CSE2DBF – CSE4DBF

Data Manipulation using

Relational Algebra

Reading:

Elmasri and Navathe, "Fundamentals of Database Systems, Chapters 1 & 2", Pearson, 2016.

Ebook: *https://ebookcentral-proquest-com.ez.library.latrobe.edu.au/lib/latrobe/detail.action?docID=5573709*

Relational Model

In lecture 2 we saw that there are 3 major aspects of a relational data model

a) Data Structure (structural part)

Consists of n-ary two dimensional (attributes, tuples) tables/relations whose attributes' domains are of atomic types.

b) Data Integrity (integrity part)

Consists of two general integrity rules, namely entity integrity and referential integrity.

c) Data Manipulation (manipulative part):

Consists of a set of algebraic operators for data manipulations.

C. Data Manipulation

(using relational database operators based on relational algebra)

Data Manipulation using Relational Algebra

An Example: “Student” Database

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Takes

SID	SNO
1108	21
1108	23
8507	23
8507	29

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
18	Algebra	Maths

Enrol

SID	Course
3936	101
1108	113
8507	101

Course

No	Name	Dept
113	BCS	CSCE
101	MCS	CSCE

- Students have names, ids, and suburbs, enrol in courses, and take subjects.
- Subjects have names and departments.
- Courses are offered by departments.

Data Manipulation using Relational Algebra

- **Which students live in Bundoora?**
  - tuples from Student, then Names from tuples.
- **Which students are doing course 113?**
  - SIDs from Enrol, then match to tuples from Student.

- **Elements of querying:**
  - From a relation choose those tuples that have a certain property.
  - Choose some attributes from a set of tuples.
  - Match one set of tuples to another.

Data Manipulation using Relational Algebra

- **Relational algebra** is a suite of operations on relations. A sequence of relational algebra operations forms a relational algebra expression (algebraic expression) that also results in a relation.

- Each **operation** takes a relation (or two relations) as input and produces a relation as output. (In arithmetic, operations take numbers and produce numbers. Relational algebra is arithmetic for relations.)

- Note that the term '**relation**' is known as '**table**' in relational database environment or '**set**' in a pure relational algebra environment.

Data Manipulation using Relational Algebra

- For example,

  "Create a relation by choosing from Student the tuples where the Suburb is 'Bundoora' "

  is an operation that takes one relation and produces another.

- Relational algebra is the basis of the implementation of most current database systems.

Data Manipulation using Relational Algebra

Relational Algebra Operations

1. PROJECTION
2. SELECTION
3. UNION
4. INTERSECTION
5. DIFFERENCE
6. PRODUCT
7. JOIN
8. OUTER JOIN

Projection

Projection: an operation that selects specified attributes from a relation.

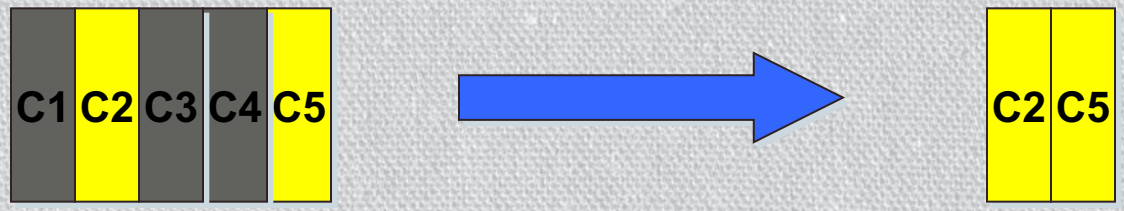
Example query is *find the names of all students*.

In English: choose from each Student tuple the value for attribute ‘Name’.
In relational algebra:

$$\pi_{\text{Name}}(\text{Student})$$

The symbol π (pi) represents projection or choose attributes.

In the general case, projection is $\pi_{\text{attributes}}(R)$ where R is a relation and the attributes are some of the attributes of R.



Relational Algebra : $\pi_{\text{<attribute list>}}(R)$

Projection *removes any duplicate tuples* of the selected attributes.

