

## 12. 관계대수식

[예제 1] 다음 릴레이션 R과 S에 대한 관계대수와 SQL문 실행 결과는?

R		S	
a	b	a	b
1	10	1	100
2	20	2	200
		3	300

- ① 관계대수  $R \times S$  //카티션곱  
 SQL문 `Select * From R, S;`

a	b	a	c
1	10	1	100
1	10	2	200
1	10	3	300
2	20	1	100
2	20	2	200
2	20	3	300

$2 \times 3 = 6$

- ② 관계대수  $\sigma_{R.a=1} (R \times S)$  //R.a=1에 대한 카티션곱  
 SQL문 `Select * From R, S Where R.a = 1;`

a	b	a	c
1	10	1	100
1	10	2	200
1	10	3	300

- ③ 관계대수  $R \bowtie_{R.a=S.a} S$  //동등조인  
 SQL문 `Select * From R, S Where R.a = S.a;`

a	b	a	c
1	10	1	100
2	20	2	200

- ④ 관계대수  $\sigma_{R.a=1} (R \bowtie_{R.a=S.a} S)$  //R.a=1에 대한 동등조인  
 SQL문 `Select * From R, S Where R.a = 1 And R.a = S.a;`

a	b	a	c
1	10	1	100

[예제 2] 다음 학생과 교수 릴레이션에 대한 관계대수와 SQL문의 결과는?

학생				교수	
학번	이름	성적	지도교수	교번	이름
1	Kim	90	P1	P1	이순신
2	Lee	80	P2	P2	강감찬
3	Han	60	P3	P3	김유신
4	Son	70	P1	P4	박문수
5	Sim	50	P1		

- ① 관계대수  $\sigma_{\text{학생.이름} = 'Kim'}$  (학생  $\times$  교수)  
 SQL문 `Select * From 학생, 교수 Where 학생.이름 = 'Kim';`

학번	이름	성적	지도교수	교번	이름
1	Kim	90	P1	P1	이순신
1	Kim	90	P1	P2	강감찬
1	Kim	90	P1	P3	김유신
1	Kim	90	P1	P4	박문수

- ② 관계대수  $\Pi_{\text{학번, 교수.이름}}(\text{학생} \bowtie_{\text{지도교수} = \text{교번}} \text{교수})$   
 SQL문 `Select 학번, 교수.이름 From 학생, 교수 Where 지도교수=교번;`

학번	이름
1	이순신
2	강감찬
3	김유신
4	이순신
5	이순신

학번의 지도교수 이름 출력

- ③ 관계대수  $\Pi_{\text{지도교수}}(\text{학생}) \cup \Pi_{\text{교번}}(\text{교수})$   
 SQL문 `Select 지도교수 From 학생 Union Select 교번 From 교수;`

지도교수
P1
P2
P3
P4

Union은 2개의 쿼리 결과를 합친다.(중복된 값은 하나만 출력, 즉 중복 제거)

[예제 3] 다음 릴레이션에 대한 관계대수식 수행 결과는?

구매			상품	
고객번호	수량	상품코드	상품코드	상품명
1	30	1	1	사과
1	60	2	2	수박
2	20	2	3	포도
2	40	3	4	오이
3	50	4	5	감자

관계대수식 :  $\Pi_{\text{고객번호, 상품명}} (\sigma_{\text{수량}=40} (\text{구매} \bowtie_N \text{상품}))$   
 ↓ 실행 순서

관계대수식 :  $\Pi_{\text{고객번호, 상품명}} (\sigma_{\text{수량}=40} (\text{구매} \bowtie_N \text{상품}))$

↓ 실행 과정

구매  $\bowtie_N$  상품

고객번호	수량	상품코드	상품명
1	30	1	사과
1	60	2	수박
2	20	2	수박
2	40	3	포도
3	50	4	오이

$\sigma_{\text{수량}=40}$

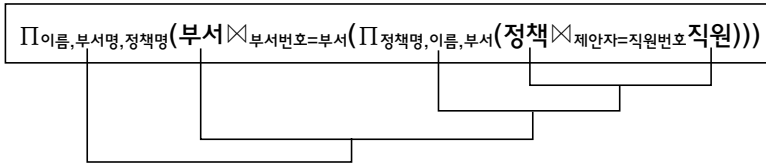
고객번호	수량	상품코드	상품명
1	30	1	사과
2	20	2	수박
2	40	3	포도

$\Pi_{\text{고객번호, 상품명}}$

고객번호	상품명
1	사과
2	수박
2	포도

[예제 4] 다음 릴레이션에 대한 관계대수식 수행 결과는?

정책			직원			부서	
정책번호	정책명	제안자	직원번호	이름	부서	부서번호	부서명
100	인력양성	40	10	김	2	1	기획과
200	주택자금	20	20	이	1	2	인사과
300	친절교육	10	30	박	1	3	총무과
400	환경정리	10	40	최	3		
500	신규고용	20					



↓ 수행 과정

정책 ⋈ 제안자=직원번호 직원					Π 정책명, 이름, 부서 (정책 ⋈ 제안자=직원번호 직원)		
정책번호	정책명	제안자	이름	부서	정책명	이름	부서
100	인력양성	40	최	3	인력양성	최	3
200	주택자금	20	이	1	주택자금	이	1
300	친절교육	10	김	2	친절교육	김	2
400	환경정리	10	김	2	환경정리	김	2
500	신규고용	20	이	1	신규고용	이	1

부서 ⋈ 부서번호=부서 (Π 정책명, 이름, 부서 (정책 ⋈ 제안자=직원번호 직원))

정책명	이름	부서	부서명
인력양성	최	3	총무과
주택자금	이	1	기획과
친절교육	김	2	인사과
환경정리	김	2	인사과
신규고용	이	1	기획과

Π 이름, 부서명, 정책명 (부서 ⋈ 부서번호=부서 (Π 정책명, 이름, 부서 (정책 ⋈ 제안자=직원번호 직원)))

이름	부서명	정책명
최	총무과	인력양성
이	기획과	주택자금
김	인사과	친절교육
김	인사과	환경정리
이	기획과	신규고용

[예제 5] 다음 릴레이션에 대한 관계대수식 수행 결과는? [관계대수식 관련 단골 출제 유형]

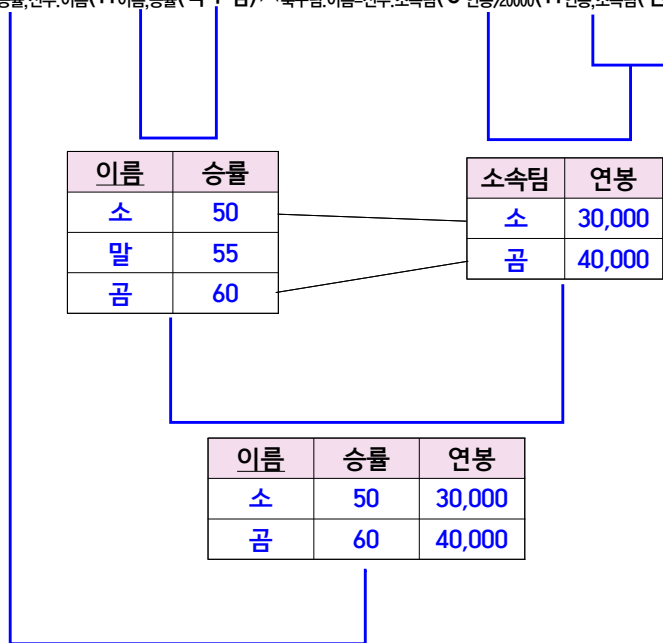
축구팀

이름	연고지	경기수	승률
소	제주	10	50
말	수원	15	55
곰	서울	20	60

선수

이름	소속팀	연봉
가	소	30,000
나	말	10,000
다	곰	40,000

• 관계대수식 :  $\Pi_{\text{축구팀.승률, 선수.이름}}(\Pi_{\text{이름, 승률}}(\text{축구팀}) \bowtie_{\text{축구팀.이름=선수.소속팀}} (\sigma_{\text{연봉} > 20000} (\Pi_{\text{연봉, 소속팀}}(\text{선수}))))$



- $\Pi_{\text{축구팀.승률, 선수.이름}}$
- 속성 "선수.이름"은 존재하지 않는다.
- 속성 "이름"은 축구팀 이름이다.
- "선수.이름"을 추출할 수 없다.
- 오류가 발생한다.

[예제 6] SQL 표현 / 관계대수 표현

〈SQL 표현〉

Select 학생.학번, 수강.과목명, 과목.학점  
 From 학생, 수강, 과목  
 Where 학생.학과='전산' And 학생.학번=수강.학번 And 수강.과목명=과목.과목명;

↓ 관계대수 표현으로

$\Pi_{\text{학생.학번, 수강.과목명, 과목.학점}}((\sigma_{\text{학생.학과='전산'}}(\text{학생})) \bowtie_{\text{학생.학번=수강.학번}}(\text{수강})) \bowtie_{\text{수강.과목명=과목.과목명}}(\text{과목}))$

↓ 릴레이션 구조로 표현하면

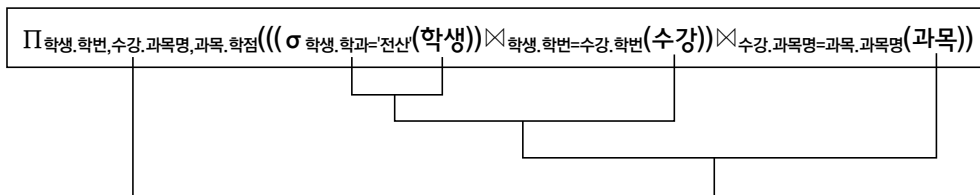
학생			수강			과목		
학번	학과	이름	학번	과목명	점수	과목명	학점	개설학과
1	전산	홍재연	1	디비	90	디비	3	전산
2	토목	김유신	2	디비	80	소공	2	전산
3	전산	강감찬	1	소공	70	컴일	1	전산
4	전산	이순신	2	소공	80	보안	3	전산
5	건축	김구	3	소공	90	역학	3	토목

