




International University, VNU-HCMC

School of Computer Science and Engineering

Lecture 4: Relational Model and Algebra

Instructor: Nguyen Thi Thuy Loan

nttloan@hcmiu.edu.vn, nthithuyloan@gmail.com
<https://nttloan.wordpress.com/>



International University, VNU-HCMC


Purpose of the Lecture

- Introduce core concepts of the Relational Model.
- Explain its role as the foundation of database systems.
- Present the primary Relational Algebra operations.

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

2


 International University, VNU-HCMC

Warm-up Question

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Why do modern database systems rely on the relational model, and how can we formally describe queries on data?

Duke CS, Fall 2024 3


 International University, VNU-HCMC


Outlines

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Relational model
- Relational algebra

Duke CS, Fall 2024 4


International University, VNU-HCMC



Assoc. Prof. Nguyen Thi Thuy Loan, PhD


Acknowledgement

- The following slides are referenced from Dr. Sudeepa Roy, Duke University.

Duke CS, Fall 2024

5

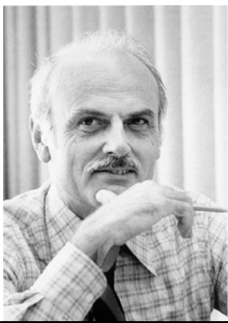

International University, VNU-HCMC


Assoc. Prof. Nguyen Thi Thuy Loan, PhD

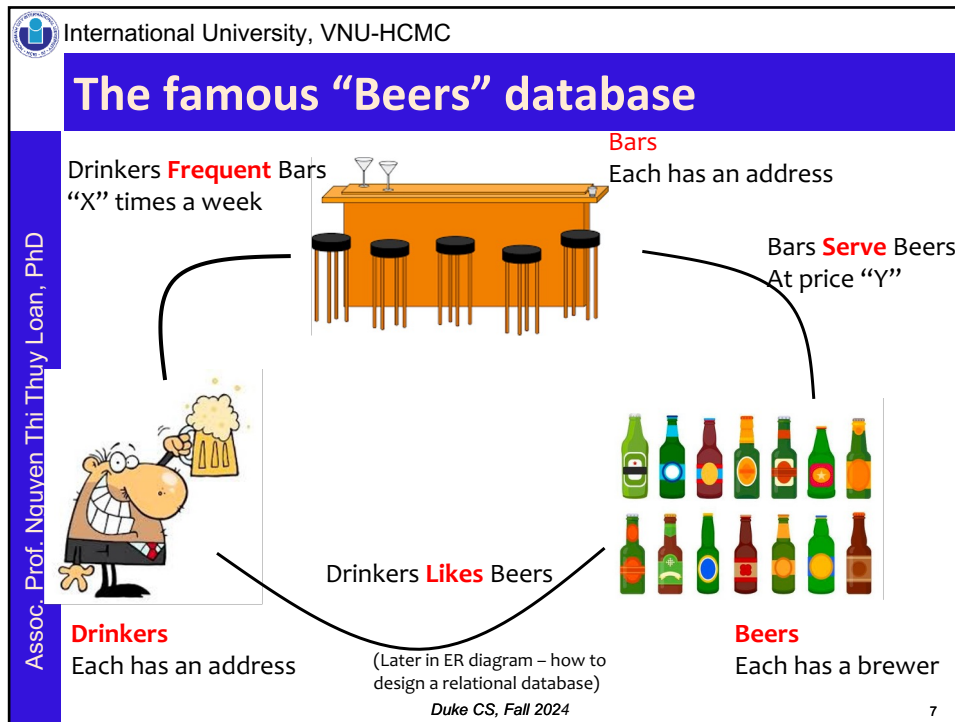
Edgar F. Codd (1923-2003)

- Served as a pilot in the Royal Air Force during WW2
- Invented the Relational Model and Algebra at IBM (1970)
- Awarded the Turing Award in 1981 for his pioneering work
- His ideas led to the development of RDBMS (Relational Database Management Systems)

http://en.wikipedia.org/wiki/File:Edgar_F_Codd.jpg



Duke CS, Fall 2024



International University, VNU-HCMC See online database for more tuples

"Beers" as a Relational Database

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Beer

Name	brewer
Budweiser	Anheuser-Busch Inc.
Corona	Grupo Modelo
Dixie	Dixie Brewing

