

두개 이상의 테이블 조회

자바 강의실

목차

I. 두 개 이상의 테이블에서 자료 조회

II. 계층형 쿼리(셀프 조인 고급)

1. 두 개 이상의 테이블에서 자료 조회

• 조인

- 조인은 두 개의 릴레이션으로부터 연관된 튜플들을 결합
- 조인의 일반적인 형식은 아래의 SELECT문과 같이 FROM절에 두 개 이상의 릴레이션들이 열거되고, 두 릴레이션에 속하는 애트리뷰트들을 비교하는 조인 조건이 WHERE절에 포함됨
- 조인 조건은 두 릴레이션 사이에 속하는 애트리뷰트 값들을 비교 연산자로 연결한 것 (보통 내부조인 이라고 함)
- 가장 흔히 사용되는 비교 연산자 : =

```
SELECT      ...  
FROM        R, S  
WHERE       R.A <비교 연산자> S.B ;
```

↑
조인조건

1. 두 개 이상의 테이블에서 자료 조회

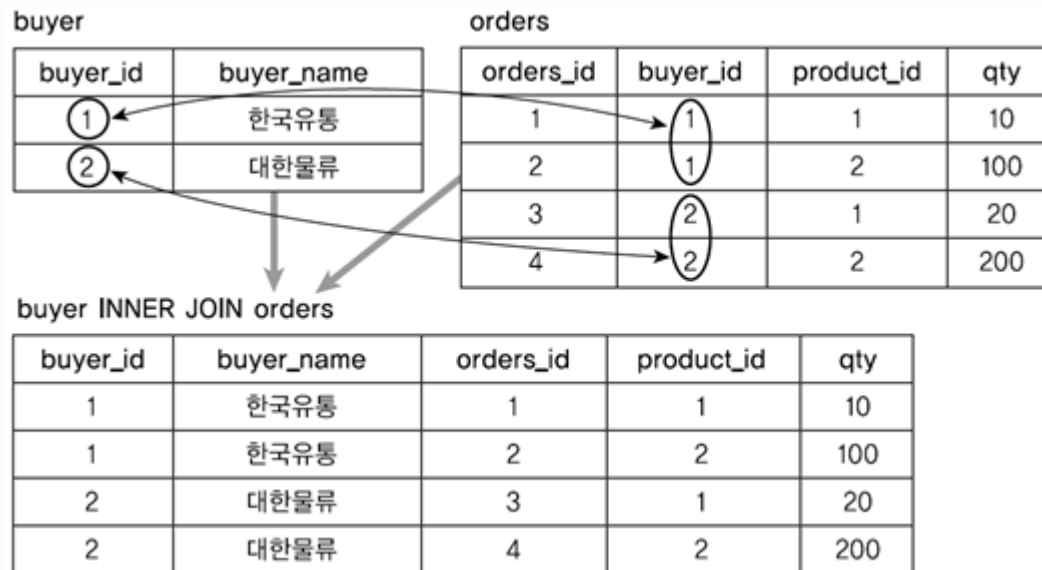
- 세타 조인과 내부 조인

- 두 릴레이션 $R(A_1, A_2, \dots, A_n)$ 과 $S(B_1, B_2, \dots, B_m)$ 의 세타 조인의 결과는 차수가 $n+m$ 이고, 애트리뷰트가 $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m)$ 이며, 조인 조건을 만족하는 튜플들로 이루어진 릴레이션
- 세타 조인은 $\{=, <>, <=, <, >=, >\}$ 중의 하나를 이용한 비교한다
- 내부 조인(동등 조인)은 세타 조인 중에서 비교 연산자가 $=$ 인 조인

1. 두 개 이상의 테이블에서 자료 조회

- 조인(계속)

- 내부 조인(inner join): 각 테이블에서 비교 연산자에 의한 조인 조건을 만족하는 행들만 조인 테이블에 포함시켜주는 조인으로, 대부분의 조인이 이에 해당됨



