UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Slide 14

Relational Algebra Part 1

CSF2600700 - BASIS DATA

SEMESTER GENAP 2019/2020

Speaker: Yova Ruldeviyani (yova@cs.ui.ac.id)

References

- Elmasri 7th edition: Chapter 8
- Stanford Database (CS145) Lecture 16

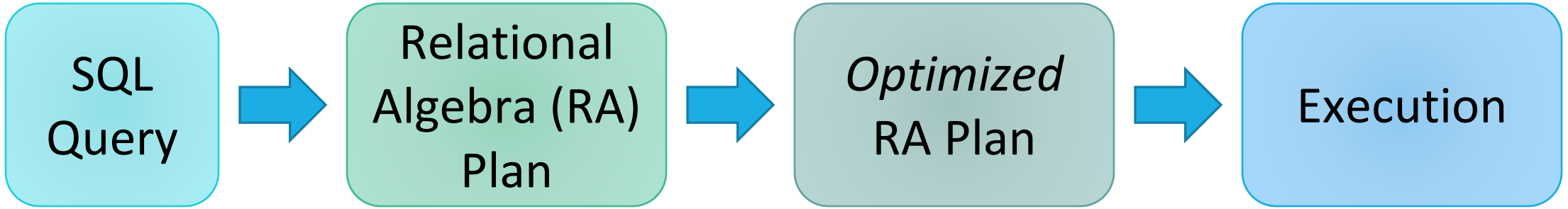
Introduction

- Relational Algebra (RA): The basic set of operations for relational model

- Important for:
  - Provides a formal foundation for relational model operations
  - Used as basis for implementing and optimizing queries by RDBMS
  - Incorporated into the SQL for RDBMS

RDBMS Architecture

How does a SQL engine work ?



Declarative query (from user)

Translate to relational algebra expression

Find logically equivalent- but more efficient- RA expression

Execute each operator of the optimized plan!

Relational Algebra Operations

- Set Operations
  - UNION ($\cup$)
  - INTERSECTION ($\cap$)
  - SET DIFFERENCE ($-$)
  - CARTESIAN/CROSS PRODUCT ($\times$)

  } Binary Operations

- Operations for specific Relational DB
  - SELECT (σ)
  - PROJECT (π)
  - RENAME (ρ)

  } Unary Operations

  - JOIN ($\bowtie$)

  } Binary Operations

- Other
  - DIVISION ($\div$)

Keep in mind: RA operates on sets!

- RDBMSs use *multisets*, however in relational algebra formalism we will consider **sets!**

- Also: we will consider the ***named perspective***, where every attribute must have a unique name
  - attribute order does not matter…

Now on to the basic RA operators…

Unary Operations

Operation:
Select
Project
Rename

SELECT Operations

Students(sid,sname,gpa)

- Returns all tuples which satisfy a condition
- Notation: $\sigma_{<condition>}(R)$
- Examples:
  - $\sigma_{Salary > 40000}$ (Employee)
  - $\sigma_{name = \text{“Smith”}}$ (Employee)
- The $< condition >$ can be:
  - $< attr\ name >< operator >< constant\ value >$
    or
  - $< attr\ name >< operator >< attr\ name >$
- Operator is one of =, <, ≤, >, ≥, <>

SQL:

```
SELECT *
FROM Students
WHERE gpa > 3.5;
```

RA:

$$\sigma_{gpa > 3.5}(Students)$$

SELECT Operations

Another example:

SSN	Name	Salary
1234545	John	200000
5423341	Smith	600000
4352342	Fred	500000

$\sigma_{Salary > 40000}$ (Employee)

SSN	Name	Salary
5423341	Smith	600000
4352342	Fred	500000

SELECT Operation

- Clauses can be connected by the standard Boolean operators **AND**, **OR**, and **NOT** to form a general selection condition.
  - For example, to select the tuples for all employees who either work in department 4 and make over $25,000 per year, or work in department 5 and make over $30,000, we can specify the following SELECT operation:

$$\sigma_{(Dno=4\ AND\ Salary>25000)\ OR\ (Dno=5\ AND\ Salary>30000)}(EMPLOYEE)$$

