HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Lesson 2: Relational Algebra

Learning points

1. Introduction to relational algebra
2. Set operators
3. Relational operators: Projection, Selection, Rename, Join
4. Common extensions

Learning objectives

- Upon completion of this lesson, students will be able to:
  - Understand relational algebra operators
  - Write relational algebraic expressions

Keywords and descriptions

Keyword	Description
Relational data model	Is data representation format as a table of values, each row in the table represents a collection of related data values
Set	Is collection of Object
Operator	Is a special token that represent computations such as union, minus, selection, join, etc
Expression	Is a expression built up from operators and operands

Database Schema

student(**student_id**, first_name, last_name, dob, gender, address, note, *clazz_id*)

clazz(**clazz_id**, name, *lecturer_id*, *monitor_id*)

subject(**subject_id**, name, credit, percentage_final_exam)

enrollment(***student_id, subject_id, semester***, midterm_score, final_score)

lecturer(**lecturer_id**, first_name, last_name, dob, gender, address, email)

teaching(***subject_id, lecturer_id***)

grade(**code**, from_score, to_score)

Database

student

student_id	first_name	last_name	dob	…	clazz_id
20160001	Ngọc An	Bùi	3/18/1987	…	
20160002	Anh	Hoàng	5/20/1987	…	20162101
20160003	Thu Hồng	Trần	6/6/1987	…	20162101
20160004	Minh Anh	Nguyễn	5/20/1987	…	20162101
20170001	Nhật Ánh	Nguyễn	5/15/1988	…	20172201

clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

subject

subject_id	name	credit	percentage_final_exam
IT1110	Tin học đại cương	4	60
IT3080	Mạng máy tính	3	70
IT3090	Cơ sở dữ liệu	3	70
IT4857	Thị giác máy tính	3	60
IT4866	Học máy	2	70

enrollment

student_id	subject_id	semester	midterm_score	final_score
20160001	IT1110	20171	9	8.5
20160001	IT3080	20172	8	
20160001	IT3090	20172	6	9
20160001	IT4857	20172	7.5	9
20160001	IT4866	20172	7	9
20160002	IT3080	20172	9	
20160003	IT1110	20171	7	6
20160004	IT1110	20171	6	5

1. Introduction to relational algebra

- Relational algebra providing a theoretical foundation for relational databases, particularly query languages for relational databases.
- Relational algebra expression is composed of one or several relational algebraic operators:
  - Operator: represent computations
    - Input: one or two relation
    - Output: a relation
  - Unary operator (one input) vs. binary operator (two inputs)

2. Set operators

2.1. Union

2.2. Intersection

2.3. Difference

2.4. Cartesian product

2.1. Union

- Combining the tuples from two *union-compatible relation* inputs (having the same set of attributes).

clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003

∪

clazz 2

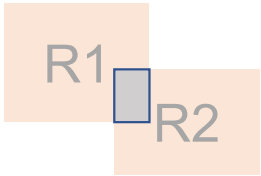
clazz_id	name	lecturer_id	monitor_id
20172201	CNTT2.01-K62	02002	20170001

result

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20172201	CNTT2.01-K62	02002	20170001

2.2. Intersection

- Keeping only common tuples from 2 input union-compatible relations.



clazz

clazz_id	name	lecturer_id	monitor_id
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

∪

clazz 2

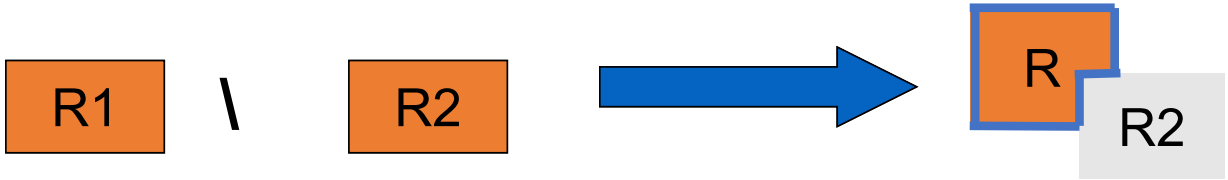
clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001

result

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001

2.3. Difference

- Containing tuples occurred in the first relation but not in the second.



clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

\

clazz 2

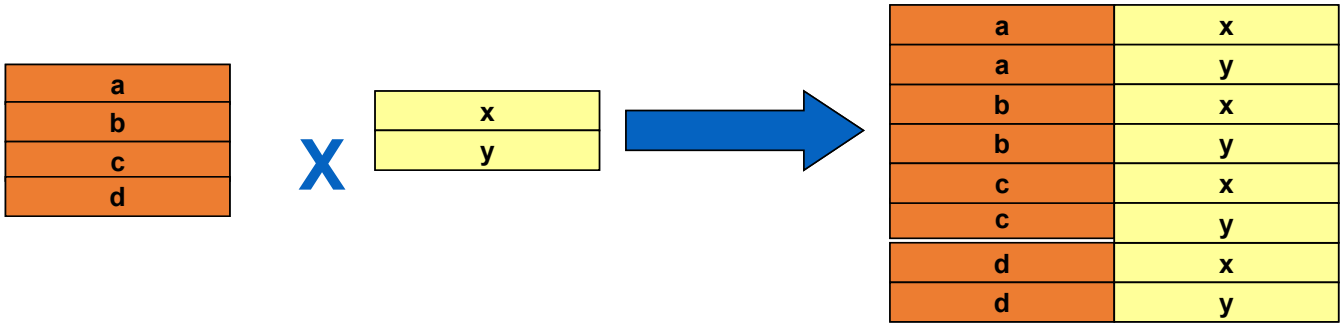
clazz_id	name	lecturer_id	monitor_id
20172202	CNTT2.02-K62		
20162102	CNTT1.02-K61		

result

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20172201	CNTT2.01-K62	02002	20170001

2.4. Cartesian Product

- The concatenation of every tuple of one relation with every tuple of the other relation.



3. Relational algebra operators

3.1. Selection

3.2. Projection

3.3. Rename

3.4. Join

3.5. Division

3.1. Selection

- Choose from R each tuple where the condition holds.

$$\sigma_{<condition>}(R)$$

clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

$$\sigma_{\text{lecturer\_id != null}}(\text{clazz})$$

result

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20172201	CNTT2.01-K62	02002	20170001

3.2. Projection

- Choose some attributes.

$$\prod_{A1, A2, \ldots}(R)$$

clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

$\prod_{\text{clazz\_id, name}}(\text{clazz})$

result

clazz_id	name
20162101	CNTT1.01-K61
20172201	CNTT2.01-K62

3.3. Rename

- Result is identical to R except that the b attribute in all tuples is renamed to an a attribute.

$$\rho_{a|b}(R)$$

clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

$$\rho_{\text{name of class} \mid \text{name}}(\text{clazz})$$

clazz

clazz_id	Name of class	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

3.4. Join

- Combine attributes from 2 tables.

$$R_1 \rhd\lhd_{<join\_condition>} R_2$$

3.4. Join

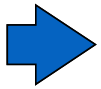
Student ⋈ clazz

student

student_id	first_name	last_name	…	clazz_id
20160001	Ngọc An	Bùi		
20160002	Anh	Hoàng		20162101
20160003	Thu Hồng	Trần		20162101
20160004	Minh Anh	Nguyễn		20162101
20170001	Nhật Ánh	Nguyễn		20172201

clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

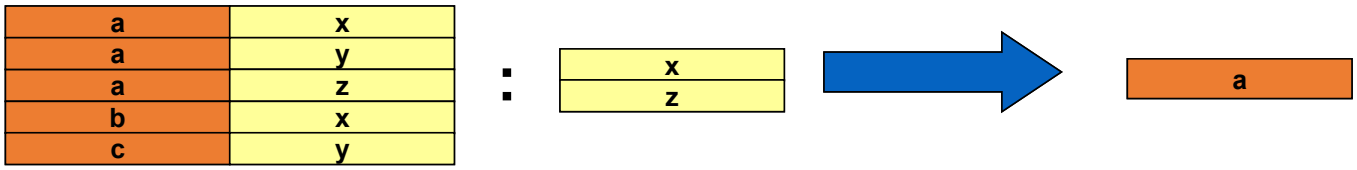


result

student_id	first_name	last_name	name
20160002	Anh	Hoàng	CNTT1.01-K61
20160003	Thu Hồng	Trần	CNTT1.01-K61
20160004	Minh Anh	Nguyễn	CNTT1.01-K61
20170001	Nhật Ánh	Nguyễn	CNTT2.01-K62

3.5. Division

- Divides a dividend relation R_1 or degree m+n by a divisor relation R_2 of degree n, and produces a quotient relation of degree m.



