



DATABASE

Chapter 06

관계 데이터 연산

Relational Data Operation

대전대학교 컴퓨터공학과

CONTENTS

01

관계 데이터 연산의 개념

데이터 모델의 구성 요소와
관계 대수 vs 관계 해석

02

관계 대수

8개 연산자: 일반 집합 연산자 +
순수 관계 연산자

SECTION 01

관계 데이터 연산의 개념



데이터 모델의 구성 요소

데이터 모델 = 데이터 구조 + 연산 + 제약조건

데이터 구조

릴레이션 (Chapter 5에서 학습)

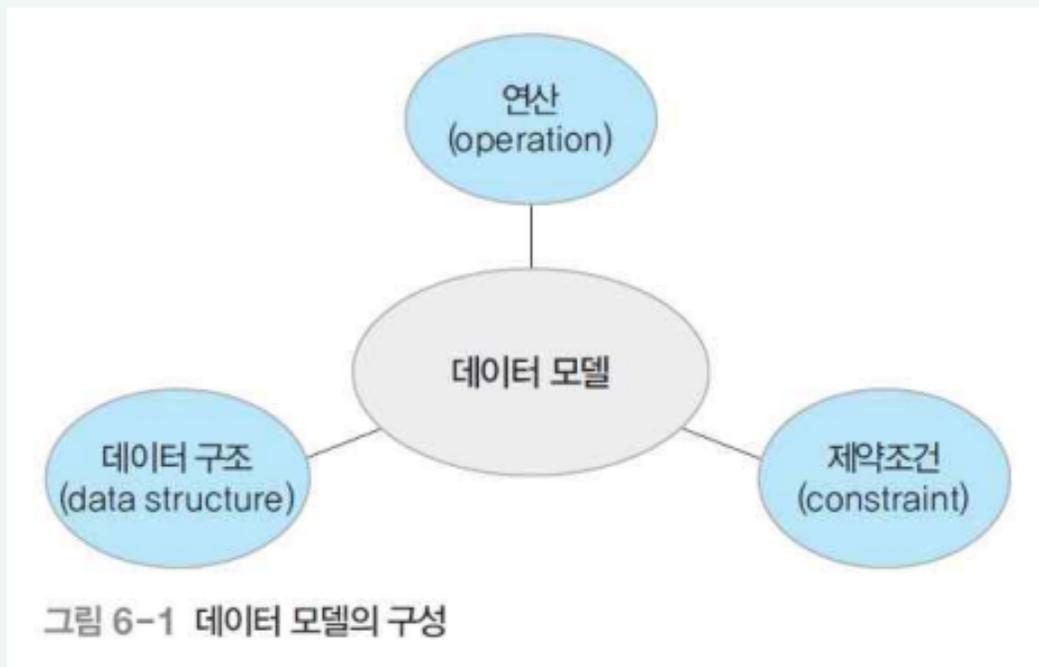
연산

릴레이션에서 원하는 데이터를 추출·가공 (이번 장)

제약조건

무결성 제약조건 (Chapter 5에서 학습)

데이터 모델의 구성 요소 다이어그램



데이터 모델을 구성하는 세 요소: 데이터 구조, 연산, 제약조건.
이번 장에서는 '연산(operation)' 부분을 다룹니다.

관계 데이터 연산

원하는 결과를 얻기 위해 릴레이션에 필요한 처리를 수행하는 것

관계 대수 (How)

절차적 방식

데이터의 처리 과정을
순서대로 기술

관계 해석 (What)

비절차적 방식

원하는 데이터가 무엇인지만 기술
(처리 과정은 기술하지 않음)