↓ SQL 실행 결과

학번	과목명	학점
1	디비	3
1	소공	2
3	소공	2

전산과 학생의 수강 과목

// 관계대수 실행 과정



[예제 7] 관계대수식에서 재명명

재명명 연산 rename operation	<ul style="list-style-type: none"> <li>• 단순히 테이블명이나 속성명을 변경하는 연산이다.</li> <li>• 수학 기호 <math>\rho</math>(rho)를 사용한다.</li> <li>• 예 : <math>\rho_f(\text{food}) \leftarrow</math> 테이블명 food을 f로 변경</li> <li>• SQL문 : FROM food AS f</li> </ul>
----------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

R		
a	b	c
a1	b1	c1
a2	b2	c2

 $\rho_{S(d,e,f)}(R)$ 

S		
d	e	f
a1	b1	c1
a2	b2	c2

• 테이블명을 R에서 S로 변경하고, 속성 a, b, c를 d, e, f로 재명명

학생		
학번	이름	학년
1	홍재연	3
2	김유신	1
3	강감찬	2
4	이순신	4
5	김구	3

수강		
학번	과목	점수
1	디비	90
2	디비	80
1	소공	70
2	소공	80
3	소공	90

SQL	SELECT 학번, 이름 FROM 학생 AS s // 테이블명 학생을 s로 재명명 WHERE s.학년 < 3;
관계대수	$\Pi_{\text{학번,이름}}(\sigma_{s.\text{학년}<3}(\rho_s(\text{학생})))$

SQL	SELECT 학번, 이름, 과목, 점수 FROM 학생 AS s, 수강 AS d // 학생은 s, 수강은 d로 재명명 WHERE s.학년 < 3 And d.점수 > 80;
관계대수	$\Pi_{\text{학번,이름,과목,점수}}(\sigma_{s.\text{학년}<3 \text{ And } d.\text{점수}>80}(\rho_s(\text{학생}) \times \rho_d(\text{수강})))$

SQL	SELECT 학번, 이름, 과목, 점수 FROM 학생 AS s, 수강 AS d // 학생은 s, 수강은 d로 재명명 WHERE s.학년 < 3 And s.학번 = d.학번;
관계대수	$\Pi_{\text{학번,이름,과목,점수}}(\sigma_{s.\text{학년}<3}(\rho_s(\text{학생}) \bowtie_{s.\text{학번}=d.\text{학번}} \rho_d(\text{수강})))$

[예제 8] 관계대수식에서 그룹화 및 집계함수 연산

집계함수 연산 F	<ul style="list-style-type: none"> <li>• 예 : <math>(name)_{F_{SUM(price)}}</math> ↓</li> <li>• 속성 name으로 그룹화하고(GROUP BY name)</li> <li>• 집계함수 SUM(price) 연산한다는 의미이다.</li> <li>• 사용하는 기호는 F, <math>\bowtie</math>, <math>\sigma</math>, <math>\pi</math> 등을 사용한다.</li> </ul>
--------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

// 그룹화한 집계함수에 대한 관계대수 표현

질의문	사원 테이블에 대한 각 부서별 급여 합을 구하라?
SQL 문	Select 부서, Sum(급여) From 사원 Group By 부서;
관계대수	$\rho_{부서} F_{sum(급여)} (사원) \leftarrow$ 속성 부서로 그룹화하고, 집계함수 Sum(급여) 연산

사원

사번	이름	급여	부서
1	홍재연	1000	생산부
2	이순신	2000	생산부
3	김유신	3000	생산부
4	강감찬	4000	경리부
5	홍하은	5000	경리부

SQL 문	Select 부서, Sum(급여) From 사원 Group By 부서;
관계대수	$\rho_{부서} F_{sum(급여)} (사원)$

부서	Sum(급여)
생산부	6,000
경리부	9,000

SQL 문	Select 부서, Sum(급여) As 총급여 From 사원 Group By 부서;
관계대수	$\rho_{(부서, 총급여)} (부서)_{F_{sum(급여)}} (사원)$

부서	총급여
생산부	6,000
경리부	9,000



**기출문제 분석**

1. 다음 관계대수식과 다른 결과를 생성하는 SQL문은? [2014년 국가 7급]

$$\Pi_{R,r2}(\sigma_{(R.r1 = S.s1 \text{ AND } S.s2 = 100)}(R \times S))$$

- R, S는 릴레이션
- r1, r2는 릴레이션 R의 애트리뷰트
- s1, s2는 릴레이션 S의 애트리뷰트

- ① SELECT DISTINCT R,r2 FROM R WHERE EXISTS  
(SELECT S,s1 FROM S WHERE S.s1=R.r1 AND S.s2=100);
- ② SELECT DISTINCT A,r2 FROM R AS A, S AS B  
WHERE A.r1=B.s1 AND B.s2=100;
- ③ SELECT DISTINCT R,r2 FROM R NATURAL JOIN S WHERE S.s2=100;
- ④ SELECT DISTINCT R,r2 FROM R WHERE R.r1 IN  
(SELECT S.s1 FROM S WHERE S.s2=100);

☞ 관계대수식을 SQL문으로 변환

//  $\Pi_{R,r2}(\sigma_{(R.r1 = S.s1 \text{ AND } S.s2 = 100)}(R \times S))$  분석

R×S	· 카티션 곱 → FROM R, S
$\sigma_{(R.r1 = S.s1 \text{ AND } S.s2 = 100)}$	· 카티션 곱 결과에서 R.r1=S.s1이고, S.s2=100인 것을 선택 · WHERE R.r1=S.s1 AND S.s2=100;
$\Pi_{R,r2}$	· 위에서 선택된 것 중에서 속성 R,r2만 추출 · SELECT DISTINCT R,r2

// 항목 ③은 다음처럼 수정하면 올바른 것이 된다.

- ③ SELECT DISTINCT R,r2 FROM R NATURAL JOIN S  
WHERE S.s2=100;  
↓ 조건 추가
- ③ SELECT DISTINCT R,r2 FROM R NATURAL JOIN S  
WHERE S.s1=R.r1 AND S.s2=100;

2. 다음과 같은 두 릴레이션 R과 S가 있다고 가정할 때, R과 S를 자연조인한 결과로 가장 적절한 것은? [2023년 군무 7급]

R			S		
A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a2	b1	c2	b1	c1	d2
a3	b2	c2	b2	c3	d3

①

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2

②

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a2	b1	c2	d1
a2	b1	c2	d2
a3	b2	c2	d2

③

A	B	C
a1	b1	c1

④

B	C
b1	c1

☞ 자연조인

R			S		
A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a2	b1	c2	b1	c1	d2
a3	b2	c2	b2	c3	d3

↓  
↓ 자연조인  
↓

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2

• 공통 속성 

B	C
---	---

 를 기준으로 자연조인한다.

3. 두 릴레이션 R과 S가 아래와 같이 주어질 때, 다음 관계대수 중에서 결과 릴레이션의 카디널리티(cardinality)가 가장 작은 것은? [2023년 군무 7급]

R			S	
A	B	C	C	D
S100	B	C0123	C0123	김철수
S200	B	C0123	C0234	이영자
S300	A	C0342	C0342	최용철
S400	B	C0342	C0445	최용철
S500	B	C0445	C0577	박진호

- ①  $R \times S$
- ②  $\Pi_C(S) - \Pi_{S,C}(R \bowtie_N S)$
- ③  $\Pi_{A,C}(R) \div \Pi_C(\sigma_{D='최용철'}(S))$
- ④  $\sigma_{D='최용철'}(R \bowtie_N S)$

☞ 관계대수 - 카디널리티(튜플수)

①  $R \times S$  : Select \* From R, S; =  $5 \times 5 = 25$ 개 ← 카티션곱

②  $\Pi_C(S) - \Pi_{S,C}(R \bowtie_N S)$  : (Select C From S;) - (Select S.C From R Natural Join S;)

C	C	
C0123	C0123	
C0234	C0123	
C0342	C0342	
C0445	C0342	
C0577	C0445	

-

C
C0234
C0577

= 2개

③  $\Pi_{A,C}(R) \div \Pi_C(\sigma_{D='최용철'}(S))$

A	C
S100	C0123
S200	C0123
S300	C0342
S400	C0342
S500	C0445

÷

C
C0342
C0445

=

A
---

0개 ←

C0342
C0445

를 동시에 갖는 속성 A는 없음

④  $\sigma_{D='최용철'}(R \bowtie_N S)$  : Select \* From R Natural Join S Where D='최용철';

C	A	B	D
C0342	S300	A	최용철
C0342	S400	B	최용철
C0445	S500	B	최용철

3개

4. 아래의 고객 릴레이션에서 등급이 gold이고 나이가 25 이상인 고객들을 검색하기 위해 기술한 관계대수 표현으로 옳은 것은? [2022년 국가 9급]

<고객 릴레이션>

고객				
고객아이디	이름	나이	등급	직업
hohoho	이순신	29	gold	교사
grace	홍길동	24	gold	학생
mango	삼돌이	27	silver	학생
juce	갑순이	31	gold	공무원
orange	강감찬	23	silver	군인

<검색결과>

고객				
고객아이디	이름	나이	등급	직업
hohoho	이순신	29	gold	교사
juce	갑순이	31	gold	공무원

- ①  $\sigma_{\text{고객}}(\text{등급} = \text{'gold'} \wedge \text{나이} \geq 25)$
- ②  $\sigma_{\text{등급} = \text{'gold'} \wedge \text{나이} \geq 25}(\text{고객})$
- ③  $\pi_{\text{고객}}(\text{등급} = \text{'gold'} \wedge \text{나이} \geq 25)$
- ④  $\pi_{\text{등급} = \text{'gold'} \wedge \text{나이} \geq 25}(\text{고객})$

☞ 관계대수

- $\sigma_{\text{등급} = \text{'gold'} \wedge \text{나이} \geq 25}(\text{고객})$



등급이 gold이고, 나이가 25 이상인 고객

// 순수 관계연산자

Select(선택, $\sigma$ )	릴레이션의 튜플 검색 연산자. 수평적 부분집합
Project(투사, $\Pi$ )	릴레이션의 속성 검색 연산자. 수직적 부분집합
Join(조인, $\bowtie$ )	두 릴레이션을 결합하여 새로운 하나의 릴레이션 생성
Division(나누기, $\div$ )	두 릴레이션의 공통 속성값에 연관된 나머지 속성의 튜플을 구한다.