1. 두 개 이상의 테이블에서 자료 조회

- 조인(계속)

- 조인 조건을 생략했을 때와 조인 조건을 틀리게 표현했을 때는 카티션 곱이 생성됨(크로스 조인이라고 함)
- 조인 질의가 수행되는 과정을 개념적으로 살펴보면 먼저 조인 조건을 만족하는 튜플들을 찾고, 이 튜플들로부터 SELECT절에 명시된 애트리뷰트들만 프로젝트하고, 필요하다면 중복을 배제하는 순서로 진행됨
- 조인 조건이 명확해지도록 애트리뷰트 이름 앞에 릴레이션 이름이나 튜플 변수를 사용하는 것이 바람직
- 두 릴레이션의 조인 애트리뷰트 이름이 동일하다면 반드시 애트리뷰트 이름 앞에 릴레이션 이름이나 튜플 변수를 사용해야 함
- 너무 많은 테이블끼리 조인을 수행할 때는 성능을 떨어뜨리므로 주의해야 함(시스템에 따라 좌우되기는 하지만, 보통 4~5개 정도까지의 테이블 사이의 조인은 크게 문제되지 않는다.

1. 두 개 이상의 테이블에서 자료 조회

- 조인(계속)

- 크로스 조인(cross join, 상호 조인): 두 테이블을 서로 곱하는 개념의 연산으로, 일명 카티션 곱(Cartesian product)이라 함
- 조인 테이블의 행 수는 두 테이블의 행 수를 곱한 값이 되며 열 수는 두 테이블의 열 수를 더한 값이 됨

