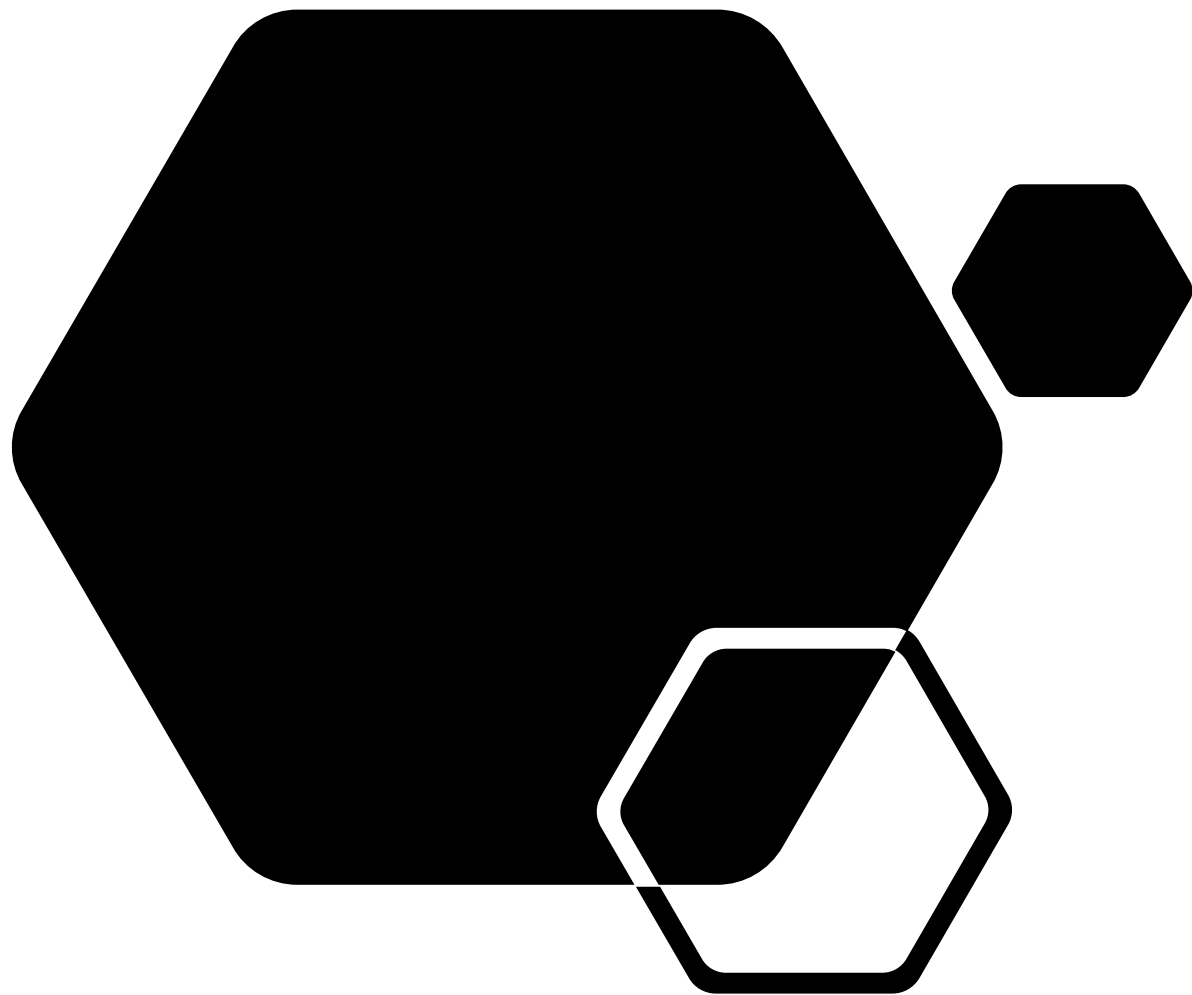


relational algebra

관계대수



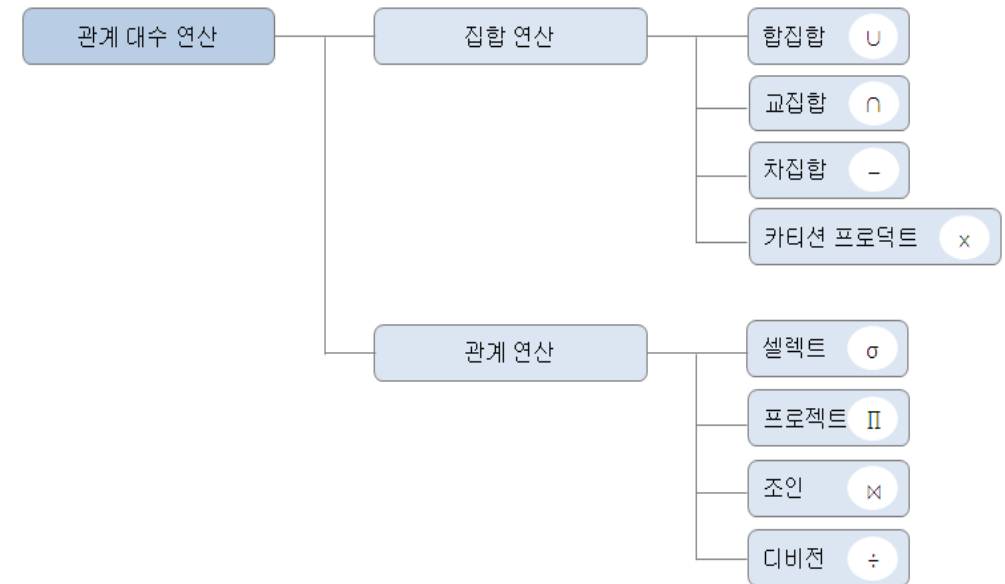
- 관계형 대수(relational algebra)

- 관계형 데이터 언어는 관계형 대수를 기초로 만들어진다
- 관계형 대수는 릴레이션들을 처리하기 위한 연산(operation)들

- 단일 릴레이션(단항연산자)이나 두 개의 릴레이션(이항연산자)을 입력으로 받아 하나의 결과 릴레이션을 생성함
- 질의를 기술하는데 사용함
- 릴레이션의 대한 연산의 결과도 릴레이션

- 관계 대수 연산의 종류

- 수학적 집합 연산
- 관계 데이터베이스를 위한 연산



- 집합 연산자

- 곱집합을 제외한 합집합, 교집합, 차집합 연산은 두 릴레이션 합병 가능
 - 합병 가능(union-compatible)이란 두 릴레이션 R과 S가 있을 때, 두 릴레이션의 차수(속성의 수)가 같고 대응되는 애트리뷰트의 도메인이 같다는 것을 의미한다

- 합집합 (Union, U)

- 두 개의 릴레이션을 합하여 하나의 릴레이션을 반환

- 수학적 정의

- $R \cup S = \{ t \mid t \in R \vee t \in S \}$