Drinker

name	address
Amy	100 W. Main Street
Ben	101 W. Main Street
Dan	300 N. Duke Street

Frequents


drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Likes

drinker	beer
Amy	Corona
Dan	Budweiser
Dan	Corona
Ben	Budweiser

Duke CS, Fall 2024

Assoc. Prof. Nguyen Thi Thuy Loan, PhD



International University, VNU-HCMC

Relational data model

Assoc. Prof. Nguyen Thi Thuy Loan, PhD


- A database is a collection of **relations** (or **tables**)
- Each relation has a set of **attributes** (or **columns**)
- Each attribute has a name and a **domain** (or **type**)
 - No set-valued attributes allowed

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Duke CS, Fall 2024

9



International University, VNU-HCMC

Relational data model

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Each relation is a set of **tuples** (or **rows**)
 - Each tuple has a value for each attribute of the relation
 - No duplicate tuples (same values across all attributes)
 - Row order doesn't matter (even though output is always in some order)
- In practice, SQL supports “bags” (allows duplicates)
- Why? Simplicity is a virtue!

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Duke CS, Fall 2024

10

International University, VNU-HCMC

Schema vs. instance

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Beer

Name	brewer
Budweiser	Anheuser-Busch Inc.
Corona	Grupo Modelo
Dixie	Dixie Brewing

Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

- Ordering of rows doesn't matter (even though output is always in some order)

Duke CS, Fall 2024 11

International University, VNU-HCMC

Schema vs. instance

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Schema (metadata)
 - Specifies the logical structure of data
 - Is defined at setup time
 - Rarely changes
- Instance
 - Represents the data content
 - Changes rapidly, but always conforms to the schema
- Compare types vs. collections of objects of these types in a programming language

Duke CS, Fall 2024 12

International University, VNU-HCMC

Example

- Schema (metadata)
 - Beer (name string, brewer string)
 - Serves (bar string, beer string, price float)
 - Frequents (drinker string, bar string, times_a_week int)
- Instance
 - Beer {<Budweiser, Anheuser-Busch Inc.>, <corona, Grupo Modelo>, ...}
 - Serves {<the Edge, Budweiser, 2.50>, <The Edge, Corona, 3.0>, ...}
 - Frequents {<Ben, Satisfaction, 2>, <Dan, The Edge, 1>, ...}

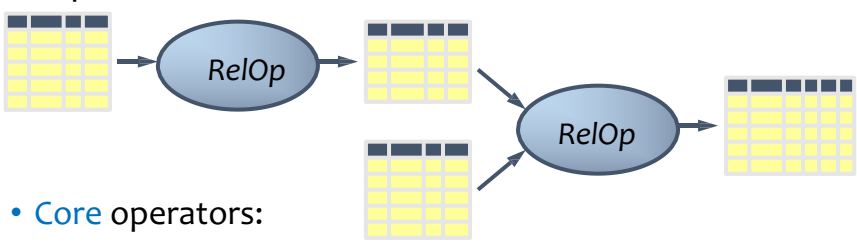
Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 13

International University, VNU-HCMC

Relational algebra


- A language for querying relational databases on “operators”



- Core operators:
 - Selection, projection, cross product, union, difference, and renaming
- Additionally, derived operators include:
 - Join, natural join, intersection, etc.
- Compose operators to make complex queries.

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 14



International University, VNU-HCMC


Selection

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Input: a table R
- Notation: $\sigma_p R$
 - p is called a selection condition (or predicate)
- Purpose: filter rows according to some criteria
- Output: same columns as R , but only rows of R that satisfy p (*set!*)

Duke CS, Fall 2024

15



International University, VNU-HCMC

Selection example

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Find beers with price < 2.75

Serves


bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

$\sigma_{price < 2.75}$ Serves

bar	beer	price
The Edge	Budweiser	2.50
Satisfaction	Budweiser	2.25

Duke CS, Fall 2024

16


International University, VNU-HCMC

More on selection


Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Selection condition can include any column of R , constants, comparison ($=$, \leq , etc.), and Boolean connectives (\wedge : and, \vee : or, \neg : not)
 - Example: Serves tuples for “The Edge” or price ≥ 2.75

$$\sigma_{\text{bar} = \text{'The Edge'} \vee \text{price} \geq 2.75}^{\text{Serves}}$$

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Duke CS, Fall 2024
17


International University, VNU-HCMC

More on selection

Assoc. Prof. Nguyen Thi Thuy Loan, PhD


- You must be able to evaluate the condition for each row of the input table.
 - Example: the most expensive beer at any bar

$$\sigma_{\text{price} \geq \text{every price in Servers}}^{\text{User}}$$

WRONG!

