

데이터베이스관리

10주차

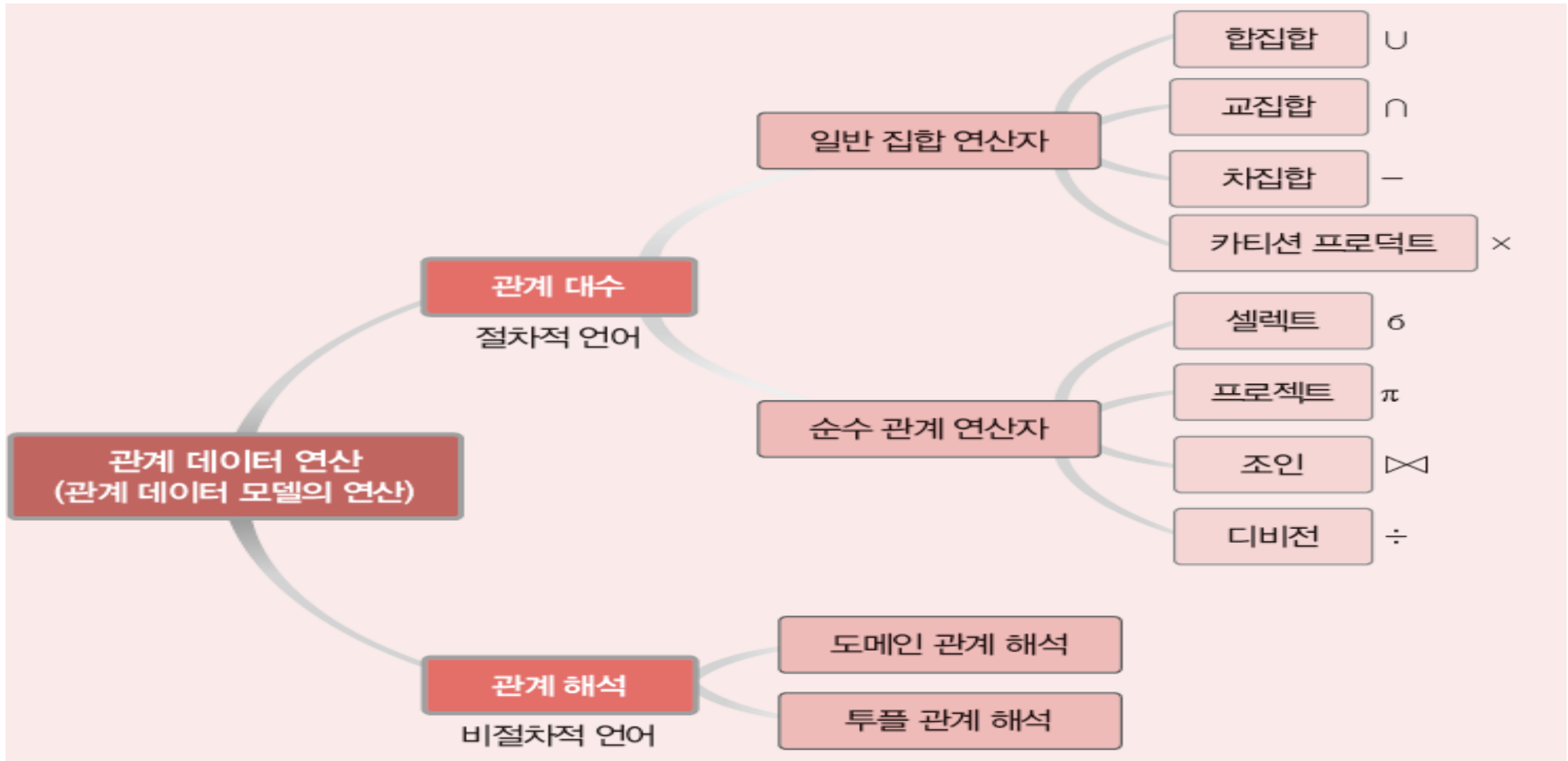
담당교수: 김희숙
(jasmin11@hanmail.net)

관계대수-01

10주차 10-01

담당교수: 김희숙
(jasmin11@hanmail.net)

관계대수



절차적 언어 vs. 비절차적 언어



* **절차적 언어**(procedural language) :

원하는 결과를 얻기 위해 수행되어야 할 일련의 절차를 명시(**how**)

* **비절차적 언어**(non procedural language) :

수행절차를 기술하지 않고 사용자가 원하는 결과만을 명시(**what**)

02 관계 대수

• 일반 집합 연산자(set operation)

- 릴레이션이 튜플의 집합이라는 개념을 이용하는 연산자

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 만들어진 새로운 튜플을 반환

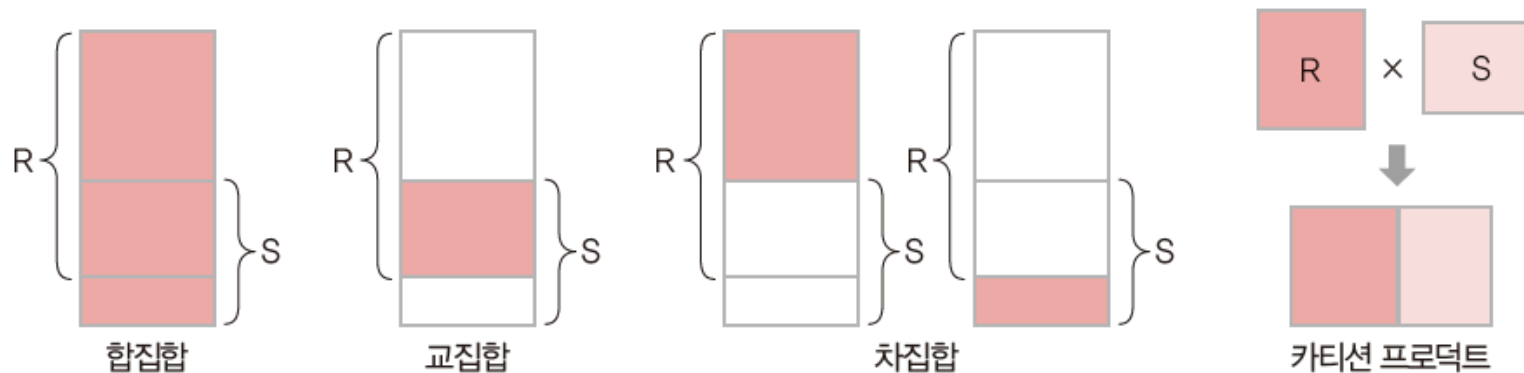


그림 6-4 일반 집합 연산자의 종류와 기능

[예제] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

$$A = \{1, 3, 6\}$$

$$B = \{2, 3, 7\}$$

$$A \cup B = \{1, 2, 3, 6, 7\}$$

$$A \cap B = \{3\}$$

$$A - B = \{1, 6\}$$

* 집합

1. 합집합 UNION
2. 교집합 INTERSECT
3. 차집합 MINUS (Oracle)
EXCEPT (MS SQL)

02 관계 대수

일반 집합 연산자의 특성

- 피연산자가 2개 필요함
 - 2개의 릴레이션을 대상으로 연산을 수행
- 합집합, 교집합, 차집합은 피연산자인 두 릴레이션이 합병 가능해야 함
 - 합병 가능(union-compatible) 조건
 - 두 릴레이션의 **차수가 같아야** 함
 - 두 릴레이션에서 서로 대응되는 속성의 **도메인이 같아야** 함

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



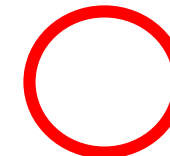
고객 릴레이션

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

직원 릴레이션

직원번호	직원이름	나이
INT	CHAR(20)	INT
10	김용욱	40
20	채광주	32
30	김수진	28

그림 6-7 합병이 가능한 예



02 관계 대수

R

번호	이름
100	정소화
200	김선우
300	고명석

S

번호	이름
100	정소화
101	채광주
102	김수진

합집합 연산

RUS	번호	이름
	100	정소화
	200	김선우
	300	고명석
	101	채광주
	102	김수진

그림 6-8 합집합 연산의 예

R

번호	이름
100	정소화
200	김선우
300	고명석

S

번호	이름
100	정소화
101	채광주
102	김수진

교집합 연산

R∩S	번호	이름
	100	정소화

그림 6-9 교집합 연산의 예

02 관계 대수

R

번호	이름
100	정소화
200	김선우
300	고명석

S

번호	이름
100	정소화
101	채광주
102	김수진

차집합 연산

R-S

번호	이름
200	김선우
300	고명석

S-R

번호	이름
101	채광주
102	김수진

그림 6-10 차집합 연산의 예

[예제] 합집합, 교집합, 차집합

[실습01] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

R1

A	B	C
a1	b1	c1
a2	b3	c3
a3	b4	c2

R2

A	B	C
a1	b1	c1
a2	b3	c3
a3	b3	c1

* 집합

1. 합집합 UNION
2. 교집합 INTERSECT
3. 차집합 MINUS (Oracle)
EXCEPT (MS SQL)

