Formal Query Languages: Relational Algebra

- Set Theory Operations
- Specific Relational Operations
- Write Queries in Relational Algebra

Relational Algebra

- Operations on entire relations
  - Operands are (constant or variable) relations
  - Result is a relation
- Set theory operations:
  - Union, Intersection, Difference and Cartesian Product (product for short)
- Specific relational operations:
  - Selection, Projection, Join and Division
- Complete set of relational algebra operations:
  - Select, project, product, union and difference
- SQL is based on concepts from relational algebra

Selection σ

- Unary operator Select, **σ**:

$$\boldsymbol{\sigma_{\text{selection-condition}}\ (r)}$$

- E.g., $\sigma_{\text{Name= 'John'} \vee \text{Name = 'Susan'}}$ (STUDENT)
  - result = {t | t∈r and (t[Name] = 'John' or t[Name] = 'Susan')}
- *Selection condition* any logical expression on attributes of **r** involving any applicable comparison operator
  {=,<, ≤, >, ≥, ≠}

Example of Selection

- $\sigma_{\text{Name='Bob'} \vee \text{ Major = 'Math'}}$ (S) = ?
- $\sigma_{\text{Name='Bob'} \wedge \text{ Major = 'Math'}}$ (S) = ?
- How can I get a copy of S?
- How can I get an empty copy of S?

Relation **S**

SID	Name	Major
1	Bob	CS
3	Ann	CoE
4	Bob	Math

Projection π

- Unary operator Project, π:

  $\boldsymbol{\pi_{\text{attribute-list}}\ (r)}$

  - Attribute-list $\subseteq$ R
- E.g., $\pi_{\text{Name, Major}}$ (STUDENT)
  - result = {t | t$\in$r and t[Name, Major]}
- What about $\pi_{\text{SID, Major}}$ (S) = ?

Relation **S**

SID	Name	Major
1	Bob	CS
3	Ann	CoE
4	Bob	Math
5	Bob	CS

Relational Algebra Expressions

- Query: *List the QPA of all students (SID) in CSD whose QPA is greater than 3.5*
- STUDENT (SID, FName, SName, Dept, Major, QPA)
- **Nesting** the operations

  $\pi_{\text{SID, QPA}}\ (\sigma_{\text{Dept = 'CSD'} \wedge \text{QPA>3.5}}\ (\text{STUDENT}))$
- **Sequence** of operations

  $\text{HS} \leftarrow \sigma_{\text{Dept = 'CSD'} \wedge \text{QPA>3.5}}\ (\text{STUDENT})$

  $\text{RESULT} \leftarrow \pi_{\text{SID, QPA}}\ (\text{HS})$
- **Query tree**
  - *leaves* nodes are relations and *internal* nodes are operations

---

Renaming Operator

- Renaming attributes of the result

  $\text{RSLT(StudentID, GPA)} \leftarrow \pi_{\text{SID, QPA}}\ (\text{HS})$
- Change the name of Attributes (in general):

  $\rho(\text{a1,a2,a3,..an})(\text{r})$
- Example:

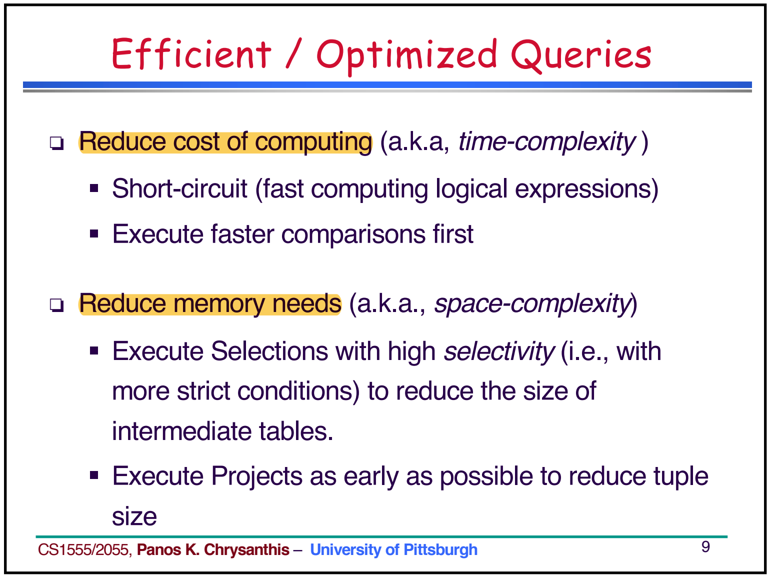
  $\rho(\text{StudentID, GPA})\ ($

  $\pi_{\text{SID, QPA}}\ (\sigma_{\text{Dept = 'CSD'} \wedge \text{QPA>3.5}}\ (\text{STUDENT}))$