buyer

buyer_id	buyer_name
1	한국유통
2	대한물류

orders

orders_id	buyer_id	product_id	qty
1	1	1	10
2	1	2	100
3	2	1	20
4	2	2	200

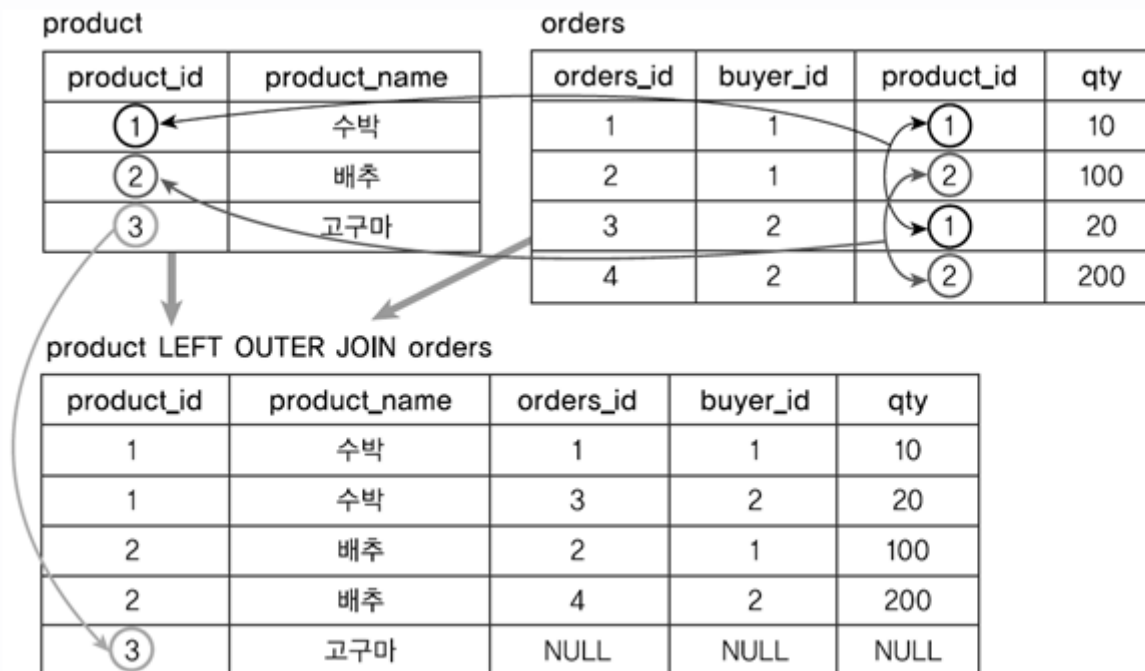
buyer * orders

buyer_id	buyer_name	orders_id	buyer_id	product_id	qty
1	한국유통	1	1	1	10
1	한국유통	2	1	2	100
1	한국유통	3	2	1	20
1	한국유통	4	2	2	200
2	대한물류	1	1	1	10
2	대한물류	2	1	2	100
2	대한물류	3	2	1	20
2	대한물류	4	2	2	200

1. 두 개 이상의 테이블에서 자료 조회

• 조인(계속)

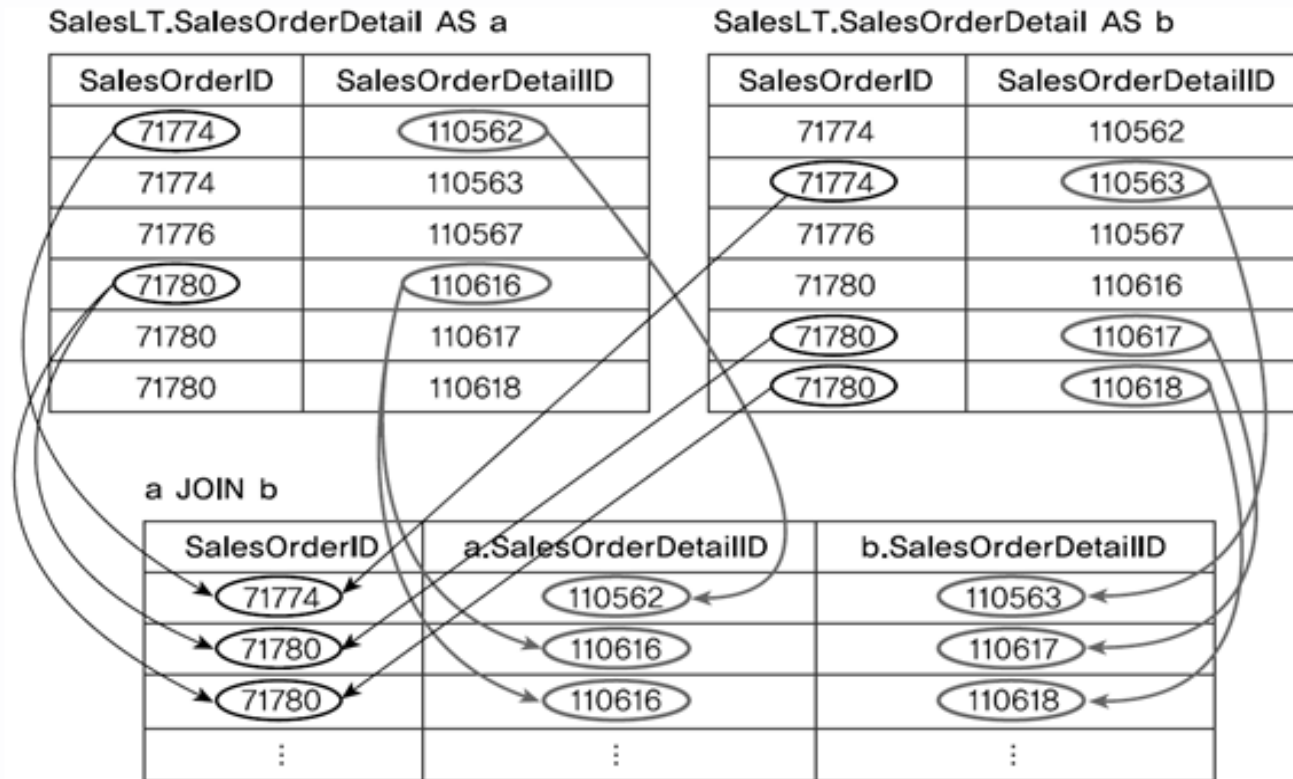
- 외부 조인(outer join): 조인 조건을 만족하지 않는 행까지 결과 집합에 포함시켜 돌려주는 특수한 조인으로, 집계할 때 종종 사용된다.
- 왼쪽 외부 조인, 오른쪽 외부 조인, 완전 외부 조인으로 나누어짐