SELECT Operation

- Based on that nature, a sequence of SELECTs can be applied in any order and any **cascade** of SELECT operations can then be combine into a single SELECT operations with conjunctive (**AND**) condition.

$$\sigma_{<cond1>}(\sigma_{<cond2>}(\ldots(\sigma_{<condn>}(R))\ldots)) =$$

$$\sigma_{<cond1>\ \textbf{AND}\ <cond2>\ \textbf{AND}\ \ldots\ \textbf{AND}<condn>}(R)$$

- The nature of SELECT Operation is **commutative**:

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

SELECT Operation

- Some domains allow additional types of comparison operators, such as in domain of character string may allow the comparison operator SUBSTRING_OF.

PROJECT Operation

Students(sid,sname,gpa)

- Select certain columns and eliminates the other columns, then removes duplicates
- Notation: $\pi_{<attribute\ list>}(R)$
- Example:
  - $\pi_{\text{sname, gpa}}$ (Students)
  - $\pi_{\text{Name, Salary}}$ (Employee)

SQL:

```
SELECT DISTINCT
  sname,
  gpa
FROM Students;
```

RA:

$$\Pi_{sname,gpa}(Students)$$

PROJECT Operation

Another example:

SSN	Name	Salary
1234545	John	200000
5423341	John	600000
4352342	John	200000

$\pi_{Name,Salary}$ (Employee)

Name	Salary
John	200000
John	600000

Note that RA Operators are Compositional!

Students(sid,sname,gpa)

```
SELECT DISTINCT
  sname,
  gpa
FROM Students
WHERE gpa > 3.5;
```

$$\Pi_{sname,gpa}(\sigma_{gpa>3.5}(Students))$$

$$\sigma_{gpa>3.5}(\Pi_{sname,gpa}(Students))$$

Are these logically equivalent?

How do we represent this query in RA?

Logical Equivalence of RA Plans

- Given relations R(A,B) and S(B,C):
  - Here, projection & selection commute:
    - $\sigma_{A=5}(\Pi_A(R)) = \Pi_A(\sigma_{A=5}(R))$
  - What about here?
    - $\sigma_{A=5}(\Pi_B(R)) \stackrel{?}{=} \Pi_B(\sigma_{A=5}(R))$

RENAME Operation

Students(sid,sname,gpa)

- RENAME: Changes the schema, not the instance
- Notation:
  - $\rho_{S(B_1,\ldots B_n)}(R)$, or $\rho_S(R)$, or $\rho_{(B_1,\ldots B_n)}(R)$
    - S: the new relation name
    - $B_1, \ldots B_n$: the new attribute names
- **Note: this is shorthand for the proper form (since names, not order matters!):**
  - $\rho_{A1\rightarrow B1,\ldots,An\rightarrow Bn}(R)$

SQL:

```
SELECT
  sid AS studId,
  sname AS name,
  gpa AS gradePtAvg
FROM Students;
```

RA:

$P_{(studId,\ name,\ gradePTAvg)}\ (STUDENTS)$

We care about this operator *because* we are working in a *named perspective*

RENAME Operation

Another example:

$$\rho_{studId,name,gradePtAvg}(Students)$$

Students

sid	sname	gpa
001	John	3.4
002	Bob	1.3

Students

studId	name	gradePtAvg
001	John	3.4
002	Bob	1.3

$$\rho_{Stud-Rel(A,B,C)}(Students)$$

Students

sid	sname	gpa
001	John	3.4
002	Bob	1.3

Stud-Rel

A	B	C
001	John	3.4
002	Bob	1.3

(Back to:) Note that RA Operators are Compositional!

Students(sid,sname,gpa)

```
SELECT DISTINCT
  sname,
  gpa
FROM Students
WHERE gpa > 3.5;
```

$$\Pi_{sname,gpa}(\sigma_{gpa>3.5}(Students))$$

$$TEMP \leftarrow \sigma_{gpa>3.5}(Students)$$
$$R(Name, GPA) \leftarrow \pi_{sname,gpa}(TEMP)$$

$$\sigma_{gpa>3.5}(\Pi_{sname,gpa}(Students))$$

$$TEMP \leftarrow \pi_{sname,gpa}(Students)$$
$$R(Name, GPA) \leftarrow \sigma_{gpa>3.5}(TEMP)$$

We can also break down a sequences/compositional RA by specifying intermediate result relations and renaming the relation and its attributes.