SECTION 02

관계 대수

Relational Algebra



관계 대수의 개념

절차 언어 — 처리 과정을 순서대로 기술하여 원하는 결과를 도출

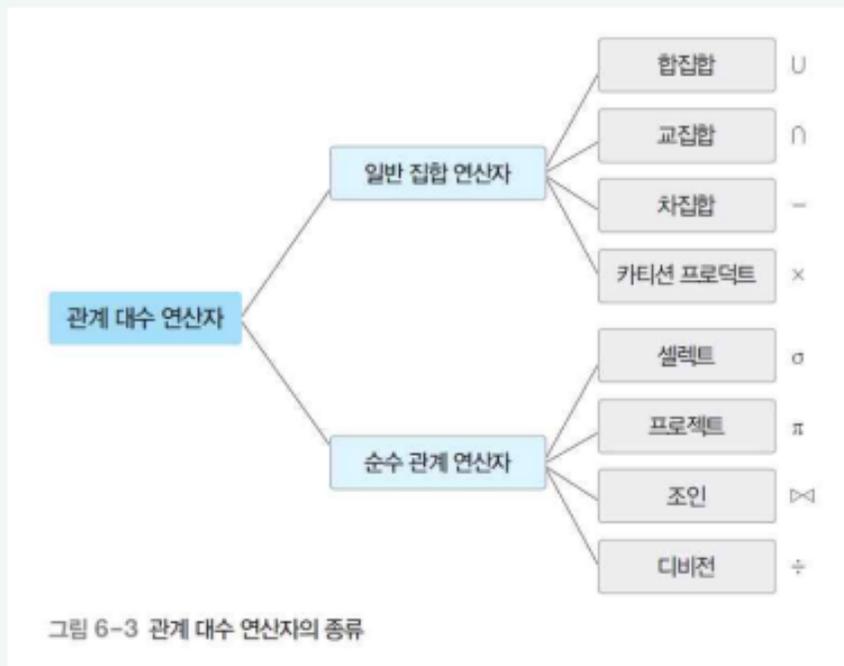
8개 연산자

일반 집합 연산자 4개 + 순수 관계 연산자 4개

폐쇄 특성 (Closure Property)

피연산자도 릴레이션, 결과도 릴레이션 → 연산의 연쇄 적용 가능

관계 대수 연산자 (8개)



관계 대수 연산자는 일반 집합 연산자(합집합 U , 교집합 \cap , 차집합 $-$, 카티션 프로덕트 \times)와 순수 관계 연산자(선택 σ , 프로젝트 π , 조인 \bowtie , 디비전 \div)로 나뉩니다.

일반 집합 연산자

합집합 (Union) \cup

R과 S에 속하는 모든 튜플

교집합 (Intersection) \cap

R과 S에 공통으로 속하는 튜플

차집합 (Difference) $-$

R에는 있고 S에는 없는 튜플

카티션 프로덕트 (Cartesian Product) \times

R의 각 튜플과 S의 각 튜플의 모든 조합

일반 집합 연산자 – 기호, 표현, 의미

연산자	기호	표현	의미
합집합	\cup	$R \cup S$	릴레이션 R과 S의 합집합을 반환
교집합	\cap	$R \cap S$	릴레이션 R과 S의 교집합을 반환
차집합	$-$	$R - S$	릴레이션 R과 S의 차집합을 반환
카티션 프로덕트	\times	$R \times S$	릴레이션 R의 각 튜플과 릴레이션 S의 각 튜플을 모두 연결하여 만든 새로운 튜플을 반환

합집합(\cup), 교집합(\cap), 차집합($-$), 카티션 프로덕트(\times)의 기호와 의미를 정리한 표입니다.
합집합, 교집합, 차집합은 '합병 가능' 조건이 필요합니다.

일반 집합 연산자의 특성

합집합, 교집합, 차집합은 두 릴레이션이 합병 가능해야 함

조건 1

두 릴레이션의 차수
(속성의 전체 개수)가
같아야 함

조건 2

대응되는 속성의
도메인이
같아야 함

합병 불가능 vs 합병 가능 예제

고객 릴레이션

고객번호	고객이름	나이
INT	CHAR(20)	INT
100	정소화	20
200	김선우	35
300	고명석	24

직원 릴레이션

직원번호	직원이름	직위
INT	CHAR(20)	CHAR(20)
10	김용욱	부장
20	채광주	과장
30	김수진	대리

그림 6-6 합병이 불가능한 예

위: 차수는 같지만 속성의 도메인이 달라 합병 불가능
아래: 차수가 같고 속성 도메인도 같아 합병 가능

합집합 (Union) $R \cup S$

R과 S에 속하는 모든 튜플로 결과 릴레이션 구성 (중복 제거)

결과 차수

R, S의 차수와 같음

결과 카디널리티

$R + S$ 의 카디널리티와 같거나 작음 (중복 제거)

교환/결합법칙

$R \cup S = S \cup R$ (교환) / $(R \cup S) \cup T = R \cup (S \cup T)$ (결합)

릴레이션 R과 S의 합집합 예제



R에 3개, S에 3개 튜플이 있고, (100,정소화)가 양쪽에 공통입니다.
RUS 결과는 중복이 제거되어 5개 튜플이 됩니다.