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Duke CS, Fall 2024
18



International University, VNU-HCMC


Projection

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Input: a table R
- Notation: $\pi_L R$
 - L is a list of columns in R
- Purpose: output chosen columns
- Output: same rows, but only the columns in L (*set!*)

Duke CS, Fall 2024

19



International University, VNU-HCMC

Projection

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Example: Find all the prices for each beer

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25


$\pi_{\text{beer, price}}$ Serves

beer	price
Budweiser	2.50
Corona	3.00
Budweiser	2.25

Output of $\pi_{\text{beer}} \text{Serves}$?

Duke CS, Fall 2024

20



International University, VNU-HCMC

More on Projection

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Duplicate output rows are removed (by definition)
- Example: beer on servers

Serves


bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

π_{beer} **Serves**

beer
Budweiser
Corona

Duke CS, Fall 2024

21



International University, VNU-HCMC

Cross product

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Input: two tables R and S
- Notation: $R \times S$
- Purpose: pairs rows from two tables
- Output: for each row r in R and each s in S , output a row rs (concatenation of r and s)

Duke CS, Fall 2024

22

International University, VNU-HCMC

Cross product

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Bar x Frequents

name	address	drinker	bar	times_a_week
The Edge	108 Morris Street	Ben	Satisfaction	2
The Edge	108 Morris Street	Dan	The Edge	1
The Edge	108 Morris Street	Dan	Satisfaction	2
Satisfaction	905 W. Main Street	Ben	Satisfaction	2
Satisfaction	905 W. Main Street	Dan	The Edge	1
Satisfaction	905 W. Main Street	Dan	Satisfaction	2

Duke CS, Fall 2024 23

International University, VNU-HCMC

Cross product

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Ordering of columns is unimportant as far as contents are concerned.


name	address	drinker	bar	times_a_week
The Edge	108 Morris Street	Ben	Satisfaction	2
The Edge	108 Morris Street	Dan	The Edge	1
The Edge	108 Morris Street	Dan	Satisfaction	2
Satisfaction	905 W. Main Street	Ben	Satisfaction	2
Satisfaction	905 W. Main Street	Dan	The Edge	1
Satisfaction	905 W. Main Street	Dan	Satisfaction	2

=

drinker	bar	times_a_week	name	address
Ben	Satisfaction	2	The Edge	108 Morris Street
Dan	The Edge	1	The Edge	108 Morris Street
Dan	Satisfaction	2	The Edge	108 Morris Street
Ben	Satisfaction	2	Satisfaction	905 W. Main Street
Dan	The Edge	1	Satisfaction	905 W. Main Street
Dan	Satisfaction	2	Satisfaction	905 W. Main Street

- So cross product is commutative, i.e., for any R and S, $R \times S = S \times R$ (up to the ordering of columns)

Duke CS, Fall 2024 24



International University, VNU-HCMC

Derived operator: join

Assoc. Prof. Nguyen Thi Thuy Loan, PhD


(Also known as “**theta-join**”: most general joins)
One of the most important operations!