1. 두 개 이상의 테이블에서 자료 조회

• 조인(계속)

- 자체 조인(self join): 자기 자신의 테이블과 하는 조인
- 자체 조인은 혼하지는 않지만, 같은 테이블의 서로 관련이 있는 행들을 조합해서 추출할 필요가 있을 때 종종 사용됨



1. 두 개 이상의 테이블에서 자료 조회

- 일반적인 조인 종류

종 류	설 명
내부조인(Equi Join)	동일 칼럼을 기준으로 조인
크로스조인(Non-Equi Join)	동일 칼럼 없이 다른 조건 사용하여 조인
외부조인(Outer Join)	조인 조건에 만족하지 않는 행도 나타남
자체조인(Seif Join)	한 테이블 내에서 조인

- 이외 많은 조인 종류들이 있으며 쿼리의 속도를 향상시키는 데 사용되는 고급 쿼리로 사용

1. 두 개 이상의 테이블에서 자료 조회

예: 조인질의

질의 : 모든 사원의 이름과 이 사원이 속한 부서 이름을 검색하라

```
SELECT      FIRST_NAME, DEPARTMENT_NAME
FROM        EMPLOYEES E, DEPARTMENTS D
WHERE       E.DEPARTMENTS_ID = D.DEPARTMENTS_ID;
```

EMPLOYEES 릴레이션의 각 튜플에 대해 부서번호를 사용하여 DEPARTMENTS 릴레이션에서 같은 부서번호를 갖는 튜플을 찾는다. EMPLOYEES 릴레이션의 튜플과 DEPARTMENTS 튜플을 결합하면 결과 릴레이션의 한 튜플이 생성된다. EMPLOYEES 릴레이션의 마지막 튜플까지 이 과정을 반복한다.

1. 두 개 이상의 테이블에서 자료 조회

EMPLOYEES

<u>EMPLOYEE_ID</u>	FIRST_NAME	...	DEPARTMENTS_ID
2106	김창섭	...	2
3426	박영권	...	1
3011	이수민	...	3
1003	조민희	...	2
3427	최종철	...	3
1365	김상원	...	1
4377	이성래	...	2

2106	김창섭	...	2
------	-----	-----	---

DEPARTMENTS

<u>DEPARTMENTS_ID</u>	DEPARTMENT_NAME	FLOOR
1	영업	8
2	기획	10
3	개발	9
4	총무	7

2	기획	10
---	----	----

2106	김창섭	...	2	기획	10
------	-----	-----	---	----	----

1. 두 개 이상의 테이블에서 자료 조회

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	JOB_ID	MANAGER_ID	SALARY	DEPARTMENT_ID	DEPARTMENT_NAME	FLOOR
2106	김창섭	대리	1077	2500000	2	영업	8
3426	박영권	과장	4377	3000000	1	기획	10
3011	이수민	부장	4377	4000000	3	개발	9
1003	조민희	과장	4377	3000000	2	영업	8
3427	최종철	사원	3011	1500000	3	개발	9
1365	김상원	사원	3426	1500000	1	기획	10
4377	이성래	사장	^	5000000	2	총무	7

1. 두 개 이상의 테이블에서 자료 조회

예: 외부조인질의

질의 : 모든 사원의 이름과 이 사원이 속한 부서 이름을 검색하되 해당 부서 이름이 없는 사원의 까지 포함하여 검색하라

```
SELECT      FIRST_NAME, DEPARTMENT_NAME
FROM        EMPLOYEES E, DEPARTMENTS D
WHERE       E.DEPARTMENTS_ID = D.DEPARTMENTS_ID(+);
```

EMPLOYEES 릴레이션의 각 튜플에 대해 부서번호를 사용하여 DEPARTMENTS 릴레이션에서 같은 부서번호를 갖는 튜플을 찾는다. EMPLOYEES 릴레이션의 튜플과 DEPARTMENTS 튜플을 결합하면 결과 릴레이션의 한 튜플이 생성된다. EMPLOYEES 릴레이션의 마지막 튜플까지 이 과정을 반복한다.

