

# Chapter 6: Formal Relational Query Languages

Database System Concepts, 6th Ed.

©Silberschatz, Korth and Sudarshan (Modified by Hyo-Sang Lim)



# **Chapter 6: Formal Relational Query Languages**

- Relational Algebra
- Tuple Relational Calculus (skip)
- Domain Relational Calculus (skip)



# **Relational Algebra**

- Procedural language
- Six basic operators

select: σ

project: ∏

• union: ∪

set difference: –

Cartesian product: x

• rename:  $\rho$ 

The operators take one or two relations as inputs and produce a new relation as a result.

**Database System Concepts - 6th Edition** 

6.3

©Silberschatz, Korth and Sudarshan



# **Select Operation – Example**

Relation r

A	В	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

 $-\sigma_{A=B \land D > 5}(r)$ 

A	В	C	D
α	α	1	7
β	β	23	10



# **Select Operation**

- Notation:  $\sigma_p(r)$
- p is called the selection predicate
- Defined as:

$$\sigma_p(\mathbf{r}) = \{t \mid t \in r \text{ and } p(t)\}$$

Where p is a formula in propositional calculus consisting of **terms** connected by :  $\land$  (and),  $\lor$  (or),  $\neg$  (not)

Each term is one of:

<attribute> op <attribute> or <constant>

where *op* is one of: =,  $\neq$ ,  $\geq$ ,  $\leq$ .  $\leq$ 

**Example of selection:** 

$$\sigma_{\textit{dept\_name} = \textit{"Physics"}}(\textit{instructor})$$

Database System Concepts - 6th Edition

6.5

©Silberschatz, Korth and Sudarshan



# **Project Operation – Example**

Relation r:

A	В	C
α	10	1
α	20	1
β	30	1
β	40	2

$$\blacksquare \prod_{A,C} (r)$$

$$\begin{array}{c|c}
A & C \\
\hline
\alpha & 1 \\
\alpha & 1 \\
\beta & 1 \\
\beta & 2 \\
\end{array}$$

$$\begin{array}{c|c}
A & C \\
\hline
\alpha & 1 \\
\beta & 1 \\
\beta & 2
\end{array}$$



### **Project Operation**

Notation:

$$\prod_{A_1,A_2,\ldots,A_k}(r)$$

where  $A_1$ ,  $A_2$  are attribute names and r is a relation name.

- The result is defined as the relation of k columns obtained by erasing the columns that are not listed
- Duplicate rows removed from result, since relations are sets
- Example: To eliminate the *dept\_name* attribute of *instructor*

 $\Pi_{ID, name, salary}$  (instructor)

Database System Concepts - 6th Edition

6.7

©Silberschatz, Korth and Sudarshan



# **Union Operation – Example**

Relations r, s:

$\boldsymbol{A}$	В
α	1
α	2
β	1
1	1

r ∪ s:

A	В
α	1
α	2
β	1
β	3



### **Union Operation**

- Notation: *r* ∪ *s*
- Defined as:

$$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$$

- For  $r \cup s$  to be valid.
  - 1. *r*, *s* must have the *same* **arity** (same number of attributes)
  - 2. The attribute domains must be **compatible** (example:  $2^{nd}$  column of r deals with the same type of values as does the  $2^{nd}$  column of s)
- Example: to find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both

$$\Pi_{course\_id}(\sigma_{semester="Fall"} \land year=2009(section)) \cup \Pi_{course\_id}(\sigma_{semester="Spring"} \land year=2010(section))$$

Database System Concepts - 6th Edition

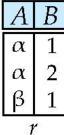
6.9

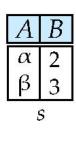
©Silberschatz, Korth and Sudarshan



#### Set difference of two relations

Relations *r*, *s*:





r − s:

A	В
α	1
β	1



#### **Set Difference Operation**

- Notation r s
- Defined as:

$$r-s = \{t \mid t \in r \text{ and } t \notin s\}$$

- Set differences must be taken between **compatible** relations.
  - r and s must have the same arity
  - attribute domains of *r* and *s* must be compatible
- Example: to find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester

$$\Pi_{course\_id} (\sigma_{semester="Fall"} \land year=2009 (section)) - \\ \Pi_{course\_id} (\sigma_{semester="Spring"} \land year=2010 (section))$$

Database System Concepts - 6th Edition

6.11

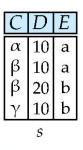
©Silberschatz, Korth and Sudarshan



# **Cartesian-Product Operation – Example**

Relations *r*, *s*:





r x s:

A	В	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b



### **Cartesian-Product Operation**

- Notation *r* x s
- Defined as:

$$r \times s = \{t \mid q \mid t \in r \text{ and } q \in s\}$$

- Assume that attributes of r(R) and s(S) are disjoint. (That is, R ∩ S = Ø).
- If attributes of *r*(*R*) and *s*(*S*) are not disjoint, then renaming must be used.