교집합 (Intersection) $R \cap S$

R과 S에 공통으로 속하는 튜플로 결과 릴레이션 구성

결과 차수

R, S의 차수와 같음

결과 카디널리티

R 또는 S의 카디널리티보다 작거나 같음

교환/결합법칙

$R \cap S = S \cap R$ (교환) / $(R \cap S) \cap T = R \cap (S \cap T)$ (결합)

릴레이션 R과 S의 교집합 예제



그림 6-9 교집합 연산의 예

R과 S 양쪽에 모두 존재하는 (100, 정소화) 하나만 결과에 남습니다.

차집합 (Difference) $R-S$

R 에는 존재하지만 S 에는 존재하지 않는 튜플로 결과 구성

주의: $R-S \neq S-R \rightarrow$ 교환법칙 성립하지 않음!

결과 카디널리티

R 의 카디널리티와 같거나 작음

릴레이션 R과 S의 차집합 예제 (R-S와 S-R)



그림 6-10 차집합 연산의 예

R-S: R에만 있는 (200,김선우),(300,고명석) → 2개
 S-R: S에만 있는 (101,채광주),(102,김수진) → 2개
 순서에 따라 결과가 다릅니다.

카티션 프로덕트 $R \times S$

R의 각 튜플과 S의 각 튜플을 모두 연결하여 새 튜플 생성

합병 가능 불필요

서로 다른 구조의 릴레이션도 가능

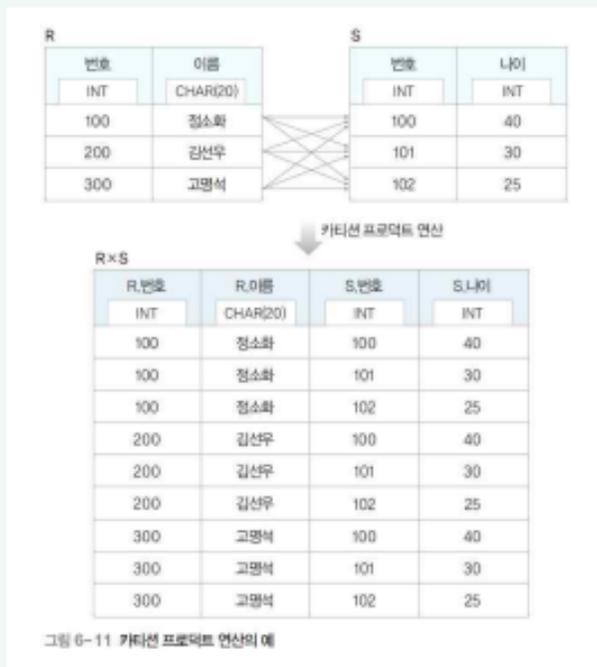
결과 차수

R의 차수 + S의 차수

결과 카디널리티

R의 카디널리티 \times S의 카디널리티

릴레이션 R과 S의 카티션 프로덕트 예제



R의 각 튜플이 S의 모든 튜플과 조합됩니다.
 $R\ 3개 \times S\ 3개 = 결과\ 9개\ 튜플.$ 차수는 $R+S$ 의 합.

순수 관계 연산자



선택 (Select) σ

조건을 만족하는 튜플을 골라냄 (행 필터링)



프로젝트 (Project) π

원하는 속성만 뽑아냄 (열 선택)



조인 (Join) \bowtie

두 릴레이션을 공통 속성으로 연결



디비전 (Division) \div

S의 모든 값과 관련된 R의 튜플을 찾음

순수 관계 연산자 – 기호, 표현, 의미

연산자	기호	표현	의미
선택	σ	$\sigma_{조건}(R)$	릴레이션 R에서 조건을 만족하는 튜플들을 반환
프로젝트	π	$\pi_{속성리스트}(R)$	릴레이션 R에서 주어진 속성들의 값으로만 구성된 튜플들을 반환
조인	\bowtie	$R \bowtie S$	공통 속성을 이용해 릴레이션 R과 S의 튜플들을 연결하여 만든 새로운 튜플들을 반환
디비전	\div	$R \div S$	릴레이션 S의 모든 튜플과 관련이 있는 릴레이션 R의 튜플들을 반환

선택(σ), 프로젝트(π), 조인(\bowtie), 디비전(\div)의 기호와 의미를 정리한 표입니다.