조인 값이 없는 테이블 [(+) 로표시] 의 튜플을 추가한다

1. 두 개 이상의 테이블에서 자료 조회

예: 조인과 ORDER BY의 결합

질의 : 모든 사원에 대해서 사원에 대해서 소속 부서이름, 사원의 이름, 직급, 급여를 검색하라. 부서 이름에 대해서 오름차순, 부서이름이 같은 경우에는 SALARY에 대해서 내림차순으로 정렬하라

```
SELECT      DEPARTMENT_NAME, FIRST_NAME, JOB_ID, SALARY
FROM        EMPLOYEES E, DEPARTMENTS D
WHERE       E.DEPARTMENTS_ID = D.DEPARTMENTS_ID
ORDER BY    DEPARTMENT_NAME, SALARY DESC;
```

오름차순	DEPARTMENT_NAME	FIRST_NAME	JOB_ID	SALARY	내림차순
	개발	이수민	부장	4000000	
	개발	최종철	사원	1500000	
	기획	이성래	사장	5000000	
	기획	조민희	과장	3000000	
	기획	김창섭	대리	2500000	
	영업	박영권	과장	3000000	
	영업	김상원	사원	1500000	

2. 계층형 쿼리

- 자체 조인(self join)

- 한 릴레이션에 속하는 튜플을 동일한 릴레이션에 속하는 튜플들과 조인하는 것
- 실제로는 한 릴레이션이 접근되지만 FROM절에 두 릴레이션이 참조되는 것처럼 나타내기 위해서 그 릴레이션에 대한 별칭을 두 개 지정해야 함

예: 자체조인

질의 : 모든 사원에 대해서 사원의 이름과 직속상사의 이름을 검색하라

```
SELECT      E.FIRST_NAME, M.FIRST_NAME
FROM        EMPLOYEES AS E, EMPLOYEES AS M
WHERE       E.MANAGER_ID = M.EMPLOYEE_ID;
```

EMPLOYEES 릴레이션의 각 튜플에 대해 상사번호 (MANAGER_ID)를 사용하여 자체릴레이션에서 그 상사번호와 같은 직원번호를 갖는 튜플을 찾는다. 두 튜플을 결합하면 결과 릴레이션의 한 튜플이 생성된다. EMPLOYEES 릴레이션의 마지막 튜플까지 이 과정을 반복한다. 사용자가 검색하려는 애트리뷰트만 프로젝션한다.

2. 계층형 쿼리

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	MANAGER_ID	...
2106	김창섭	1003	...
3426	박영권	4377	...
3011	이수민	4377	...
1003	조민희	4377	...
3427	최종철	3011	...
1365	김상원	3426	...
4377	이성래	^	...

2106	김창섭	1003	...
------	-----	------	-----

EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	MANAGER_ID	...
2106	김창섭	1003	...
3426	박영권	4377	...
3011	이수민	4377	...
1003	조민희	4377	...
3427	최종철	3011	...
1365	김상원	3426	...
4377	이성래	^	...

1003	조민희	4377	...
------	-----	------	-----

2106	김창섭	1003	...	조민희	4377	...
------	-----	------	-----	-----	------	-----

2. 계층형 쿼리

최종결과 릴레이션

E.FIRST_NAME	M.FIRST_NAME
김창섭	조민희
박영권	이성래
이수민	이성래
조민희	이성래
최종철	이수민
김상원	박영권