- Input: two tables R and S
- Notation: $R \bowtie_p S$
 - p is called a join condition (or predicate)
- Purpose: relate rows from two tables according to some criteria
- Output: for each row r in R and each row s in S , output a row rs if r and s satisfy p
- Shorthand for $\sigma_p(R \times S)$
- Predicate p only has equality ($A = 5 \wedge B = 7$): equijoin

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

25



International University, VNU-HCMC

Join example

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Extend Frequent relation with addresses of the bars

Frequent $\bowtie_{\text{bar} = \text{name } Bar}$

Ambiguous attribute? Prefix a column reference with table name and “.” to disambiguate identically named columns from different tables. Ex. Use Bar.name

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street

Frequent


drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

name	address	drinker	bar	times_a_week
The Edge	108 Morris Street	Ben	Satisfaction	2
The Edge	108 Morris Street	Dan	The Edge	1
The Edge	108 Morris Street	Dan	Satisfaction	2
Satisfaction	905 W. Main Street	Ben	Satisfaction	2
Satisfaction	905 W. Main Street	Dan	The Edge	1
Satisfaction	905 W. Main Street	Dan	Satisfaction	2

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

26



International University, VNU-HCMC

Join Types

- Theta Join
- Equi-Join
- Natural Join
- Later, (left/right) outer join, semi-join

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024
27


International University, VNU-HCMC

Derived operator: natural join

- Input: two tables R and S
- Notation: $R \bowtie S$ (i.e. no subscript)
- Purpose: relate rows from two tables, and
 - Enforce equality between identically named columns
 - Eliminate one copy of identically named columns
- Shorthand for $\pi_L(R \bowtie_p S)$, where
 - p equates each pair of columns common to R and S
 - L is the union of column names from R and S (with duplicate columns removed)

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024
28

International University, VNU-HCMC

Natural join example

Serves \bowtie Likes
 $= \pi_{\text{bar, beer, price, drinker}}(\text{Serves} \bowtie_{\text{Serves.beer} = \text{Likes.beer}} \text{Likes})$

Serves

bar	beer	price
The Edge	Budweiser	2.50
The Edge	Corona	3.00
Satisfaction	Budweiser	2.25

Likes

drinker	beer
Amy	Corona
Dan	Budweiser
Dan	Corona
Ben	Budweiser

Serves \bowtie Likes

bar	beer	price	drinker
The Edge	Budweiser	2.50	Dan
The Edge	Budweiser	2.50	Ben
The Edge	Corona	3.00	Amy
The Edge	Corona	3.00	Dan
...	

Natural Join is on beer. Only one column for beer in the output
 What happens if the tables have two or more common columns?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 29

International University, VNU-HCMC

Union

- Input: two tables R and S
- Notation: $R \cup S$
 - R and S must have identical schema
- Output:
 - Has the same schema as R and S
 - Contains all rows in R and all rows in S (with duplicate rows removed)

Important for set operations:
 Union Compatibility

Example on board

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 30

International University, VNU-HCMC

Example

StudentID	Name
3	Chi
4	Dũng
5	Hạnh
6	Khoa

StudentID	Name
1	An
2	Bình
3	Chi
4	Dũng

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 31

International University, VNU-HCMC


Derived operator: intersection

- Input: two tables R and S
- Notation: $R \cap S$
 - R and S must have identical schema
- Output:
 - Has the same schema as R and S
 - Contains all rows that are in both R and S
- How can you write it using other operators?
- Shorthand for $R - (R - S)$
- Also equivalent to $S - (S - R)$
- And to $R \bowtie S$

Important for set operations:
Union Compatibility

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 32



International University, VNU-HCMC

Example

Student_A

StudentID	Name
3	Chi
4	Dũng
5	Hạnh
6	Khoa


Student_B

StudentID	Name
1	An
2	Bình
3	Chi
4	Dũng

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

33



International University, VNU-HCMC

Difference

- Input: two tables R and S
- Notation: $R - S$
 - R and S must have identical schema
- Output:
 - Has the same schema as R and S
 - Contains all rows in R that are not in S


Important for set operations:
Union Compatibility

Example on board

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

34



International University, VNU-HCMC

Example

Student_A

StudentID	Name
3	Chi
4	Dũng
5	Hạnh
6	Khoa


Student_B

StudentID	Name
1	An
2	Bình
3	Chi
4	Dũng

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

35



International University, VNU-HCMC


Renaming

- Input: a table R
- Notation: $\rho_S R$, $\rho_{(A_1, A_2, \dots)} R$, or $\rho_{S(A_1, A_2, \dots)} R$
- Purpose: “rename” a relation and/or its attributes
- Output: a new relation with the same rows as R , but a new name and/or new column names
- Example:
 - $\rho(\text{Tempsids}, \text{Student})$
 - $\rho(S(\text{SID}, \text{SName}), \text{Student})$