Projection

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
28	Algebra	CSCE
18	Algebra	Maths

π Name,Dept (Subject) produces Output

Name	Dept
Systems	CSCE
Database	CSCE
VB	CSCE
Algebra	CSCE
Algebra	Maths

π Name (Subject) produces Output

Name
Systems
Database
VB
Algebra

Selection

Selection: Where the project operator takes a vertical subset (columns) of a relation (R), the selection takes a horizontal subset (rows).

The selection operation σ (sigma) means: choose from R each tuple where the condition holds.

Relational Algebra : σ <selection condition> **(R)**

Selection

Subject

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE
18	Algebra	Maths

The query $\sigma_{Dept="CSCE"}$ (Subject) produces Output

No	Name	Dept
21	Systems	CSCE
23	Database	CSCE
29	VB	CSCE

Selection

Example: Which students (name) live in Bundoora?

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

π Name (σ Suburb="Bundoora" (Student))

Output

Name
Glen
Norman

Selection

With intermediate or temporary relations:

R1 ← σ Suburb="Bundoora" (Student)

R1

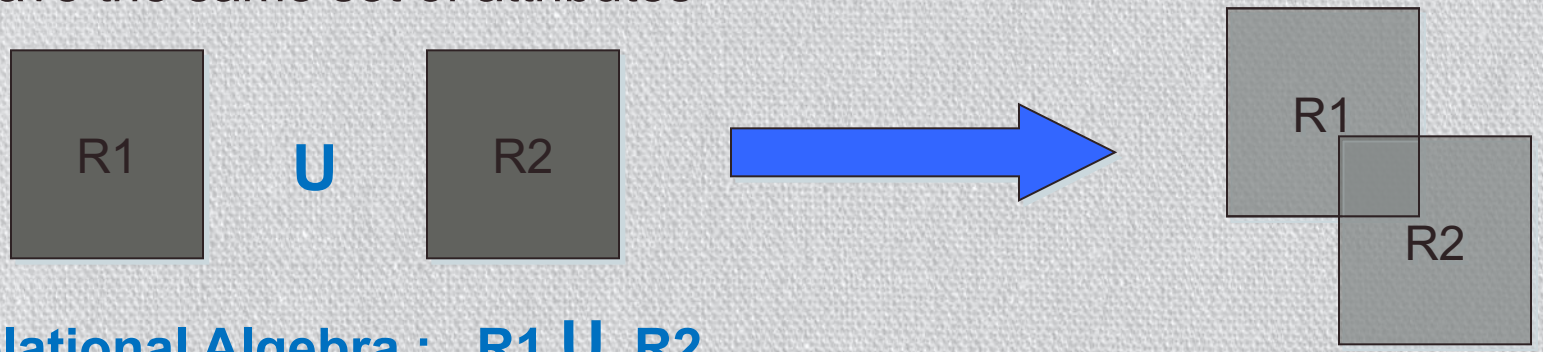
Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

R2 ← π Name (R1)

Name
Glen
Norman

Union

UNION: The union of two relations is formed by combining the tuples from one relation (R1) with those of a second relation (R2) to produce a third. R1 and R2 have to be union-compatible, that is, they have the same set of attributes



Relational Algebra : R1 U R2

EXAMPLE

SubjectCode
CSE21DB
CSE21SDT

UNION

SubjectCode
CSE30PRJ
CSE31DB
CSE42ADB
CSE42SPM

SubjectCode
CSE21DB
CSE21SDT
CSE30PRJ
CSE31DB
CSE42ADB
CSE42SPM

Union

If two relations have the same set of attributes (that is, if they are about the same kinds of information), they can be added together.
Example: What are the SIDs of students doing subject 21 **or** subject 29?