**Database System Concepts - 6th Edition** 

6.13

©Silberschatz, Korth and Sudarshan



# **Composition of Operations**

- Can build expressions using multiple operations
- **Example**:  $\sigma_{A=C}(r x s)$
- rxs

A	В	C	D	Ε
α	1	$\alpha$	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	$\alpha$	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

 $\sigma_{A=C}(r \times s)$ 

$\boldsymbol{A}$	В	C	D	Ε
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b



#### **Rename Operation**

- Allows us to name, and therefore to refer to, the results of relationalalgebra expressions.
- Allows us to refer to a relation by more than one name.
- Example:

$$\rho_{x}(E)$$

returns the expression *E* under the name *X* 

■ If a relational-algebra expression *E* has arity *n*, then

$$\rho_{x(A_1,A_2,\dots,A_n)}(E)$$

returns the result of expression E under the name X, and with the attributes renamed to  $A_1$ ,  $A_2$ , ...,  $A_n$ .

Database System Concepts - 6th Edition

6.15

©Silberschatz, Korth and Sudarshan



#### **Example Query**

- Find the largest salary in the university
  - Step 1: find instructor salaries that are less than some other instructor salary (i.e. not maximum)
    - using a copy of instructor under a new name d
    - $\sqcap$  instructor.salary  $(\sigma$  instructor.salary < d.salary  $(instructor \times \rho_d$  (instructor)))
  - Step 2: Find the largest salary
    - $\begin{array}{c} \Pi_{\textit{salary}} \textit{ (instructor)} \\ \Pi_{\textit{instructor.salary}} (\sigma_{\textit{instructor.salary}} < \textit{d.salary} \\ \textit{(instructor } x \ \rho_{\textit{d}} \textit{(instructor)))} \end{array}$



#### **Example Queries**

- Find the names of all instructors in the Physics department, along with the course\_id of all courses they have taught
  - Query 1

```
\prod_{instructor.ID,course\_id} (\sigma_{dept\_name="Physics"} (\sigma_{instructor.ID=teaches.ID} (instructor x teaches)))
```

Query 2

```
\Pi_{instructor.ID,course\_id} (\sigma_{instructor.ID=teaches.ID} (\sigma_{dept\_name="Physics"} (instructor) x teaches))
```

Database System Concepts - 6th Edition

6.17

©Silberschatz, Korth and Sudarshan



#### **Formal Definition**

- A basic expression in the relational algebra consists of either one of the following:
  - A relation in the database
  - A constant relation
- Let  $E_1$  and  $E_2$  be relational-algebra expressions; the following are all relational-algebra expressions:
  - $E_1 \cup E_2$
  - $E_1 E_2$
  - $E_1 \times E_2$
  - σ<sub>p</sub> (E<sub>1</sub>), P is a predicate on attributes in E<sub>1</sub>
  - $\prod_{S}(E_1)$ , S is a list consisting of some of the attributes in  $E_1$
  - $\rho_{x}(E_{1})$ , x is the new name for the result of  $E_{1}$



#### **Additional Operations**

We define additional operations that do not add any power to the relational algebra, but that simplify common queries.

- Set intersection
- Natural join
- Assignment
- Outer join

Database System Concepts - 6th Edition

6.19

©Silberschatz, Korth and Sudarshan



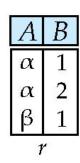
# **Set-Intersection Operation**

- Notation:  $r \cap s$
- Defined as:
- $r \cap s = \{t \mid t \in r \text{ and } t \in s\}$
- Assume:
  - r, s have the same arity
  - attributes of *r* and *s* are compatible
- Note:  $r \cap s = r (r s)$



# **Set-Intersection Operation – Example**

Relation *r*, *s*:



A	В
α	2
β	3

 $r \cap s$ 



Database System Concepts - 6th Edition

6.21

©Silberschatz, Korth and Sudarshan



### **Natural-Join Operation**

- Notation: r ⋈ s
- Let r and s be relations on schemas R and S respectively. Then,  $r \bowtie s$  is a relation on schema  $R \cup S$  obtained as follows:
  - Consider each pair of tuples  $t_r$  from r and  $t_s$  from s.
  - If  $t_r$  and  $t_s$  have the same value on each of the attributes in  $R \cap S$ , add a tuple t to the result, where
    - ightharpoonup t has the same value as  $t_r$  on r
    - t has the same value as  $t_s$  on s
- Example:

$$R = (A, B, C, D)$$

$$S = (E, B, D)$$

- Result schema = (A, B, C, D, E)
- $r \bowtie s$  is defined as:

$$\prod_{r.A, r.B, r.C, r.D, s.E} (\sigma_{r.B = s.B \land r.D = s.D} (r \times s))$$



#### **Natural Join Example**

Relations r, s:

<u>B</u>	C	$\mathcal{D}$
1	α	a
2	γ	a
4	β	b
1	γ	a
2	β	b
	1 2 4 1	1 α 2 γ 4 β 1 γ

В	D	Ε	
1	a	α	
3	a	β	
1	a	γ	
2	b	δ	
3	b	3	
S			

■ r⋈s

A	В	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Database System Concepts - 6th Edition

6.23

©Silberschatz, Korth and Sudarshan



#### **Natural Join and Theta Join**

- Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses that the instructors teach
  - Π name, title (σ dept\_name="Comp. Sci." (instructor ⋈ teaches ⋈ course))
- Natural join is associative
  - (instructor ⋈ teaches) ⋈ course is equivalent to instructor ⋈ (teaches ⋈ course)
- Natural join is commutative
  - instruct ⋈ teaches is equivalent to teaches ⋈ instructor
- The **theta join** operation  $r \bowtie_{\theta} s$  is defined as
  - $r \bowtie_{\theta} s = \sigma_{\theta} (r \times s)$



#### **Assignment Operation**

- The assignment operation (←) provides a convenient way to express complex queries.
  - Write query as a sequential program consisting of
    - a series of assignments
    - followed by an expression whose value is displayed as a result of the query.
  - Assignment must always be made to a temporary relation variable.