합집합 (U)	$R1 \cup R2$	<table> <tr> <th>A</th><th>B</th><th>C</th></tr> <tr> <td>a1</td><td>b1</td><td>c1</td></tr> <tr> <td>a2</td><td>b3</td><td>c3</td></tr> <tr> <td>a3</td><td>b4</td><td>c2</td></tr> <tr> <td>a3</td><td>b3</td><td>c1</td></tr> </table>	A	B	C	a1	b1	c1	a2	b3	c3	a3	b4	c2	a3	b3	c1
A	B	C															
a1	b1	c1															
a2	b3	c3															
a3	b4	c2															
a3	b3	c1															
차집합 (-)	$R1 - R2$	<table> <tr> <th>A</th><th>B</th><th>C</th></tr> <tr> <td>a3</td><td>b4</td><td>c2</td></tr> </table>	A	B	C	a3	b4	c2									
A	B	C															
a3	b4	c2															

[실습02] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라 (member_set.sql)

정회원

	번호	이름	주민번호	휴대폰번호	이메일	등록일
▶	1	홍길동	820416-1234567	(011) 123-1231	gdhong@hitel.net	20070302
	2	임꺽정	830507-2345678	(010) 122-1222	jung@hanmail.net	20050422

준회원

	번호	이름	주민번호	휴대폰번호	이메일	등록일
▶	1	홍길동	820416-1234567	(011) 123-1231	gdhong@hitel.net	20070302
	3	박찬호	850321-1456789	(010) 133-1231	chpark@hanmail.net	20090512
	4	선동열	761122-1889911	(010) 144-1222	sun@naver.com	20080605

-- [Quiz] 정회원과 준회원을 **UNION** 연산자를 사용한 것과, **UNION ALL** 연산자를 사용한 것의 차이점을 확인하라

* 집합

1. 합집합 **UNION**
2. 교집합 **INTERSECT**
3. 차집합 **MINUS (Oracle)**
EXCEPT (MS SQL)

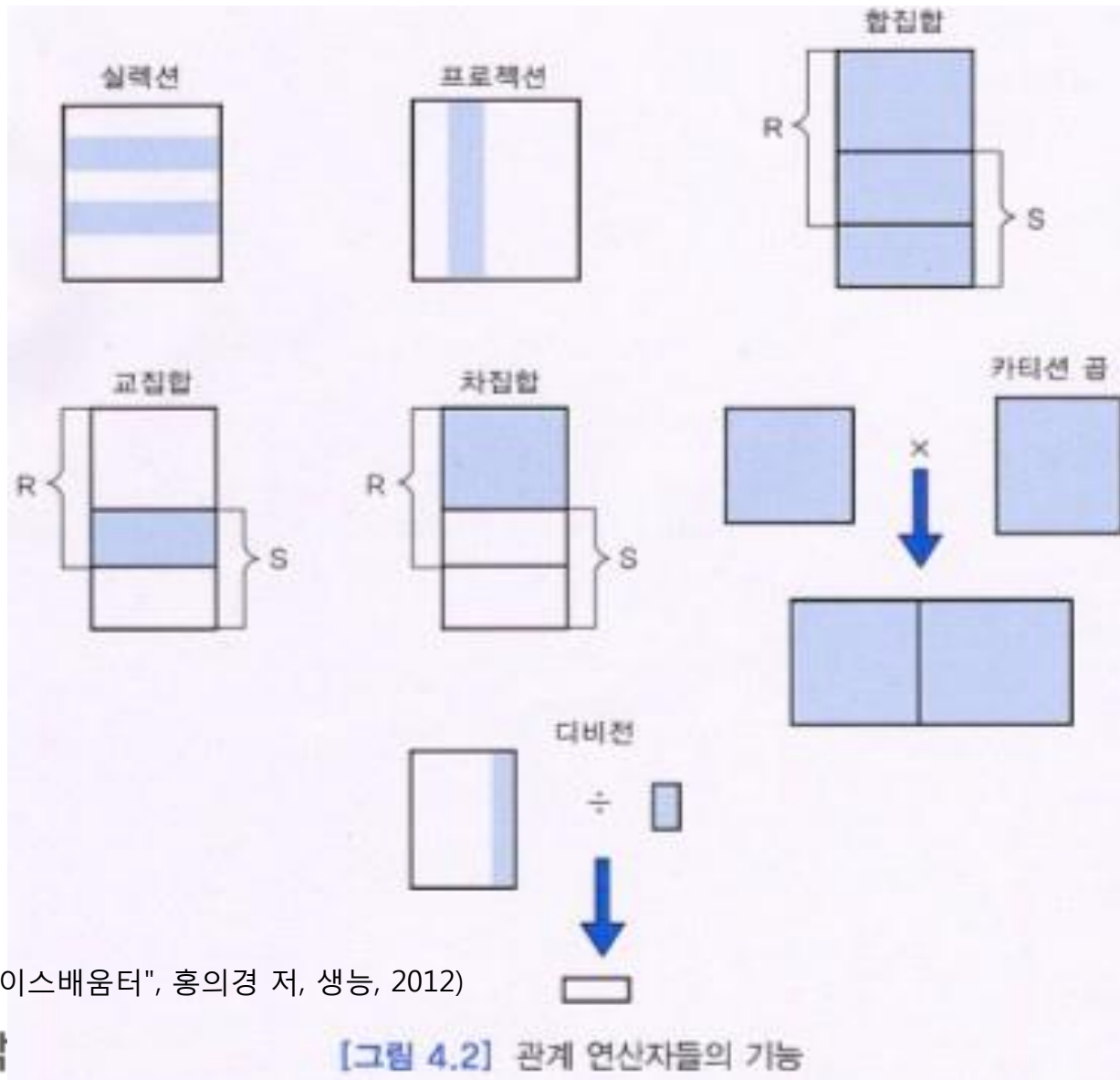
-- 합집합

```
select 번호, 이름 from 정회원
UNION
select 번호, 이름 from 준회원;
```

-- 합집합(중복 포함)

```
select 번호, 이름 from 정회원
UNION ALL
select 번호, 이름 from 준회원;
```

관계대수



* 집합

1. 합집합 UNION
2. 교집합 INTERSECT
3. 차집합 MINUS (Oracle)
EXCEPT (MS SQL)

(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)

[요약] 관계대수



❖ 관계대수: 수학 수식구조와 거의 동일, 관계대수의 실행결과는 새로운 테이블이 된다

1. 실렉션(selection) : 선택연산(시그마) σ

SELECT

FROM

WHERE

2. 프로젝션(projection) : 추출연산(파이) π

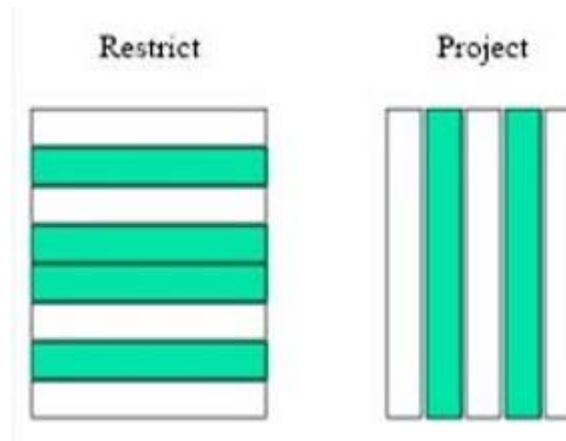
SELECT

FROM

3. 합집합(union)

교집합(intersect)

차집합(difference)



[실습01] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

R1

A	B	C
a1	b1	c1
a2	b3	c3
a3	b4	c2

R2

A	B	C
a1	b1	c1
a2	b3	c3
a3	b3	c1

* 관계대수

1. 선택(selection)
2. 투영(projection)

주요 연산자	사용 예	결과									
선택 (σ)	$\sigma_{A=a1 \text{ or } A=a2} (R1)$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a2</td><td>b3</td><td>c3</td></tr></table>	A	B	C	a1	b1	c1	a2	b3	c3
A	B	C									
a1	b1	c1									
a2	b3	c3									
프로젝션 (π)	$\pi_{A,B} (R2)$	<table><tr><th>A</th><th>B</th></tr><tr><td>a1</td><td>b1</td></tr><tr><td>a2</td><td>b3</td></tr><tr><td>a3</td><td>b3</td></tr></table>	A	B	a1	b1	a2	b3	a3	b3	
A	B										
a1	b1										
a2	b3										
a3	b3										

$$\sigma_{A=a1 \text{ or } A=a2}(R1)$$

$$\pi_{A,B}(R2)$$

[실습03] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

(quiz-madang_book-k-mysql)

질의 2-1 마당서점에서 판매하는 도서 중 8,000원 이하인 도서를 검색하시오.

