RELATIONAL ALGEBRA

Relational Querying

- Relational model helps in simple and powerful data retrieval
- Output of query modelled as a relation.
- Based on formal mathematical model.
 - First order predicate Logic
 - Eg : Book('B101')
- Allows for much optimization

Relational Data Languages

- Manipulation and Retrieval of data
- Two Types of Query Languages
 - Relational Algebra
 - Procedural
 - Set of operators operating on relations
 - Relational Calculus
 - NonProcedural
 - Users describe what they want rather than how to compute

Formal Relational Query Languages

- Procedural (Relational Algebra)
 - User specifies what data is required and how to get those data
 - Operational
 - Execution plans can be represented
- Nonprocedural (Relational calculus)
 - User specifies what data is required without specifying how to get those data
 - Declarative
 - Query semantics can be represented
- SQL is the most widely used query language based on Relational calculus.

Relational Algebra

- Procedural language
- Six basic operators
 - select: σ
 - project: ∏
 - union: \cup
 - set difference: -
 - Cartesian product: x
 - rename: ρ
- The operators take one or two relations as inputs and produce a new relation as a result.

Relational Algebra

- o Algebra ?
- Operands
 - Variables or values from which new values can be constructed.
- Operators
 - Symbols denoting procedures to construct new values from Operands.
- Relational Algebra has relations as operands and set operations as operators.
- Satisfies Closure property
 - Output of an operation on relations is a relation itself
 - Operations can be composed.

Types of Operations

- Unary Relational Operations
 - \triangleright Select (σ)
 - \triangleright Project (Π)
 - \triangleright Rename (ρ)
- Binary Relational Operations
 - > Join
 - > natural,semi,Θ-join
 - > Division (÷)

- Set theory Operations
 - > Union (U)
 - \triangleright Intersection (\cap)
 - > Difference (-)
 - ➤ Cartesian Product (X)
- Additional Relational Operations
 - > Outer Joins
 - > Outer Union
 - > Aggregate Functions
 - Eg. Sum, Count, Avg..

Select (denoted by σ (sigma))

- Retrieval of subset of the tuples from a relation based on a selection condition
- Selection condition acts as filter

$$\sigma_{< selection_condition>}(R)$$
 ; R is a relation

Selection condition is a boolean formula.

Tuples satisfying the condition are retained.

Ex:
$$\sigma_{ISBN='B110'}$$
 (Book)

Selection

 $\sigma_F(R)$

| Class | Faculty |
|-------|-------------|
| S7 CS | Dr. Albert |
| S7 CS | Dr. Neumann |
| S7 CS | Dr. Raj |
| S5 CS | Dr. Biswas |

- Select * from class_tab where Class='S7CS'
- \bullet $\sigma_{\text{Class='S7CS'}}$ (class_tab)

Properties of Selection operation

- \circ $\sigma_{\text{selection condition}}$ (R) = S; R and S have same schema
- Number of tuples in S <= Number of tuples in R
- Is commutative

$$\sigma_{}(\sigma_{}(R)) = \sigma_{}(\sigma_{}(R))$$

Cascade sequence of SELECT operations may be applied in any order:

$$\sigma_{\text{cn1}} (\sigma_{\text{cn2}} (\sigma_{\text{cn3}} (R))) = \sigma_{\text{cn2}} (\sigma_{\text{cn3}} (R))$$

Cascade equivalent to conjunction of all the conditions

$$\sigma_{\text{cn1}} (\sigma_{\text{cn2}} (\sigma_{\text{cn3}} (R))) = \sigma_{\text{cn2}} (R))$$

Project (denoted by Π (pi))

- Retrieval of the subset of columns from a relation based on a **specified list of attributes**
- Specified lists forms a projection of attributes

$$\Pi_{\text{attr_list}}(R)$$
; R is a relation

All the tuples of R with only the specified attribute values are retrieved.

Ex:
$$\Pi_{ISBN. Title}(Book)$$

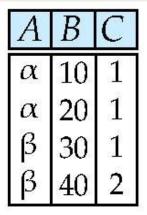
Is
$$\Pi_{ISBN}(\sigma_{Publ_code='PO10'}(Book))$$
 valid?

Which property of relational algebra?