5. 학생과 교수 릴레이션에 대한 아래 관계대수 중에서 결과 릴레이션의 카디널리티(cardinality)가 가장 큰 것은? [2022년 군무원 7급]

학생			
학번	이름	성적	지도교수
100	Kim	92	P1
200	Park	88	P2
300	Lee	95	P3
400	Choi	73	P1
500	Kang	82	P1

교수	
id	학과
P1	컴퓨터공학
P2	컴퓨터공학
P3	기계공학
P4	교양학부

- ①  $\sigma_{\text{성적} > 90}$  (학생)
- ②  $\Pi_{\text{학번}, \text{학과}} (\text{학생} \bowtie_{\text{지도교수} = \text{id}} \text{교수})$
- ③  $\Pi_{\text{지도교수}} (\text{학생}) \cup \Pi_{\text{id}} (\text{교수})$
- ④  $\sigma_{\text{이름} = 'Kim'}$  (학생  $\times$  교수)

☞ 관계대수 - 카디널리티 : 튜플수

①  $\sigma_{\text{성적} > 90}$  (학생) : **Select \* From 학생 Where 성적>90;**

학번	이름	성적	지도교수
100	Kim	92	P1
300	Lee	95	P3

카디널리티 = 2

②  $\Pi_{\text{학번}, \text{학과}} (\text{학생} \bowtie_{\text{지도교수} = \text{id}} \text{교수})$  : **Select 학번, 학과 From 학생, 교수 Where 지도교수=id;**

학번	학과
100	컴퓨터공학
200	컴퓨터공학
300	기계공학
400	컴퓨터공학
500	컴퓨터공학

카디널리티 = 5

③  $\Pi_{\text{지도교수}} (\text{학생}) \cup \Pi_{\text{id}} (\text{교수})$  : **Select 지도교수 From 학생 Union Select id From 교수;**

지도교수
P1
P2
P3
P4

카디널리티 = 4

④  $\sigma_{\text{이름} = 'Kim'}$  (학생  $\times$  교수) : **Select \* From 학생, 교수 Where 이름 = 'Kim';**

학번	이름	성적	지도교수	id	학과
100	Kim	92	P1	P1	컴퓨터공학
100	Kim	92	P1	P2	컴퓨터공학
100	Kim	92	P1	P3	기계공학
100	Kim	92	P1	P4	교양학부

카디널리티 = 4

정답 : ②

6. <보기>의 릴레이션 Student(sno, sname, address, score, year, dept)에 대해, 여러 관계대수식을 수행하려고 한다. 관계대수식의 결과로 생성되는 릴레이션의 차수(degree)와 카디널리티(cardinality)가 바르게 짝지어지지 않은 것은? [2018년 서울 7급]

-----<보기>-----

sno	sname	address	score	year	dept
10	홍길동	서울	50	4	100
11	김철수	대전	90	3	200
12	김영희	대구	55	2	100
13	황이수	부산	60	3	200
14	정순희	서울	70	1	300
15	이태백	서울	100	4	300
16	최지현	부산	70	2	100
17	김현정	인천	65	3	100

- ①  $\sigma_{year \geq 3}(\text{Student}) - \sigma_{dept = 100}(\text{Student}) =$  차수 : 6, 카디널리티 : 2
- ②  $\Pi_{sno, sname, score}(\sigma_{score > 60}(\text{Student})) =$  차수 : 3, 카디널리티 : 5
- ③  $\sigma_{year \geq 4}(\text{Student}) \cup \sigma_{year \leq 1}(\text{Student}) =$  차수 : 6, 카디널리티 : 3
- ④  $\sigma_{address = '대전' \text{ and } year = 3}(\Pi_{sno, address, score, year}(\text{Student})) =$  차수 : 4, 카디널리티 : 1

♣ 관계대수식

- ①  $\sigma_{year \geq 3}(\text{Student}) - \sigma_{dept = 100}(\text{Student}) =$  차수 : 6, 카디널리티 : 3

sno	sname	address	score	year	dept
10	홍길동	서울	50	4	100
11	김철수	대전	90	3	200
13	황이수	부산	60	3	200
15	이태백	서울	100	4	300
17	김현정	인천	65	3	100

sno	sname	address	score	year	dept
10	홍길동	서울	50	4	100
12	김영희	대구	55	2	100
16	최지현	부산	70	2	100
17	김현정	인천	65	3	100

=

sno	sname	address	score	year	dept
11	김철수	대전	90	3	200
13	황이수	부산	60	3	200
15	이태백	서울	100	4	300

• 차수 : 6, 카디널리티 : 3

7. 다음은 어느 기관의 데이터베이스 테이블을 나타낸 것이다. [2010년 국가 9급]

직원			부서		정책		
직원번호	이름	부서	부서번호	부서명	정책번호	정책명	제안자
10	김	B20	A10	기획과	100	인력양성	40
20	이	A10	B20	인사과	200	주택자금	20
30	박	A10	C30	총무과	300	친절교육	10
40	최	C30			400	성과금	10
					500	신규고용	20

다음 관계대수식을 적용한 결과의 카디널리티(cardinality)로 옳은 것은?

$\Pi_{\text{이름, 부서명, 정책명}}(\text{부서} \bowtie \text{부서번호} = \text{부서} (\Pi_{\text{정책명, 이름, 부서}}(\text{정책} \bowtie \text{제안자} = \text{직원번호 직원})))$

- ① 3                                      ② 4                                      ③ 5                                      ④ 6

♣ 관계대수의 관계연산자

- Project(투사,  $\Pi$ ) : 릴레이션의 속성 검색 연산자. 수직적 부분집합
- Join( $\bowtie$ ) : 두 릴레이션을 결합하여 새로운 하나의 릴레이션을 생성하는 연산이다.

㉠  $\Pi_{\text{정책명, 이름, 부서}}(\text{정책} \bowtie \text{제안자} = \text{직원번호 직원}) \rightarrow$  정책과 직원 테이블을 조인

정책명	이름	부서
인력양성	최	C30
주택자금	이	A10
친절교육	김	B20
성과금	김	B20
신규고용	이	A10

↓  
↓ 조인 결과와 부서 테이블 다시 조인함  
↓

㉡  $\Pi_{\text{이름, 부서명, 정책명}}(\text{부서} \bowtie \text{부서번호} = \text{부서} (\Pi_{\text{정책명, 이름, 부서}}(\text{정책} \bowtie \text{제안자} = \text{직원번호 직원})))$

이름	부서명	정책명
최	총무과	인력양성
이	기획과	주택자금
김	인사과	친절교육
김	인사과	성과금
이	기획과	신규고용

→ 튜플 수는 5개이다.

- 적용 결과 카디널리티(cardinality)는 5이다. 카디널리티는 튜플 수이다.

8. 데이터베이스 스키마가 <보기>와 같을 때, 결과가 다른 질의는? [2022년 서울 7급]

----<보기>-----

Student(id, name, dept\_id)

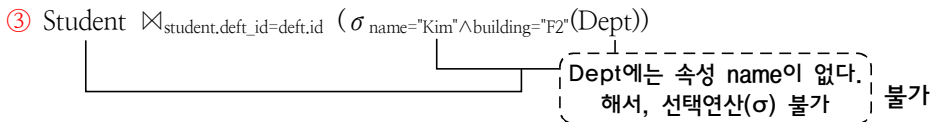
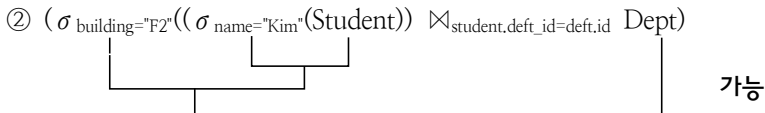
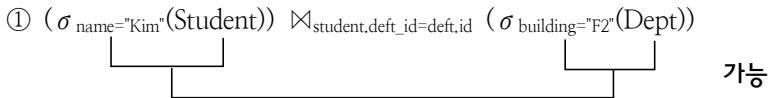
Dept(id, dept\_name, building)

- ①  $(\sigma_{name="Kim"}(Student)) \bowtie_{student.dept\_id=dept.id} (\sigma_{building="F2"}(Dept))$
- ②  $(\sigma_{building="F2"}((\sigma_{name="Kim"}(Student)) \bowtie_{student.dept\_id=dept.id} Dept))$
- ③  $Student \bowtie_{student.dept\_id=dept.id} (\sigma_{name="Kim" \wedge building="F2"}(Dept))$
- ④  $\sigma_{name="Kim" \wedge building="F2"}(Student \bowtie_{student.dept\_id=dept.id} Dept)$

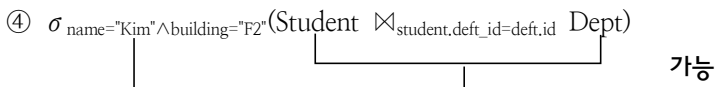
☞ 관계대수식

// 먼저, 질의를 분석하면 다음과 같다.

