역시
2	신한식품 (주)	서울특별시
3	대양농산 (주)	서울특별시
4	서울무역 (주)	서울특별시
5	현진식품 (주)	서울특별시
6	유미식품 (주)	인천광역시

제품		
업체번호	제품명	단가
2	사과	15000
3	바나나	6000
4	수박	25000
7	참외	10000
8	키위	12000
9	배	6500

업체번호
1
2
3
4
5
6
7
8
9

SELECT 업체번호 **FROM** 공급처
UNION
SELECT 업체번호 **FROM** 제품;

3. 유니온

UNION ALL : 중복을 포함하는 합집합

UNION과 같으나 두 테이블의 중복되는 값 까지 반환 한다.

공급처		
업체번호	회사명	도시명
1	태양식품 (주)	부산광역시
2	신한식품 (주)	서울특별시
3	대양농산 (주)	서울특별시
4	서울무역 (주)	서울특별시
5	현진식품 (주)	서울특별시
6	유미식품 (주)	인천광역시

제품		
업체번호	제품명	단가
2	사과	15000
3	바나나	6000
4	수박	25000
7	참외	10000
8	키위	12000
9	배	6500

SELECT 업체번호 **FROM** 공급처
UNION ALL
SELECT 업체번호 **FROM** 제품;

업체번호
1
2
2
3
3
4
4
5
...