셀렉트 (Select) σ

릴레이션에서 조건을 만족하는 튜플만 선택
→ 행(row)을 필터링하는 연산 · SQL의 WHERE절에 해당

표기: σ 조건(릴레이션)

결과 릴레이션의 차수는 원래와 같고, 카디널리티만 줄어듦
교환법칙 성립: 조건을 적용하는 순서를 바꿔도 결과 동일

선택 예제: 등급이 gold인 투플 검색

예제 6-1

고객 릴레이션에서 등급이 gold인 투플을 검색하시오.

▶▶ $\sigma_{\text{등급}='gold'}(\text{고객})$ 또는 `고객 where 등급 = 'gold'`

결과 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
carrot	원유선	28	gold	교사	4500

$\sigma_{\text{등급}='gold'}(\text{고객}) \rightarrow$ 등급이 gold인 apple과 carrot만 결과에 남습니다.
열(속성)은 그대로, 행(투플)만 줄어드는 것이 선택의 핵심입니다.

선택 예제: AND 조건 결합

예제 6-2

고객 릴레이션에서 등급이 gold이고, 적립금이 2000 이상인 튜플을 검색하시오.

▶▶ $\sigma_{\text{등급}='gold' \wedge \text{적립금} \geq 2000}(\text{고객})$ 또는 고객 where 등급 = 'gold' and 적립금 \geq 2000

결과 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
carrot	원유선	28	gold	교사	4500

등급='gold' AND 적립금 \geq 2000 \rightarrow 두 조건을 동시에 만족하는 carrot만 결과에 포함.
선택트는 교환법칙이 성립합니다.

프로젝트 (Project) π

릴레이션에서 원하는 속성만 선택
→ 열(column)을 고르는 연산 · SQL의 SELECT절에 해당

표기: π 속성리스트(릴레이션)

중요: 결과에서 동일한 튜플은 중복 제거됨 (릴레이션 = 집합)

프로젝트 예제: 고객이름, 등급, 적립금 검색

예제 6-3

고객 릴레이션에서 고객이름, 등급, 적립금을 검색하시오.

▶▶ $\pi_{\text{고객이름, 등급, 적립금}}(\text{고객})$ 또는 $\text{고객}[\text{고객이름, 등급, 적립금}]$

결과 릴레이션

고객이름	등급	적립금
정소화	gold	1000
김선우	vip	2500
고명석	gold	4500
김용욱	silver	0

$\pi_{\text{고객이름, 등급, 적립금}}(\text{고객}) \rightarrow$ 6개 속성에서 3개만 남습니다.
투플 수는 그대로 유지됩니다.

프로젝트 예제: 등급 속성만 검색 (중복 제거)

예제 6-4

고객 릴레이션에서 등급을 검색하시오.

▶▶ $\pi_{\text{등급}}(\text{고객})$ 또는 $\text{고객}[\text{등급}]$

결과 릴레이션

등급
gold
vip
silver

$\pi_{\text{등급}}(\text{고객}) \rightarrow \text{gold, vip, gold, silver}$ 에서 중복 gold가 제거되어 결과는 gold, vip, silver 3개만 남습니다.