질의 : 학생(Student) 이름이 Kim이고, 학과(Dept)는 빌딩 F2에 있는 것을 조인



Select(선택, $\sigma$ )	<ul style="list-style-type: none"> <li>• 형식 : <math>\sigma_{조건}</math> (릴레이션)</li> <li>• 주어진 조건에 만족하는 튜플들을 선택하는 연산자이다.(연산 대상이 튜플)</li> <li>• 조건에는 반드시 릴레이션에 있는 속성을 사용해야 한다.</li> </ul>
-----------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------





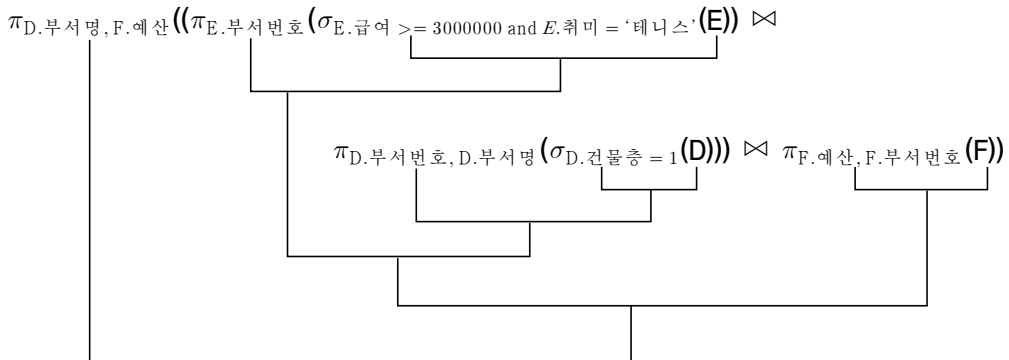
9. 다음 데이터베이스 스키마를 보고, SQL 질의문과 동등한 관계대수식은? [2021년 국가 7급]

스키마	사원(사원번호, 부서번호, 급여, 취미) 부서(부서번호, 부서명, 건물층, 전화번호) 재정(부서번호, 예산, 매출, 비용)
질의문	Select D.부서명, F.예산 From 사원 E, 부서 D, 재정 F Where E.부서번호=D.부서번호 and D.부서번호=F.부서번호 and D.건물층=1 and E.급여>=3000000 and E.취미='테니스'

- ①  $\pi_{D.부서명, F.예산} ((\pi_{E.부서번호} (\sigma_{E.급여 \geq 3000000 \text{ and } E.취미 = '테니스'}(E))) \bowtie \pi_{D.부서번호, D.부서명} (\sigma_{D.건물층 = 1}(D))) \bowtie \pi_{F.예산, F.부서번호}(F))$
- ②  $\pi_{E.부서번호} ((\pi_{D.부서명, F.예산} (\sigma_{E.급여 \geq 3000000 \text{ and } E.취미 = '테니스'}(E))) \bowtie \pi_{D.부서번호, D.부서명} (\sigma_{D.건물층 = 1}(D))) \bowtie \pi_{F.예산, F.부서번호}(F))$
- ③  $\pi_{D.부서명, F.예산} ((\pi_{F.부서번호, D.부서명} (\sigma_{D.건물층 = 1}(D))) \bowtie \pi_{E.부서번호} (\sigma_{E.급여 \geq 3000000 \text{ and } E.취미 = '테니스'}(E))) \bowtie \pi_{F.예산}(F))$
- ④  $\pi_{D.부서명, F.예산} ((\pi_{F.부서번호} (\sigma_{F.예산, F.부서번호}(F))) \bowtie \pi_{D.부서번호, D.부서명} (\sigma_{D.건물층 = 1}(D))) \bowtie \sigma_{E.급여 \geq 3000000 \text{ and } E.취미 = '테니스'}(E))$

☞ 관계대수식

// 수행 과정은 다음과 같다



- 사원 테이블에서 급여>=3000000 and 취미='테니스'인 튜플을 선택 후에 부서번호를 추출
- 부서 테이블에서 건물층=1인 튜플을 선택 후에 부서번호, 부서명을 추출
- 위의 두 결과를 조인한다.
- 그리고, 재정 테이블에서 예산과 부서번호를 추출하고 조인한다.
- 마지막으로 부서명과 예산을 추출한다.

10. <보기>의 릴레이션 R과 S에 대하여 주어진 관계대수식을 수행하려고 한다. 관계대수식의 수행 결과로 가장 옳지 않은 것은? [2020년 서울 7급]

-----<보기>-----

R	sid	name	address	year
	1	문재훈	광양	7
	2	김형진	대구	6
	3	유제현	서울	9
	4	박진우	서울	8
	5	김현호	서울	2
	6	정승완	서울	5
	7	박성욱	부산	1

S	sid	score	dept
	1	80	100
	2	80	200
	3	90	300
	4	70	400
	6	100	500

①  $\sigma_{\text{address}='서울'}(R) - \sigma_{\text{year} \geq 6}(R)$

sid	name	address	year
5	김현호	서울	2
6	정승완	서울	5

②  $\pi_{\text{name, score}}(\sigma_{\text{score} \geq 80 \text{ and year} > 5}(R \times S))$

name	score
문재훈	80
김형진	80
유제현	90
정승완	100

③  $\pi_{\text{name, address, year, score}}(\sigma_{\text{year} \geq 5}(R \times S))$

name	address	year	score
문재훈	광양	7	80
김형진	대구	6	80
유제현	서울	9	90
박진우	서울	8	70
정승완	서울	5	100

④  $\pi_{\text{score, dept}}(\sigma_{\text{score} \geq 90}(S))$

score	dept
90	300
100	500

☞ 관계대수식

②  $\pi_{name, score}(\sigma_{score \geq 80 \text{ and } year \neq 5}(R \bowtie S))$

R	sid	name	address	year
	1	문재훈	광양	7
	2	김형진	대구	6
	3	유제현	서울	9
	4	박진우	서울	8
	5	김현호	서울	2
	6	정승완	서울	5
	7	박성욱	부산	1

S	sid	score	dept
	1	80	100
	2	80	200
	3	90	300
	4	70	400
	6	100	500

↓  
 $R \bowtie S \rightarrow$  주어진 문제에 조인조건이 없다.  
 ↓ 자연조인으로 처리하면 다음과 같다.  
 ↓

sid	name	address	year	score	dept
1	문재훈	광양	7	80	100
2	김형진	대구	6	80	200
3	유제현	서울	9	90	300
4	박진우	서울	8	70	400
6	정승완	서울	5	100	500

↓  
 $\sigma_{score \geq 80 \text{ and } year \neq 5}(R \bowtie S)$   
 ↓

sid	name	address	year	score	dept
1	문재훈	광양	7	80	100
2	김형진	대구	6	80	200
3	유제현	서울	9	90	300

↓  
 $\pi_{name, score}(\sigma_{score \geq 80 \text{ and } year \neq 5}(R \bowtie S))$   
 ↓

name	score
문재훈	80
김형진	80
유제현	90

11. 다음 주어진 릴레이션 R과 S에 대해서 T의 결과를 얻기 위해 적용 가능한 관계대수식으로 옳지 않은 것은? (단,  $\bowtie$  : 조인,  $*$  : 자연조인,  $\times$  : 카티션 프로덕트) [2010년 국가 7급]

R		
A	B	C
1	b1	c1
2	b2	c2
3	b1	c3

S		
D	E	F
1	b2	f1
4	b3	f2
3	b2	f3

 $=$ 

T		
A	B	F
2	b2	f1
2	b2	f3

- ①  $\Pi_{A, B, F}(R \bowtie_{R,B=S,E} (\sigma_{S,D>1}(S)))$
- ②  $\Pi_{A, B, F}(R *_{\langle R,B, S,E \rangle} S)$
- ③  $\Pi_{A, B, F}(\sigma_{R,B=S,E}(R \times S))$
- ④  $\Pi_{A, B, F}(R \bowtie_{R,B=S,E} S)$

☞ 관계대수식

①  $\Pi_{A, B, F}(R \bowtie_{R,B=S,E} (\sigma_{S,D>1}(S)))$   
 $\downarrow \sigma_{S,D>1}(S)$

D	E	F
4	b3	f2
3	b2	f3

$\downarrow R \bowtie_{R,B=S,E} (\sigma_{S,D>1}(S))$

A	B	C	D	E	F
2	b2	c2	3	b2	f3

$\downarrow \Pi_{A, B, F}(R \bowtie_{R,B=S,E} (\sigma_{S,D>1}(S)))$

A	B	F
2	b2	f3

②  $\Pi_{A, B, F}(R \bowtie_{R,B=S,E} S)$

A	B	C	D	E	F
2	b2	c2	1	b2	f1
2	b2	c2	3	b2	f3

 $\Rightarrow$ 

A	B	F
2	b2	f1
2	b2	f3

12. 다음은 회원과 동호회 관리를 위해 모델링된 릴레이션 스키마와 SQL 질의문이다. 이 SQL 문을 관계대수로 변환한 것으로 옳지 않은 것은? (단, 밑줄은 기본키이다) [2012년 국가 7급]

릴레이션 스키마	<ul style="list-style-type: none"> <li>· 회원(회원번호, 주민등록번호, 회원명, 관심분야)</li> <li>· 동호회(동호회ID, 동호회이름, 개설일, 관리자회원번호)</li> <li>· 가입(회원번호, 동호회ID)</li> </ul>
SQL 질의문	<pre>SELECT a.회원명, a.관심분야 FROM 회원 a, 가입 b, 동호회 c WHERE c.동호회이름='자연사랑'       AND c.동호회ID=b.동호회ID       AND b.회원번호=a.회원번호</pre>