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

36



International University, VNU-HCMC


Renaming

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Why use it?
 - Avoid confusion when column names are identical
 - Create consistent attribute names for natural joins
- Notes:
 - Renaming does not modify the original database
 - Think of it as creating a temporary copy of R with new names

Duke CS, Fall 2024

37



International University, VNU-HCMC

Renaming example

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Find drinkers who frequent both “The Edge” and “Satisfaction”

Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

WRONG!

$$\pi_{\text{drinker}} \left(\text{Frequents} \bowtie \begin{array}{l} \text{Bar} = \text{'The Edge'} \wedge \\ \text{Bar} = \text{'Satisfaction'} \wedge \\ \text{drinker} = \text{drinker} \end{array} \text{Frequents} \right)$$

Renamed!

$$\pi_{\text{uid}_1} \left(\begin{array}{c} \bowtie \\ \text{b1} = \text{'The Edge'} \wedge \text{b2} = \text{'Satisfaction'} \wedge \text{d1} = \text{d2} \end{array} \begin{array}{c} \rho_{(\text{d1}, \text{b1}, \text{t1})} \text{Frequents} \\ \rho_{(\text{d2}, \text{b2}, \text{t2})} \text{Frequents} \end{array} \right)$$

Duke CS, Fall 2024

38

International University, VNU-HCMC

Exercise

Given a database

- Student(SID, Sname, age, add, phone, ID_card)
- Course(CID, Cname, credit)
- Enrolled (SID, CID, grade)

- Find all the students (SID, Sname) who enroll in the PDM course?
- Find the students (SID, Sname) who enroll in both the OOP and PDM courses?
- Find the course name that didn't have any students to enroll?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 39

International University, VNU-HCMC

Expression tree notation

- Find addresses of all bars that 'Dan' frequents

What if you move σ to the top?
Still correct?
More or less efficient?

Also called logical Plan tree

Bar

name	address
The Edge	108 Morris Street
Satisfaction	905 W. Main Street


Frequents

drinker	bar	times_a_week
Ben	Satisfaction	2
Dan	The Edge	1
Dan	Satisfaction	2

Equivalent to

$\pi_{\text{address}}(\text{Bar} \bowtie_{\text{bar=name}} (\sigma_{\text{drinker='Dan'}} \text{Frequents}))$

Duke CS, Fall 2024 40



International University, VNU-HCMC

Summary of core operators

- Selection: $\sigma_p R$
- Projection: $\pi_L R$
- Cross product: $R \times S$
- Union: $R \cup S$
- Difference: $R - S$
- Renaming: $\rho_{S(A_1, A_2, \dots)} R$
 - Does not really add “processing” power

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024
41


International University, VNU-HCMC

Summary of derived operators

- Join: $R \bowtie_p S$
- Natural join: $R \bowtie S$
- Intersection: $R \cap S$
- Many more
 - Semijoin, anti-semijoin, quotient, ...

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024
42

International University, VNU-HCMC

Exercise

Frequents(drinker, bar, times_of_week)
Bar(name, address)
Drinker(name, address)

- Find the bars that are not frequented by drinkers who live at the address “300 N. Duke Street”.

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 43

International University, VNU-HCMC

Exercise

Frequents(drinker, bar, times_of_week)
Bar(name, address)
Drinker(name, address)

- Find the bars that are not frequented by drinkers who live at the address “300 N. Duke Street”.