An example

- List student_id who enroll in all subjects.

enrollment

student_id	subject_id	semester	midterm_ score	final_ score
20160001	IT1110	20171	9	8.5
20160001	IT3080	20172	8	
20160001	IT3090	20172	6	9
20160001	IT4857	20172	7.5	9
20160001	IT4866	20172	7	9
20160002	IT3080	20172	9	
20160003	IT1110	20171	7	6
20160004	IT1110	20171	6	5

subject

subject_id	name	credit	percentage_ final_exam
IT1110	Tin học đại cương	4	60
IT3080	Mạng máy tính	3	70
IT3090	Cơ sở dữ liệu	3	70
IT4857	Thị giác máy tính	3	60
IT4866	Học máy	2	70

An example

- List student_id who enroll in all subjects.

enrollment

student_id	subject_id	semester	midterm_ score	final_ score
20160001	IT1110	20171	9	8.5
20160001	IT3080	20172	8	
20160001	IT3090	20172	6	9
20160001	IT4857	20172	7.5	9
20160001	IT4866	20172	7	9
20160002	IT3080	20172	9	
20160003	IT1110	20171	7	6
20160004	IT1110	20171	6	5

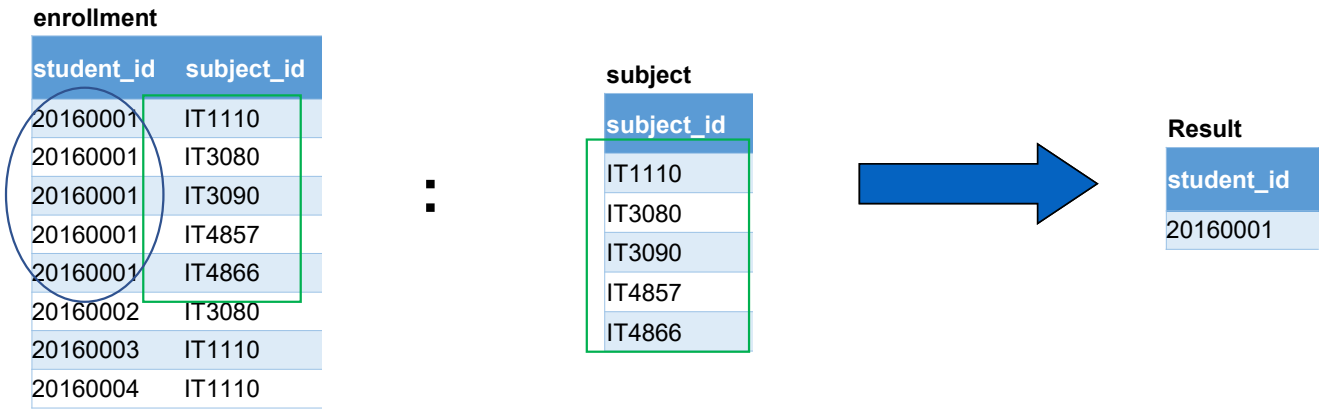
subject

subject_id	name	credit	percentage_ final_exam
IT1110	Tin học đại cương	4	60
IT3080	Mạng máy tính	3	70
IT3090	Cơ sở dữ liệu	3	70
IT4857	Thị giác máy tính	3	60
IT4866	Học máy	2	70

$\Pi_{\text{student\_id, subject\_id}}(\text{enrollement})$ $\Pi_{\text{subject\_id}}(\text{subject})$

An example

- List student_id who enroll in all subjects.



$(\Pi_{\text{student\_id, subject\_id}}(\text{enrollement})) \quad : \quad (\Pi_{\text{subject\_id}}(\text{subject}))$

4. Common extension

4.1. Natural join

4.2. Outer join

4.3. Aggregation

4.1. Natural join

- Special join operation with equal join condition on their common attributes, noted *

student

student_id	first_name	last_name	dob	…	clazz_id
20160001	Ngọc An	Bùi	3/18/1987	…	
20160002	Anh	Hoàng	5/20/1987	…	20162101
20160003	Thu Hồng	Trần	6/6/1987	…	20162101
20160004	Minh Anh	Nguyễn	5/20/1987	…	20162101
20170001	Nhật Ánh	Nguyễn	5/15/1988	…	20172201

