

Division Usage: Review Example

- ◆ Query: “List the usernames of people who have in common **all** Susan’s friends on Facebook.”

Relation **Friends**

Username	FNFriend	LNFriend
Susan	Alex	L
Mark	Alex	L
Kirk	Mary	K
Shi	Alex	L
Susan	Mary	K
Shi	Lory	M
Kirk	Chia	S
Kirk	Alex	L

$SF \leftarrow \sigma_{\text{Username} = \text{'Susan'}}(\text{Friends})$

$SSF \leftarrow \text{Friends} \div \pi_{\text{FN, LN}}(SF)$

$\text{RSLT} \leftarrow \sigma_{\text{Username} \neq \text{'Susan'}}(SSF)$

Extended Relational Operations

- Extended set and Relational operations:
 - Outer Union
 - Outer Joins
- Aggregate operations:
 - MAX, MIN, AVG, SUM
 - Count
 - Subset: grouping
- Arithmetic operations and other functions:

Outer Union

- it is defined on partially union compatible relations
 - Non union-compatible attributes are kept in $r \cup^* s$
 - Non union-compatible attributes without value are set to NULL
 - Tuples are “matched” over common named attributes like in natural join

relation **r**

FN	LN	MJ
a	b	cs
d	a	ce
c	b	cs

relation **s**

FN	LN	CL
b	g	f
d	a	sr

$r \cup^* s$

FN	LN	MJ	CL
a	b	cs	Null
d	a	ce	sr
c	b	cs	Null
b	g	Null	f

- ◆ what about outer intersection or outer difference?

Outer Join

- Join selects only tuples satisfying the join-condition
- **Outer Join:**
 - Left outer join ($r \bowtie^L s$) keeps every tuple in the left relation
 - Right outer join ($r \bowtie^R s$) keeps every tuple in the right relation
 - Full outer join ($r \bowtie^F s$) keeps every tuple
- Attributes of tuples with no matching tuples are set to NULL
- With out a join-condition they behave like natural join

Outer Join ("natural")

relation r				relation s			
FN	LN	MJ	CL	FN	LN	CL	
a	b	cs		b	g	f	
d	a	ce	sr	d	a	sr	
c	b	cs					

r ⋈ s			
FN	LN	MJ	CL
a	b	cs	Null
d	a	ce	sr
c	b	cs	Null

r ⋈ s			
FN	LN	MJ	CL
d	a	ce	sr
b	g	Null	f

r ⋈ s			
FN	LN	MJ	CL
a	b	cs	Null
d	a	ce	sr
c	b	cs	Null
b	g	Null	f

CS1555/2055, Panos K. Chrysanthos & Constantinos Costa – University of Pittsburgh

34

Aggregate Functions

- Mathematical and Statistical aggregate functions on collections of values

- SUM, MAXIMUM, MINIMUM, AVERAGE
- COUNT number of tuples (cardinality)

$f_{\text{function list}}(<\text{relation}>)$

- Function list is a list of pairs
(*function, attribute*)

- E.g., $f_{\text{count SID, AVERAGE GPA}}(\text{STUDENT})$

CS1555/2055, Panos K. Chrysanthos & Constantinos Costa – University of Pittsburgh

35

Aggregate Functions: Example

- $\text{RSLT} \leftarrow f_{\text{count SID, AVERAGE GPA}}(\text{STUDENT})$

Student

SID	Name	Age	GPA
546007	Susan	18	3.8
546100	Bob	19	3.65
546500	Bill	20	3.7

RSLT

Count_SID	AVERAGE_GPA
3	3.72

CS1555/2055, Panos K. Chrysanthos & Constantinos Costa – University of Pittsburgh

36

Example of Aggregation Query

- Q: Find the students with the highest GPA.

- Student** (SID, Name, Age, GPA)

- A:

$\text{MG}(\text{MGPA}) \leftarrow f_{\text{MAX GPA}}(\text{Student});$

$\text{RSLT} \leftarrow \text{MG} \bowtie \text{MGPA} = \text{GPA}(\text{Student});$

CS1555/2055, Panos K. Chrysanthos & Constantinos Costa – University of Pittsburgh

37

Grouping

- ❑ Grouping the tuple in a relation
 $\langle \text{grouping attributes} \rangle \text{ } \mathcal{F} \langle \text{function list} \rangle (\langle \text{relation} \rangle)$
 - Tuples are grouped based on the values of *grouping attributes*
- E.g., Q1: $\text{major } \mathcal{F} \text{ count SID, AVERAGE GPA (STUDENT)}$

major	Count_SID	AVERAGE_GPA
-------	-----------	-------------
- ❑ Q2: $\text{class, major } \mathcal{F} \text{ count SID, AVERAGE GPA (STUDENT)}$?

class	major	Count_SID	AVERAGE_GPA
-------	-------	-----------	-------------

Recursive closure

- ❑ It is applied to a recursive relationship between tuples of the same relation
- ❑ E.g., find all the ancestors or descendants
- ❑ How do we express it?
- ❑ What about the join operation?
- ❑ Need control statements...iteration

Write Queries in Relational Algebra

- ❑ Deletion:
 - $r \leftarrow r - \text{Relational_Expression}$
 - $\text{STUDENT} \leftarrow \text{STUDENT} - (\sigma_{\text{Dept} = \text{'CSD'} \wedge \text{QPA} < 2.5} (\text{STUDENT}))$
- ❑ Insertion:
 - $r \leftarrow r \cup \text{Relational_Expression}$
 - $\text{STUDENT} \leftarrow \text{STUDENT} \cup \{(365, \text{'Smith'}, \text{'John'})\}$
- ❑ Updating:
 - $r \leftarrow \Pi_{\text{attributes-to-be-updated}} (r)$
 - $\text{STUDENT} \leftarrow \Pi_{\text{Dept} = \text{'CSD'}} (\sigma_{\text{Dept} = \text{'CS'}} (\text{STUDENT}))$

Discussion

- ❑ The relational algebra is **procedural**
- ❑ The queries in relational algebra specify **how** to produce a result, BUT...
- ❑ The **how** should be the responsibility of the system
- ❑ User queries should be **declarative** specifying what is to be retrieved
 - Textual query languages (SQL, QUEL)
 - Graphical query languages (QBE)
 - Visual iconic languages (QBI)
- ❑ Other formal query languages:
 - Relational tuple calculus
 - Relational domain calculus

Steps in Processing a Query