BS
$$<$$
 $\sigma_{Publ_code='PO10'}$ (Book); BS2 $<$ Γ_{ISBN} (BS))

Project Operation – Example

■ Relation *r*:



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

| A | C | A | C |
|---|---|-----|---|
| α | 1 | α | 1 |
| α | 1 | β | 1 |
| β | 1 | ß | 2 |
| ß | 2 | للث | |

Projection

 \blacksquare $\Pi_{A,B}(R)$

| COURSE | воок | FACULTY MEMBER |
|-----------------------|---------------|-------------------|
| Distributed Databases | Pelaggatti | Dr. Albert |
| OS | Tanenbau m | Dr. Neumann |
| DBMS | Date | Dr. Susan |
| CN | Tanenbau m | Dr. Lekshmi |
| DBMS | Corth | Dr. Raj |

- Select course, book from course_fac where course = 'DBMS'
- $\blacksquare \quad \Pi_{\text{course,book}}(\sigma_{\text{Course='DBMS'}}(\text{course_fac}))$

| COURSE | ВООК |
|--------|-------|
| DBMS | Date |
| DBMS | Corth |

Properties of Projection operation

- Removes duplicate tuples. True ?
- Number of tuples in S <= Number of tuples in R

| ISBN | Title | Category | Publ_code |
|------|-------|----------|-----------|
| B111 | FISH | ARTICLE | P010 |
| B112 | GLOW | ARTICLE | P212 |
| B110 | FERT | NEWS | P010 |

 $\Pi_{ISBN, Title}(Book)$

Attribute list contains Key.

 $\Pi_{Publ_code}(Book)$

Removes duplicate tuples.

Not commutative

 $\Pi_{ISBN, Title}(\Pi_{category,publ_code}(Book))$

 $\Pi_{ISBN, Title}(\Pi_{category, ISBN}(Book))$

Results

| ISBN | Title | Category | Publ_code |
|------|-------|----------|-----------|
| B111 | FISH | ARTICLE | P010 |
| B112 | GLOW | ARTICLE | P212 |
| B110 | FERT | NEWS | P010 |

 $\blacksquare \quad \Pi_{\text{category,ISBN}}(\text{Book}) \qquad \qquad \Pi_{\text{category}}(\text{Book})$

| Category | ISBN |
|----------|------|
| ARTICLE | B111 |
| ARTICLE | B112 |
| NEWS | B110 |

Category
ARTICLE
NEWS

 $\blacksquare \quad \Pi_{\text{ISBN, category}}(\sigma_{\text{Publ_code='P010'}}(\text{Book}))$

| ISBN | Category | Publ_code |
|------|----------|-----------|
| B111 | ARTICLE | P010 |
| B110 | NEWS | P010 |

Rename (denoted by ρ (rho))

- The general RENAME operation ρ can be
 - $\rho_{S(B1,B2,...,Bn)}(R)$ changes

the relation name to S, and the column (attribute) names to B1, B1,Bn

 \blacksquare $\rho_s(R)$ changes:

the relation name only to S

$$\rho_{(B1,B2,...,Bn)}(R)$$
 changes:

the column (attribute) names only to B1, B1,Bn

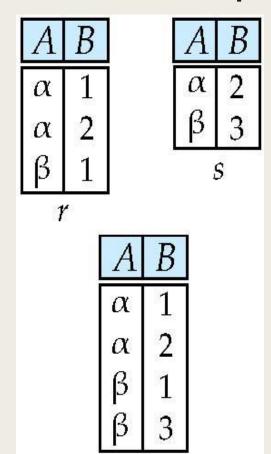
Union (denoted by U)

- RUS is a Binary Operation
- R and S should be type compatible
 - R and S should have same number of attributes
 - Each pair of corresponding attributes must be type compatible (have same or compatible domains)
- Tuples present in R or S or both are retrieved.
- Duplicate tuples are eliminated.
- Ex: Purchase_Invoice U Sales_Invoice

Union Operation – Example

Relations r, s:

 \Box $r \cup s$:



Union Operation

- Notation: $r \cup s$
- Defined as:

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

- \blacksquare 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

$$\Pi_{\text{course_id}} (\sigma_{\text{semester="Fall"} \ \Lambda \ \text{year=2009}} (\text{section})) \cup \Pi_{\text{course_id}} (\sigma_{\text{semester="Spring"} \ \Lambda \ \text{year=2010}} (\text{section}))$$

Union Example

■ Find ISBN, title of the books that were published in 2009 or belongs to NEWS category

| ISBN | Title | Year | Category | Publ_code |
|------|-----------------|------|----------|-----------|
| B111 | FISH | 2007 | ARTICLE | P010 |
| B112 | GLOW | 2009 | ARTICLE | P212 |
| B110 | FERT | 2010 | NEWS | P010 |
| B113 | FINE ARTS | 2009 | NEWS | P010 |
| B114 | INDU – THE MAID | 2008 | NOVEL | P201 |