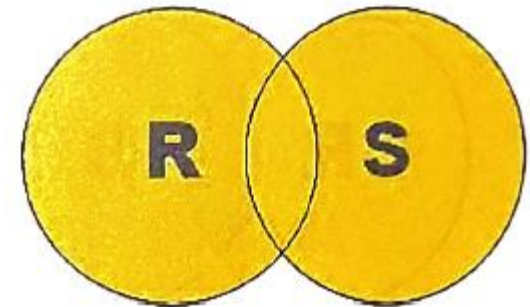
- 합집합의 예

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 |

```
SELECT 번호, 이름      FROM R
UNION [ALL]
SELECT 번호, 이름      FROM S;
```

- 교집합 (Intersect, n)

- 두 릴레이션이 공통으로 가지 있는 튜플을 반환

- 수학적 정의

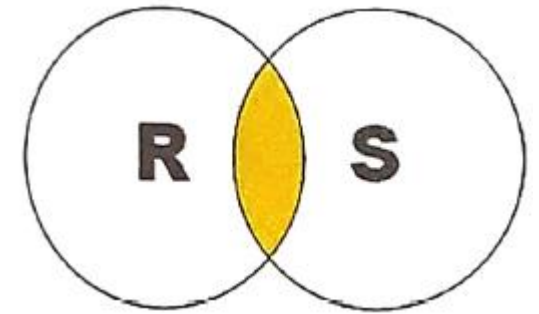
- $R \cap S = \{t \mid t \in R \wedge t \in S\}$

- 교집합의 예

| R | | S | |
|-----|-----|-----|-----|
| 번호 | 이름 | 번호 | 이름 |
| 100 | 정소화 | 100 | 정소화 |
| 200 | 김선우 | 101 | 채광주 |
| 300 | 고명석 | 102 | 김수진 |

↓ 교집합 연산

| R ∩ S | |
|-------|-----|
| 번호 | 이름 |
| 100 | 정소화 |



INNER JOIN....

```
SELECT 번호, 이름 FROM R
INTERSECT
SELECT 번호, 이름 FROM S;
```

- 차집합 (Difference, -)

