



RELATIONAL DATA LANGUAGES



Part 2_2

Outer Join

- An extension of the join operation that avoids loss of information.
- Computes the join and then adds tuples from one relation that do 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*

<i>loan-number</i>	<i>branch-name</i>	<i>amount</i>
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

□ Relation *borrower*

<i>customer-name</i>	<i>loan-number</i>
Jones	L-170
Smith	L-230
Hayes	L-155



Left Outer Join

■ Join

loan ⋈ *Borrower*

<i>loan-number</i>	<i>branch-name</i>	<i>amount</i>	<i>customer-name</i>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

□ Left Outer Join

loan ⋈_L *Borrower*

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



Right Outer Join, Full Outer Join

■ Right Outer Join

loan ⋈_r *borrower*

<i>loan-number</i>	<i>branch-name</i>	<i>amount</i>	<i>customer-name</i>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	<i>null</i>	<i>null</i>	Hayes

Outer Join

loan ⋈_f *borrower*

<i>loan-number</i>	<i>branch-name</i>	<i>amount</i>	<i>customer-name</i>
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	<i>null</i>
L-155	<i>null</i>	<i>null</i>	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 ?
 - $R = \text{Book} ; S = \text{Category}$
 - $R \div S = \{(x) \mid \exists (x,y) \in R, \forall y \in S\}$
 - $= \Pi_A(R) - \Pi_A((\Pi_A(R) \times S) - R)$
 - $A : \text{set of attributes not in } S$

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

Category
ARTICLE
NEWS



Another Division Example

Relations r, s :

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

r

D	E
a	1
b	1

s

$r \div s$:

A	B	C
α	a	γ
γ	a	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

is an extended relational algebra operation denoted by Calligraphic \mathcal{G} (letter G written in calligraphic font) .

Very useful to apply a function to a collection of values to generate a single result



- Most common aggregate functions:
- Sum :sums the values in the collection
- Avg : computes average of values in the collection
- Count : counts number of elements in the collection
- Min : returns minimum value in the collection
- Max : returns maximum value in the collection
- Example
- “Find the total amount owed to the credit company.
- G sum(balance)(credit_acct)



Aggregate Operation – Example

- Relation r :

A	B	C
α	α	7
α	β	7
β	β	3
β	β	10

$G_{\text{sum}(c)}(r)$

$sum-C$
27



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.