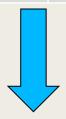
- Result $1 < \Pi_{ISBN, Title}(\sigma_{year=2009}(Book))$
- $\blacksquare \quad Result2 <- \Pi_{ISBN, Title}(\sigma_{Category='NEWS'}(Book))$
- Result <- Result1 U Result2</p>

| ISBN | Title | Year | Category | Publ_code |
|------|-----------|------|----------|-----------|
| B112 | GLOW | 2009 | ARTICLE | P212 |
| B110 | FERT | 2010 | NEWS | P010 |
| B113 | FINE ARTS | 2009 | NEWS | P010 |

Intersection

- Find ISBN, title of the books that were published in 2009 and belongs to NEWS category
- Result $1 < \Pi_{ISBN, Title}(\sigma_{year=2009}(Book))$
- $\blacksquare \quad Result2 <- \Pi_{ISBN, Title}(\sigma_{Category='NEWS'}(Book))$
- Result <- Result1 ∩ Result2

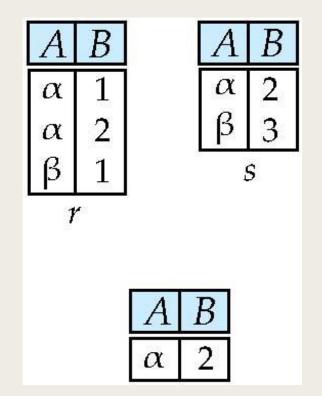
| ISBN | Title | Year | Category | Publ_code |
|------|-----------------|------|----------|-----------|
| B111 | FISH | 2007 | ARTICLE | P010 |
| B112 | GLOW | 2009 | ARTICLE | P212 |
| B110 | FERT | 2010 | NEWS | P010 |
| B113 | FINE ARTS | 2009 | NEWS | P010 |
| B114 | INDU – THE MAID | 2008 | NOVEL | P201 |



| ISBN | Title | Year | Category | Publ_code |
|------|-----------|------|----------|-----------|
| B113 | FINE ARTS | 2009 | NEWS | P010 |

Set-Intersection Operation – Example

■ Relation *r*, s:



 $r \cap s$

Set Difference Operation

- Notation r s
- Defined as:

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

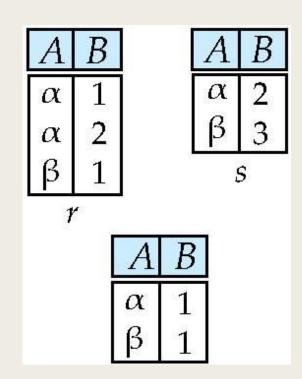
- Set differences must be taken between **compatible** relations.
 - r and s must have the same arity
 - attribute domains of r and s must be compatible
- Example: to find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester

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

Set difference of two relations

■ Relations *r*, s:

 \Box r-s:



Set Difference

- Find ISBN, title of the books that were published in 2009 and does not belong to NEWS category
- Result1 <- $\Pi_{ISBN, Title}(\sigma_{year=2009}(Book))$
- $\blacksquare \quad Result2 <- \Pi_{ISBN, Title}(\sigma_{Category='NEWS'}(Book))$
- Result <- Result1 Result2

| ISBN | Title | Year | Category | Publ_code |
|------|-----------------|------|----------|-----------|
| B111 | FISH | 2007 | ARTICLE | P010 |
| B112 | GLOW | 2009 | ARTICLE | P212 |
| B110 | FERT | 2010 | NEWS | P010 |
| B113 | FINE ARTS | 2009 | NEWS | P010 |
| B114 | INDU – THE MAID | 2008 | NOVEL | P201 |



| ISBN | Title | Year | Category | Publ_code |
|------|-------|------|----------|-----------|
| B112 | GLOW | 2009 | ARTICLE | P212 |