Properties of σ and π

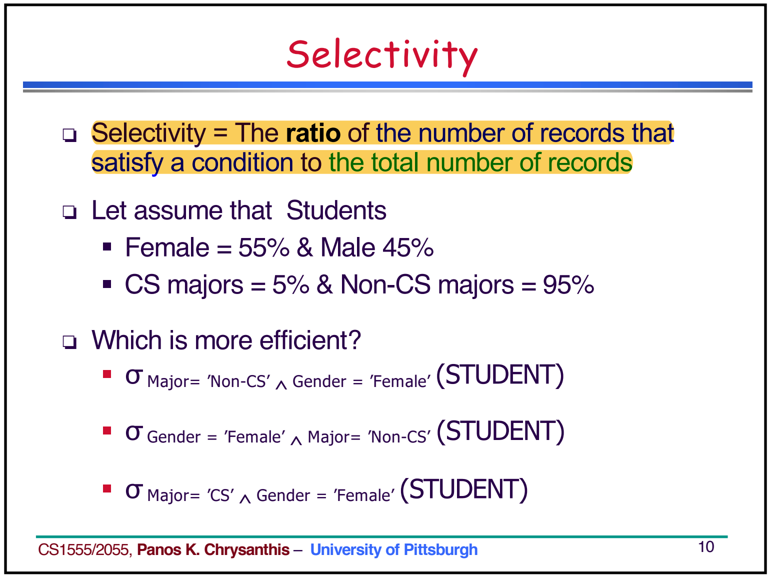
- $\sigma_{\text{cond1}}\ (\sigma_{\text{cond2}}\ (R)) = \sigma_{\text{cond2}}\ (\sigma_{\text{cond1}}\ (R))$
- $\sigma_{\text{cond1}}\ (\sigma_{\text{cond2}}\ (R)) = \sigma_{\text{cond2} \wedge \text{cond1}}\ (R)$

  $= \sigma_{\text{cond1} \wedge \text{cond2}}\ (R)$
- $\pi_{\text{list1}}\ (\pi_{\text{list2}}\ (R)) = \pi_{\text{list1}}\ (R)$ When?



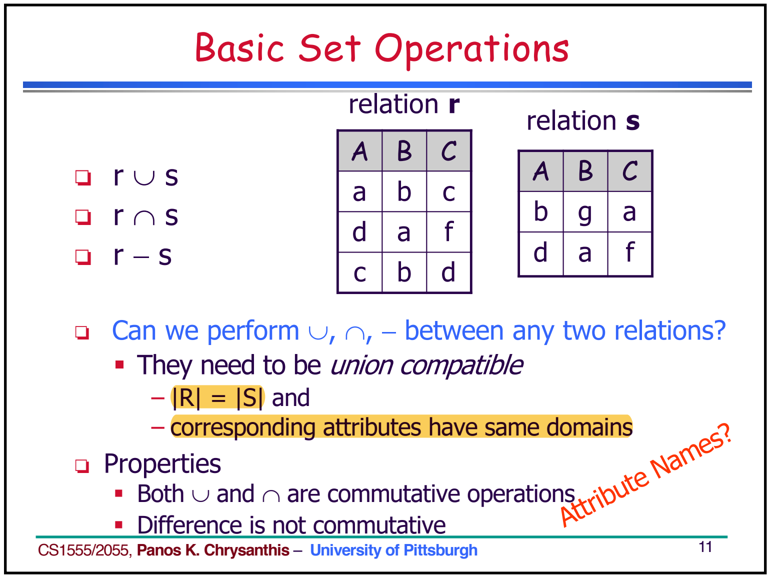
Efficient / Optimized Queries

- Reduce cost of computing (a.k.a, *time-complexity*)
  - Short-circuit (fast computing logical expressions)
  - Execute faster comparisons first
- Reduce memory needs (a.k.a., *space-complexity*)
  - Execute Selections with high *selectivity* (i.e., with more strict conditions) to reduce the size of intermediate tables.
  - Execute Projects as early as possible to reduce tuple size



Selectivity

- Selectivity = The **ratio** of the number of records that satisfy a condition to the total number of records
- Let assume that Students
  - Female = 55% & Male 45%
  - CS majors = 5% & Non-CS majors = 95%
- Which is more efficient?
  - $\sigma_{\text{Major= 'Non-CS'} \wedge \text{Gender = 'Female'}}$ (STUDENT)
  - $\sigma_{\text{Gender = 'Female'} \wedge \text{Major= 'Non-CS'}}$ (STUDENT)
  - $\sigma_{\text{Major= 'CS'} \wedge \text{Gender = 'Female'}}$ (STUDENT)