Generalized PROJECT Operation

- Also allows functions of attributes to be included in the projection list.

As an example, consider the relation

EMPLOYEE (Ssn, Salary, Deduction, Years_service)

A report may be required to show

Net Salary = Salary – Deduction,
Bonus = 2000 * Years_service, and
Tax = 0.25 * Salary.

Then a generalized projection combined with renaming may be used as follows:

$$\text{REPORT} \leftarrow \rho_{(\text{Ssn, Net\_salary, Bonus, Tax})}(\pi_{\text{Ssn, Salary} - \text{Deduction, } 2000 * \text{Years\_service, } 0.25 * \text{Salary}}(\text{EMPLOYEE})).$$

Binary Operations

- Two relations $R_1(A_1, A_2, \ldots A_n)$ and $R_2(B_1, B_2, \ldots B_n)$ are said to be **union compatible** if they have the same degree n and if $dom(A_i) = dom(B_i)$ for $1 \leq i \leq n$

- Need union compatibility:
  - UNION, INTERSECTION, SET DIFFERENCE

- Do not need union compatibility:
  - CARTESIAN PRODUCT, JOIN

Example:

STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

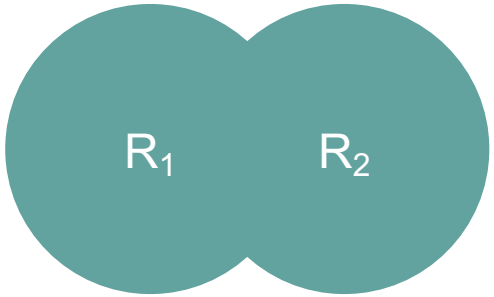
INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

Both have two attributes with the same domain

UNION Operation

- Includes all tuples that are either in R_1 or R_2 or both without duplicate tuples.
- Notation: $R_1 \cup R_2$



- Commutative, Associative

Example: Student ∪ Instructor

STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

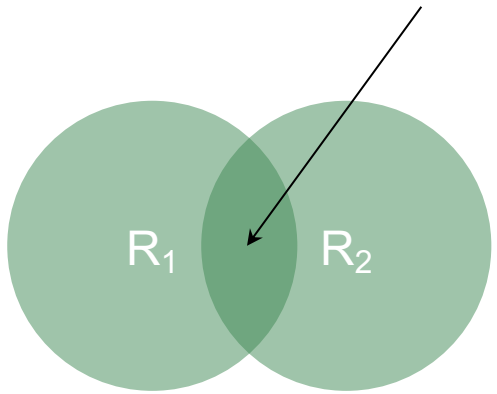
INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

INTERSECTION Operation

- Includes all tuples that are in both R_1 and R_2
- Notation: $R_1 \cap R_2$



- $R_1 \cap R_2 = R_1 - (R_1 - R_2)$
- Commutative, Associative

Example: Student ∩ Instructor

STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

Fn	Ln
Susan	Yao
Ramesh	Shah

DIFFERENCE/MINUS/EXCEPT Operation

- Includes all tuples that are in R but not in S
- Notation: $R - S$

$$R - S =$$
$$\big((R \cup S) - (R - S)\big) - (S - R)$$

Example:

STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

Student – Instructor

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

Instructor – Student

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

CARTESIAN/CROSS PRODUCT

- Combining each tuple in R_1 with each tuple in R_2
- Notation: $R_1 \times R_2$
- Example:
  - Employee × Dependents
  - People × Students
- Rare in practice; mainly used to express joins
- Useful when followed by a selection condition (see example in the next page)

People(ssn,pname,address)
Students(sid,sname,gpa)

SQL:

```
SELECT *
FROM People, Students
```

RA: $People \times Students$

Another example:

```
SELECT *
FROM People, Students
WHERE pname = sname;
```

People

ssn	pname	address
1234545	John	216 Rosse
5423341	Bob	217 Rosse

×

Students

sid	sname	gpa
001	John	3.4
002	Bob	1.3

$PeopStud \leftarrow People \times Students$

$Result \leftarrow \sigma_{pname=sname}(PeopStud)$

PeopStud

ssn	pname	address	sid	sname	gpa
1234545	John	216 Rosse	001	John	3.4
5423341	Bob	217 Rosse	001	John	3.4
1234545	John	216 Rosse	002	Bob	1.3
5423341	Bob	216 Rosse	002	Bob	1.3