Set operations on different relations

| P_Inv_No | Date | Publ_code |
|----------|----------------|-----------|
| PI_1001 | 29/10/20 09 | P010 |
| PI_2001 | 1/2/2001 | P212 |
| PI_1002 | 12/4/200 7 | P010 |
| PI_1045 | 5/2/2006 | P010 |

| S_Inv_No | Date | Cust_code |
|----------|----------------|-----------|
| SI_1001 | 29/10/20 09 | C010 |
| SI_2001 | 1/2/2001 | C212 |
| SI_1002 | 12/4/200 7 | C010 |
| SI_1045 | 5/2/2006 | C010 |

| _ | P_Inv_No | Date | Publ_code |
|---|----------|----------------|-----------|
| | PI_1001 | 29/10/20 09 | P010 |
| | PI_2001 | 1/2/2001 | P212 |
| | PI_1002 | 12/4/200 7 | P010 |
| | PI_1045 | 5/2/2006 | P010 |
| | SI_1001 | 29/10/20 09 | C010 |
| | SI_2001 | 1/2/2001 | C212 |
| | SI_1002 | 12/4/200 7 | C010 |
| | SI_1045 | 5/2/2006 | C010 |

Properties of Union, Intersect, Difference

- Commutative
 - Satisfied by Union and Intersect
- Associative
 - Satisfied by Union and Intersect
- Distributive
 - RU(S-T) = (RUS) (RUT)
- \blacksquare R-(R-S) = Which Operation?
- $(R \cup S) ((R S) \cup (S R)) = Which operation?$
- R-S ≠ S-R

Cartesian Product

- Combine tuples from two different relations
- Combinatorial manner
- *RXS*
- R(A1, A2, ..., An) **X** S(B1, B2, ..., Bm)
- Q(A1, A2, ..., An, B1, B2, ..., Bm) is the result
- Number of columns in Q

$$cQ = cR + cS$$

Number of tuples in Q

$$nQ = nR * nS$$

Cartesian-Product Operation - Example

☐ Relations *r*, *s*:

| \boldsymbol{A} | В |
|------------------|---|
| α | 1 |
| β | 2 |
| 1 | |

| C | D | E |
|---|----|---|
| α | 10 | a |
| β | 10 | a |
| β | 20 | b |
| γ | 10 | b |
| | s | |

 \Box $r \times s$:

| 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

| ISBN | Title | Category | Publ_code |
|------|-------------|---------------|------------|
| B1: | 11 FISH | I ARTICLE | E P010 |
| B1: | 12 GLO | W ARTIC | LE P212 |
| B1: | 10 FER | T NEWS | P010 |

| Publ_code | Name | Address | |
|-----------|------|---------|--|
| | | | |
| P011 | Pub1 | Add1 | |
| P212 | Pub2 | Add2 | |
| P010 | Pub3 | Add3 | |

Book X Publisher

| ISBN Title C | Category | Publ_code | Publ_code | Name | Address | |
|--------------|----------|-----------|-----------|------|---------|--|
| | | | | | | |
| B111 FISH | ARTICLE | P010 | P011 | Pub1 | Add1 | |
| B111 FISH | ARTICLE | P010 | P212 | Pub2 | Add2 | |
| B111 FISH | ARTICLE | P010 | P010 | Pub3 | Add3 | |
| B112 GLOW | ARTICLE | P212 | P011 | Pub1 | Add1 | |
| B112 GLOW | ARTICLE | P212 | P212 | Pub2 | Add2 | |
| B112 GLOW | ARTICLE | P212 | P010 | Pub3 | Add3 | |
| | | | | | | |

Composition of Operations

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

 \blacksquare rxs

| A | В | C | D | Ε |
|---|---|----------|----|---|
| 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 |

| A | В | C | D | Ε |
|---|---|---|----|---|
| α | 1 | α | 10 | a |
| β | 2 | β | 10 | a |
| β | 2 | β | 20 | b |

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

Joins

• To give meaningful representation for the cartesian product.

| ISBN | Title | Category | Publ_code |
|------|---------|----------|-----------|
| | | | |
| B1: | 11 FISH | l ARTICL | E P010 |
| B1: | 12 GLO | W ARTIC | LE P212 |
| B1: | 10 FER | T NEWS | P010 |

| Publ_code | Name | Address | |
|-----------|------|---------|--|
| | | | |
| P011 | Pub1 | Add1 | |
| P212 | Pub2 | Add2 | |
| P010 | Pub3 | Add3 | |

| Book | \bowtie | Publisher |
|------|-----------|-----------|
|------|-----------|-----------|

| Category | Publ_code |
|-----------|-------------------------|
| | |
| H ARTICLE | P010 |
| W ARTICLE | P212 |
| T NEWS | P010 |
| | H ARTICLE OW ARTICLE |

| Publ_code | Name | Address | |
|-----------|------|---------|--|
| | | | |
| P010 | Pub3 | Add3 | |
| P212 | Pub2 | Add2 | |
| P010 | Pub3 | Add3 | |