Database System Concepts - 6th Edition

6.25

©Silberschatz, Korth and Sudarshan



#### **Outer Join**

- An extension of the join operation that avoids loss of information.
- Computes the join and then adds tuples form one relation that does not match tuples in the other relation to the result of the join.
- Uses null values:
  - null signifies that the value is unknown or does not exist
  - All comparisons involving null are (roughly speaking) false by definition.
    - We shall study precise meaning of comparisons with nulls later



# **Outer Join – Example**

#### Relation instructor1

ID	name	dept_name
10101	Srinivasan	Comp. Sci.
12121	Wu	Finance
15151	Mozart	Music

#### Relation teaches1

ID	course_id
10101	CS-101
12121	FIN-201
76766	BIO-101

Database System Concepts - 6th Edition

6.27

©Silberschatz, Korth and Sudarshan



# **Outer Join – Example**

#### Join

#### *instructor* ⋈ *teaches*

ID	name	dept_name	course_id
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201

#### ■ Left Outer Join

#### 

ID	name	dept_name	course_id
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	null



#### **Outer Join – Example**

#### ■ Right Outer Join

instructor ⋈ teaches

ID	name	dept_name	course_id
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
76766	null	null	BIO-101

#### ■ Full Outer Join

instructor □×□ teaches

ID	name	dept_name	course_id
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	null
76766	null	null	BIO-101

**Database System Concepts - 6th Edition** 

6.29

©Silberschatz, Korth and Sudarshan



#### **Null Values**

- It is possible for tuples to have a null value, denoted by null, for some of their attributes
- null signifies an unknown value or that a value does not exist.
- The result of any arithmetic expression involving *null* is *null*.
- Aggregate functions simply ignore null values (as in SQL)
- For duplicate elimination and grouping, null is treated like any other value, and two nulls are assumed to be the same (as in SQL)



#### **Division Operator**

Given relations r(R) and s(S), such that  $S \subset R$ ,  $r \div s$  is the largest relation t(R-S) such that

$$t \times s \subseteq r$$

■ E.g. let  $r(ID, course\_id) = \prod_{ID, course\_id} (takes)$  and

 $s(course\_id) = \prod_{\textit{course\_id}} (\sigma_{dept\_name="Biology"}(\textit{course}~) \\$  then  $r \div s$  gives us students who have taken all courses in the Biology department

**Can write**  $r \div s$  as

$$temp1 \leftarrow \prod_{R-S} (r)$$

$$temp2 \leftarrow \prod_{R-S} ((temp1 \times s) - \prod_{R-S,S} (r))$$

$$result = temp1 - temp2$$

- The result to the right of the ← is assigned to the relation variable on the left of the ←.
- May use variable in subsequent expressions.

Database System Concepts - 6th Edition

6.31

©Silberschatz, Korth and Sudarshan



#### **Extended Relational-Algebra-Operations**

- Generalized Projection
- Aggregate Functions



### **Aggregate Functions and Operations**

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

avg: average valuemin: minimum valuemax: maximum valuesum: sum of valuescount: number of values

■ Aggregate operation in relational algebra

$$G_1,G_2,...,G_n \ G_{F_1(A_1),F_2(A_2,...,F_n(A_n)}(E)$$

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<sub>i</sub> is an attribute name
- Note: Some books/articles use  $\gamma$  instead of  $\mathcal G$  [Calligraphic G)

Database System Concepts - 6th Edition

6.33

©Silberschatz, Korth and Sudarshan



### **Aggregate Operation – Example**

Relation r:

Α	В	С
α	α	7
$\alpha$	β	7
β	β	3
β	β	10

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





### **Aggregate Operation – Example**

Find the average salary in each department

 $dept\_name\ G\ {
m avg}(salary)\ (instructor)$ 

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

Database System Concepts - 6th Edition

6.35

©Silberschatz, Korth and Sudarshan



### **Aggregate Functions (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  $G_{avg}(salary)$  as  $avg_{sal}(instructor)$ 



# **SQL** and Relational Algebra

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

is equivalent to the following expression in multiset relational algebra

$$\prod_{A1, ..., An} (\sigma_P(r1 \times r2 \times ... \times rm))$$

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

is equivalent to the following expression in multiset relational algebra

A1, A2 
$$G_{sum(A3)}$$
 ( $\sigma_P(r1 \times r2 \times .. \times rm)$ ))

Database System Concepts - 6th Edition