- ①  $\Pi_{\text{회원명, 관심분야}}(\sigma_{\text{동호회이름='자연사랑'}}((\Pi_{\text{회원번호, 회원명, 관심분야}} \text{회원}) \bowtie_{\text{회원번호=회원번호}} (\text{가입} \bowtie_{\text{동호회ID=동호회ID}} \text{동호회})))$
- ②  $\Pi_{\text{회원명, 관심분야}}(\sigma_{\text{동호회이름='자연사랑'}}(\Pi_{\text{회원명, 관심분야}}(\text{회원} \bowtie_{\text{회원번호=회원번호}} (\text{가입} \bowtie_{\text{동호회ID=동호회ID}} \text{동호회}))))$
- ③  $\Pi_{\text{회원명, 관심분야}}(\text{회원} \bowtie_{\text{회원번호=회원번호}} (\text{가입} \bowtie_{\text{동호회ID=동호회ID}} (\sigma_{\text{동호회이름='자연사랑'}} \text{동호회})))$
- ④  $\Pi_{\text{회원명, 관심분야}}(\sigma_{\text{동호회이름='자연사랑'}}(\text{회원} \bowtie_{\text{회원번호=회원번호}} (\text{가입} \bowtie_{\text{동호회ID=동호회ID}} \text{동호회})))$

☞ 관계대수

· SQL 질의문 : 자연사랑 동호회에 가입된 회원명과 관심분야를 검색한다.

$$\textcircled{2} \Pi_{\text{회원명, 관심분야}}(\sigma_{\text{동호회이름='자연사랑'}}(\Pi_{\text{회원명, 관심분야}}(\text{회원} \bowtie_{\text{회원번호=회원번호}} (\text{가입} \bowtie_{\text{동호회ID=동호회ID}} \text{동호회}))))$$



$$\Pi_{\text{회원명, 관심분야}}(\text{회원} \bowtie_{\text{회원번호=회원번호}} (\text{가입} \bowtie_{\text{동호회ID=동호회ID}} \text{동호회}))$$

↳ "회원명, 관심분야"만 프로젝션 되었다. "동호회이름"은 추출되지 않았다.



$$\sigma_{\text{동호회이름='자연사랑'}}(\Pi_{\text{회원명, 관심분야}}(\text{회원} \bowtie_{\text{회원번호=회원번호}} (\text{가입} \bowtie_{\text{동호회ID=동호회ID}} \text{동호회})))$$

↳ "동호회이름"은 추출되지 않아서, 비교할 것이 없다.(오류 발생)

정답 : ②

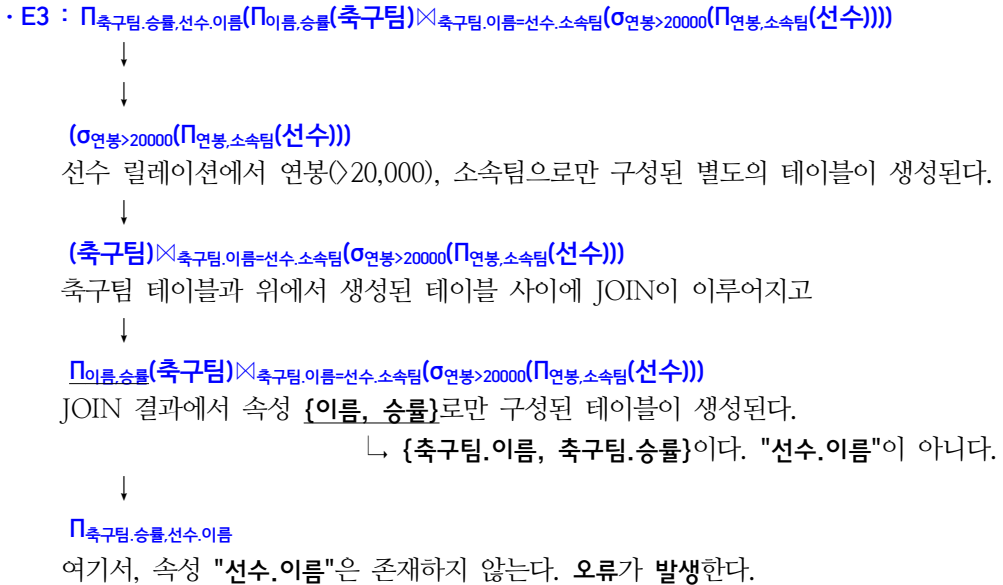
13. 축구팀과 선수들에 대한 릴레이션 스키마와 관계대수식이 다음과 같을 때, 동등한 관계대수식을 모두 고른 것은? (단, 밑줄은 기본키이다) [2012년 국가 7급]

릴레이션 스키마	<ul style="list-style-type: none"> <li>· 축구팀(이름, 연고지, 경기수, 승률)</li> <li>· 선수(이름, 소속팀, 연봉)</li> </ul>
관계대수식	<ul style="list-style-type: none"> <li>· E1 : <math>\Pi_{\text{축구팀.승률, 선수.이름}}(\sigma_{\text{축구팀.이름=선수.소속팀 \text{ and } \text{선수.연봉}&gt;20000}(\text{축구팀} \times \text{선수}))</math></li> <li>· E2 : <math>\Pi_{\text{축구팀.승률, 선수.이름}}(\Pi_{\text{이름, 승률}}(\text{축구팀}) \times \Pi_{\text{이름=선수.소속팀}}(\sigma_{\text{연봉}&gt;20000}(\text{선수})))</math></li> <li>· E3 : <math>\Pi_{\text{축구팀.승률, 선수.이름}}(\Pi_{\text{이름, 승률}}(\text{축구팀}) \times \Pi_{\text{이름=선수.소속팀}}(\sigma_{\text{연봉}&gt;20000}(\Pi_{\text{연봉, 소속팀}}(\text{선수}))))</math></li> <li>· E4 : <math>\Pi_{\text{축구팀.승률, 선수.이름}}(\text{축구팀} \times \Pi_{\text{이름=선수.소속팀}}(\sigma_{\text{선수.연봉}&gt;20000}(\text{선수})))</math></li> </ul>

- ① E1, E2
- ② E2, E3
- ③ E1, E2, E4
- ④ E1, E2, E3, E4

☞ 관계대수식

- E1, E2, E4 : 연봉이 20,000 이상인 선수와 해당 팀의 승률을 구한다.
- E3 : 오류가 발생한다.



14. 사원 데이터베이스에서 '부양가족이 없는 사원들의 성과 이름을 검색하라'라는 질의에 대한 관계대수식은? (단,  $\bowtie$ 는 자연조인이고, 조인 애트리뷰트는 주민번호이다) [2015년 국가 7급]

---

사원	주민번호(PK)	성	이름	주소	성별	연봉	부서번호(FK)
부서	부서번호(PK)	부서명	관리자주민번호(FK)	관리시작일			
부양가족	주민번호(FK)	가족이름	관계	생년월일			

---

- ①  $\Pi_{성,이름}((\Pi_{주민번호}(사원) - \Pi_{주민번호}(부양가족)) \bowtie 사원)$
- ②  $\Pi_{성,이름}((\Pi_{주민번호}(사원) \bowtie \Pi_{주민번호}(부양가족)) \times 사원)$
- ③  $\Pi_{성,이름}((\Pi_{주민번호}(사원) \times \Pi_{주민번호}(부양가족)) \bowtie 사원)$
- ④  $\Pi_{성,이름}((\Pi_{주민번호}(사원) \bowtie \Pi_{주민번호}(부양가족)) - 사원)$

☞ 관계대수식

- 
- ①  $\Pi_{성,이름}((\Pi_{주민번호}(사원) - \Pi_{주민번호}(부양가족)) \bowtie 사원)$ 
    - ↓
    - $(\Pi_{주민번호}(사원) - \Pi_{주민번호}(부양가족))$ 
      - "사원"의 주민번호 집합에서 "부양가족"의 주민번호를 뺀다.
      - 부양가족이 없는 사원의 주민번호만 남는다.
    - $(부양가족이 없는 사원의 주민번호) \bowtie 사원$ 
      - (부양가족이 없는 사원의 주민번호)와 "사원"을 조인한다.
      - 자연조인하면, 부양가족이 없는 사원 정보만 남는다.
    - $\Pi_{성,이름}(부양가족이 없는 사원)$ 
      - ↑
      - "부양가족이 없는 사원"에서 성과 이름 컬럼만 추출한다.

15. 다음 두 질의에 대한 설명으로 옳은 것은? (단, 테이블 R(a, b)과 S(b, c)가 있고, 각 애트리뷰트는 NULL을 포함하지 않지만 중복 값을 가지고 있다고 가정한다) [2013년 국가 7급]

q1	SELECT R.a FROM R, S WHERE R.b = S.b;
q2	SELECT R.a FROM R WHERE R.b IN (SELECT S.b FROM S);

- ① q1과 q2의 결과는 같다.
- ② q1의 결과가 항상 q2의 결과를 포함한다.
- ③ q2의 결과가 항상 q1의 결과를 포함한다.
- ④ q1과 q2는 실행할 수 없다.

♣ SQL 비교

• q1 : SELECT R.a FROM R, S WHERE R.b = S.b;

R	
a	b
10	1
20	2
25	2

S	
b	c
1	100
2	200
1	150

→

a
10
20
25
10

• q2 : SELECT R.a FROM R WHERE R.b IN (SELECT S.b FROM S);

R	
a	b
10	1
20	2
25	2

S	
b	c
1	100
2	200
1	150

→

a
10
20
25

- q1의 조인(join) 결과가 q2의 결과를 모두 포함한다.
- 중복 값을 가지고 있으므로 조인 결과가 중복된 값만큼 더 포함한다.
- 해서, q1의 결과가 항상 q2의 결과를 포함한다.