- 첫번째 릴레이션에는 속하고 두번째 릴레이션에는 속하는 않는 튜플을 반환

- 수학적 정의

- $R - S = \{t \mid t \in R \wedge t \notin S\}$

- 차집합의 예

R

| 번호 | 이름 |
|-----|-----|
| 100 | 정소화 |
| 200 | 김선우 |
| 300 | 고명석 |

S

| 번호 | 이름 |
|-----|-----|
| 100 | 정소화 |
| 101 | 채광주 |
| 102 | 김수진 |



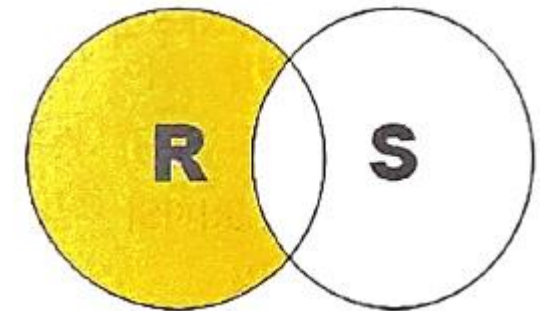
차집합 연산

R-S

| 번호 | 이름 |
|-----|-----|
| 200 | 김선우 |
| 300 | 고명석 |

S-R

| 번호 | 이름 |
|-----|-----|
| 101 | 채광주 |
| 102 | 김수진 |



```
SELECT 번호, 이름 FROM R  
EXCEPT  
SELECT 번호, 이름 FROM S;  
  
SELECT 번호, 이름 FROM S  
EXCEPT  
SELECT 번호, 이름 FROM R;
```

- 곱집합 (Cartesian Product, \times)

- 다른 집합 연산자와는 달리 곱집합은 합병 가능할 필요는 없다.

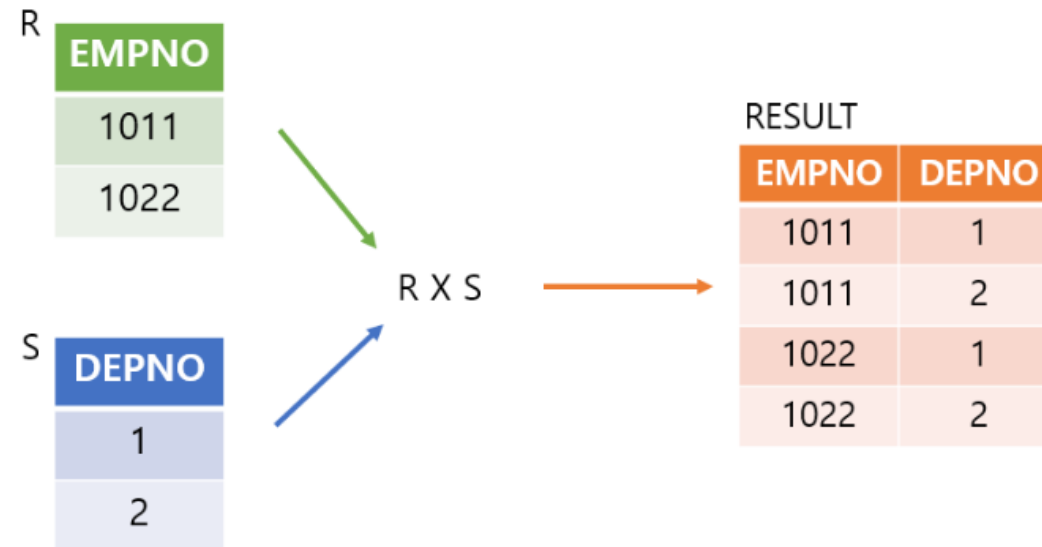
- 결과 릴레이션의 차수 : 두 릴레이션 속성의 합

- 결과 릴레이션의 카디널리티 : 두 릴레이션 튜플의 곱

- 수학적 정의

- $R \times S = \{rs | r \in R \wedge s \in S\}$

- 곱집합의 예



- 곱집합 (Cartesian Product, \times)

Student(name, score, dept)

Department(name, chair, phone, average)

| | |
|------|---|
| 관계대수 | $\Pi_{\text{name, phone}}(\text{Student} \times \text{Department})$ |
| SQL | SELECT Student.name, Department.phone FROM Student, Department |
| | SELECT R.번호, R.이름, S.번호, S.이름 FROM R, S; |

- 선택(트) 연산자 σ
 - 원하는 데이터를 행(튜플)으로 출력함
 - 관계 대수식
 - $\sigma_{\langle \text{선택조건} \rangle}(R)$
 - 데이터 언어적 표현 : 릴레이션 where 조건식

- 선택(트) 연산자 σ

학생 릴레이션에서 학과가 "컴퓨터"인 튜플을 찾아라.