$\sigma_{\text{가격} \leq 8000}$ (도서)

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

$\sigma_{\text{가격} \leq 8000}$ (도서)

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000

* 실렉션(selection) σ

SELECT

FROM

WHERE

-- 가격이 8,000원 이하인 도서 검색

SELECT *

FROM 도서

WHERE 가격 <= 8000;

그림 2-18 셀렉션의 예

[실습03] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

(quiz-madang_book-k-mysql)

질의 2-2 고객 릴레이션에서 이름, 주소, 핸드폰을 검색하시오.

$\pi_{\text{이름, 주소, 핸드폰}}(\text{고객})$

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001
2	김연아	900905-2222222	대한민국 서울	000-6000-0001
3	장미란	831009-2333333	대한민국 강원도	000-7000-0001
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001

$\pi_{\text{이름, 주소, 핸드폰}}(\text{고객})$

이름	주소	핸드폰
박지성	영국 맨체스타	000-5000-0001
김연아	대한민국 서울	000-6000-0001
장미란	대한민국 강원도	000-7000-0001
추신수	미국 클리블랜드	000-8000-0001

* 프로젝션(projection) π

SELECT

FROM

-- 이름, 주소, 핸드폰을 검색

SELECT 이름, 주소, 핸드폰

FROM 도서

그림 2-19 프로젝션의 예

관계대수-02

10주차 10-02

담당교수: 김희숙
(jasmin11@hanmail.net)

[실습04] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

student

stu_id	resident_id	name	year	address	dept_id
1292001	900424-1825409	김광식	3	서울	920
1292002	900305-1730021	김정현	3	서울	920
1292003	891021-2308302	김현정	4	대전	920
1292301	890902-2704012	김현정	2	대구	923
1292303	910715-1524390	박광수	3	광주	923
1292305	921011-1809003	김우주	4	부산	923
1292501	900825-1506390	박철수	3	대전	925
1292502	911011-1809003	백태성	3	서울	925

professor

prof_id	resident_id	name	dept_id	position	year_emp
92001	590327-1839240	이태규	920	교수	1997
92002	690702-1350026	고희석	920	부교수	2003
92301	741011-2765501	최성희	923	부교수	2005
92302	750728-1102458	김태석	923	교수	1999
92501	620505-1400546	박철재	925	조교수	2007
92502	740101-1830264	장민석	925	부교수	2005

1. 실렉션(Selection): 선택 연산

질의1) student 테이블에서 address 가 서울인 레코드 검색하라

질의2) 2000년 이후에 임용된 교수들에 대한 레코드 검색하라

질의3) 2000년 이후에 임용된 부교수들의 레코드를 검색하라

[실습04] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

student

stu_id	resident_id	name	year	address	dept_id
1292001	900424-1825409	김광식	3	서울	920
1292002	900305-1730021	김정현	3	서울	920
1292003	891021-2308302	김현정	4	대전	920
1292301	890902-2704012	김현정	2	대구	923
1292303	910715-1524390	박광수	3	광주	923
1292305	921011-1809003	김우주	4	부산	923
1292501	900825-1506390	박철수	3	대전	925
1292502	911011-1809003	백태성	3	서울	925

professor

prof_id	resident_id	name	dept_id	position	year_emp
92001	590327-1839240	이태규	920	교수	1997
92002	690702-1350026	고희석	920	부교수	2003
92301	741011-2765501	최성희	923	부교수	2005
92302	750728-1102458	김태석	923	교수	1999
92501	620505-1400546	박철재	925	조교수	2007
92502	740101-1830264	장민석	925	부교수	2005

2. 프로젝션(Projection): 추출 연산

질의4) student 테이블에서 stu_id 와 name 를 검색하라

질의5) student 테이블에서 dept_id 를 검색하라

질의8) professor 테이블에서 name, position 을 검색하라

[실습04] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

student

stu_id	resident_id	name	year	address	dept_id
1292001	900424-1825409	김광식	3	서울	920
1292002	900305-1730021	김정현	3	서울	920
1292003	891021-2308302	김현정	4	대전	920
1292301	890902-2704012	김현정	2	대구	923
1292303	910715-1524390	박광수	3	광주	923
1292305	921011-1809003	김우주	4	부산	923
1292501	900825-1506390	박철수	3	대전	925
1292502	911011-1809003	백태성	3	서울	925

professor

prof_id	resident_id	name	dept_id	position	year_emp
92001	590327-1839240	이태규	920	교수	1997
92002	690702-1350026	고희석	920	부교수	2003
92301	741011-2765501	최성희	923	부교수	2005
92302	750728-1102458	김태석	923	교수	1999
92501	620505-1400546	박철재	925	조교수	2007
92502	740101-1830264	장민석	925	부교수	2005

3. 실렉션과 프로젝션의 혼합:

질의10) 2000년 이후에 임용된 부교수 의 이름을 검색하라

관계대수

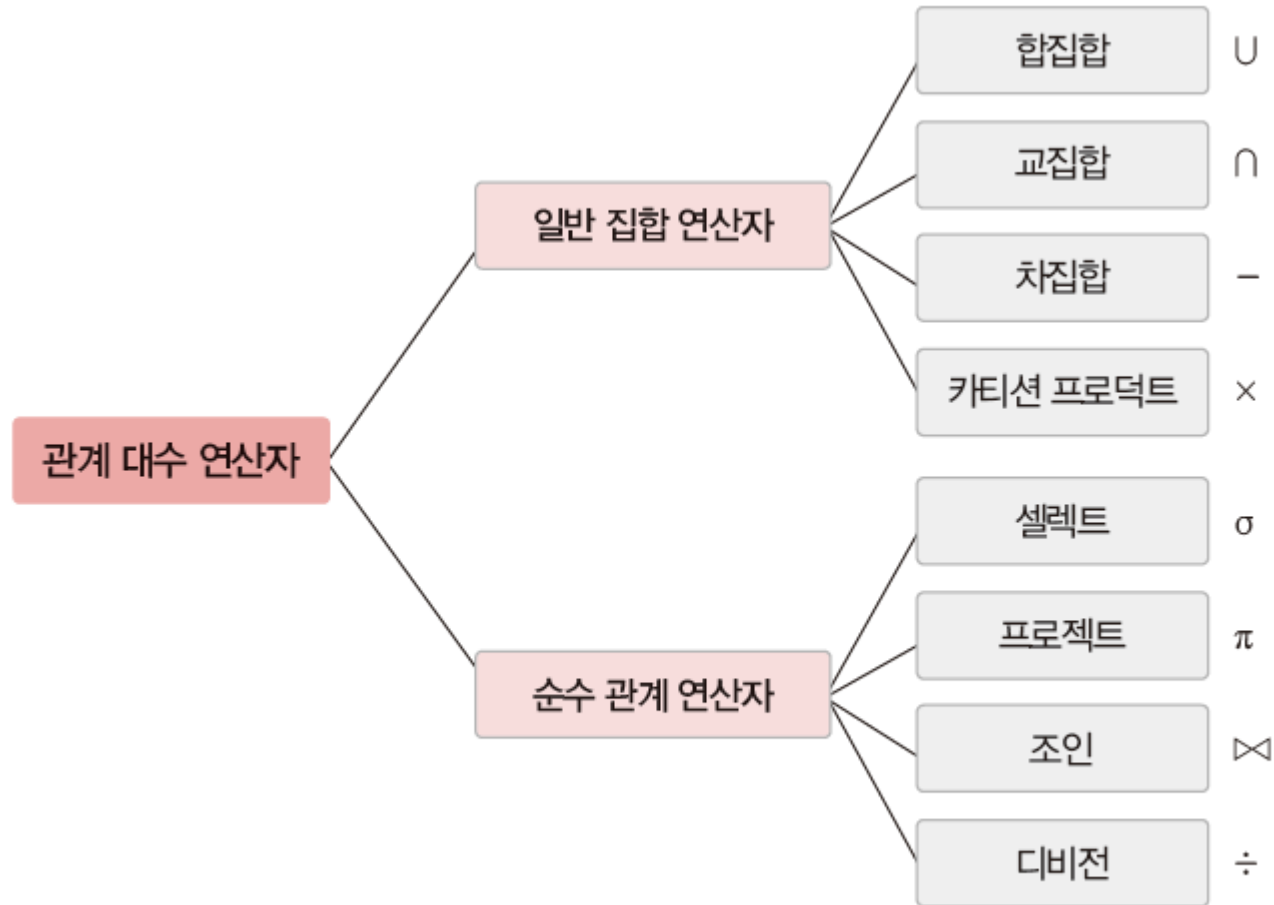


그림 6-3 관계 대수 연산자의 종류



[실습05] (테이블 3개) 고객, 도서, 주문 테이블 생성(기본키, 외래키)