조인 (Join) ⚡

두 릴레이션을 공통 속성을 기준으로 연결
→ 가장 강력하고 자주 사용되는 연산 · SQL의 JOIN에 해당

세타 조인 (Theta Join)

일반적인 비교 조건(=, <, > 등)으로 연결

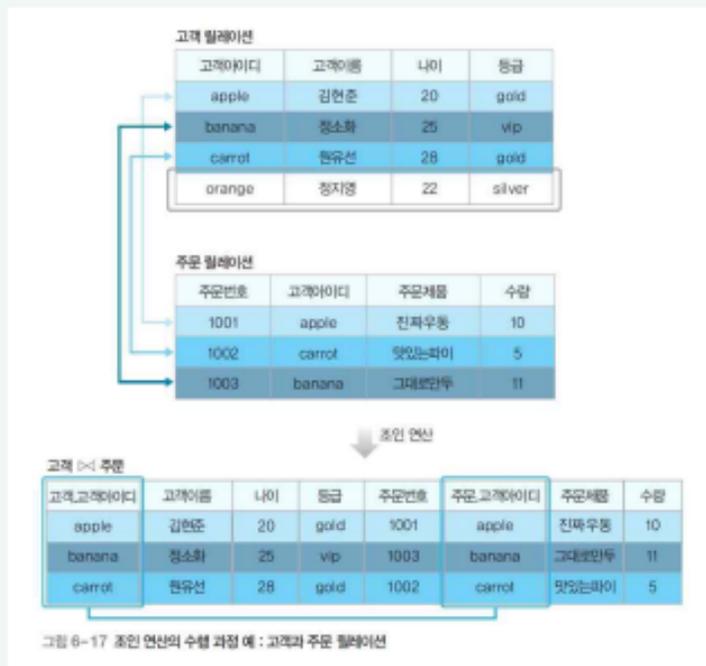
동등 조인 (Equi Join)

= 조건으로 연결 (가장 일반적)

자연 조인 (Natural Join)

같은 이름의 속성을 자동 연결 + 중복 속성 제거

조인 예제: 고객 ✕ 주문



고객 릴레이션과 주문 릴레이션을 고객아이디 기준으로 조인합니다.
매칭되는 튜플끼리 연결되어 양쪽 속성이 합쳐진 결과가 나옵니다.

자연 조인 예제

고객 릴레이션

고객아이디	고객이름	나이	등급
apple	김현준	20	gold
banana	정소화	25	vip
carrot	원유선	28	gold
orange	정지영	22	silver

주문 릴레이션

주문번호	고객아이디	주문제품	수량
1001	apple	진짜우동	10
1002	carrot	맛있는파이	5
1003	banana	그대로만두	11

↓ 자연 조인 연산

고객 \bowtie_N 주문

고객아이디	고객이름	나이	등급	주문번호	주문제품	수량
apple	김현준	20	gold	1001	진짜우동	10
banana	정소화	25	vip	1003	그대로만두	11
carrot	원유선	28	gold	1002	맛있는파이	5

그림 6-18 자연 조인 연산의 예 : 고객과 주문 릴레이션

자연 조인은 같은 이름의 속성을 자동으로 연결하고 중복 속성을 제거하여 더 깔끔한 결과를 만듭니다.

조인 비교 예제: 원본 릴레이션

직원 릴레이션

직원번호	이름	부서	연차
E1	김철수	개발	3
E2	이영희	연구	7
E3	박민수	개발	5

- 같은 이름 속성 2개: 직원번호, 부서
- 직원.부서 = 소속 부서
프로젝트.부서 = 주관 부서
- P2 챗봇: 연구부서 프로젝트에 개발부서 E3 박민수가 교차 배정된 상황

프로젝트 릴레이션

프로젝트코드	프로젝트명	직원번호	부서	최소연차
P1	홈페이지	E1	개발	2
P2	챗봇	E3	연구	4

세타 조인 예제: 연차 \geq 최소연차 \rightarrow 5행

직원번호	이름	연차	프로젝트코드	프로젝트명	최소연차	연차 \geq 최소연차?
E1	김철수	3	P1	홈페이지	2	O 3 \geq 2
E1	김철수	3	P2	챗봇	4	X 3 $<$ 4
E2	이영희	7	P1	홈페이지	2	O 7 \geq 2
E2	이영희	7	P2	챗봇	4	O 7 \geq 4
E3	박민수	5	P1	홈페이지	2	O 5 \geq 2
E3	박민수	5	P2	챗봇	4	O 5 \geq 4

카디션 프로젝트 6행 중 조건 만족하는 5행만 남음 (김철수-챗봇 탈락)
 배정 여부와 무관하게 연차 자격만 판단 \rightarrow 이영희-홈페이지도 자격이 되니까 포함됨