Also can be represented as this single line RA:

$Result \leftarrow \sigma_{pname=sname}(People \times Students)$

Result

ssn	pname	address	sid	sname	gpa
1234545	John	216 Rosse	001	John	3.4
5423341	Bob	216 Rosse	002	Bob	1.3

(Theta) JOIN

- **To combine related tuples** from two relations into single tuples. In database **JOIN** operation is important because it allows us to process relationship among relations.
- Notation: $R_1 \bowtie_{<\text{join condition}>} R_2$

  <join condition> = <condition> AND <condition> AND … AND <condition>
- Each <condition> in the form $A_i \theta B_j$ and θ is one of =, <, <=, >, >=, <>

  A_i is attribute in R_1, Bj is attribute in R_2.
- Previous CROSS PRODUCT's example can be replaced by this JOIN operation.

  $\text{Result} = (\text{PEOPLE}) \bowtie_{\text{Pname=Sname}} (\text{STUDENTS})$
- This JOIN example is called EQUIJOIN = a JOIN operation uses only **equality** comparison operator

(Natural) JOIN

- A theta JOIN + PROJECT operation.
  - Join two relations, then use projection to select desired columns
- An EQUIJOIN with the removal of superfluous attribute.
  - See example in the next page.

- Notation: $R_1 * R_2$ (or $R_1 \bowtie R_2$ without join condition)

- The attributes to be joined must have the same name in both relations, otherwise we have to rename one of them first.

(Natural) JOIN

People

ssn	pname	address
1234545	John	216 Rosse
5423341	Bob	217 Rosse

Students

sid	sname	gpa
001	John	3.4
002	Bob	1.3

- Example:
  - EQUIJOIN: $People \bowtie_{pname=sname} Students$

ssn	pname	address	sid	sname	gpa
1234545	John	216 Rosse	001	John	3.4
5423341	Bob	216 Rosse	002	Bob	1.3

Superfluous Attribute

  - Natural Join: $People * \rho_{(sid,pname,gpa)} Students$

ssn	pname	address	sid	gpa
1234545	John	216 Rosse	001	3.4
5423341	Bob	216 Rosse	002	1.3

If the two joined relations already have the same attribute name as join attribute, then renaming is not necessary.

(Semi) JOIN ⋉

Students(sid,sname,gpa)
People(ssn,pname,address)

- Notation: R ⋉< join condition > S
- $R \ltimes_{< join\ condition >} S = \Pi_{A1, A2, \ldots An} (R \bowtie S)$
  Where A1, A2, … An are attributes in R
- Example:
  People $\ltimes_{pname=sname}$ *Students*

ssn	pname	address	sid	gpa
1234545	John	216 Rosse	001	3.4
5423341	Bob	216 Rosse	002	1.3

SQL:

```
SELECT DISTINCT
  ssn,pname,address
FROM
  People, Students
WHERE
  pname = sname;
```

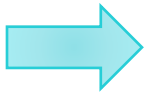
RA:

People $\ltimes_{pname=sname}$ *Students*

Example: Converting SFW Query -> RA

Students(sid,sname,gpa)
People(ssn,sname,address)

```
SELECT DISTINCT
  gpa,
  address
FROM Students S,
     People P
WHERE gpa > 3.5 AND
   sname = pname;
```



$$\Pi_{gpa,address}(\sigma_{gpa>3.5}((\rho_S(\text{STUDENTS})) \bowtie_{Sname = Pname} (\rho_P(\text{PEOPLE}))))$$

How do we represent this query in RA?

DIVISION

- Notation: R ÷ S
- is used when we wish to express queries with “all”.
- For example:
  - “Which persons have a loyal customer's card at ALL the clothing b outiques in town X?”
  - “Which persons have a bank account at ALL the banks in the country?”
  - “Which students are registered on ALL the courses given by Soni”
  - “Which boys are registered on those courses that are taken by ALL the girls?”