❖ 기본키가 복합키인 경우

(quiz-madang_book-k-mysql)

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810101-1111111	영국 맨체스터	000-5000-0001
2	김연아	900101-2222222	대한민국 서울	000-6000-0001
3	장미란	830101-2333333	대한민국 강원도	000-7000-0001
4	추신수	820101-1444444	미국 클리블랜드	000-8000-0001

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

주문

고객번호	도서번호	판매가격	주문일자
1	1	7000	2013-07-01
1	2	13000	2013-07-03
2	5	8000	2013-07-03
3	2	13000	2013-07-04
4	4	35000	2013-07-05
1	3	22000	2013-07-07
4	3	22000	2013-07-07

/* 한빛 DB (박우창, 남송희)

```
drop table 고객;
drop table 도서;
drop table 주문;
```

```
--고객 (고객번호, 이름, 주민번호, 주소, 핸드폰)
--도서 (도서번호, 도서이름, 출판사, 가격)
--주문 (고객번호, 도서번호, 판매가격, 주문일자)
```

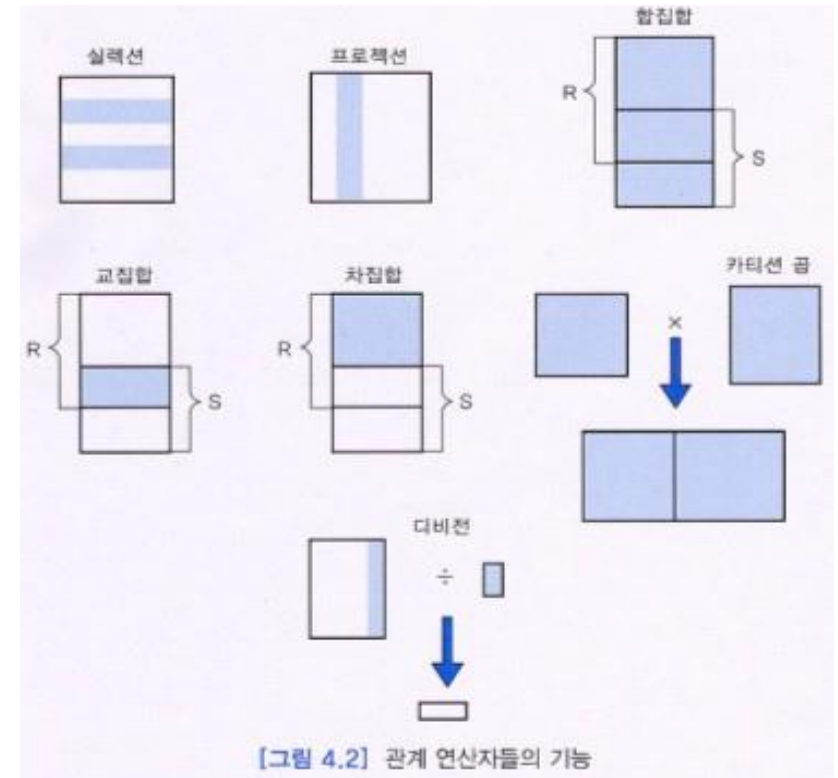
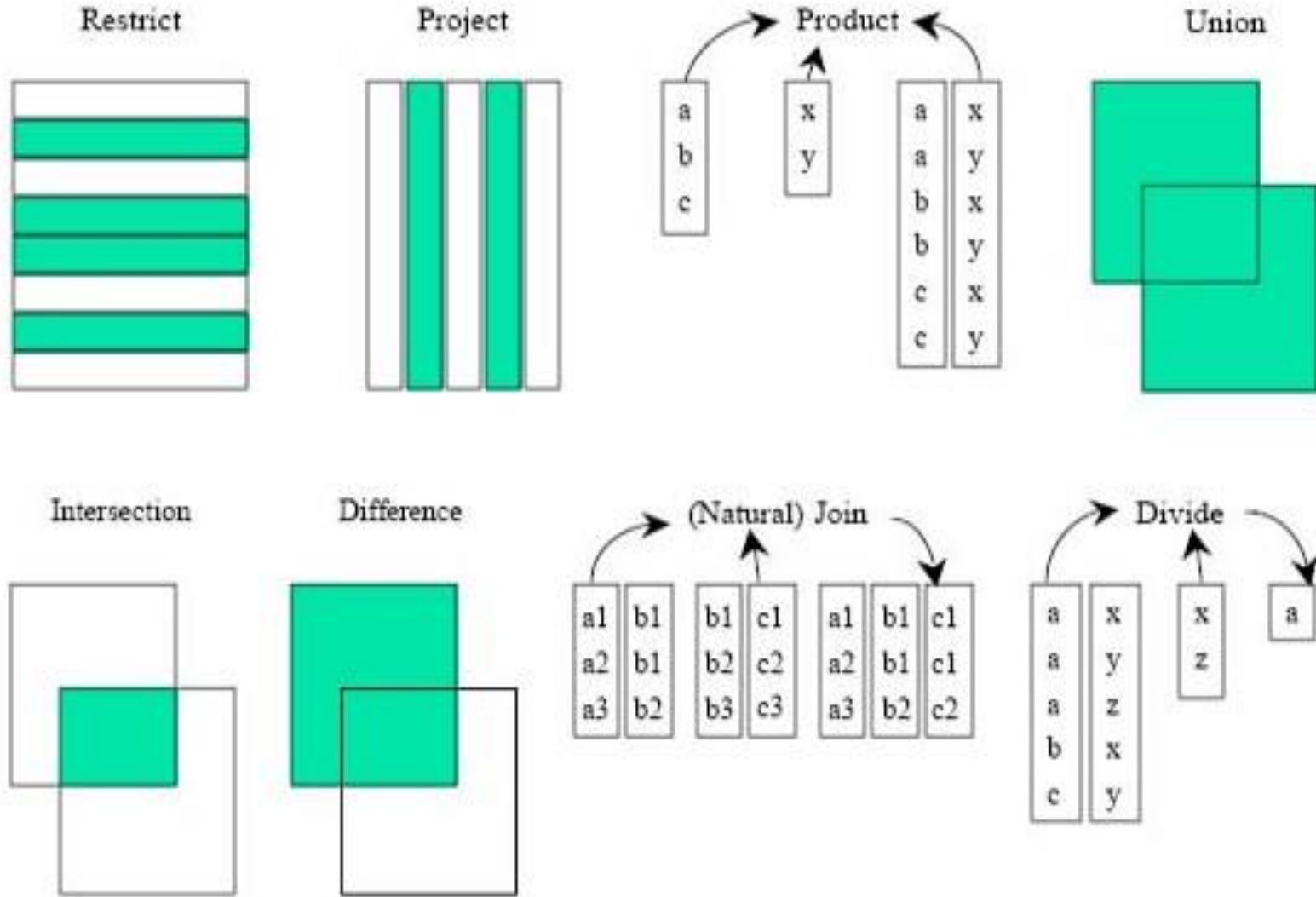
```
CREATE TABLE 고객 (
  고객번호 INT PRIMARY KEY,
  이름 VARCHAR(40),
  주민번호 CHAR(14),
  주소 VARCHAR(40),
  핸드폰 VARCHAR(30)
);
```

```
CREATE TABLE 도서 (
  도서번호 INT PRIMARY KEY,
  도서이름 VARCHAR(40),
  출판사 VARCHAR(40),
  가격 INT
);
```

```
CREATE TABLE 주문 (
  고객번호 INT NOT NULL,
  도서번호 INT NOT NULL,
  판매가격 INT,
  주문일자 DATE,
  PRIMARY KEY (고객번호, 도서번호),
  FOREIGN KEY (고객번호) REFERENCES 고객 (고객번호),
  FOREIGN KEY (도서번호) REFERENCES 도서 (도서번호)
);
```