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Diagram illustrating the query logic:

```

graph TD
    Root[ ] --- L1[ ]
    L1 --- L2[ ]
    L2 --- L3[ ]
    L3 --- L4[ ]
    L3 --- L5[ ]
    L4 --- L6[ ]
    L5 --- L7[ ]
    L6 --- L8[ ]
    L7 --- L9[ ]
    L8 --- L10[ ]
    L9 --- L11[ ]
    L10 --- L12[ ]
    L11 --- L13[ ]
    L12 --- L14[ ]
    L13 --- L15[ ]
    L14 --- L16[ ]
    L15 --- L17[ ]
    L16 --- L18[ ]
    L17 --- L19[ ]
    L18 --- L20[ ]
    L19 --- L21[ ]
    L20 --- L22[ ]
    L21 --- L23[ ]
    L22 --- L24[ ]
    L23 --- L25[ ]
    L24 --- L26[ ]
    L25 --- L27[ ]
    L26 --- L28[ ]
    L27 --- L29[ ]
    L28 --- L30[ ]
    L29 --- L31[ ]
    L30 --- L32[ ]
    L31 --- L33[ ]
    L32 --- L34[ ]
    L33 --- L35[ ]
    L34 --- L36[ ]
    L35 --- L37[ ]
    L36 --- L38[ ]
    L37 --- L39[ ]
    L38 --- L40[ ]
    L39 --- L41[ ]
    L40 --- L42[ ]
    L41 --- L43[ ]
    L42 --- L44[ ]
    L43 --- L45[ ]
    L44 --- L46[ ]
    L45 --- L47[ ]
    L46 --- L48[ ]
    L47 --- L49[ ]
    L48 --- L50[ ]
    L49 --- L51[ ]
    L50 --- L52[ ]
    L51 --- L53[ ]
    L52 --- L54[ ]
    L53 --- L55[ ]
    L54 --- L56[ ]
    L55 --- L57[ ]
    L56 --- L58[ ]
    L57 --- L59[ ]
    L58 --- L60[ ]
    L59 --- L61[ ]
    L60 --- L62[ ]
    L61 --- L63[ ]
    L62 --- L64[ ]
    L63 --- L65[ ]
    L64 --- L66[ ]
    L65 --- L67[ ]
    L66 --- L68[ ]
    L67 --- L69[ ]
    L68 --- L70[ ]
    L69 --- L71[ ]
    L70 --- L72[ ]
    L71 --- L73[ ]
    L72 --- L74[ ]
    L73 --- L75[ ]
    L74 --- L76[ ]
    L75 --- L77[ ]
    L76 --- L78[ ]
    L77 --- L79[ ]
    L78 --- L80[ ]
    L79 --- L81[ ]
    L80 --- L82[ ]
    L81 --- L83[ ]
    L82 --- L84[ ]
    L83 --- L85[ ]
    L84 --- L86[ ]
    L85 --- L87[ ]
    L86 --- L88[ ]
    L87 --- L89[ ]
    L88 --- L90[ ]
    L89 --- L91[ ]
    L90 --- L92[ ]
    L91 --- L93[ ]
    L92 --- L94[ ]
    L93 --- L95[ ]
    L94 --- L96[ ]
    L95 --- L97[ ]
    L96 --- L98[ ]
    L97 --- L99[ ]
    L98 --- L100[ ]
  
```

The diagram shows a query tree structure. The root node branches into two main paths. The left path represents "All bars" and consists of a projection operator ρ_{bar} followed by a selection operator π_{name} applied to the *Bar* table. The right path represents "Bars that the drinkers at this address frequent" and consists of a projection operator π_{bar} followed by a join operator \bowtie with the condition $\text{drinker} = \text{name}$. This join operator connects to the *Frequents* table and a selection operator $\sigma_{\text{address}='300 N.Duke Street'}$ applied to the *Drinker* table.

Duke CS, Fall 2024 44

International University, VNU-HCMC

Frequents(drinker, bar,
times_of_week) Bar(name, address)
Drinker(name, address)

41

A trickier Exercise

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- For each bar, find the drinkers who frequent it the maximum number of times per week.

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

45

International University, VNU-HCMC

Frequents(drinker, bar,
times_of_week) Bar(name, address)
Drinker(name, address)

42

A trickier Exercise

Assoc. Prof. Nguyen Thi Thuy Loan, PhD


- For each bar, find the drinkers who frequent it the maximum number of times per week.
 - Who do NOT visit a bar max no. of times?
 - Whose times_of_weeks is lower than somebody else's for a given bar

A deeper question:
When (and why) is “-” needed?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024

46



International University, VNU-HCMC


Expressions in a Single Assignment

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Example: the theta-join $R3 = R1 \bowtie_C R2$ can be written: $R3 := \sigma_C (R1 \times R2)$
- Precedence of relational operators:
 1. $[\sigma, \pi, \rho]$ (highest)
 2. $[X, \bowtie]$
 3. \cap
 4. $[\cup, -]$

Duke CS, Fall 2024

47



International University, VNU-HCMC

Example


Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Find the names of sailors who have reserved boat number 103.

