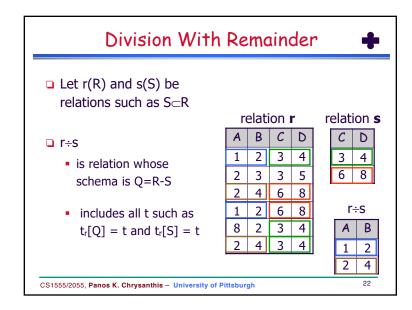
## Examples from Library DB

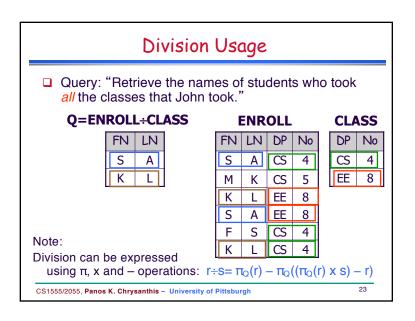
- □ Library DB schema:
  - LIBRARIAN(Name, SSN, SNO, BirthPlace)
  - SECTION(SName, 3NO, Head)
  - OUTREACH(Pname(PNO, SNUM, Location))
  - WORKSON(LSSN,PNO,Hours)
- located in PGH, list its project number, the responsible section name and the name of its head.

```
□ For every outreach activity PP \leftarrow \sigma_{Location = 'Pgh'}(OUTREACH);
                                              SPP \leftarrow PP \bowtie SNUM = SNO SECTION;
                                              \mathsf{HSPP} \leftarrow \mathsf{SPP} * \mathsf{LIBRARIAN};
                                              RSLT \leftarrow \pi_{PNO, Sname, Name}(HSPP);
```

CS1555/2055. Panos K. Chrysanthis - University of Pittsburgh

#### Division □ Let r(R) and s(S) be relations such as S⊂R relation s relation r □ The division of r by s, 1 2 3 4 3 4 denoted by r+s, 4 6 8 is relation whose 2 1 6 8 schema is Q=R-S and r÷s 2 4 3 4 includes all t such as $t_r[Q] = t$ and $t_r[S] = t$ 1 2 CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh





### Division Usage: Review Example

• Query: "List the names of people who have in common all Susan's friends on Facebook."

#### Relation Friends

Username	FNFriend	LNFrind
Susan	Alex	L
Mark	Alex	L
Kirk	Mary	K
Shi	Alex	L
Susan	Mary	K
Shi	Lory	М
Kirk	Chia	S
Kirk	Alex	L

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

## Division Usage: Review Example

• Query: "List the names of people who have in common all Susan's friends on Facebook."

Username

Susan

**FNFriend** 

Alex

			_
Relation <b>Friends</b>	Mark	Alex	L
	Kirk	Mary	K
	Shi	Alex	L
$SF \leftarrow \sigma_{Username = `Susan'}(Friends)$	Susan	Mary	K
$SSF \leftarrow Friends \div \pi_{FN,LN}(SF)$	Shi	Lory	М
$RSLT \leftarrow \sigma_{Username \neq `Susan'}(SSF)$	Kirk	Chia	S
RSEI ( O∪sername ≠ Susan (BSI )	Kirk	Alex	L

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

LNFrind

## Extended Relational Operations

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

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

#### Outer Union

- □ it is defined on partially union compatible relations
  - Non union-compatible attributes are kept in r ∪\* s
  - Non union-compatible attributes without value are set
  - Tuples are "matched" over common named attributes like in natural join r ∪\* s

volation v

relation I			
FN LN		MJ	
а	b	cs	
d	а	ce	
С	b	CS	

relation s

LN	CL
g	f
а	sr

FN LN MJ CL Null b CS а d а ce b Null С CS

what about outer intersection or outer difference?

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

#### Outer Join

- □ Join selects only tuples satisfying the join-condition
- □ Outer Join:
  - Left outer join (r) s) keeps every tuple in the left relation
  - Right outer join (r⋈ s) keeps every tuple in the right relation
  - Full outer join (r) 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

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

28

# Aggregate Functions

- Mathematical and Statistical aggregate functions on collections of values
  - SUM, MAXIMUM, MINIMUM, AVERAGE
  - COUNT number of tuples (cardinality)

f <function list> (< relation>)

- Function list is a list of pairs
   (< function, attribute >)
- □ E.g., \$ count SID, AVERAGE GPA (STUDENT)

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

30

## Outer Join ("natural")

r**]⊠**s

FN	LN	MJ	CL
а	b	CS	Null
d	а	ce	sr
С	b	CS	Null

relation <b>r</b>				
FΝ	LN	MJ		
а	b	cs		Ī
d	a	ce		Ī
C	b	cs	ļ '	_

relation **s**FN LN CL

b g f

d a sr

' NATS

' IXL's			
FN	LN	MJ	CL
d	а	ce	sr
b	g	Null	f

FN	LN	MJ	CL
а	b	cs	Null
d	а	ce	sr
С	b	cs	Null
b	g	Null	f

r**]⋈[**s

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

29

### Aggregate Functions: Example

RSLT  $\leftarrow f$  count SID, AVERAGE GPA (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. Chrysanthis - University of Pittsburgh

31

## Example of Aggregation Query

- Q: Find the students with the highest GPA.
- □ Student (SID, Name, Age, GPA)

□ A:

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

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

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

32

#### Grouping

Grouping the tuple in a relation

<grouping attributes> f<function list> (<relation>)

- Tuples are grouped based on the values of grouping attributes
- E.g., major & count SID, AVERAGE GPA (STUDENT)

major	Count_SID	AVERAGE_GPA

CS1555/2055. Panos K. Chrysanthis - University of Pittsburgh

00

#### 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

CS1555/2055, Panos K. Chrysanthis – University of Pittsburgh

34

### Write Queries in Relational Algebra

- □ Deletion:
  - r ← r Relational Expression
  - STUDENT  $\leftarrow$  STUDENT ( $\sigma$  Dept = 'CSD'  $\wedge$  QPA<2.5 (STUDENT))
- □ Insertion:
  - $r \leftarrow r \cup \textit{Relational Expression}$
  - STUDENT ← STUDENT ∪ {(365, `Smith', `John')}
- Updating:
  - $r \leftarrow \Pi$  attributes-to-be-updated (r)
  - STUDENT  $\leftarrow \Pi$  Dept = 'CSD' ( $\sigma$  Dept = 'CS' (STUDENT))

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

35

#### **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

CS1555/2055, Panos K. Chrysanthis - University of Pittsburgh

36