Joined Relations

- Join operations take two relations and return as a result another relation.
- A join operation is a Cartesian product which requires that tuples in the two relations match (under some condition). It also specifies the attributes that are present in the result of the join
- The join operations are typically used as subquery expressions in the from clause

Joined Relations

- Join operations take two relations and return as a result another relation.
- These additional operations are typically used as subquery expressions in the from clause
- Join condition defines which tuples in the two relations match, and what attributes are present in the result of the join.
- Join type defines how tuples in each relation that do not match any tuple in the other relation (based on the join condition) are treated.

inner join left outer join right outer join full outer join

Join Conditions

natural

on < predicate>
using $(A_1, A_1, ..., A_n)$

Join (denoted by ⋈)

- Derivative of Cartesian product
- Allows to combine tuples from different relations based on some meaningful condition
- Θ-join
 - Join based on any of the binary comparison operators (>,=,<,>=,>=,!= et. al)
 - Any Boolean formula
- \blacksquare R $\bowtie_{<F>}$ S; F is a join condition
- \blacksquare F = R.a Θ S.b
- Can you Express $R \bowtie_F S$ in terms of other operation?

Join Example

■ Get the publishers name of each book

| ISBN | Title | Category | Pub_cd |
|------|-------|----------|--------|
| B111 | FISH | ARTICLE | P010 |
| B112 | GLOW | ARTICLE | P212 |
| B110 | FERT | NEWS | P010 |

| Pbl_code | Publ_name | Publ_phone |
|----------|-----------|------------|
| P212 | Pearson | 3452198 |
| P010 | McGraw | 8930287 |
| | | |

- Book ⋈_{pub_cd=pbl_code} Publisher

Natural Join

■ Get the publishers name of each book

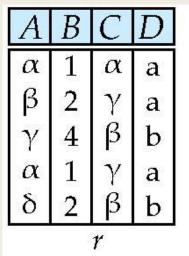
| ISBN | Title | Category | Publ_code |
|------|-------|----------|-----------|
| B111 | FISH | ARTICLE | P010 |
| B112 | GLOW | ARTICLE | P212 |
| B110 | FERT | NEWS | P010 |

| Publ_code | Publ_name | Publ_phone |
|-----------|-----------|------------|
| P212 | Pearson | 3452198 |
| P010 | McGraw | 8930287 |
| | | |

- Book * Publisher
- The join condition is dependent on the columns with same attribute names

Natural Join Example

Relations r, s:



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

 \Box $r \bowtie s$

| A | В | C | D | Ε |
|---|---|----------|---|---|
| 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 |

Natural 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
 - $\Pi_{\text{name, title}}$ ($\sigma_{\text{dept_name="Comp. Sci."}}$ (instructor \bowtie teaches \bowtie course))
- Natural join is associative
 - (instructor \bowtie teaches) \bowtie course is equivalent to instructor \bowtie (teaches \bowtie course)
- Natural join is commutative
 - instruct ⋈ teaches is equivalent to teaches ⋈ instructor

Joins - Example

Relation course

| course_id | title | dept_name | credits |
|-----------|-------------|------------|---------|
| BIO-301 | Genetics | Biology | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |

Relation prereq

| course_id | prereq_id |
|-----------|-----------|
| BIO-301 | BIO-101 |
| CS-190 | CS-101 |
| CS-347 | CS-101 |

 Observe that prereq information is missing for CS-315 and course information is missing for CS-437

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.

Outer Join - Example

Relation loan

| loan-number | branch-name | amount |
|-------------|-------------|--------|
| L-170 | Downtown | 3000 |
| L-230 | Redwood | 4000 |
| L-260 | Perryridge | 1700 |

Relation borrower

| customer-name | loan-number | |
|---------------|-------------|--|
| Jones | L-170 | |
| Smith | L-230 | |
| Hayes | L-155 | |

Outer Join - Example

loan _⋈ Borrower

| loan-number | branch-name | amount | customer-name |
|-------------|-------------|--------|---------------|
| L-170 | Downtown | 3000 | Jones |
| L-230 | Redwood | 4000 | Smith |

☐ Left Outer Join

| loan-number | branch-name | amount | customer-name |
|-------------|-------------|--------|---------------|
| L-170 | Downtown | 3000 | Jones |
| L-230 | Redwood | 4000 | Smith |
| L-260 | Perryridge | 1700 | null |