- Sailors(sid, sname, rating, age)
- Boats(bid, bname, color)
- Reserves(sid, bid, day)

Duke CS, Fall 2024

48



International University, VNU-HCMC

Find sailors who've reserved a red or a green boat.

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Sailors(sid, sname, rating, age)

Boats(bid, bname, color) Use of rename operation

Reserves(sid, bid, day)

- Can identify all red or green boats, then find sailors who've reserved one of these boats:


$$\rho \quad (Tempboats, (\sigma_{color='red' \vee color='green'} Boats))$$

$$\pi_{sname}(Tempboats \bowtie Reserves \bowtie Sailors)$$

Can also define Tempboats using union. Try the “AND” version yourself

Duke CS, Fall 2024

49



International University, VNU-HCMC

Division

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Not supported as a primitive operator, but useful for expressing queries like:
 - Find sailors who have reserved all boats.
- Let A have 2 fields, x and y ; B have only field y :

$$A/B = \pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$$

- i.e., A/B contains all x tuples (sailors) such that for every y tuple (boat) in B , there is an xy tuple in A .
- Or: If the set of y values (boats) associated with an x value (sailor) in A contains all y values in B , the x value is in A/B .

Duke CS, Fall 2024

50

International University, VNU-HCMC

Examples of Division A/B

$\pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

sno	pno
s1	p1
s1	p
s1	2
s1	p
s2	3
s2	p
s3	4
s4	p
s4	1
s4	2
s4	p

$\pi_x(A)$

pno
p2

B1

$\pi_x(A)$

pno
p2
p4

B2

$\pi_x(A)$

pno
p1
p2
p4

B3

sno
s1
s2
s3
s4

A/B1

sno
s1
s4

A/B2

sno
s1

A/B3

Duke CS, Fall 2024 51

International University, VNU-HCMC

Expressing A/B Using Basic Operators

- Division is not essential operator; just a useful shorthand.
 - (Also true of joins, but joins are so common that systems implement joins specially)
- **Idea:** For A/B, find all x values that are not disqualified by any y in B.
 - An x value is disqualified if, when combined with some y from B, the tuple (x,y) does not appear in A.

Disqualified x values: all disqualified tuples

$A/B:$ $\pi_x(A) - \pi_x((\pi_x(A) \times B) - A)$

Duke CS, Fall 2024 52

International University, VNU-HCMC

Find the name of sailors who've reserved all boats

Sailors(sid, sname, rating, age)
Boats(bid, bname, color)
Reserves(sid, bid, day)

- Uses **division**; schemas of the input relations to/ must be carefully chosen:

$$\rho \text{ (TempSids, } (\pi_{sid, bid} \text{Reserves}) / (\pi_{bid} \text{Boats}))$$

$$\pi_{sname} (\text{TempSids} \bowtie \text{Sailors})$$

- To find sailors who've reserved all 'Interlake' boats:

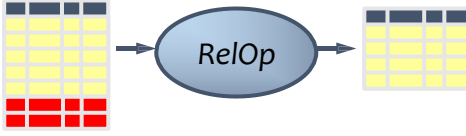
$$\dots / \pi_{bid} (\sigma_{bname = 'Interlake'} \text{Boats})$$

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 53

International University, VNU-HCMC

Monotone operators



Add more rows to the input...

What happens to the output?

- If we add more rows to the input... What happens to the output?
- An operator is monotone if adding rows to the input never removes existing output rows.
- An operator is non-monotone if adding rows to the input may cause some old output rows to disappear.

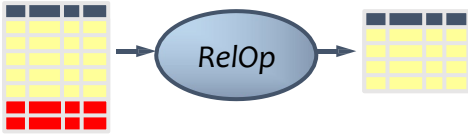
Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Duke CS, Fall 2024 54

International University, VNU-HCMC 43

Monotone operators

Assoc. Prof. Nguyen Thi Thuy Loan, PhD



Add more rows to the input... What happens to the output?

- If old results always remain valid when more data is added, the operator is **monotone**.

Example: Union, Join, Selection.