| | |
|------|---|
| 관계대수 | $\sigma_{\text{학과}='컴퓨터'}(\text{학생})$ |
| SQL | <pre>SELECT * FROM 학생 WHERE 학과='컴퓨터';</pre> |

학생 릴레이션에서 나이가 30보다 많고 성별이 "남"인 튜플을 찾아라.

| | |
|------|--|
| 관계대수 | $\sigma_{\text{나이}>30 \wedge \text{성별}='남'}(\text{학생})$ |
| SQL | <pre>SELECT * FROM 학생 WHERE 나이>30 AND 성별='남';</pre> |

- 프로젝션 연산자 Π

- 원하는 데이터를 수직적으로 출력함
- 결과 릴레이션은 애트리뷰트 리스트에 명시된 R의 애트리뷰트들만 가짐
- 선택의 결과 릴레이션에는 중복 튜플이 존재할 수 없지만, 프로젝션 연산의 결과 릴레이션에는 중복된 튜플들이 존재할 수 있음 -> 중복 제거
- 관계 대수식
 - $\Pi_{\langle \text{애트리뷰트 리스트} \rangle}(R)$
 - 데이터 언어적 표현 : 릴레이션[속성리스트]
 - $\Pi_{\text{TITLE, SALARY}}(\text{EMP}) \gg \text{EMP}[\text{TITLE, SALARY}]$

■ 프로젝션 연산자 π

생산품 릴레이션에서 판매가가 8,000원을 초과하는 모든 생산품코드를 찾아라

| | |
|------|--|
| 관계대수 | $\pi_{\text{생산품코드}}(\sigma_{\text{판매가} > 8000}(\text{생산품}))$ |
|------|--|

| | |
|-----|---|
| SQL | SELECT 생산품코드 FROM 생산품 WHERE 판매가 > 8000; |
|-----|---|

직급이 사원인 이름과 월급을 검색

| | |
|------|---|
| 관계대수 | $\Pi_{\text{EMPNAME, SALARY}}(\sigma_{\text{TITLE} = \text{'사원'}}(\text{EMP}))$ |
|------|---|

| | |
|-----|---|
| SQL | SELECT EMPNAME, SALARY FROM EMP WHERE TITLE = '사원'; |
|-----|---|

■ 조인(join) 연산자 ⋈

대출, 고객 릴레이션에서 “혜화” 지점에 대출이 있는 모든 고객들의 이름을 찾아라.

| | |
|------|---|
| 관계대수 | $\pi_{\text{이름}}((\sigma_{\text{지점명}=\text{“혜화”}}(\text{대출})) \bowtie \text{고객})$ |
| SQL | SELECT 고객.이름 FROM 고객, 대출 WHERE 고객.대출번호=대출.대출번호 AND 대출.지점명='혜화'; |

- 조인(join) 연산자 \bowtie

- 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 : $R1 \bowtie R2$
- 조인 연산의 구분
 - 기본연산 : 세타조인(\bowtie_{θ}), 동등조인(\bowtie), 자연조인(\bowtie_N)
 - 확장된 조인 연산 : 세미조인($\bowtie, \bowtie_{\text{semi}}$), 외부조인($\bowtie_{\text{left}}, \bowtie_{\text{right}}, \bowtie_{\text{full}}$)

- 자연조인(natural join)

- 동등조인에서 조인에 참여한 속성이 두 번 나오지 않도록 두 번째 속성을 제거한 결과를 반환함

- 형식 : $R \bowtie N(r, s) S$

고객과 고객의 주문 사항을 모두 보여주되 같은 속성은 한 번만 표시하시오.

관계대수

고객 \bowtie N(고객.고객번호=주문.고객번호) 주문

SQL

```
SELECT *  
FROM 고객, 주문  
WHERE 고객.고객번호=주문.고객번호;
```

- 다음 세 개의 릴레이션 스키마를 보고 각 물음에 답하시오.

학생(학번, 이름, 학년)

과목(과목번호, 과목이름)

수강(학번, 과목번호, 중간성적, 기말성적, 학점)

1. 모든 과목의 이름을 검색하는 질의문을 관계 대수로 표현하시오.
2. 1학년 학생의 학번과 이름을 검색하는 질의문을 관계 대수로 표현하시오.
3. 중간성적이 80점 이상이고 기말성적이 70점 이상인 학생의 학번과 수강한 과목번호, 학점을 검색하는 질의문을 관계 대수로 표현하시오.
4. 3번 과목에서 A0 학점을 받은 학생의 이름과 학년을 검색하는 질의문을 관계 대수로 표현하시오.