16. SQL 질의문은 동치인 관계대수식으로 변환할 수 있다. 다음 SQL 질의문을 관계대수식으로 변환한 것 중 옳은 것은? (단, 제품 테이블 P, 판매 테이블 S, 집계함수 표현을 위한 관계대수 연산자 기호는 F로 한다) [2018년 국가 7급]

스키마	· S(sId, sDate, pId, sAmount) · P(pId, pName)
SQL	SELECT SUM(sAmount) FROM P, S WHERE S.pId = P.pId AND P.pName='P1' GROUP BY sDate;

- ①  $F_{SUM(sAmount)}(\pi_{P.pName}(\sigma_{P.pName='P1'} \bowtie_{S.pId=P.pId} P))$
- ②  $sDate F_{SUM(sAmount)}(\pi_{S.sAmount}(\sigma_{P.pName='P1'}(S \bowtie_{S.pId=P.pId} P)))$
- ③  $F_{SUM(sAmount)}(S \bowtie_{S.pId=P.pId} (\sigma_{P.pName='P1'}(P)))$
- ④  $sDate F_{SUM(sAmount)}(\sigma_{P.pName='P1'}(P) \bowtie_{P.pId=S.pId} S)$

☞ SQL 질의문은 동치인 관계대수식으로 변환

// Schema 구조

제품 테이블 P		판매 테이블 S			
pId(제품번호)	pName(제품명)	sId(판매번호)	pId(제품번호)	sDate(판매일)	sAmount(판매량)
1	P1	100	1	8월 1일	1000
2	P2	200	1	8월 1일	2000
3	P3	300	2	8월 2일	3000

↳ 속성 sDate별 그룹화에 대한 sAmount의 총합을 구함

·  $sDate F_{SUM(sAmount)}(\sigma_{P.pName='P1'}(P) \bowtie_{P.pId=S.pId} S)$   
 ↓  
 ↳ 제품 테이블 P와 판매 테이블 S를 조인  
 속성 sDate를 기준으로 그룹화

// 그룹화한 집계함수에 대한 관계대수 표현

- 질의문 : 직원 테이블에 대한 각 부서별 급여 합을 구하라?
- SQL : SELECT SUM(급여) FROM 직원 GROUP BY 부서;
- 관계대수 :  $\text{부서} F_{sum(급여)}(\text{직원})$   
 ↓  
 그룹화할 속성(부서)을 F 앞에 기술한다.

17. 다음 릴레이션 R과 S에 대한 조인 연산 결과가 옳지 않은 것은? (단,  $\bowtie_N$ 은 자연조인,  $\bowtie$ 은 왼쪽 외부조인,  $\bowtie$ 은 오른쪽 외부조인,  $\bowtie$ 은 세미조인을 의미한다) [2021년 국가 7급]

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b3	c3
a5	b5	c5

B	C	D
b1	c1	d1
b2	c2	d2
b2	c2	d3
b3	c3	d4
b4	c4	d4

①  $R \bowtie_N S$

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	d2
a2	b2	c2	d3
a3	b3	c3	d4
a4	b3	c3	d4

②  $R \bowtie_{R.B=S.B \text{ and } R.C=S.C} S$

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	d2
a2	b2	c2	d3
a3	b3	c3	d4
a4	b3	c3	d4
a5	b5	c5	null

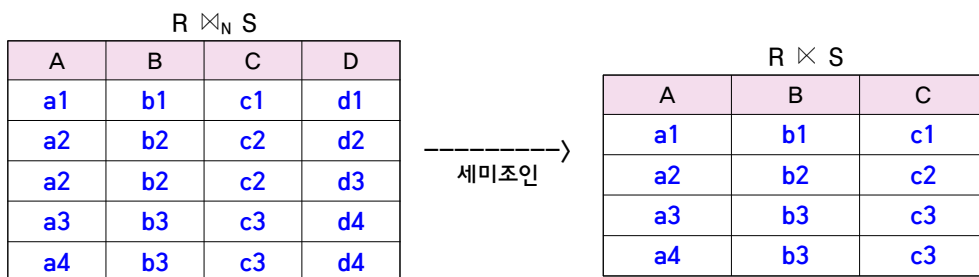
③  $R \bowtie_{R.B=S.B \text{ and } R.C=S.C} S$

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	d2
a2	b2	c2	d3
a3	b3	c3	d4
a4	b3	c3	d4
null	b4	c4	d5

④  $R \bowtie_{R.B=S.B \text{ and } R.C=S.C} S$

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

☞ 세미조인 -  $R \bowtie S$



• 세미조인은 R, S의 자연조인 결과에서 R의 튜플만 선택하는 것이다.

18. 세미조인과 가장 관계없는 것은? [2020년 군무 7급 복원]

- ① 교집합                      ② 셀렉트
- ③ 프로젝트                  ④ 자연조인

☞ 세미조인

// 사원과 부서 테이블의 세미조인

사원		
사번	이름	부서번호
1	홍재연	1
2	이순신	1
3	김유신	null
4	강감찬	3

사원 × 부서

부서	
부서번호	부서명
1	생산
2	영업
3	개발
4	경리



사원 ⋈<sub>N</sub> 부서

사번	이름	부서번호	부서명
1	홍재연	1	생산
2	이순신	1	생산
4	강감찬	3	개발



사원 × 부서

사번	이름	부서번호
1	홍재연	1
2	이순신	1
4	강감찬	3

- 좌측 사원 테이블의 속성만으로 구성
- 우측 부서 테이블의 부서명이 제거된다.
- 속성과 튜플 수는 각각 3개이다.

- 세미조인은 자연조인 결과에서 좌측테이블에 있는 내용만을 선택하는 것과 같다.
- SQL에서 세미조인은 IN이나 Exists를 이용할 수 있다.
- 세미조인 결과를 얻기 위해, SQL에서 내부적으로 자연조인을 실시하는 것은 아니다.

// 연산자 Exists를 사용한 세미조인

Select \* From 사원 Where Exists

(Select \* From 부서 Where 사원.부서번호 = 부서.부서번호);

// 연산자 In을 사용한 세미조인

Select \* From 사원 Where 부서번호 In

(Select 부서번호 From 부서 Where 사원.부서번호 = 부서.부서번호);

// 문제 분석

- 주어진 문제 정답은 1번으로 체크되는 것 같다. 그런데, 엄격하게 말하면 4번도 관계가 없다.
- 그 이유를 적으면, 다음과 같다.

↓ 세미조인 결과를 얻는 방법

- 세미조인은 내부적으로 자연조인을 실시하는 것은 아니다.
- 세미조인은 내부적으로 셀렉트( $\sigma$ )와 프로젝트( $\Pi$ )가 적용된다.

// 세미조인 분석

- 세미조인은 질의 처리를 효율적으로 수행하기 위해서 꼭 필요한 필드(속성)만을 추출해서
- 즉, 데이터 양을 최소화 해서 세미조인을 실시한다.
- 세미조인에서 중요한 것은 데이터 양을 최소로 줄이는 것이다.
- 세미조인을 위해서 내부적으로 자연조인을 실시하는 것은 아니다.
- 만약, 세미조인을 위해서 내부적으로 자연조인을 실시하면 불필요한 데이터도 적용된다.
- 불필요한 데이터가 많이 적용된다는 것은 세미조인을 실시하는 취지에 위배된다.
- SQL에서 IN 또는 Exists를 이용하면, 최소 데이터 양으로 세미조인 결과를 얻을 수 있다.
- 세미조인은 원래의 테이블에 대해 프로젝트(project)와 셀렉트(select)한 결과가 된다.
- 세미조인은 서브쿼리를 사용하여 서브쿼리에 존재하는 데이터만 메인쿼리에서 추출한다.
- 세미조인은 분산질의를 효율적으로 수행하기 위해 도입된 개념이다.
- 세미조인은 네트워크를 통해 전송되는 데이터 양을 줄이고자 하는 개념으로 도입된 것이다.

// 문제에 대한 이의신청

- 먼저, 세미조인은 내부적으로 셀렉트( $\sigma$ )와 프로젝트( $\Pi$ )가 적용된다.
- 세미조인과 관계가 없는 것은 교집합과 자연조인이다. 즉, 정답은 2개가 될 수 있다.
- 그런데, 출제자는 정답을 1번으로 체크한 것으로 예측된다.
- 내부적으로 세미조인은 전혀 자연조인을 이용하지 않는데
- 외관상으로 세미조인이 자연조인을 이용하는 것처럼 보인다고
- 세미조인이 자연조인과 관계가 있는 것으로 출제자는 문제를 제시한 것이다.
- 주어진 문제는 이의신청을 하였다. 지금 이의신청한 결과는 알 수 없다.
- 현재, 군무원 시험은 문제와 정답을 공개하지 않는다.
- 해서, 군무원 시험의 정답이 무엇인지? 정확하게 알 수는 없다.
- 하지만, 충분히 예측은 할 수 있다.
- 이 문제의 정답은 1번으로 결정되어 있을 것으로 예측한다. 해서, 이의신청을 하였다.