그림 2

관계대수



02 관계 대수

- 순수 관계 연산자(relational operation)
 - 릴레이션의 구조와 특성을 이용하는 연산자

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

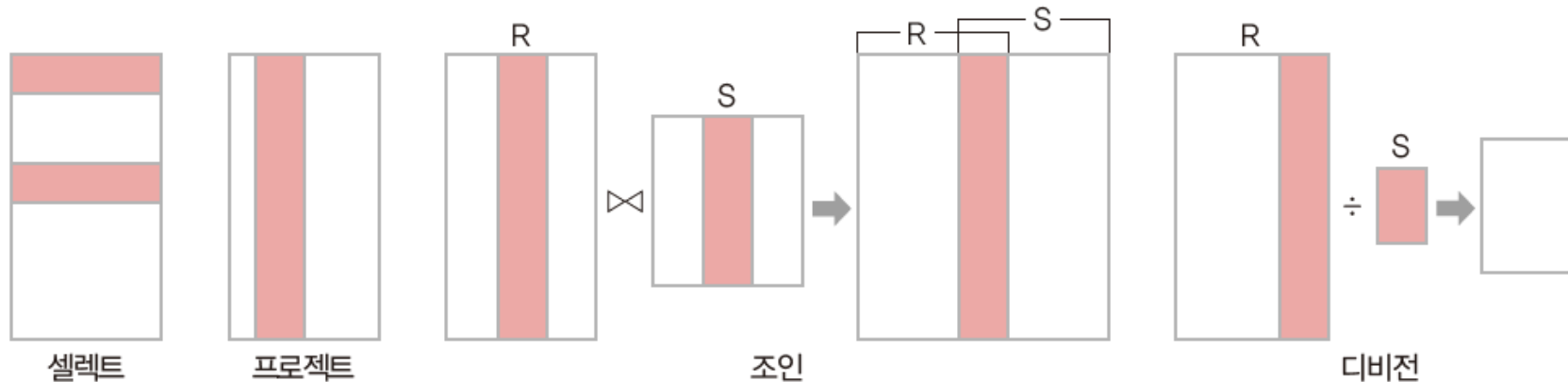
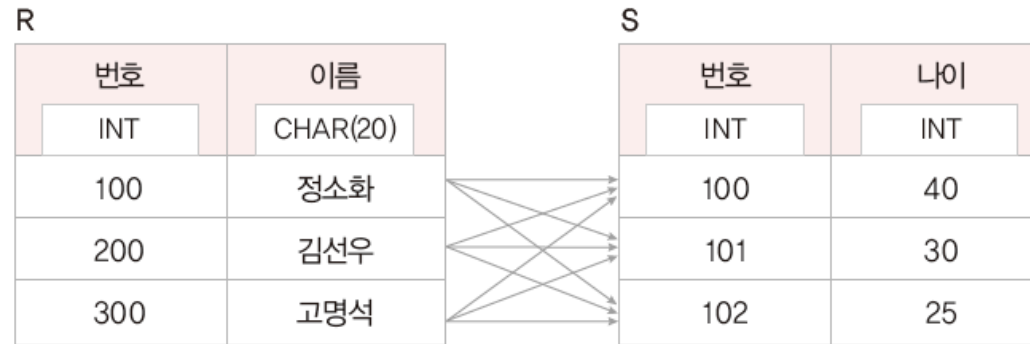


그림 6-5 순수 관계 연산자의 종류와 기능

02 관계 대수



카티션 프로덕트 연산

R × S

R.번호	R.이름	S.번호	S.나이
INT	CHAR(20)	INT	INT
100	정소화	100	40
100	정소화	101	30
100	정소화	102	25
200	김선우	100	40
200	김선우	101	30
200	김선우	102	25
300	고명석	100	40
300	고명석	101	30
300	고명석	102	25

카티션 프로덕트
(cartesian product)

$R \times S$

[요약] 관계대수



❖ 관계대수: 수학 수식구조와 거의 동일, 관계대수의 실행결과는 새로운 테이블이 된다

4. 카티션 프로덕트(product)

5. **조인** : 두 개 이상의 테이블에서
원하는 결과를 검색
(공통 필드)

- 1) **내부조인**: 두 개의 릴레이션에서
일치하는 행을 검색
- 2) **외부조인**: 두 개의 릴레이션에서
일치하지 않는 행도 검색

조인 연산의 구분

기본연산 : 세타조인(\bowtie_θ), 동등조인(\bowtie), 자연조인(\bowtie_N)

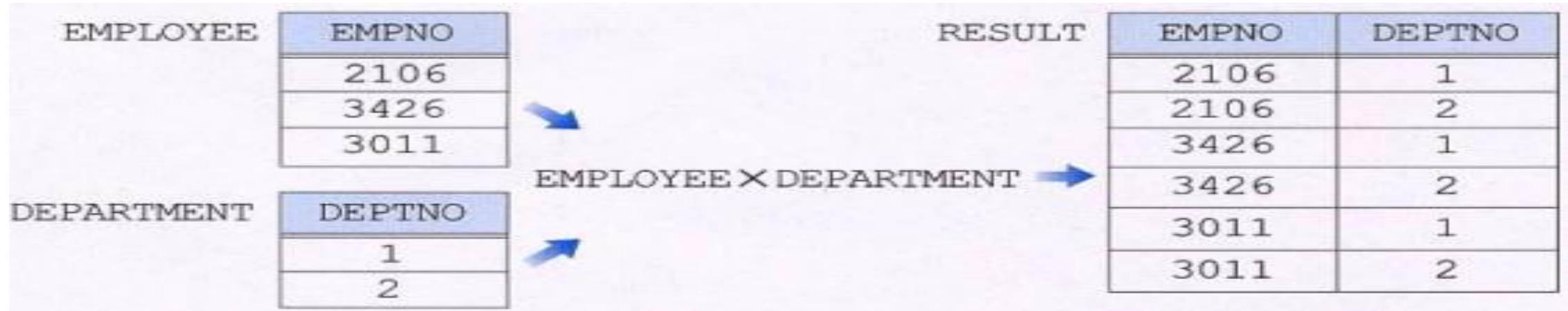
확장된 조인 연산 : 세미조인(\ltimes, \rtimes), 외부조인($\ltimes, \rtimes, \bowtie$)

내부조인

(theta join, equi join, natural join)

외부조인

(left outer join, right outer join, full outer join)



EMPLOYEE X DEPARTMENT

[실습01] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

R1

A	B	C
a1	b1	c1
a2	b3	c3
a3	b4	c2

R2

A	B	C
a1	b1	c1
a2	b3	c3
a3	b3	c1

❖ 조인 (join)

- 동등조인(equi join)
- 자연조인(natural join)

조인 (\bowtie)	$R1 \bowtie_{R1.C=R2.C} R2$						
		R1.A	R1.B	R1.C	R2.A	R2.B	R2.C
		a1	b1	c1	a1	b1	c1
		a1	b1	c1	a3	b3	c1
		a2	b3	c3	a2	b3	c3

$R1 \bowtie_{R1.C=R2.C} R2$

[실습06] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

EMPLOYEE	EMPNO	EMPNAME	DNO
	2106	김창섭	2
	3426	박영권	1
	3011	이수민	3
	1003	조민희	2
	3427	최종철	3

DEPARTMENT	DEPTNO	DEPTNAME
	1	영업
	2	기획
	3	개발
	4	총무

EMPLOYEE ⋈_{DNO=DEPTNO} DEPARTMENT

RESULT	EMPNO	EMPNAME	DNO	DEPTNO	DEPTNAME
	2106	김창섭	2	2	기획
	3426	박영권	1	1	영업
	3011	이수민	3	3	개발
	1003	조민희	2	2	기획
	3427	최종철	3	3	개발

❖ 조인 (join)

- 동등조인(equi join)
- 자연조인(natural join)

RESULT	EMPNO	EMPNAME	DNO	DEPTNAME
	2106	김창섭	2	기획
	3426	박영권	1	영업
	3011	이수민	3	개발
	1003	조민희	2	기획
	3427	최종철	3	개발