동등 조인 예제: 직원번호 = 직원번호 → 2행

직원번호	이름	부서	연차	프로젝트 코드	프로젝트 명	직원번호	부서	최소 연차	일치?
E1	김철수	개발	3	P1	홈페이지	E1	개발	2	O E1=E1
E1	김철수	개발	3	P2	챗봇	E3	연구	4	X E1+E3
E2	이영희	연구	7	P1	홈페이지	E1	개발	2	X E2+E1
E2	이영희	연구	7	P2	챗봇	E3	연구	4	X E2+E3
E3	박민수	개발	5	P1	홈페이지	E1	개발	2	X E3+E1
E3	박민수	개발	5	P2	챗봇	E3	연구	4	O E3=E3

직원번호가 같은 2행만 남음 → 결과에 직원번호 열이 양쪽 다 나와서 9열 (중복)
박민수(개발)가 챗봇(연구)에 배정된 교차 배정도 직원번호만 맞으면 포함됨

자연 조인 예제 + 비교 요약 → 1행

동등 조인 2행에서 부서까지 매칭 검증:

직원번호	이름	직원 부서	프로젝트코드	프로젝트명	직원번호	프로젝트 부서	판정
E1	김철수	개발	P1	홈페이지	E1	개발	O 둘 다 일치
E3	박민수	개발	P2	챗봇	E3	연구	X 부서 불일치

▼ 자연 조인 최종 결과 (1행, 중복 열 합쳐서 7열)

직원번호	이름	부서	연차	프로젝트코드	프로젝트명	최소연차
E1	김철수	개발	3	P1	홈페이지	2

비교 요약

구분	조건	결과 행	결과 열
카티션 프로젝트	없음	6행	9열
세타 조인	연차 ≥ 최소연차	5행	9열
동등 조인	직원번호 = 직원번호	2행	9열
자연 조인	직원번호+부서 자동	1행	7열

관계 대수를 이용한 질의 표현

선택, 프로젝트, 조인을 조합하여 다양한 질의를 표현할 수 있습니다.

폐쇄 특성 덕분에 연산 결과에 또 연산을 적용할 수 있어요.

질의 예제에서 사용하는 릴레이션 데이터



그림 6-22 질의 표현에 사용할 예제 릴레이션들 : 고객과 주문 릴레이션

고객 릴레이션 4명, 주문 릴레이션 3건의 데이터입니다.
이 두 테이블을 가지고 여러 가지 질의를 표현해봅니다.

관계 대수 질의 예제 (1)

예제 6-5

등급이 gold인 고객의 이름과 나이를 검색하시오.

▶▶ $\pi_{\text{고객이름, 나이}}(\sigma_{\text{등급='gold'}}(\text{고객}))$

결과 릴레이션

고객이름	나이
김현준	20
원유선	28

선택, 프로젝트, 조인을 단계적으로 적용하여 원하는 결과를 도출합니다.
폐쇄 특성 덕분에 연산을 연쇄 적용할 수 있습니다.

관계 대수 질의 예제 (2)

예제 6-6

고객이름이 원유선인 고객의 등급과, 원유선 고객이 주문한 주문제품, 수량을 검색하시오.

▶▶ $\pi_{\text{등급, 주문제품, 수량}}(\sigma_{\text{고객이름='원유선'}}(\text{고객} \bowtie \text{주문}))$

결과 릴레이션

등급	주문제품	수량
gold	맛있는파이	5

다른 방식으로 같은 결과를 얻을 수도 있습니다.
연산 순서에 따라 중간 결과의 크기가 달라질 수 있습니다.

관계 대수 질의 예제 (3)

예제 6-7

주문수량이 10개 미만인 주문 내역을 제외하고 검색하시오.

▶▶ 주문 - ($\sigma_{\text{수량} < 10}(\text{주문})$)

결과 릴레이션

주문번호	주문고객	주문제품	수량
1001	apple	진짜우동	10
1003	banana	그대로만두	11

같은 질문이라도 관계 대수 표현 방법이 여러 가지일 수 있습니다.
이것이 나중에 '쿼리 최적화'의 출발점이 됩니다.