$R1 \leftarrow \pi_{SID}(\sigma_{SNo=21}(Takes))$

SID
1108

$R2 \leftarrow \pi_{SID}(\sigma_{SNo=29}(Takes))$

SID
8507

The union of R1 and R2 is $R3 \leftarrow R1 \cup R2$

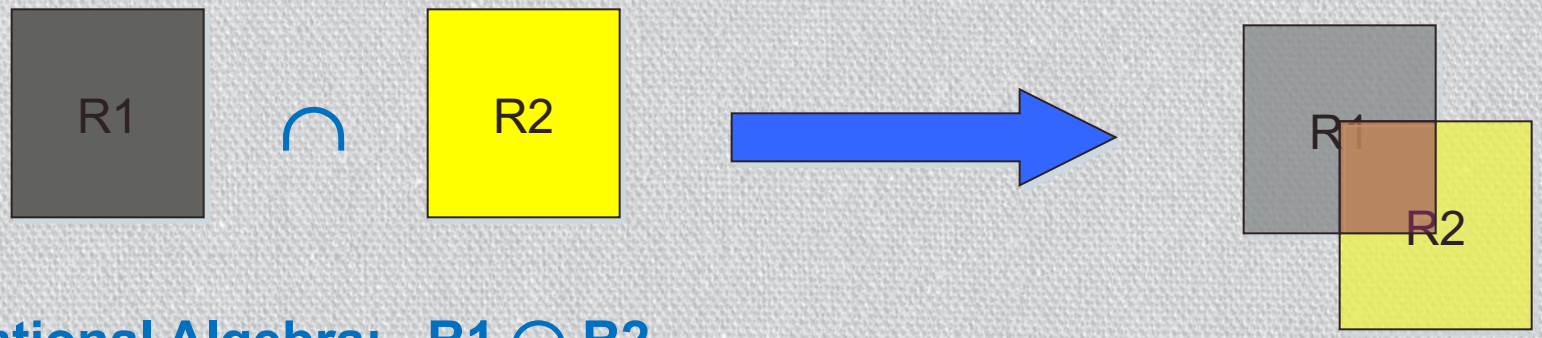
SID
1108
8507

Note: how do we rewrite R3 into one single algebraic expression?

$$R3 \leftarrow \pi_{SID}(\sigma_{SNo=21}(Takes)) \cup \pi_{SID}(\sigma_{SNo=29}(Takes))$$

Intersection

Intersection: The intersection of two relations (R1 and R2) is a third relation containing **common** tuples. Both R1 and R2 must be union-compatible.



Relational Algebra: R1 ∩ R2

EXAMPLE

SubjectCode
CSE21DB
CSE21SDT
CSE30PRJ

INTERSECT

SubjectCode
CSE30PRJ
CSE31DB
CSE21DB
CSE42ADB

SubjectCode
CSE21DB
CSE30PRJ

Intersection

If two relations have the same set of attributes, they can be compared to see which rows they have in common.

Example: Show IDs of students doing both subject 21 **and** subject 23?

$R1 \leftarrow \pi_{SID}(\sigma_{SNo=21}(Takes))$

SID
1108

$R2 \leftarrow \pi_{SID}(\sigma_{SNo=23}(Takes))$

SID
1108
8507

The intersection of R1 and R2 is $R3 \leftarrow R1 \cap R2$

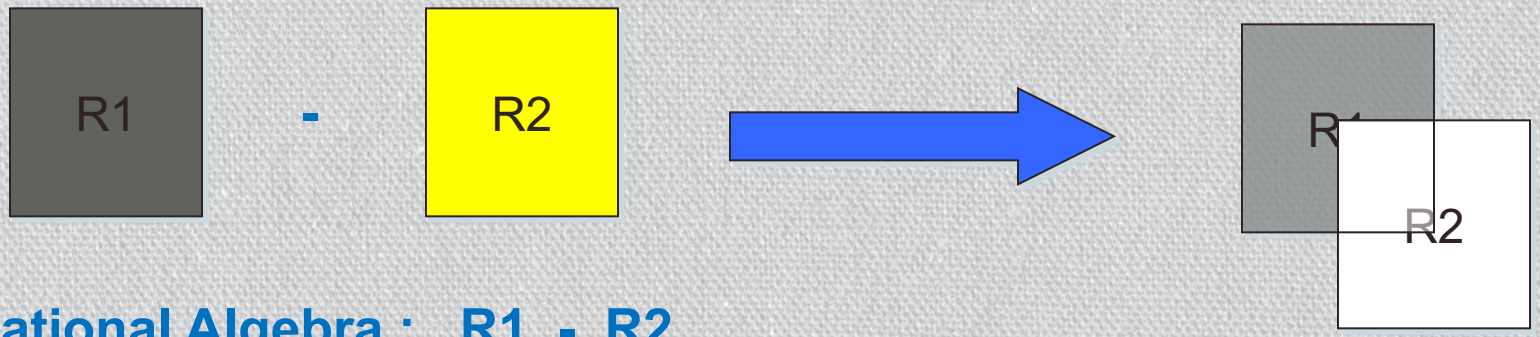
SID
1108

Note: how do we rewrite R3 into one single algebraic expression?