EMPLOYEE ⋈_{DNO=DEPTNO} DEPARTMENT

[Quiz 01] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

R1	A	B	S1	A	C	D	다음 결과는? 1. R1 X S1 2. R1 ⋈ S1 3. R1 ⋈ _N S1 4. R1 ⋈ _{A >= C} S1
	1	4		1	3	4	
	2	5		2	2	6	
				3	1	9	

-- 1. 카티션 프로덕트

select *

from R1, S1;

-- 2. 동등조인(equi join)

select *

from R1, S1

where R1.A = S1.A;

[Quiz 01] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

R1	<table><tr><th>A</th><th>B</th></tr><tr><td>1</td><td>4</td></tr><tr><td>2</td><td>5</td></tr></table>	A	B	1	4	2	5	S1	<table><tr><th>A</th><th>C</th><th>D</th></tr><tr><td>1</td><td>3</td><td>4</td></tr><tr><td>2</td><td>2</td><td>6</td></tr><tr><td>3</td><td>1</td><td>9</td></tr></table>	A	C	D	1	3	4	2	2	6	3	1	9	다음 결과는?
A	B																					
1	4																					
2	5																					
A	C	D																				
1	3	4																				
2	2	6																				
3	1	9																				
				1. R1 X S1																		
				2. R1 ⋈ S1																		
				3. R1 ⋈ _N S1																		
				4. R1 ⋈ _{A ≥ C} S1																		

-- 2. 동등조인(equi join)

select *

from R1, S1

where R1.A = S1.A;

-- 3. 자연조인(natural join)

select R1.A, B, C, D

from R1, S1

where R1.A = S1.A;

[Quiz 01] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

R1

A	B
1	4
2	5

S1

A	C	D
1	3	4
2	2	6
3	1	9

다음 결과는?

1. R1 X S1

2. R1 ⋈ S1

3. R1 ⋈_N S1

4. R1 ⋈_{A >= C} S1

-- 관계 연산자

>, >=, =, !=, <=, <

-- 4. 세타조인(theta join)

select *

from R1, S1

where R1.A >= C;

[실습03] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

질의 2-7 고객과 고객의 주문 사항을 모두 보이시오.

고객 ⋈ 고객.고객번호=주문.고객번호 주문

고객

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	2	1	7000	2013-07-01
2	1	2	13000	2013-07-03
3	2	5	8000	2013-07-03
4	1	2	13000	2013-07-04
5	4	4	35000	2013-07-05
6	5	3	22000	2013-07-07
7	4	3	22000	2013-07-07

고객 ⋈ 고객.고객번호=주문.고객번호 주문

고객번호	이름	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	영국 맨체스터	000-5000-0001	2	1	2	13000	2013-07-03
1	박지성	영국 맨체스터	000-5000-0001	4	1	2	13000	2013-07-04
2	김연아	대한민국 서울	000-6000-0001	1	2	1	7000	2013-07-01
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2013-07-03
4	추신수	미국 클리블랜드	000-8000-0001	5	4	4	35000	2013-07-05
4	추신수	미국 클리블랜드	000-8000-0001	7	4	3	22000	2013-07-07

그림 2-24 동등조인의 예

❖ 조인 (join)

- 동등조인(equi join)
- 자연조인(natural join)

[실습03] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

질의 2-8 고객과 고객의 주문 사항을 모두 보이시오.

고객 $\bowtie_{N(\text{고객.고객번호}=\text{주문.고객번호})}$ 주문

고객

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	2	1	7000	2013-07-01
2	1	2	13000	2013-07-03
3	2	5	8000	2013-07-03
4	1	2	13000	2013-07-04
5	4	4	35000	2013-07-05
6	5	3	22000	2013-07-07
7	4	3	22000	2013-07-07

고객 $\bowtie_{N(\text{고객.고객번호}=\text{주문.고객번호})}$ 주문

고객번호	이름	주소	핸드폰	주문번호	도서번호	판매가격	주문일자
1	박지성	영국 맨체스터	000-5000-0001	2	2	13000	2013-07-03
1	박지성	영국 맨체스터	000-5000-0001	4	2	13000	2013-07-04
2	김연아	대한민국 서울	000-6000-0001	1	1	7000	2013-07-01
2	김연아	대한민국 서울	000-6000-0001	3	5	8000	2013-07-03
4	추신수	미국 클리블랜드	000-8000-0001	5	4	35000	2013-07-05
4	추신수	미국 클리블랜드	000-8000-0001	7	3	22000	2013-07-07

그림 2-25 자연조인의 예

❖ 조인 (join)

- 동등조인(equi join)
- 자연조인(natural join)

[요약] 관계대수



❖ 관계대수: 수학 수식구조와 거의 동일, 관계대수의 실행결과는 새로운 테이블이 된다

4. 카티션 프로덕트(product)

5. **조인** : 두 개 이상의 테이블에서
원하는 결과를 검색
(공통 필드)

- 1) **내부조인**: 두 개의 릴레이션에서
일치하는 행을 검색
- 2) **외부조인**: 두 개의 릴레이션에서
일치하지 않는 행도 검색

조인 연산의 구분

기본연산 : 세타조인(\bowtie_{θ}), 동등조인(\bowtie), 자연조인(\bowtie_N)

확장된 조인 연산 : 세미조인(\ltimes, \rtimes), 외부조인($\ltimes, \rtimes, \bowtie$)

내부조인

(theta join, equi join, natural join)

외부조인

(left outer join, right outer join, full outer join)

왼쪽(left) 외부조인 - $R \bowtie_{(r, s)} S$

완전(full) 외부조인 - $R \bowtie_{(r, s)} S$

오른쪽(right) 외부조인 - $R \bowtie_{(r, s)} S$

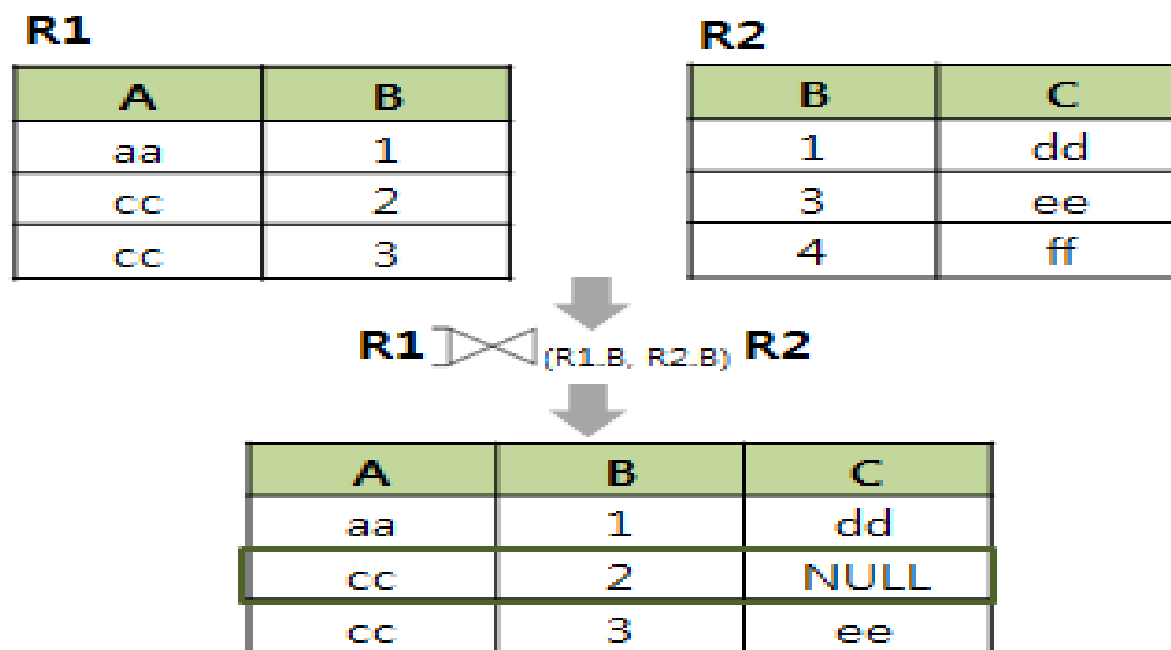


그림 2-26 왼쪽 외부조인의 예

관계대수-03

10주차 10-03

담당교수: 김희숙
(jasmin11@hanmail.net)

[Quiz 02] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

관계대수 / SQL

freshman

name	address
김광식	서울
김현정	대전
조영수	대전

dmember

name	dept_name
김광식	컴퓨터공학과
김현정	산업공학과
이진영	전자공학과

- 1-1) 관계대수
- 1-2) SQL
- 1-3) 실행결과

- 1. 카티션 프로덕트(product)
- 2. 동등조인(equi join)
- 3. 자연조인(natural join)
- 4. 왼쪽외부조인(left outer join)
- 5. 오른쪽외부조인(right outer join)
- 6. 완전외부조인(full outer join)

-- 1. 카티션 프로덕트

```
select *
from freshman, dmember;
```

[Quiz 02] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

관계대수 / SQL

freshman

name	address
김광식	서울
김현정	대전
조영수	대전

dmember

name	dept_name
김광식	컴퓨터공학과
김현정	산업공학과
이진영	전자공학과

- 1-1) 관계대수
- 1-2) SQL
- 1-3) 실행결과

- 1. 카티션 프로덕트(product)
- 2. 동등조인(equi join)
- 3. 자연조인(natural join)
- 4. 왼쪽외부조인(left outer join)
- 5. 오른쪽외부조인(right outer join)
- 6. 완전외부조인(full outer join)

-- 1. 카티션 프로덕트

```
select *
from freshman, dmember;
```

-- 2. 동등조인(equi join)

```
select *
from freshman, dmember
where freshman.name=dmember.name;
```

-- 2. 동등조인(equi join)

```
select *
from freshman INNER JOIN dmember
on freshman.name=dmember.name;
```

[Quiz 02] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

관계대수 / SQL

freshman

name	address
김광식	서울
김현정	대전
조영수	대전

dmember

name	dept_name
김광식	컴퓨터공학과
김현정	산업공학과
이진영	전자공학과

1-1) 관계대수

1-2) SQL

1-3) 실행결과

1. 카티션 프로덕트(product)

2. 동등조인(equi join)

3. 자연조인(natural join)

4. 왼쪽외부조인(left outer join)

5. 오른쪽외부조인(right outer join)

6. 완전외부조인(full outer join)

-- 3. 자연조인(natural join)

```
select freshman.name, address, dept_name
from freshman, dmember
where freshman.name=dmember.name;
```

-- 3. 자연조인(natural join)

```
select freshman.name, address, dept_name
from freshman INNER JOIN dmember
on freshman.name=dmember.name;
```


[Quiz 02] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

관계대수 / SQL

freshman

name	address
김광식	서울
김현정	대전
조영수	대전

dmember

name	dept_name
김광식	컴퓨터공학과
김현정	산업공학과
이진영	전자공학과

- 1-1) 관계대수
- 1-2) SQL
- 1-3) 실행결과

- 1. 카티션 프로덕트(product)
- 2. 동등조인(equi join)
- 3. 자연조인(natural join)
- 4. 왼쪽외부조인(left outer join)
- 5. 오른쪽외부조인(right outer join)
- 6. 완전외부조인(full outer join)

-- 4. 왼쪽 외부조인(left outer join)

```
select freshman.name, address, dept_name
from freshman LEFT OUTER JOIN dmember
on  freshman.name=dmember.name;
```

[Quiz 02] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

관계대수 / SQL			
freshman		dmember	
name	address	name	dept_name
김광식	서울	김광식	컴퓨터공학과
김현정	대전	김현정	산업공학과
조영수	대전	이진영	전자공학과

1-1) 관계대수	1. 카티션 프로덕트(product)
1-2) SQL	2. 동등조인(equi join)
1-3) 실행결과	3. 자연조인(natural join)
	4. 왼쪽외부조인(left outer join)
	5. 오른쪽외부조인(right outer join)
	6. 완전외부조인(full outer join)

-- 5. 오른쪽 외부조인(right outer join)

```
select freshman.name, address, dept_name
from freshman RIGHT OUTER JOIN dmember
on  freshman.name=dmember.name;
```

[Quiz 02] (관계대수) 다음을 관계대수로 나타내고 SQL과 실행결과를 작성하라

관계대수 / SQL

freshman

name	address
김광식	서울
김현정	대전
조영수	대전

dmember

name	dept_name
김광식	컴퓨터공학과
김현정	산업공학과
이진영	전자공학과

- 1-1) 관계대수
- 1-2) SQL
- 1-3) 실행결과

- 1. 카티션 프로덕트(product)
- 2. 동등조인(equi join)
- 3. 자연조인(natural join)
- 4. 왼쪽외부조인(left outer join)
- 5. 오른쪽외부조인(right outer join)
- 6. 완전외부조인(full outer join)

-- 6. 완전 외부조인(full outer join)

02 관계 대수

❖ 순수 관계 연산자 – 선택

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

그림 6-12 선택 연산을 적용할 릴레이션 예 : 고객 릴레이션

예제 6-1

고객 릴레이션에서 등급이 gold인 튜플을 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

02 관계 대수

❖ 순수 관계 연산자 – 선택

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

그림 6-12 선택 연산을 적용할 릴레이션 예 : 고객 릴레이션

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 ≥ 2000

결과 릴레이션

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

02 관계 대수

❖ 순수 관계 연산자 - 선택

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

등급 = 'gold'

↓ 선택 연산

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

그림 6-13 선택 연산의 수행 과정 : 고객 릴레이션

결과 릴레이션은 연산 대상 릴레이션의 **수평적** 부분집합

02 관계 대수

• 순수 관계 연산자 – 프로젝트

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

그림 6-14 프로젝트 연산을 적용할 릴레이션 예

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

02 관계 대수

• 순수 관계 연산자 – 프로젝트

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

그림 6-14 프로젝트 연산을 적용할 릴레이션 예 : 고객 릴레이션

-- 고객 릴레이션에서 등급을 검색

SELECT **DISTINCT** 등급

FROM 고객;

예제 6-4

고객 릴레이션에서 등급을 검색하시오.

▶▶ $\pi_{\text{등급(고객)}}$ 또는 고객[등급]

결과 릴레이션

등급
gold
vip
silver

결과 릴레이션에서 동일한 튜플은 중복되지 않고
한 번만 나타남

02 관계 대수

❖ 순수 관계 연산자 – 프로젝트

π 고객이름,등급,적립금(고객)

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

프로젝트 연산

고객이름	등급	적립금
김현준	gold	1000
정소화	vip	2500
원유선	gold	4500
정지영	silver	0

결과 릴레이션은
연산 대상 릴레이션의
수직적 부분집합

그림 6-15 프로젝트 연산의 수행 과정 예 : 고객 릴레이션

02 관계 대수

- **순수 관계 연산자 – 조인(join)**

- 조인 속성을 이용해 두 릴레이션을 조합하여 결과 릴레이션을 구성
 - 조인 속성의 값이 같은 튜플만 연결하여 생성된 튜플을 결과 릴레이션에 포함
 - 조인 속성 : 두 릴레이션이 공통으로 가지고 있는 속성
- 표현법 : 릴레이션1 \bowtie 릴레이션2
- 자연 조인(natural join)이라고도 함
 - 표현법 : 릴레이션1 \bowtie_N 릴레이션2

02 관계 대수

- 순수 관계 연산자 – 조인

고객 릴레이션

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

주문 릴레이션

<u>주문번호</u>	주문고객	주문제품	수량
1001	apple	진짜우동	10
1002	carrot	맛있는파이	5
1003	banana	그대로만두	11

주문 릴레이션의 외래키

조인 속성:

고객 릴레이션의 고객아이디,
주문 릴레이션의 주문고객

그림 6-16 조인 연산을 적용할 릴레이션 예들의 관계

02 관계 대수

❖ 순수 관계 연산자 - 조인

-- 내부조인(inner join)

```
select 고객아이디,고객이름,나이,등급,  
       주문번호,주문제품,수량  
from   고객 INNER JOIN 주문  
on     고객아이디 = 주문고객;
```

-- 내부조인(inner join)

```
select 고객아이디,고객이름,나이,등급,  
       주문번호,주문제품,수량  
from   고객 , 주문  
where  고객아이디 = 주문고객;
```

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

↓ 조인 연산

고객 ⋈ 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 6-17 조인 연산의 수행 과정 예 : 고객과 주문 릴레이션