clazz

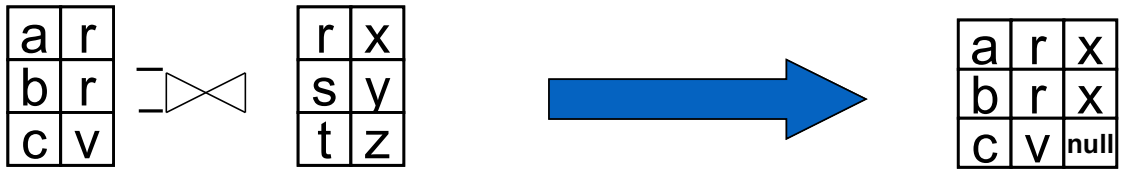
clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

results

student_id	first_name	last_name	dob	…	clazz_id	Name	lecturer_id	Monitor_id
20160002	Anh	Hoàng	5/20/1987	…	20162101	CNTT1.01-K61	02001	20160003
20160003	Thu Hồng	Trần	6/6/1987	…	20162101	CNTT1.01-K61	02001	20160003
20160004	Minh Anh	Nguyễn	5/20/1987	…	20162101	CNTT1.01-K61	02001	20160003
20170001	Nhật Ánh	Nguyễn	5/15/1988	…	20172201	CNTT2.01-K62	02002	20170001

4.2. Outer join

- Left Outer join



- Right Outer join

4.2. Outer join

- **Example of left-outer join:** List all students and class information if any

student

student_id	first_name	last_name	dob	…	clazz_id
20160001	Ngọc An	Bùi	3/18/1987	…	
20160002	Anh	Hoàng	5/20/1987	…	20162101
20160003	Thu Hồng	Trần	6/6/1987	…	20162101
20160004	Minh Anh	Nguyễn	5/20/1987	…	20162101
20170001	Nhật Ánh	Nguyễn	5/15/1988	…	20172201

clazz

clazz_id	name	lecturer_id	monitor_id
20162101	CNTT1.01-K61	02001	20160003
20162102	CNTT1.02-K61		
20172201	CNTT2.01-K62	02002	20170001
20172202	CNTT2.02-K62		

results

student_id	first_name	last_name	dob	…	clazz_id	Name	lecturer_id	Monitor_id
20160001	Ngọc An	Bùi	3/18/1987					
20160002	Anh	Hoàng	5/20/1987	…	20162101	CNTT1.01-K61	02001	20160003
20160003	Thu Hồng	Trần	6/6/1987	…	20162101	CNTT1.01-K61	02001	20160003
20160004	Minh Anh	Nguyễn	5/20/1987	…	20162101	CNTT1.01-K61	02001	20160003
20170001	Nhật Ánh	Nguyễn	5/15/1988	…	20172201	CNTT2.01-K62	02002	20170001

4.3. Aggregation

- Aggregation

$$G_1, G_2, \ldots G_n \, \mathcal{G} \, F_1(A_1), F_2(A_2), \ldots F_n(A_n) \, (\mathcal{R})$$

- $G_1, G_2, \ldots G_n$ is a list of attributes on which to group $F_1 (A_1), F_2 (A_2), \ldots F_n (A_n)$ is a list of aggregation function on attribute $A_1, A_2, \ldots A_n$

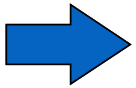
4.3. Aggregation

- Example of Aggregation

$$G_{student\_id}\ Gcount(subject\_id)\ (enrollment\ )$$

enrollment

student_id	subject_id	semester	midterm_ score	final_ score
20160001	IT1110	20171	9	8.5
20160001	IT3080	20172	8	
20160001	IT3090	20172	6	9
20160001	IT4857	20172	7.5	9
20160001	IT4866	20172	7	9
20160002	IT3080	20172	9	
20160003	IT1110	20171	7	6
20160004	IT1110	20171	6	5



results

student_id	count(subject_id)
20160001	5
20160002	1
20160003	1
20160004	1

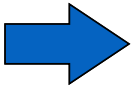
4.3. Aggregation

- Example of Aggregation

$\mathcal{G}$ count(student_id) (student)

student

student_id	first_name	last_name	dob	…	clazz_id
20160001	Ngọc An	Bùi	3/18/1987	…	
20160002	Anh	Hoàng	5/20/1987	…	20162101
20160003	Thu Hồng	Trần	6/6/1987	…	20162101
20160004	Minh Anh	Nguyễn	5/20/1987	…	20162101
20170001	Nhật Ánh	Nguyễn	5/15/1988	…	20172201



value not a relation

5

Summary

- Introduction to relational algrebra
  - Procedural langue
- Set operators
  - Union, intersection, difference
- Relational operators
  - Projection, Selection, Rename, Join
- Common extensions
  - Natural join, Outer join, Aggregation

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict