Chapter 6: Formal Relational Query Languages – Part 2

Revision by Gun-Woo Kim

Dept. of Computer Science and Engineering

Hanyang University

Database System Concepts, 6th Ed.

Contents

6.2 Extended Relational-Algebra Operations

- Generalized Projection
- Aggregate Functions
- Division

⑪ Generalized Projection

Extends the projection operation by allowing arithmetic functions to be used in the projection list.

$$\Pi_{F_1, F_2, \ldots, F_n}(E)$$

F1은 각종 산술연산된 것들

E is any relational-algebra expression

Each of F_1, F_2, …, F_n are arithmetic expressions involving constants and attributes in the schema of *E*.

Given relation *instructor(ID, name, dept_name,* salary) where salary is annual salary, get the same information but with monthly salary

$$\Pi_{ID,\ name,\ dept\_name,\ salary/12}\ (instructor)$$

⑫ Aggregate Functions and Operations

Aggregation function takes a collection of values and returns a single value as a result.

avg: average value
min: minimum value
max: maximum value
sum: sum of values
count: number of values

Aggregate operation in relational algebra

캘리그레피(체 폰트) G라고함

$$_{G_1, G_2, \ldots, G_n}\ \mathcal{G}\ _{F_1(A_1), F_2(A_2, \ldots, F_n(A_n)}(E)$$

G는 group by에 사용될 변수

E is any relational-algebra expression

- G_1, G_2 …, G_n is a list of attributes on which to group (can be empty)
- Each F_i is an aggregate function
- Each A_i is an attribute name

Note: Some books/articles use γ instead of $\mathcal{G}$ (Calligraphic G)

Aggregate Operation – Example

Relation *r*:

A	*B*	*C*
α	α	7
α	β	7
β	β	3
β	β	10

$\mathcal{G}_{\textbf{sum(c)}}(\mathrm{r})$

sum(*c*)
27

Aggregate Operation – Example (Cont.)

Find the average salary in each department

$$\underbrace{\textit{dept\_name}}_{\text{group by}} \mathcal{G}\ \underbrace{\textbf{avg}(\textit{salary})}_{\text{Aggregate op.}}\ (\textit{instructor})$$

ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

dept_name	*avg_salary*
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

Aggregate Operation (Cont.)

Result of aggregation does not have a name

- Can use rename operation to give it a name
- For convenience, we permit renaming as part of aggregate operation

$$_{dept\_name}\mathcal{G}_{\textbf{avg}(salary)\ \textbf{as}\ avg\_sal}(instructor)$$

⑬ Division Operation

Notation: $r \div s$

Let $q = r \div s$

Then q is the largest relation satisfying $q \times s \subseteq r$

Let r and s be relations on schemas R and S respectively where

$R = (A_1, \ldots, A_m, B_1, \ldots, B_n)$

$S = (B_1, \ldots, B_n)$

The result of $r \div s$ is a relation on schema

$R - S = (A_1, \ldots, A_m)$

$$r \div s = \{ t \mid t \in \Pi_{R-S}(r) \land \forall u \in s\ (tu \in r) \}$$

굳이 안외어도됨

Where tu means the concatenation of tuples t and u to produce a single tuple

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

Division Operation – Example

Relations *r, s*:

R

A	B
s1	b1
s1	b2
s1	b3
s1	b4
s2	b1
s2	b2
s2	b4
s3	b1
s3	b2
s4	b4
s4	b1
s4	b2

R÷S1

S1

B
b1

→

결과

A
s1
s2
s3
s4

R÷S2

S2

B
b1
b3

→

A
s1

R÷S3

S3

B
b1
b2
b4

→

A
s1
s2
s4

Another Division Example

Relations *r, s*:

A	*B*	*C*	*D*	*E*
α	a	α	a	*1*
α	a	γ	a	*1*
α	a	γ	b	*1*
β	a	γ	a	*1*
β	a	γ	b	*3*
γ	a	γ	a	*1*
γ	a	γ	b	*1*
γ	a	β	b	*1*

r

D	*E*
a	*1*
b	*1*

s

$r \div s$:

A	*B*	*C*
α	a	γ
γ	a	γ

SQL and Relational Algebra

select *A1, A2, .. An*
from *r1, r2, …, rm*
where P

is equivalent to the following expression

$$\Pi_{A1, .., An} (\sigma_{P} (r1 \times r2 \times .. \times rm))$$

select *A1, A2,* **sum***(A3)*
from *r1, r2, …, rm*
where P
group by *A1, A2*

is equivalent to the following expression

$$_{A1, A2}\mathcal{G}_{\mathbf{sum}(A3)} (\sigma_{P} (r1 \times r2 \times .. \times rm)))$$

SQL and Relational Algebra (Cont.)

More generally, the non-aggregated attributes in the **select** clause may be a subset of the **group by** attributes, in which case the equivalence is as follows:

select *A1, A2,* **sum***(A3)*
from *r1, r2, …, rm*
where P
group by *A1, A2*

is equivalent to the following expression

$$\Pi_{A1,A2,sumA3}(\ {}_{A1,A2}\mathcal{G}_{\textbf{sum}(A3)\ \textbf{as}\ sumA3}(\sigma_P(r1 \times r2 \times .. \times rm)))$$

이것이 더 정직한 표현

Relational Calclus

관계 해석 (Relational Calculus)

- "어떻게 검색할 것인가" 보다 "무엇을 검색할 것인가" 만을 기술하는 선언적 표현법을 사용하는 비절차적 질의어
- 투플 관계 해석(tuple relational calculus)과 도메인 관계 해석 (domain relational calculus)으로 구분됨

관계 대수와의 차이점

- 관계 해석은 하나의 선언적(declarative) 해석식으로 검색 질의를 명시하며, 비절차적인 언어임
- 관계 대수에서는 연산들을 순차적으로 사용하므로 절차적인 성질을 가짐
- 두 언어의 표현력(expressive power)은 동등함

* 즉 관계대수 식은 관계해석으로 표현 가능

또한 관계해석식을 관계대수식으로 변환해 표현가능

6.3 Tuple Relational Calculus

튜플 관계 해석은 다음과 같은 형태로 표시

$$\{t \mid P(t)\}$$

It is the set of all tuples *t* such that predicate *P* is true for *t*

t is a *tuple variable*, *t* [*A*] denotes the value of tuple *t* on attribute *A*

튜플 변수: 릴레이션의 투플들을 값으로 취할 수 있는 변수

$t \in r$ denotes that tuple *t* is in relation *r*

P is a *formula* similar to that of the predicate calculus

- 예제: 봉급이 $50,000를 넘는 모든 사원을 검색하라.

P= $\{t \mid t \in \text{EMPLOYEE} \wedge t[\text{SALARY}] > 50000\}$ **t[SALARY]: t(row)에 속해있는 속성 salary**

- 여기서, $t \in$ EMPLOYEE는 투플 변수 t가 릴레이션 EMPLOYEE 투플들을 범위로 함을 나타냄
- 투플 t에 대하여 t[SALARY] > 50000을 만족하는 투플 만이 검색된다.
- 투플 t의 모든 애트리뷰트 값들이 리턴된다

Predicate Calculus Formula

$$\{t \mid t \in instructor \land t\,[salary\,] > 80000\}$$

1. Set of attributes and constants
2. Set of comparison operators: (e.g., $<$, $\leq$, $=$, $\neq$, $>$, $\geq$)
3. Set of connectives: and ($\land$), or (v), not ($\neg$)
4. Implication ($\Rightarrow$): x $\Rightarrow$ y, if x is true, then y is true

   $$x \Rightarrow y \equiv \neg x \lor y$$

5. Set of quantifiers:
   - $\exists\, t \in r\,(Q\,(t\,)) \equiv$ ”there exists” a tuple in *t* in relation *r* such that predicate *Q* (*t*) is true
     - 존재 정량자(existential quantifier) (∃) (their exists라 읽음)
   - $\forall t \in r\,(Q\,(t\,)) \equiv Q$ is true “for all” tuples *t* in relation *r*
     - 전체 정량자(universal quantifier) (∀) (for all이라 읽음)

시험에서 all은 안나오고 some이나옴

Example Queries

Find the *ID, name, dept_name, salary* for instructors whose salary is greater than $80,000

$$\{t \mid t \in instructor \land t[salary] > 80000\}$$

As in the previous query, but output only the *ID* attribute value

$$\{t \mid \exists\, s \in \text{instructor}\ (t[ID] = s[ID] \land s[salary] > 80000)\}$$

Notice that a relation on schema (*ID*) is implicitly defined by the query

- 다음 SQL 질의와 동일한 의미를 가진다. (표현력이 동일하다)

```
SELECT      S.ID
FROM        INSTRUCTOR  S
WHERE       S.SALARY > 80000;
```

Example Queries (Cont.)

Find the names of all instructors whose department is in the Watson building

$$\{t \mid \exists s \in \textit{instructor}\ (t\,[\textit{name}\,] = s\,[\textit{name}\,] \\ \land \exists u \in \textit{department}\ (u\,[\textit{dept\_name}\,] = s[\textit{dept\_name}]\text{ “} \\ \land\ u\,[\textit{building}] = \text{“Watson” }))\}$$

select name from instructor where dept_name in (select dept_name from department where building=“Watson”)
= select name from instructor join select dept_name from department where building=“Watson” using (dept_name)

Find the set of all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or both

$$\{t \mid \exists s \in \textit{section}\ (t\,[\textit{course\_id}\,] = s\,[\textit{course\_id}\,] \land \\ s\,[\textit{semester}] = \text{“Fall”} \land s\,[\text{year}] = \textit{2009}) \\ \lor \exists u \in \textit{section}\ (t\ \ [\textit{course\_id}\,] = u\,[\textit{course\_id}\,] \land \\ u\,[\textit{semester}] = \text{“Spring”} \land u\,[\text{year}] = \textit{2010})\}$$

Example Queries (Cont.)

Find the set of all courses taught in the Fall 2009 semester, and in the Spring 2010 semester

$$\{t \mid \exists s \in section\ (t[course\_id] = s[course\_id] \land s[semester] = \text{“Fall”} \land s[\text{year}] = 2009$$
$$\land\ \exists u \in section\ (t[course\_id] = u[course\_id] \land u[semester] = \text{“Spring”} \land u[\text{year}] = 2010)\}$$

Find the set of all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester

$$\{t \mid \exists s \in section\ (t[course\_id] = s[course\_id] \land s[semester] = \text{“Fall”} \land s[\text{year}] = 2009$$
$$\land \neg \exists u \in section\ (t[course\_id] = u[course\_id] \land u[semester] = \text{“Spring”} \land u[\text{year}] = 2010)\}$$

6.4 Domain Relational Calculus

A nonprocedural query language equivalent in power to the tuple relational calculus

Each query is an expression of the form:

$$\{ < x_1, x_2, \ldots, x_n > \mid P(x_1, x_2, \ldots, x_n)\}$$

앞쪽 x는 내가 찾고자하는 변수를 넣음

$x_1, x_2, \ldots, x_n$ represent domain variables

P represents a formula similar to that of the predicate calculus

- 투플 변수 대신 도메인 변수(domain variables)를 사용하는 관계 해석
- 도메인 변수는 한 애트리뷰트의 도메인을 범위로 가짐
  - 투플 관계 해석에서는? → 투플의 도메인을 범위로 가졌음
  - 투플 관계 해석에서는 투플이 중심인 반면, 도메인 관계 해석에서는 애트리뷰트가 중심임
- 차수가 n인 릴레이션의 경우 n개의 도메인 변수를 사용함

Example Queries

Find the *ID, name, dept_name, salary* for instructors whose salary is greater than $80,000

$$\boxed{\{< i, n, d, s> \mid \ < i, n, d, s> \in instructor \land s > 80000\}}$$

$$\{t \mid t \in instructor \land t\,[salary\,] > 80000\}$$

As in the previous query, but output only the *ID* attribute value

$$\boxed{\{< i> \ \mid \exists\ n, d, s\ (< i, n, d, s> \in instructor \land s > 80000)\}}$$

보고 싶은 값을 제외

$$\{t \mid \exists\ s\ \in \text{instructor}\ (t\,[ID\,] = s\,[ID\,] \land s\,[salary\,] > 80000)\}$$

Find the names of all instructors whose department is in the Watson building

$$\boxed{\begin{array}{l}\{< n > \mid \ \exists\ i, d, s\ (< i, n, d, s > \in instructor \\ \qquad\qquad \land \exists\ b, \text{a}\ (< d, b, a> \in department \ \land \ b = \text{“Watson”}\ ))\}\end{array}}$$

여기서 d는 상위d랑 같은 d

$$\begin{array}{l}\{t \mid \exists s \in instructor\ (t\,[name\,] = s\,[name\,] \\ \qquad \land \exists u \in department\ (u\,[dept\_name\,] = s[dept\_name]\ \text{“} \\ \qquad\qquad\qquad \land \ \ u\,[building] = \text{“Watson”}\ ))\}\end{array}$$

Example Queries (Cont.)

Find the set of all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or both

$$\{<c> \mid \exists\, a, s, y, b, r, t\ (<c, a, s, y, b, r, t> \in section \land s = \text{“Fall”} \land y = 2009) \lor \exists\, a, s, y, b, r, t\ (<c, a, s, y, b, r, t> \in section \land s = \text{“Spring”} \land y = 2010)\}$$

This case can also be written as

$$\{<c> \mid \exists\, a, s, y, b, r, t\ (<c, a, s, y, b, r, t> \in section \land ((s = \text{“Fall”} \land y = 2009) \lor (s = \text{“Spring”} \land y = 2010))\}$$

Find the set of all courses taught in the Fall 2009 semester, and in the Spring 2010 semester

$$\{<c> \mid \exists\, a, s, y, b, r, t\ (<c, a, s, y, b, r, t> \in section \land s = \text{“Fall”} \land y = 2009) \land \exists\, a, s, y, b, r, t\ (<c, a, s, y, b, r, t> \in section \land s = \text{“Spring”} \land y = 2010)\}$$