DIVISION

Division $T \leftarrow R \div S$ can be expressed as a sequence of these operations:

- Example:
  - $T \leftarrow R \div S$

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

1 $T_1 \leftarrow \pi_B(R)$

T_1

B
b1
b2
b3
b4

2 $T_2 \leftarrow \pi_B((S \times T_1) - R)$

$S \times T_1$

A	B
a1	b1
a2	b1
a3	b1
a1	b2
a2	b2
a3	b2
a1	b3
a2	b3
a3	b3
a1	b4
a2	b4
a3	b4

$-$ R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

$=$ T_2

B
b2
b3

3 $T \leftarrow T_1 - T_2$

T

B
b1
b4

DIVISION

- Example:
  - *Retrieve the names of employees who work on all projects that 'John Smith' works on*

$$SMITH \leftarrow \sigma_{Fname='John' \text{ AND } Lname='Smith'}(EMPLOYEE)$$
$$SMITH\_PNOS \leftarrow \pi_{Pno}(WORKS\_ON \bowtie_{Essn=Ssn} SMITH)$$

$$SSN\_PNOS \leftarrow \pi_{Essn,\ Pno}(WORKS\_ON)$$

$$SSNS(Ssn) \leftarrow SSN\_PNOS \div SMITH\_PNOS$$
$$RESULT \leftarrow \pi_{Fname,\ Lname}(SSNS * EMPLOYEE)$$

SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

SMITH_PNOS

Pno
1
2

SSNS

Ssn
123456789
453453453

Exercise

- Retrieve the name and address of all employees who work for the 'Research' department.
- Retrieve the names of all employees in department 5 who work more than 10 hours per week on the ProductX project.
- List the names of all employees who have a dependent with the same first name as themselves.
- Find the names of all employees who are directly supervised by 'Franklin Wong'.
- Find the names of employees who work on *all* the projects controlled by department number 5.

Exercise (Answer)

- $\pi_{Fname,Address}\left(Employee \bowtie_{Dno=Dnumber} \left(\sigma_{Dname='Research'}(Department)\right)\right)$
- $\pi_{Fname,} \left(\sigma_{Hours>10}\left(Works\_On \quad \bowtie_{Essn=Ssn} \left(\pi_{Ssn,Fname}\left(\sigma_{Dno=5}(Employee)\right)\right)\right) \bowtie_{Pno=P}\right.$
- $\pi_{Fname}\left(\sigma_{Fname=Name}\left(Employee \bowtie_{Ssn=Essn} (Dependent)\right)\right)$
- $\pi_{Fname}\left(Employee \bowtie_{Super_{Ssn}=Ssn} \left(\pi_{Ssn}\left(\sigma_{Fname='Franklin' \text{ and } Lname='Wong'}(Employee)\right)\right)\right)$
- $\pi_{Fname}\left(Employee \bowtie_{Ssn=ESsn} \left(\left(\pi_{Essn,Pno}(Works\_On)\right) \div \rho_{Pno}\left(\pi_{Pnumber}\left(\sigma_{Dno=5}(Project)\right)\right)\right)\right)$

Exercise for Discussion

1. Diberikan schema R(A, B, C, D) dan S(A, C, E). Berikan hasil keluaran skema dari expression $R \bowtie S$.
2. Diberikan schema R(A, B, C) dan S(D, E). Berikan hasil keluaran skema dari expression $R \bowtie S$.
3. Diberikan schema R(A, B) dan S(C, D). Berikan hasil keluaran skema dari expression $R \ltimes_{B=C} S$.
4. Tampilkan fname, lname semua employee yang menjadi supervisor.
5. Tampilkan fname, lname semua employee yang menjadi supervisor, namun tidak menjabat sebagai manager.

Exercise for Discussion

6. Tampilkan fname dan lname manager dari department yang mengontrol project dan fname dan lname supervisor dimana department tempat supervisor bekerja mengontrol project.
7. Tampilkan pname project yang dikerjakan oleh semua employee.
8. Tampilkan fname dan lname dari employee yang mengerjakan semua project yang dikerjakan oleh supervisor dari employee tersebut.