Basic Set Operations

- r ∪ s
- r ∩ s
- r – s

relation **r**

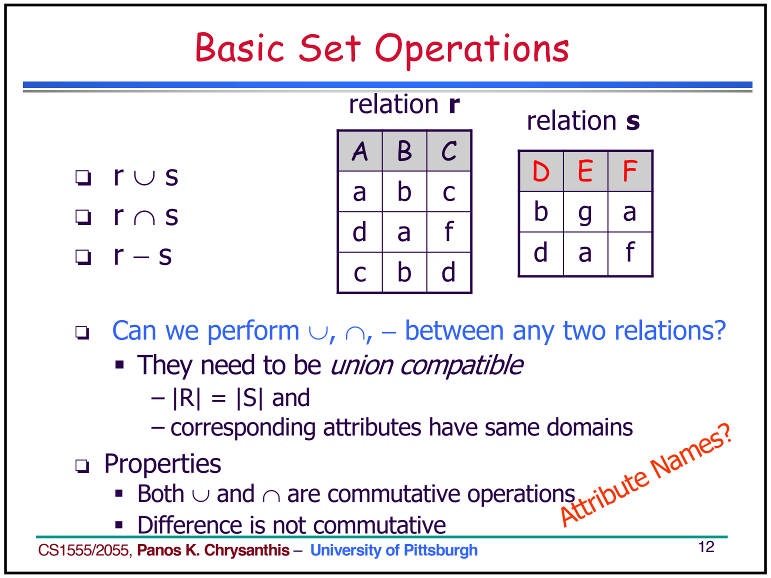
A	B	C
a	b	c
d	a	f
c	b	d

relation **s**

A	B	C
b	g	a
d	a	f

- Can we perform ∪, ∩, – between any two relations?
  - They need to be *union compatible*
    - |R| = |S| and
    - corresponding attributes have same domains
- Properties
  - Both ∪ and ∩ are commutative operations
  - Difference is not commutative

Attribute Names?



Basic Set Operations

- r ∪ s
- r ∩ s
- r – s

relation **r**

A	B	C
a	b	c
d	a	f
c	b	d

relation **s**

D	E	F
b	g	a
d	a	f

- Can we perform ∪, ∩, – between any two relations?
  - They need to be *union compatible*
    - |R| = |S| and
    - corresponding attributes have same domains
- Properties
  - Both ∪ and ∩ are commutative operations
  - Difference is not commutative

Attribute Names?

Cartesian Product

- r x s
- Let p(P) = r(R) x s(S)
- |P| = ? and |p| = ?
  - $|P| = |R| + |S| = \alpha_r + \alpha_s$
  - $|p| = |r| * |s|$
- Name conflicts are resolved by using the relations names as prefixes: r.A, r.B, S.A, S.B

relation **r**

A	B	C
a	b	c
d	a	f
c	b	d

α_r

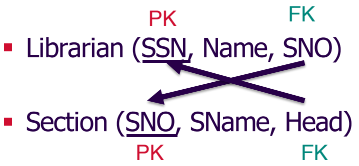
relation **s**

A	B
b	g
d	a

α_s

Common Query

- Library microDB:



  - Librarian (SSN, Name, SNO)
  - Section (SNO, SName, Head)

- *List the names of head librarians.*
- How?

List the names of head librarians

- L: Librarian, S: Section

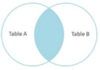
LS ← Librarian X Section;

- LS schema:
  (L.SSN,L.Name,L.SNO, S.SNO,S.Sname, S.Head)

HL ← $\sigma_{L.SSN = S.Head}$ (LS);

RSLT ← $\Pi_{L.name}$ (HL);

Equi-Join



- $r \bowtie_{r.Ai = s.Aj} s$
- =-join is a macro of
  $\sigma_{r.Ai = s.Aj}(r \times s)$
- =-join of r(R) and s(S):
  $r \bowtie_{r.B = s.D} s = ?$

relation **s**

C	D
3	4
6	8

relation **r**

A	B	C	D
1	2	3	4
2	4	6	8
1	2	4	8
2	6	6	8
8	2	3	4
2	4	3	4

Θ-Join

- $\Theta = \{=, <, \leq, >, \geq, \neq\}$
- Θ-join of r(R) and s(S) on attributes r.A$_i$ and s.A$_j$

$$r \bowtie_{r.Ai\ \theta\ s.Aj} s$$

$$= \sigma_{r.Ai\ \theta\ s.Aj} (r \times s)$$

- ≥-join of r(R) and s(S):

$$r \bowtie_{r.B \geq\ s.D} s = ?$$

relation **r**

A	B	C	D
1	2	3	4
2	4	6	8
1	2	4	8
2	6	6	8
8	2	3	4
2	4	3	4

relation **s**

C	D
3	4
6	8

Example of Θ-Join

- ≥-join of r(R) and s(S):

$$r \bowtie_{r.B \geq\ s.D} s = ?$$

r.A	r.B	r.C	r.D	s.C	s.D
2	4	6	8	3	4
2	4	3	4	3	4
2	6	6	8	3	4

- $r \bowtie s = r \times s,\ \Theta = \Phi$

relation **r**

A	B	C	D
1	2	3	4
2	4	6	8
1	2	4	8
2	6	6	8
8	2	3	4
2	4	3	4

relation **s**

C	D
3	4
6	8

Natural-Join

- Equi-join without duplicate columns

$$r *_P s$$

- P=list of attributes: $P = R \cap S$
- $r * s = \Pi_{R \cup S} (r \bowtie_{r.P =\ s.P} s)$
- $r * s = ?$
- Note other notations & meanings
  - $r \bowtie s = r * s,\ \ R \cap S \neq \Phi$
  - $r * s = r \times s,\ \ R \cap S = \Phi$

relation **r**

A	B	C	D
1	2	3	4
2	4	6	8
1	2	4	8
2	6	6	8
8	2	3	4
2	4	3	4

relation **s**

C	D	E
3	4	6
6	8	8