$$R3 \leftarrow \pi_{SID}(\sigma_{SNo=21}(Takes)) \cap \pi_{SID}(\sigma_{SNo=23}(Takes))$$

Difference

Difference: The difference of two relations (R1 and R2) is a third relation containing tuples which occur in the first relation but not in the second. Both R1 and R2 must be union-compatible.



Relational Algebra : R1 - R2

EXAMPLE

SubjectCode
CSE21DB
CSE21SDT
CSE30PRJ

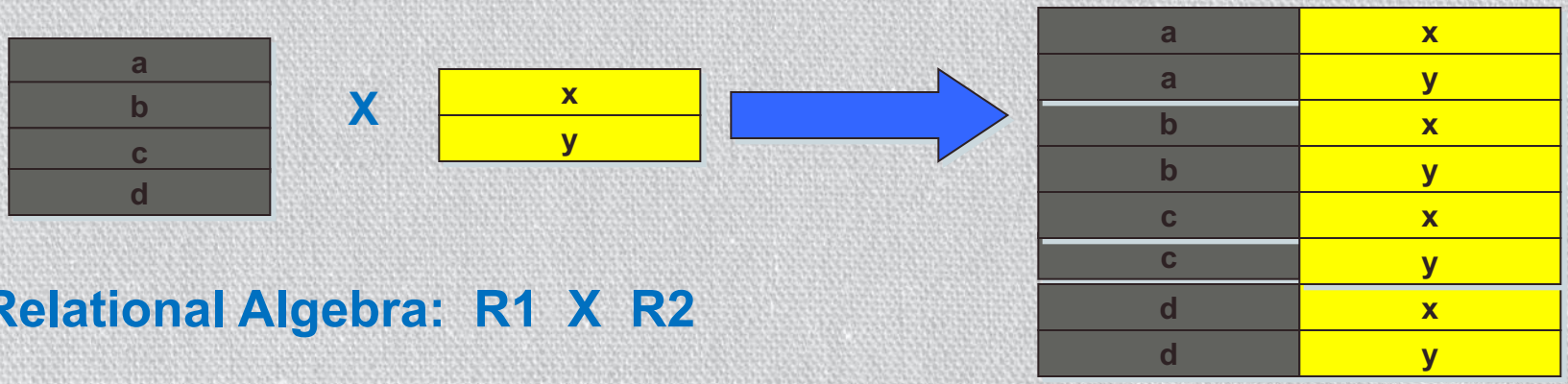
DIFFERENCE

SubjectCode
CSE30PRJ
CSE31DB
CSE21ADC
CSE42ADB

SubjectCode
CSE21DB
CSE21SDT

Cartesian Product

Product: The product of two relations (R1 and R2) is the concatenation of every tuple of one relation with every tuple of the other relation.



a	x
a	y
b	x
b	y
c	x
c	y
d	x
d	y

Relational Algebra: R1 X R2

The relations on which a Product operation is applied **do not have to be union compatible.**

Cartesian Product

An example of Cartesian Product: **Takes X Student**

(The relation TAKES has been changed for this example.)

