



Chapter 6 Review

(Formal Relational Query Languages)



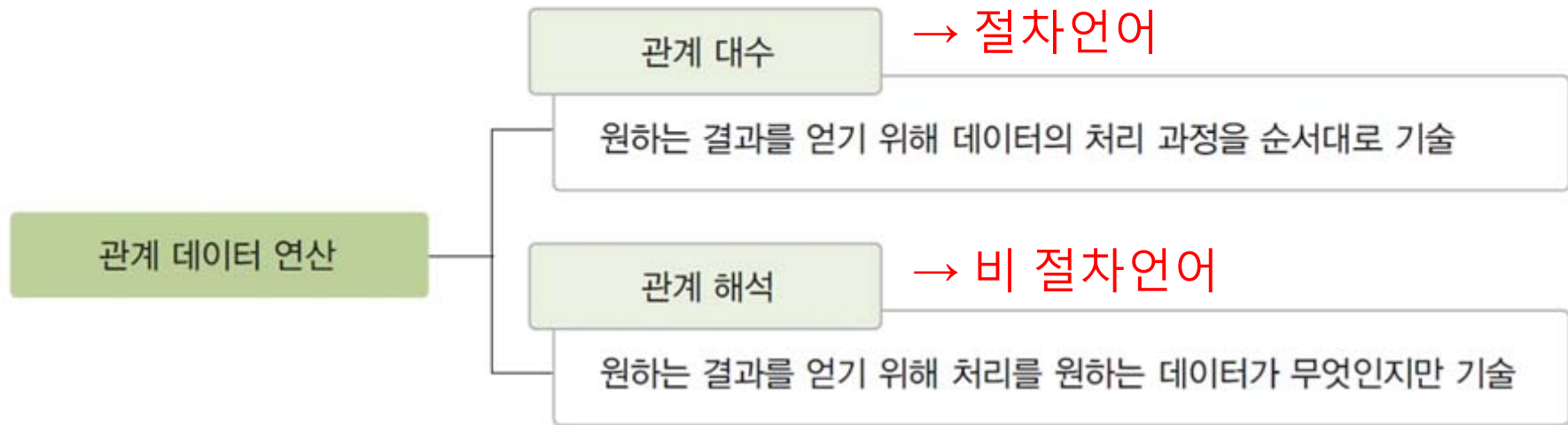
Relational Algebra vs. Relational Calculus

■ 정형 관계 질의 언어 (Formal Relational Query Language)

- 원하는 데이터를 얻기 위해 릴레이션에 필요한 처리 요구를 수행하는 것으로 데이터베이스 시스템의 구성 요소 중 데이터 언어의 역할을 함
- 데이터 관계 모델 연산의 종류 : 관계 대수와 관계 해석이 있음

- 기능과 표현력 측면에서 능력이 동등함

관계대수로 혹은 관계해석으로
어떻게 잘 표현하느냐가 문제로 나옴, 개념문제 안나옴



■ 관계적 완전성(Relationally Completeness)

- 어떤 관계 질의어를 관계 대수 또는 관계 해석으로 표현할 수 있으면 "관계적으로 완전(relationally complete)하다"라고 함









Operators of Relational Algebra

차집합 연산 많이나옴(이때 속성의 갯수를 맞춰야함)

■ 관계대수 기본 연산자 (6 Basic Operators)

| 연산자 종류 | 대상 | 연산자 이름 | 기호 | 설명 |
|--------|----|--------------------------|----------|------------------------|
| 기본 | 단항 | Selection | σ | 릴레이션에서 조건에 만족하는 튜플을 선택 |
| 기본 | 단항 | Projection | π | 릴레이션의 속성을 선택 |
| 기본 | 이항 | Union | \cup | 두 릴레이션의 합집합 |
| 기본 | 이항 | Set Difference | $-$ | 두 릴레이션의 차집합 |
| 기본 | 이항 | Cartesian Product | \times | 두 릴레이션에 속한 모든 튜플의 집합 |
| 기본 | 단항 | Rename | ρ | 릴레이션이나 속성의 이름을 변경 |

■ 관계대수 추가 연산자 아우터 조인은 안나옴

| 연산자 종류 | 대상 | 연산자 이름 | | 기호 | 설명 | |
|--------|----|------------------|---------|---|---|---|
| 추가 | 이항 | Set Intersection | |  | 두 릴레이션의 교집합 | |
| 추가 | 이항 | Assignment | | \leftarrow | 릴레이션 결과 값을 일시적인 변수 형태로 표현 | |
| 추가 | 이항 | 조인 | Natural | |  | 두 릴레이션 간의 같은 속성을 기준으로 조인 (중복 속성 제거) |
| | | | Theta | |  | 두 릴레이션 간의 비교 조건에 만족하는 집합 |
| | | | Outer | left |  | <ul style="list-style-type: none">자연 조인 후 각각 왼쪽(left), 오른쪽(right), 양쪽(full)의 모든 값을 결과로 추출조인이 실패(또는 값이 없을 경우)한 쪽의 값을 NULL로 채움 |
| | | | | right |  | |
| | | | | full |  | |