19. 다음 관계대수 연산의 수행 결과로 옳은 것은?(단,  $\pi$ 는 프로젝트,  $\sigma$ 는 선택,  $\bowtie_N$ 은 자연 조인을 나타내는 연산자이다) [2014년 계리직]

관계대수 :  $\pi_{\text{고객번호, 상품코드}} (\sigma_{\text{가격} \leq 40} (\text{구매} \bowtie_N \text{상품}))$

구매		상품		
고객번호	상품코드	상품코드	비용	가격
100	P1	P1	20	35
200	P2	P2	50	65
100	P3	P3	10	27
100	P2	P4	20	45
200	P1	P5	30	50
300	P2	P6	40	55

- | <p>①</p> <table border="1"> <thead> <tr> <th>고객번호</th> <th>상품코드</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>P1</td> </tr> <tr> <td>100</td> <td>P3</td> </tr> </tbody> </table>                                     | 고객번호 | 상품코드 | 100 | P1 | 100 | P3 | <p>②</p> <table border="1"> <thead> <tr> <th>고객번호</th> <th>상품코드</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>P1</td> </tr> <tr> <td>200</td> <td>P1</td> </tr> </tbody> </table> | 고객번호 | 상품코드                                                                                                                                                                                                                      | 100  | P1   | 200 | P1 |     |    |     |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|-----|----|-----|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|-----|----|-----|----|-----|----|
| 고객번호                                                                                                                                                                                                                      | 상품코드 |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 100                                                                                                                                                                                                                       | P1   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 100                                                                                                                                                                                                                       | P3   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 고객번호                                                                                                                                                                                                                      | 상품코드 |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 100                                                                                                                                                                                                                       | P1   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 200                                                                                                                                                                                                                       | P1   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| <p>③</p> <table border="1"> <thead> <tr> <th>고객번호</th> <th>상품코드</th> </tr> </thead> <tbody> <tr> <td>100</td> <td>P1</td> </tr> <tr> <td>100</td> <td>P3</td> </tr> <tr> <td>200</td> <td>P1</td> </tr> </tbody> </table> | 고객번호 | 상품코드 | 100 | P1 | 100 | P3 | 200                                                                                                                                                                                   | P1   | <p>④</p> <table border="1"> <thead> <tr> <th>고객번호</th> <th>상품코드</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>P2</td> </tr> <tr> <td>100</td> <td>P2</td> </tr> <tr> <td>300</td> <td>P2</td> </tr> </tbody> </table> | 고객번호 | 상품코드 | 200 | P2 | 100 | P2 | 300 | P2 |
| 고객번호                                                                                                                                                                                                                      | 상품코드 |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 100                                                                                                                                                                                                                       | P1   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 100                                                                                                                                                                                                                       | P3   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 200                                                                                                                                                                                                                       | P1   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 고객번호                                                                                                                                                                                                                      | 상품코드 |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 200                                                                                                                                                                                                                       | P2   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 100                                                                                                                                                                                                                       | P2   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |
| 300                                                                                                                                                                                                                       | P2   |      |     |    |     |    |                                                                                                                                                                                       |      |                                                                                                                                                                                                                           |      |      |     |    |     |    |     |    |

☞ 관계대수의 관계연산자

// 자연조인(natural join) :  $\text{구매} \bowtie_N \text{상품}$

구매  $\bowtie_N$  상품

고객번호	상품코드	비용	가격
100	P1	20	35
200	P2	50	65
100	P3	10	27
100	P2	50	65
200	P1	20	35
300	P2	50	65

• 자연조인은 조인 결과 존재하는 중복되는 속성을 모두 제거해 주는 연산이다.

// Select(선택,  $\sigma$ ) :  $\sigma_{\text{가격} \leq 40}$  (구매  $\bowtie_N$  상품)

$\sigma_{\text{가격} \leq 40}$  (구매  $\bowtie_N$  상품)

고객번호	상품코드	비용	가격
100	P1	20	35
100	P3	10	27
200	P1	20	35

- 가격  $\leq 40$ 인 행을 추출한다.

// Project(투사,  $\Pi$ ) :  $\Pi_{\text{고객번호, 상품코드}}$  ( $\sigma_{\text{가격} \leq 40}$  (구매  $\bowtie_N$  상품))

고객번호	상품코드
100	P1
100	P3
200	P1

- 고객번호, 상품코드 열을 추출한다.

정답 : ③

20. 데이터베이스를 분산시키기 위한 방법으로 수직 분할(vertical fragmentation), 수평 분할(horizontal fragmentation), 그리고 중복이 있다. 이 방법들에 대한 설명으로 옳지 않은 것은?  
[2011년 국가 7급]

- ① 2단계완료규약(2-Phase commit protocol)은 수직 분할, 수평 분할 또는 중복으로 분산된 데이터베이스의 일관성을 유지하기 위한 트랜잭션의 수행 방법이다.
- ② 데이터베이스를 중복으로 분산시키면 검색과 갱신 성능이 향상된다.
- ③ 테이블 R을 R1, R2, ..., Rn으로 수평 분할하였다면, 분산 데이터베이스 시스템에 주어진 선택 질의  $\sigma$ 은  $\sigma(R1) \cup \sigma(R2) \cup \dots \cup \sigma(Rn)$ 로 수행된다. (단, U는 합집합 연산자를 의미한다)
- ④ 테이블 R을 R1, R2, ..., Rn으로 수직 분할하였다면,  $R=R1 \bowtie R2 \dots \bowtie Rn$ 으로 표현될 수 있다. (단,  $\bowtie$ 는 조인 연산자를 의미한다)

☞ 데이터베이스 분산

- 데이터베이스를 중복으로 분산시키면 검색과 갱신 성능이 향상된다.(×)  
→ 데이터가 중복으로 분산되면 갱신 성능은 저하된다.

정답 : ②

21. 다음 릴레이션 R과 S에 대한 관계대수식의 결과로 옳지 않은 것은? (단, 기호  $\bowtie_N$ 은 자연조인을 의미하고, 릴레이션에서 A, B, C, D는 속성 이름이다) [2023년 국가 7급]

R		
A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3
a4	b4	c5

S		
B	C	D
b1	c1	d1
b3	c2	d2
b3	c3	d3
b5	c5	d5

①  $\sigma_{B='b3'}(S)$

B	C	D
b3	c2	d2
b3	c3	d3

②  $\pi_{B,C}(R \bowtie_N S)$

B	C
b1	c1
b3	c3

③  $\pi_{A,B}(\sigma_{C='c3'}(R))$

A	B
a3	b3

④  $\pi_{A,C}(\sigma_{B='b3'}(R \bowtie_N S))$

A	C
a2	c2

☞ 자연조인

④  $\pi_{A,C}(\sigma_{B='b3'}(R \bowtie_N S))$

↓  $R \bowtie_N S$

↓ Select \* From R Natural Join S;

A	B	C	D
a1	b1	c1	d1
a3	b3	c3	d3

↓  $\sigma_{B='b3'}(R \bowtie_N S)$

↓ Select \* From R Natural Join S Where B = 'b3';

A	B	C	D
a3	b3	c3	d3

↓  $\pi_{A,C}(\sigma_{B='b3'}(R \bowtie_N S))$

↓ Select A, C From R Natural Join S Where B = 'b3';

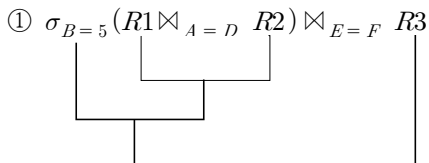
A	C
a3	c3

22. 다음 릴레이션 R1, R2, R3에 대한 관계대수식의 의미가 다른 것은? [2023년 국가 7급]

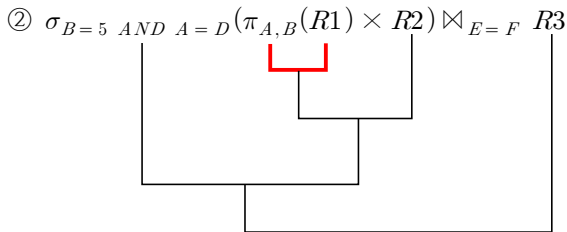
R1(A, B, C)	R2(D, E)	R3(F, G, H)
-------------	----------	-------------

- ①  $\sigma_{B=5}(R1 \bowtie_{A=D} R2) \bowtie_{E=F} R3$
- ②  $\sigma_{B=5 \text{ AND } A=D}(\pi_{A,B}(R1) \times R2) \bowtie_{E=F} R3$
- ③  $\sigma_{A=D}(\sigma_{B=5}(R1) \times R2) \bowtie_{E=F} R3$
- ④  $\sigma_{B=5}((R2 \bowtie_{E=F} R3) \bowtie_{A=D} R1)$

☞ 관계대수식의 의미가 다른 것



- 조인 결과에서 조건 B=5 AND A=D AND E=F를 만족하는 레코드가 출력된다.
- 그리고, R1(A, B, C) R2(D, E) R3(F, G, H)가 가지는 모든 속성이 출력된다.



- 조인 결과에서 조건 B=5 AND A=D AND E=F를 만족하는 레코드가 출력된다.(①과 같음)
- 연산  $\pi_{A,B}(R1)$ 에서 R1(A, B, C)의 속성 중에 속성 C는 제외된다.(①과 다름)

③ ④는 ①과 같다.



23. 음식점 관리시스템에서 고객(customer), 주문(order), 음식(food)을 관리하기 위해 릴레이션을 다음과 같이 설계하였다. 이 릴레이션에 대하여, SQL문에 대응하는 관계대수식으로 옳지 않은 것은? (단, 밑줄 친 속성은 기본키이고, 관계대수 연산자 중에서  $\rho$ 는 재명명(rename) 연산자이며,  $\mathcal{J}$ 은 집계함수(aggregate function) 연산을 정의하는 연산자이다) [2023년 국가 7급]

```
customer(name, gender, phone, address, year)
order(cname, fname, price, date)
food(name, price, stock)
```

①	SQL	SELECT name FROM food AS f WHERE f.stock < 10 OR f.price > 20000;
관계대수		$\pi_{name}(\sigma_{f.stock < 10 \text{ OR } f.price > 20000}(\rho_f(food)))$
②	SQL	SELECT c.name, c.phone FROM customer AS c, order AS o, food AS f WHERE o.fname = f.name AND c.name = o.cname AND f.price > 10000;
관계대수		$\pi_{c.name, c.phone} \left( \begin{matrix} \sigma_{o.fname = f.name \\ \text{AND } c.name = o.cname} \left( \begin{matrix} \rho_c(customer) \\ \times \rho_o(order) \\ \times \rho_f(food) \end{matrix} \right) \right)$
③	SQL	SELECT name, SUM(price) AS orderPrice FROM customer, order WHERE name = cname GROUP BY name;
관계대수		$\rho_{(name, orderPrice)}(name \mathcal{J}_{SUM(price)}(customer \bowtie_{name=cname} order))$
④	SQL	SELECT name, AVG(food.price) AS avgPrice FROM food, order GROUP BY name;
관계대수		$\rho_{(name, avgPrice)}(name \mathcal{J}_{AVG(food.price)}(food \times order))$

☞ SQL문에 대응하는 관계대수식

- 먼저, 항목 ②는 SQL문과 관계대수식의 결과는 다르게 검색된다.
- SQL문에는 조건 f.price>10000이 있고, 관계대수식에는 조건 f.price>10000이 없다.