02 관계 대수

❖ 순수 관계 연산자 – 조인

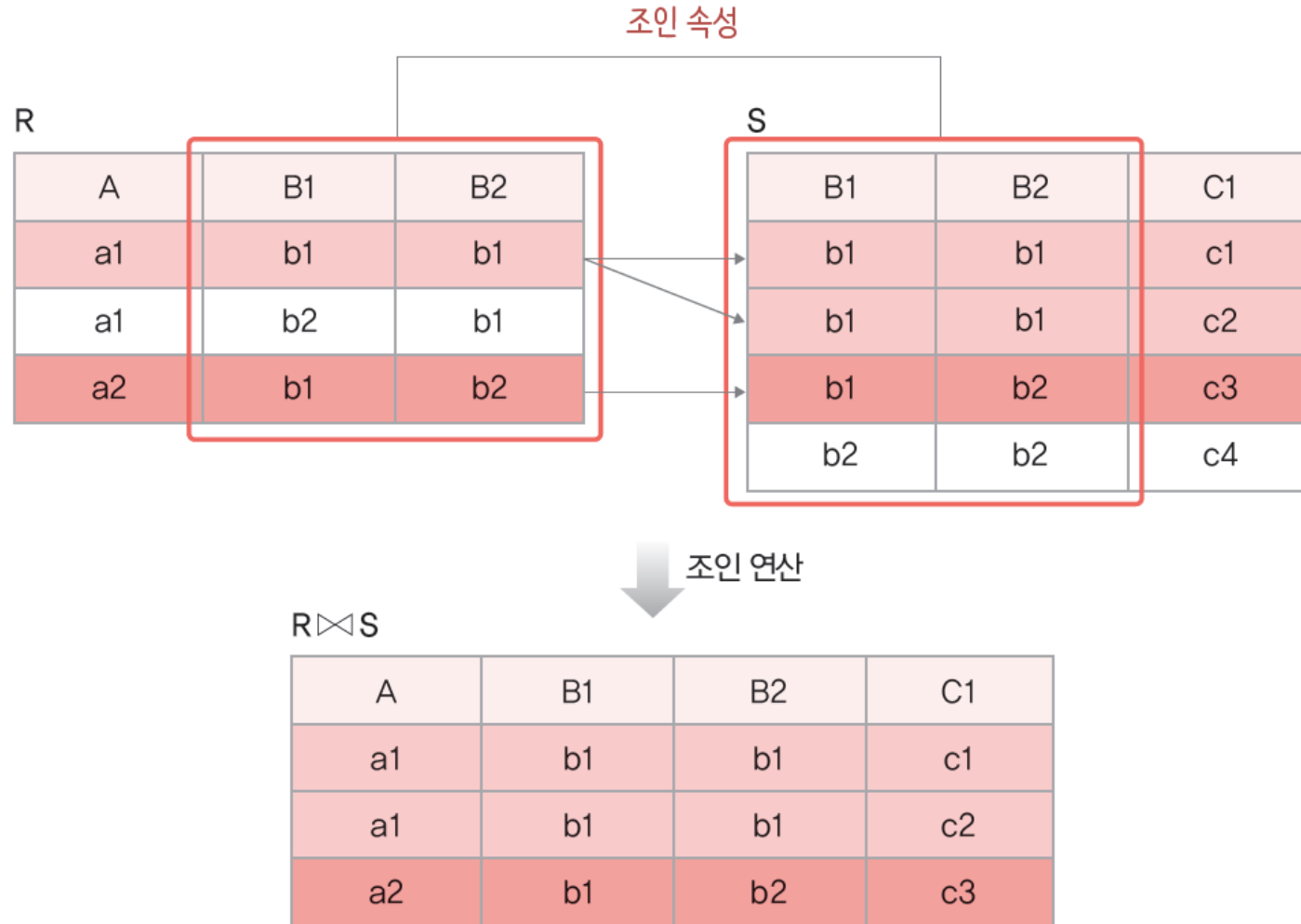


그림 6-18 2개의 속성으로 이루어진 조인 속성을 이용하는 조인 연산의 예 : R과 S 릴레이션

02 관계 대수

- 세타 조인(theta join, θ -join)

- 자연 조인에 비해 더 일반화된 조인
- 주어진 조인 조건을 만족하는 두 릴레이션의 모든 튜플을 연결하여 생성된 새로운 튜플로 결과 릴레이션을 구성
- 결과 릴레이션의 차수는 두 릴레이션의 차수를 더한 것과 같음
- 표현법 : 릴레이션1 $\bowtie_{A\theta B}$ 릴레이션2
 - θ 는 비교 연산자(>, \geq , <, \leq , =, \neq)를 의미

- 동일 조인(equi-join)

- θ 연산자가 “=”인 세타 조인을 의미

02 관계 대수

❖ 동일 조인

-- 내부조인(inner join)

select *

from 고객 INNER JOIN 주문

on 고객아이디 = 주문고객;

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

동일 조인 연산

고객 ⋈ 고객아이디=주문고객 주문

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

그림 6-19 동일 조인 연산의 예: 고객과 주문 릴레이션

02 관계 대수

❖ 순수 관계 연산자 – 디비전

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

02 관계 대수

❖ 순수 관계 연산자 – 디비전

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 ÷ 제품1

주문고객	제조업체
apple	한빛식품

주문내역 ÷ 제품2

주문고객
banana
apple
carrot

02 관계 대수

❖ 관계 대수를 이용한 질의 표현 예

고객 릴레이션

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

주문 릴레이션

<u>주문번호</u>	주문고객	주문제품	수량
1001	apple	진짜우동	10
1002	carrot	맛있는파이	5
1003	banana	그대로만두	11

주문 릴레이션의 외래키

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

02 관계 대수

- 관계 대수를 이용한 질의 표현 예

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

```
-- 등급이 gold인 고객의 이름과 나이를 검색
select 고객이름, 나이
from 고객
where 등급 = 'gold';
```

02 관계 대수

❖ 관계 대수를 이용한 질의 표현 예

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

```
-- 고객이름이 원유선인 고객의 등급과
-- 원유선 고객이 주문한 주문제품, 수량을 검색
select 등급, 주문제품, 수량
from 고객, 주문
where 고객아이디 = 주문고객
and 고객이름 = '원유선';
```

02 관계 대수

- 확장된 관계 대수 연산자 – 외부 조인(outer-join)
 - 자연 조인 연산에서 제외되는 튜플도 결과 릴레이션에 포함시키는 조인
 - 두 릴레이션에 있는 모든 튜플을 결과 릴레이션에 포함시킴
 - 표현법 : 릴레이션1 \bowtie^+ 릴레이션2

02 관계 대수

❖ 확장된 관계 대수 연산자 – 외부 조인

-- 외부조인

```
select 고객아이디,고객이름,나이,  
       주문번호,주문제품  
from   고객 LEFT OUTER JOIN 주문  
on     고객아이디 = 주문고객;
```

고객 릴레이션

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

주문 릴레이션

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

↓ 외부 조인 연산

고객 ⋈⁺ 주문

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

그림 6-26 고객과 주문 릴레이션의 외부 조인 연산

03 관계 해석

- **관계 해석(relational calculus)**
 - 처리를 원하는 데이터가 무엇인지만 기술하는 언어
 - 비절차 언어(nonprocedural language)
 - 수학의 프레디킷 해석(predicate calculus)에 기반을 두고 있음
 - 분류
 - 튜플 관계 해석(tuple relational calculus)
 - 도메인 관계 해석(domain relational calculus)

-- 고객(고객아이디, 고객이름, 나이, 등급)

-- 주문(주문번호, 주문고객, 주문제품, 수량)

CREATE TABLE 고객 (

 고객아이디 VARCHAR(20) NOT NULL,

 고객이름 VARCHAR(10),

 나이 INT,

 등급 VARCHAR(10) NOT NULL,

 PRIMARY KEY(고객아이디)

);

CREATE TABLE 주문 (

 주문번호 CHAR(4) NOT NULL,

 주문고객 VARCHAR(20),

 주문제품 varCHAR(20),

 수량 INT,

 PRIMARY KEY(주문번호),

 FOREIGN KEY(주문고객) REFERENCES 고객(고객아이디)

);

-- 고객(고객아이디, 고객이름, 나이, 등급)
-- 주문(주문번호, 주문고객, 주문제품, 수량)

-- 고객(고객아이디, 고객이름, 나이, 등급)
INSERT INTO 고객 VALUES ('apple', '김현준', 20, 'gold');
INSERT INTO 고객 VALUES ('banana', '정소화', 25, 'vip');
INSERT INTO 고객 VALUES ('carrot', '원유선', 28, 'gold');
INSERT INTO 고객 VALUES ('orange', '정지영', 22, 'silver');

-- 주문(주문번호, 주문고객, 주문제품, 수량)
INSERT INTO 주문 VALUES ('1001', 'apple', '진짜우동', 10);
INSERT INTO 주문 VALUES ('1002', 'carrot', '맛있는파이', 5);
INSERT INTO 주문 VALUES ('1003', 'banana', '그대로만두', 45);

select * from 고객;
select * from 주문;