Operators of Relational Algebra

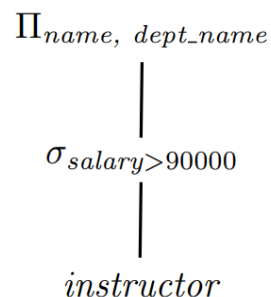
■ 관계대수 확장 연산자

| 연산자 종류 | 대상 | 연산자 이름 | 기호 | 설명 |
|--------|----|-------------------------------|---------------------------------|--|
| 확장 | 단항 | Generalized Projection | $\Pi_{F_1, F_2, \dots, F_n}(E)$ | Projection 연산에 산술연산(Arithmetic Operation)을 추가 |
| 확장 | 단항 | Aggregate Functions | G | Sum, Average, Max, Min, Count와 같은 집계함수를 관계대수로 표현 |
| 확장 | 이항 | Division | \div | 부모 릴레이션에 포함된 튜플의 값을 모두 갖고 있는 튜플을 분자 릴레이션에서 추출 |

집계함수 쓸때 최종결과물이 아니면 집계함수 쓴부분을 이름 재정의해줘야함

■ 관계대수 식 표현 방법

● 수식트리 (Expression Tree) 활용



- represents

$$\Pi_{name, dept_name}(\sigma_{salary > 90000}(instructor))$$

1. 정보를 추출하고자 하는 Relation을 선택 또는 만듦
(FROM 절; 스키마의 의미를 잘 보도록!!!)
* 조인, 차집합, 합집합 등으로 relation을 결합함
2. 추출하고자 하는 정보가 담긴 튜플만 선택
(Where 절; 조건에 해당하는 튜플만 선택 함)
3. 원하는 데이터 추출
(Select 절: 질의에서 원하는 데이터를 출력)



참고: 합집합, 차집합, 교집합

■ 합집합, 교집합, 차집합은 합병 가능(union-compatible)하기 위해 아래의 2가지 조건을 만족해야 함

- 두 릴레이션의 **차수(속성 개수)**가 같아야 함
- 두 릴레이션에서 서로 대응되는 속성의 도메인이 같아야 함, 단 도메인이 같으면 이름은 달라도 된다.

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

- 속성이 모두 3개로 차수가 같다.(O)
- 나이(INT)와 직위(CHAR) 도메인이 다름 (X)

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



Tuple Relational Calculus

안나올수있음(하지만 100%참은 아니니 알아두기)

■ 튜플 관계 해석

- 원하는 릴레이션을 튜플해석식(Tuple calculus expression)으로 정의할 수 있는 표기법

■ 튜플 해석식의 구성 요소

$$\{t \mid t \in \text{EMPLOYEE} \wedge t[\text{SALARY}] > 50000\}$$

- 튜플 변수 t
- $t[A]$: 튜플변수 t 가 나타내는 튜플의 어떤 Attribute A 의 값
- 막대 (|) 왼편에 나온 튜플변수 t 는 목표 튜플이며, 막대(|) 오른편에 명세 된 조건을 만족하는 결과로 추출됨

$\text{employee}(\underline{\text{person_name}}, \text{street}, \text{city})$
 $\text{works}(\underline{\text{person_name}}, \text{company_name}, \text{salary})$
 $\text{company}(\underline{\text{company_name}}, \text{city})$
 $\text{manages}(\underline{\text{person_name}}, \text{manager_name})$

■ Question

- Find the names of all employees who work for First Bank Corporation ?

$$\{t \mid \text{some } s \text{ in works } (t[\text{person_name}] = s[\text{person_name}] \wedge s[\text{company_name}] = \text{"First Bank Corporation"})\}$$

- Find the names and cities of residence of all employees who work for First Bank Corporation ?

$$\{t \mid \text{some } r \text{ in Employee } (\text{some } s \text{ in works } (t[\text{person_name}] = r[\text{person_name}] \wedge t[\text{city}] = r[\text{city}] \wedge r[\text{person_name}] \wedge s[\text{company_name}] = \text{"First Bank"}))\}$$



Domain Relational Calculus

■ 도메인 관계 해석

- 원하는 릴레이션을 도메인 해석식(Domain calculus expression)으로 정의할 수 있는 표기법

■ 도메인 해석식의 구성 요소

$$\{ \langle x_1, x_2, \dots, x_n \rangle \mid P(x_1, x_2, \dots, x_n) \}$$

- 도메인 변수 x_1, x_2, \dots, x_n : 지정된 Attribute의 도메인을 값으로 취하는 변수
- 막대 (|) 왼편에 나온 도메인 변수들은 목표 리스트이며, 막대 (|) 오른편에 명세 된 조건을 만족하는 도메인 값으로 만들어지는 튜플

employee (person_name, street, city)
works (person_name, company_name, salary)
company (company_name, city)
manages (person_name, manager_name)

■ Question

- Find the names of all employees who work for First Bank Corporation ?
 $\{ \langle \text{person_name} \rangle \mid \text{some city, s} \langle \text{s, p, city, salary} \rangle \text{ in works} \wedge \text{city} = \text{"First Bank com"} \}$
- Find the names and cities of residence of all employees who work for First Bank Corporation ?
 $\{ \langle \text{person_name, city} \rangle \mid \text{some c, s, st} \langle \text{person_name, company_name, salary} \rangle \text{ in works} \wedge \langle \text{person_name, street, city} \rangle \text{ in employee} \wedge \text{company_name} = \text{"First Bank Corp"} \}$