Outer Join - Example

Right Outer Join

loan ⋈_ borrower

| Ioan-number | branch-name | amount | customer-name |
|-------------|-------------|--------|---------------|
| L-170 | Downtown | 3000 | Jones |
| L-230 | Redwood | 4000 | Smith |
| L-155 | null | null | Hayes |

☐ Full Outer Join

loan Derrower

| loan-number | branch-name | amount | customer-name |
|-------------|-------------|--------|---------------|
| L-170 | Downtown | 3000 | Jones |
| L-230 | Redwood | 4000 | Smith |
| L-260 | Perryridge | 1700 | null |
| L-155 | null | null | Hayes |

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
- For duplicate elimination and grouping, null is treated like any other value, and two nulls are assumed to be the same

Division(Quotient)

- Select those rows which are sufficient to provide all values in certain columns
 - Which publishers publishes all types of books?
 - R = Book; S = Category
 - $R \div S = \{(x) \mid \exists (x,y) \in R, \forall y \in S\}$
 - $= \Pi_A(R) \Pi_A((\Pi_A(R)XS) R)$
 - A: set of attributes not in S
 - Let r and s be relations on schemas R and S respectively where
 - R = (A1, ..., Am, B1, ..., Bn)
 - S = (B1, ..., Bn)
 - Then r/s is a relation on schema R S = (A1, ..., Am)

| ISBN | Title | Category | Publ_code |
|------|-------|----------|-----------|
| B111 | FISH | ARTICLE | P010 |
| B112 | GLOW | ARTICLE | P212 |
| B110 | FERT | NEWS | P010 |

| Category |
|----------|
| ARTICLE |
| NEWS |

Example of Division A/B

| A_{\perp} | | B1 | B2 | В3 |
|-------------|--|------|-----------------|------|
| sno | pno | pno | pno | pno |
| s1 | p1 | p2 | p2 | p1 |
| s1 | p2 | | p4 | p2 |
| s1 | p3 | A/B1 | | p4 |
| s1 | p4 | | | |
| s2 | p1 | sno | A/B2 | |
| s2 | p2 | s1 | sno | A/B3 |
| s3 | p2 | s2 | s1 | sno |
| s4 | p1 p2 p3 p4 p1 p2 p2 p2 p4 | s3 | $\overline{s4}$ | s1 |
| s4 | p4 | s4 | <u> </u> | |

Division Example

Relations r, s:

| Α | В | С | D | Ε |
|--|---|---------|---|----------------------------|
| α | а | α | а | 1 |
| $egin{array}{c} lpha \\ lpha \\ eta \\ eta \\ eta \\ \gamma \end{array}$ | а | γ | а | 1 1 1 3 1 1 |
| α | а | γ | b | 1 |
| β | а | γ | а | 1 |
| β | а | γ | b | 3 |
| γ | а | γ | а | 1 |
| γ | а | γ | b | 1 |
| γ | а | β | b | 1 |
| | | r | | |

D E
a 1
b 1

r÷s:

| Α | В | С |
|---|---|----------|
| α | а | γ |
| γ | а | γ |

Questions

- Find names of Publishers who have published book B101
- Find the names of sales representatives who work for the publishers who have published books in News as well as in Article category

Aggregate functions

- Max
- Min
- Sum
- Count
- Average

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 value

sum: sum of values

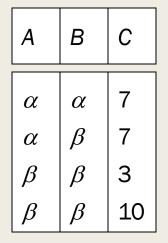
count: number of values

Aggregate operation in relational algebra

- 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

Aggregate Operation - Example

■ Relation *r*:



 $g_{\text{sum(c)}}(\mathbf{r})$

sum-C

27

Aggregate Operation - Example

■ Relation account grouped by branch-name:

| branch-name | account-number | balance |
|-------------|----------------|---------|
| Perryridge | A-102 | 400 |
| Perryridge | A-201 | 900 |
| Brighton | A-217 | 750 |
| Brighton | A-215 | 750 |
| Redwood | A-222 | 700 |

branch-name $g_{sum(balance)}$ (account)

| branch-name | balance | |
|-------------|---------|--|
| Perryridge | 1300 | |
| Brighton | 1500 | |
| Redwood | 700 | |

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

branch-name **g** sum(balance) as sum-balance (account)

Advantages of Relational Algebra

- Rigorously defined simple and yet powerful query language
- More operational
- Useful for query evaluation plans
- Several ways of expressing a query. Query optimizer should choose the most efficient one.

THANK YOU

References

■ Silberschatz A Korth H F and SudharshanS, "Database System Concepts", 6th Edition, TMH publishing company limited, 2011.