재명명 연산 rename operation	<ul style="list-style-type: none"> <li>• 단순히 테이블명이나 속성명을 변경하는 연산이다.</li> <li>• 수학 기호 <math>\rho</math>(rho)를 사용한다.</li> <li>• 예 : <math>\rho_f(\text{food}) \leftarrow</math> 테이블명 food을 f로 변경</li> <li>• SQL문 : FROM food AS f</li> </ul>
----------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

R		
a	b	c
a1	b1	c1
a2	b2	c2

 $\rho_{S(d,e,f)}(R)$ 

S		
d	e	f
a1	b1	c1
a2	b2	c2

- 테이블명을 R에서 S로 변경하고, 속성 a, b, c를 d, e, f로 재명명

집계함수 연산 $\bowtie$	<ul style="list-style-type: none"> <li>• 예 : <math>(\text{name} \bowtie \text{SUM}(\text{price}))</math> ↓</li> <li>• 속성 name으로 그룹화하고(GROUP BY name)</li> <li>• 집계함수 SUM(price) 연산한다는 의미이다.</li> </ul>
----------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

SQL	<pre>SELECT c.name, c.phone FROM customer AS c, order AS o, food AS f WHERE o.fname = f.name AND c.name = o.cname AND f.price &gt; 10000;</pre>
결과	<p>조인 결과에서 o.fname = f.name AND c.name = o.cname이면서 f.price &gt; 10000인 레코드의 c.name, c.phone이 검색된다.</p>
관계대수	$\pi_{c.name, c.phone} \left( \begin{matrix} \sigma_{o.fname = f.name} \\ AND c.name = o.cname \end{matrix} \left( \begin{matrix} \rho_c(customer) \\ \times \rho_o(order) \\ \times \rho_f(food) \end{matrix} \right) \right)$
② 결과	<p>카티션곱(<math>\times</math>) 결과에서 o.fname = f.name AND c.name = o.cname인 레코드의 c.name, c.phone이 검색된다.</p>

- 카티션곱은 where절에 기술하는 조인(join) 조건이 없는 경우이다.
- 조건이 없으므로 각 테이블에 대한 모든 데이터를 조인하게 된다.
- 카티션곱은 where절에 조인 조건이 잘못 기술되었을 때도 발생한다.
- 카티션곱은 조인에 참조되는 테이블 행수를 모두 곱한 값의 결과가 만들어진다.

24. 다음의 릴레이션 R1과 R2에 대한 관계대수 R1 ÷ R2의 결과 릴레이션으로 옳은 것은? (단, 릴레이션 R1.C2와 R2.C2는 동일한 도메인에서 정의되었다) [2010년 국가 7급]

C1	C2
1	A
2	C
1	E
1	B
3	J
4	R
3	B
2	B
5	R
3	A
4	A

C2
A
B

① 

C1
3
4

② 

C1	C2
1	A
2	B

③ 

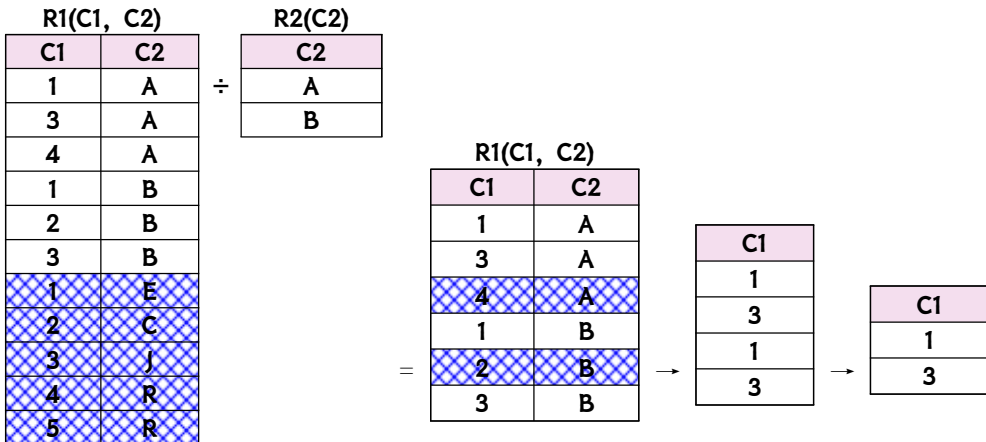
C1
1
3

④ 

C1
2
3

♣ 디비전(division, ÷)

- 디비전은 먼저, 두 릴레이션의 공통 속성값을 가지는 튜플을 구한다.
- 그리고, 그 나머지 속성에 대해서 같은 속성값을 가지는 것만 남기면 된다.



정답 : ③

25. 데이터베이스 스키마(database schema)가 <보기>와 같을 때, 문법적으로 가장 옳지 않은 관계대수(relational algebra) 식은? [2021년 서울 7급]

-----<보기>-----  
professor(id, name, department, tel)  
teach(id, course\_id, section\_id, semester, year)  
section(course\_id, section\_id, semester, year, classroom\_no)  
-----

- ①  $\pi_{name}(\sigma_{department = 'Biology'}(\text{professor}))$
- ②  $\pi_{id, name, tel}(\text{professor})$
- ③  $\pi_{course\_id}(\sigma_{semester = 'Fall' \wedge year = 2020}(\text{section})) \cup \pi_{course\_id, section\_id}(\sigma_{semester = 'Spring' \wedge year = 2021}(\text{section}))$
- ④  $\sigma_{professor.id = teach.id}(\text{professor} \times \text{teach})$

☞ 관계대수식 - 합병 가능 조건

- 두 테이블의 **차수가 같아야** 한다. - 항목 ③은 차수가 다르다.
- 여기서, **차수가 같다는** 것은 'Select 학번, 이름'에 기술되는 속성 수가 같아야 하는 것이다.
- 두 테이블에서 서로 대응되는 속성의 **도메인이 같아야** 한다.

정답 : ③

26. 조인 연산을 구현하는 방법에 대한 설명으로 옳지 않은 것은? [2013년 국가 7급]

- ① 정렬-합병 조인방법은 두 릴레이션이 각각 조인 애트리뷰트값에 따라 정렬되어 있는 경우에 유용하다.
- ② 중첩루프(nested loop) 조인방법은 모든 조인 비교연산자를 처리할 수 있다.
- ③ 세타조인(theta-join)의 경우 메모리가 충분하면 해시조인 방법이 유리하다.
- ④ 중첩루프 조인방법에서 내부 릴레이션의 조인 애트리뷰트에 인덱스가 존재할 경우 조인 연산을 효율적으로 처리할 수 있다.

☞ 조인 연산

- 세타조인(theta-join)의 경우 메모리가 충분하면 해시조인 방법이 유리하다.(x)
  - 세타조인은 모든 연산자 사용이 가능하지만,
  - 해시조인은 동등 비교(equi-join) 조인만 적용 가능하다.
  - 즉, equi만 가능한 해시조인에서 모든 연산자 사용이 가능한 세타조인은 불가능하다.

정답 : ③

27. <보기 1>의 두 릴레이션 C, O에 대한 연산 후 <보기 2>의 결과 릴레이션을 얻기 위한 연산자로 가장 옳은 것은? [2021년 서울 7급]

-----<보기 1>-----

C

ID	NAME	AGE	GRADE
AAA	TOM	22	S
BBB	JAMES	23	A
CCC	AMY	21	B
DDD	ANNE	20	B

O

NO	ID	ORDER	AMOUNT
10001	AAA	BOOK	2
10002	BBB	PIZZA	1
10003	CCC	SNACK	10

-----<보기 2>-----

결과

ID	NAME	AGE	GRADE	NO	ORDER	AMOUNT
AAA	TOM	22	S	10001	BOOK	2
BBB	JAMES	23	A	10003	SNACK	10
CCC	AMY	21	B	10002	PIZZA	1

- ①  $\sigma$                       ②  $\bowtie$                       ③  $\div$                       ④  $\pi$

♣ **조인** -  $\bowtie$

• 결과 릴레이션은 조인 연산 결과이다. C  $\bowtie_{ID=ID}$  O

정답 : ②

28. 관계 데이터베이스 스키마 STUDENT(SNO, NAME, AGE)에 대하여 다음과 같은 SQL 질의 문장을 사용한다고 할 때, 이 SQL 문장과 동일한 의미의 관계대수식은? (단, STUDENT 스키마에서 밑줄 친 속성은 기본키 속성을, 관계대수식에서 사용하는 관계대수 연산자 기호  $\pi$ 는 프로젝트 연산자를,  $\sigma$ 는 선택 연산자를 나타낸다) [2020년 지방 9급]

----<SQL 질의문>-----

SELECT SNO, NAME  
FROM STUDENT WHERE AGE > 20;

- ①  $\sigma_{SNO.NAME}(\pi_{AGE > 20}(STUDENT))$
- ②  $\pi_{SNO.NAME}(\sigma_{AGE > 20}(STUDENT))$
- ③  $\sigma_{AGE > 20}(\pi_{SNO.NAME}(STUDENT))$
- ④  $\pi_{AGE > 20}(\sigma_{SNO.NAME}(STUDENT))$

♣ SQL 질의문을 관계대수식으로 변경

---

· SQL 질의문 : SELECT SNO, NAME FROM STUDENT WHERE AGE > 20;

↓

↓ 관계대수식으로

↓ Select( $\sigma$ ) - 수평적 선택 :  $\sigma_{AGE > 20}(STUDENT)$

↓ Project( $\pi$ ) - 수직적 추출 :  $\pi_{SNO.NAME}$

↓

· 관계대수식 :  $\pi_{SNO.NAME}(\sigma_{AGE > 20}(STUDENT))$

---

정답 : ②