- Adding data only increases (or keeps) the results. Old answers are never invalidated.
- Formal definition, for a monotone operator op :
If $R \subseteq R'$ then $op(R) \subseteq op(R')$ for any R, R'

Duke CS, Fall 2024 55


International University, VNU-HCMC

Which operators are non-monotone?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

• Selection: $\sigma_p R$	Monotone
• Projection: $\pi_L R$	Monotone
• Cross product: $R \times S$	Monotone
• Join: $R \bowtie_p S$	Monotone
• Natural join: $R \bowtie S$	Monotone
• Union: $R \cup S$	Monotone
• Difference: $R - S$	Monotone w.r.t. R ; non-monotone w.r.t S
• Intersection: $R \cap S$	Monotone

Duke CS, Fall 2024 56



International University, VNU-HCMC

45

Why is “–” needed for “highest”?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Monotone queries:


- If you add more rows, the old results stay correct.
- Example: simple selections or joins.

Is “highest” monotone?

- No!
- Suppose the current highest price = 3.0.
- If we add a new row with price = 3.01,
 The old answer (3.0) is wrong.
- So it must use a difference!

Duke CS, Fall 2024

57



International University, VNU-HCMC

Why do we need each core relational operator (X)?


Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Difference ($R - S$)

- It is the **only non-monotone operator**.
- Adding new rows to the input can remove results (because rows might now be subtracted).
- That’s why we can’t build it from only monotone operators.

Duke CS, Fall 2024

58


International University, VNU-HCMC

Why do we need each core relational operator (X)?


Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Projection ($\pi_L R$)

- It is the **only operator that removes columns**.
- Other operators only combine or filter rows, but don't reduce attributes.
- Without projection, we couldn't simplify a relation to fewer attributes.

Duke CS, Fall 2024

59


International University, VNU-HCMC

Why do we need each core relational operator (X)?


Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Cross product ($R \times S$)

- It is the **only operator that adds new columns**.
- Combining attributes from two relations creates a wider schema.

Duke CS, Fall 2024

60

 International University, VNU-HCMC


Why do we need each core relational operator (X)?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Union ($R \cup S$)

- It is the **only operator that allows you to add rows** from another relation.
- It's the building block for combining datasets.

Duke CS, Fall 202461

 International University, VNU-HCMC


Why do we need each core relational operator (X)?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Selection ($\sigma_p R$)

- It is the only operator that **removes rows** based on a condition.
- It's the core operator for **filtering tuples** in a relation.

Duke CS, Fall 202462

 International University, VNU-HCMC

46


Extensions to relational algebra

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Duplicates → Bag algebra (handle repeated tuples)
- Grouping & Aggregation → SUM, COUNT, AVG, etc.
- Extended Projection → Compute new column values

Duke CS, Fall 2024

63

 International University, VNU-HCMC


Why is RA a good query language?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Simple:
 - A few core operators
 - Easy-to-understand semantics
- Declarative?
 - More declarative than older models (e.g., CODASYL)
 - But combining operators can feel somewhat “procedural”

Duke CS, Fall 2024

64



International University, VNU-HCMC


Why is RA a good query language?

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

- Complete
 - Expressive enough to represent all basic relational queries
 - (Completeness depends on the formal definition used)

Duke CS, Fall 2024

65




International University, VNU-HCMC

Operations of Relational Algebra

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R .	$\sigma_{\langle \text{selection condition} \rangle}(R)$
PROJECT	Produces a new relation with only some of the attributes of R and removes duplicate tuples.	$\pi_{\langle \text{attribute list} \rangle}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\langle \text{join condition} \rangle} R_2$, OR $R_1 \bowtie_{\langle \text{join attributes 1} \rangle} R_2$ $R_1 \bowtie_{\langle \text{join attributes 2} \rangle} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{\langle \text{join condition} \rangle} R_2$, OR $R_1 *_{\langle \text{join attributes 1} \rangle} R_2$, OR $R_1 *_{\langle \text{join attributes 2} \rangle} R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2 ; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2 ; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2 ; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2 .	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $\{X\}$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

66

International University, VNU-HCMC

Assoc. Prof. Nguyen Thi Thuy Loan, PhD

Thank you for your attention!

Duke CS, Fall 202467