디비전 (Division) $R \div S$

S의 모든 튜플과 관련 있는 R의 튜플을 찾는 연산
 → '전부', '모든' 조건이 붙는 질의에 사용

고객 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
NULL	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
NULL	정지영	22	silver	학생	0

우수등급 릴레이션

등급
gold

디비전 연산

고객 ÷ 우수등급

고객아이디	고객이름	나이	직업	적립금
apple	김현준	20	학생	1000
carrot	원유선	28	교사	4500

그림 6-20 디비전 연산의 예 1 : 고객과 우수등급 릴레이션

디비전 (Division) $R \div S$

주문내역 릴레이션

주문고객	제품이름	제조업체
apple	진짜우동	한빛식품
carrot	맛있는파이	마포과자
banana	그대로만두	한빛식품
apple	그대로만두	한빛식품
carrot	그대로만두	한빛식품

제품1 릴레이션

제품이름
진짜우동
그대로만두

제품2 릴레이션

제품이름	제조업체
그대로만두	한빛식품

↓ 디비전 연산

주문내역 \div 제품1

주문고객	제조업체
apple	한빛식품

주문내역 \div 제품2

주문고객
banana
apple
carrot

그림 6-21 디비전 연산의 예 2 : 주문내역, 제품1, 제품2 릴레이션

세미 조인 (Semi Join)

자연 조인 후 한쪽 릴레이션의 속성만 남기는 연산
분산 데이터베이스에서 전송 데이터양을 줄이기 위해 사용

세미 조인 예제



자연 조인 결과에서 한쪽 릴레이션의 속성만 프로젝트한 것과 같은 결과가 나옵니다.

외부 조인 (Outer Join)

자연 조인에서 제외되는 튜플도 결과에 포함 (매칭 안 되는 부분은 NULL)

왼쪽 외부 조인 (LEFT)

왼쪽 릴레이션의 모든 튜플을 결과에 포함

표현법 : 릴레이션1 \bowtie 릴레이션2

오른쪽 외부 조인 (RIGHT)

오른쪽 릴레이션의 모든 튜플을 결과에 포함

표현법 : 릴레이션1 \bowtie 릴레이션2

완전 외부 조인 (FULL)

양쪽 릴레이션의 모든 튜플을 결과에 포함

표현법 : 릴레이션1 \bowtie 릴레이션2

외부 조인 예제: LEFT / RIGHT / FULL

고객 릴레이션 **릴레이션1**

고객아이디	고객이름	나이
apple	김현준	20
banana	정소화	25
carrot	원유선	28
orange	정지영	22

주문 릴레이션 **릴레이션2**

주문번호	고객아이디	주문제품
1001	apple	진짜우동
1002	carrot	맛있는파이
1003	banana	그대로만두

고객 >< 주문 **left 외부 조인**

고객아이디	고객이름	나이	주문번호	주문제품
apple	김현준	20	1001	진짜우동
banana	정소화	25	1003	그대로만두
carrot	원유선	28	1002	맛있는파이
orange	정지영	22	NULL	NULL

고객 <> 주문 **right 외부 조인**

고객아이디	고객이름	나이	주문번호	주문제품
apple	김현준	20	1001	진짜우동
banana	정소화	25	1003	그대로만두
carrot	원유선	28	1002	맛있는파이

고객 >< 주문 **full 외부 조인**

고객아이디	고객이름	나이	주문번호	주문제품
apple	김현준	20	1001	진짜우동
banana	정소화	25	1003	그대로만두
carrot	원유선	28	1002	맛있는파이
orange	정지영	22	NULL	NULL

왼쪽 외부 조인: 왼쪽의 모든 튜플 포함 (매칭 안 되면 NULL)

오른쪽 외부 조인: 오른쪽의 모든 튜플 포함

완전 외부 조인: 양쪽 모두 포함

요약

관계 데이터 연산

릴레이션에서 원하는 데이터를 추출·가공하는 방법

일반 집합 연산자

합집합(\cup), 교집합(\cap), 차집합($-$), 카티션 프로덕트(\times)

순수 관계 연산자

선택(σ), 프로젝트(π), 조인(\bowtie), 디비전(\div)

확장 연산자

세미 조인, 외부 조인 (LEFT/RIGHT/FULL)

요약 – 관계 대수 전체 구조