Takes

SID	SNO
1108	21
8507	23

X

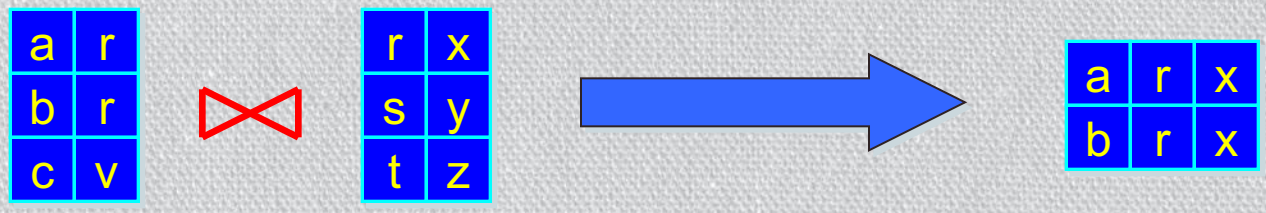
Student

ID	Name
1108	Robert
3936	Glen
8507	Norman
8452	Mary

SID	SNO	ID	Name
1108	21	1108	Robert
1108	21	3936	Glen
1108	21	8507	Norman
1108	21	8452	Mary
8507	23	1108	Robert
8507	23	3936	Glen
8507	23	8507	Norman
8507	23	8452	Mary

Join

Join: The join operation is the combination of the product, selection and projection operations. It is used to combine related tuples from two relations into single “longer” tuples by matching attribute values.



Relational Algebra : R1 ⋈ **<join condition> R2**

Join

Enrol

SID	Course
3936	101
1108	113
8507	101

Student

Id	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Example: Which students (name) are doing course 113?

1. R1 ← σ Course="113" (Enrol)

SID	Course
1108	113

2. R2 ← R1 ⋈ SID=ID Student

SID	Course	Name	Suburb
1108	113	Robert	Kew

3. R3 ← π Name (R2)

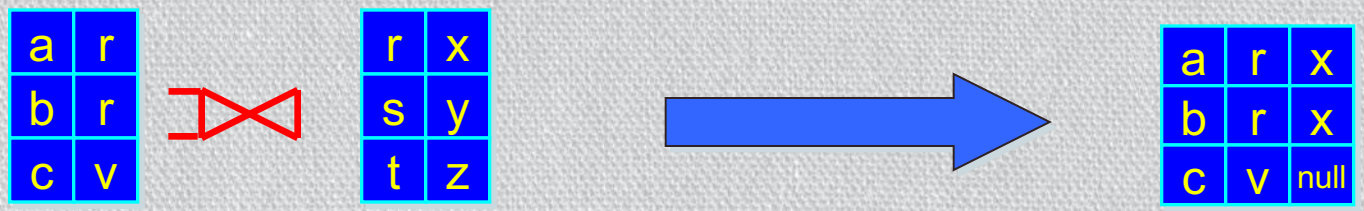
Name
Robert

The whole algebraic expression for the above example can be rewritten as: **R3 ← π Name (σ Course="113" (Enrol) ⋈ SID=ID Student)**

Join: Outer Join

Outer Join: While a (natural) join operation only gives matching tuples that satisfy the join condition, the outer join operation will produce both matched and unmatched tuples from either one or both of the participating relations in the operation.

Example: **LEFT OUTER JOIN** (keeping unmatched tuples from the left hand side relation)



Relational Algebra : R1 ⟕ **<join condition> R2**

Note that we can also use **RIGHT OUTER JOIN**, where we can keep unmatched tuples from right hand side relation, and **FULL OUTER JOIN** where we can keep unmatched tuples from both sides.

Join: Outer Join

Example: List all student IDs along with their enrolled course codes, make sure to include also those students who haven't enrolled in any course.

Without going through the steps again, we can write the final algebraic expression for the above query as follows:

$$R \leftarrow \pi_{ID,\ Course}\ (\ (Student) \bowtie_{ID=SID} Enrol)$$

Student

ID	Name	Suburb
1108	Robert	Kew
3936	Glen	Bundoora
8507	Norman	Bundoora
8452	Mary	Balwyn

Enrol

SID	Course
3936	101
1108	113
8507	101

R

ID	Course
1108	113
3936	101
8507	101
8452	null

Next Lecture

Relational Database Language - SQL

Reading:

Elmasri and Navathe, "Fundamentals of Database Systems, Chapters 1 & 2", Pearson, 2016.

Ebook: *https://ebookcentral-proquest-com.ez.library.latrobe.edu.au/lib/latrobe/detail.action?docID=5573709*