CSCI4333 Database Design & Implement

Lecture 13 – Relational Algebra 2

Instructor: Dr. Yifeng Gao

Relational Algebra

- Five basic operators
  - **selection** $\sigma_{name='Lee'}(professor)$
  - **projection** $\pi_{name}(professor)$
  - **union** $professor \cup lecturer$
  - **set difference** *student - researchAssistant*
  - **cross product** $professor \times courses$

Rename Operation

- Rename: Allows us to name, and therefore to refer to, the results of relational-algebra expressions ρ.

Renaming columns: $\rho\ (A\text{->}E,\ B\text{->}K,t1)$

Renaming table: $\rho\ (t2,\ (r - s))$

Combining table: $\rho\ (t3(A\text{->}E,\ B\text{->}K),(r\text{-}s))$

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

E	K
α	1
α	2
β	1

t1

A	B
α	1
β	1

t2

E	K
α	1
β	1

t3

Quick note on notation (Special case for ρ)

good_customers

customer-name	***loan-number***
Patty	1234
Apu	3421
Selma	2342
Ned	4531

bad_customers

customer-name	***loan-number***
Seymour	3432
Marge	3467
Selma	7625
Abraham	3597

If we have two or more relations which feature the same attribute names, we could confuse them.

Classical Case: Use ρ(*customer-name->c, loan-number>l, good_customers*)
Simplify Case: Assume attribute is associated with table:

good_customers.loan-number

Additional Operations

We define additional operations that do not add any power to the relational algebra, but that simplify common queries.

- Natural join
- Conditional Join
- Equi-Join
- Division

All joins are really special cases of conditional join

Also, we've already seen "Set intersection":

$$r \cap s = r - (r - s)$$

Natural-Join Operation: Motivation

Very often, we have a query, and the answer is not contained in a single relation. For example:

return a table contains Park name and the state capital.

NP

name	***state***
YellowStone	WY
Big Bend	TX

State

state	***capital***
WY	Cheyenne
TX	Austin

name	***state***	***state***	***captial***
YellowStone	WY	WY	Cheyeene
YellowStone	WY	TX	Austin
Big Bend	TX	WY	Cheyeene
Big Bend	TX	TX	Austin

The classic relational algebra way to do such queries is a cross product, followed by a selection which tests for equality on some pair of fields.

$\sigma_{NP.state = State.state}(NP \times State)))$

While this works…

- it is unintuitive
- the notation is cumbersome

cust-name	***borrower.l-number***	***loan.l-number***	***branch***
YellowStone	WY	WY	Cheyeene
Big Bend	TX	TX	Austin

Note that in this example the two relations are the same size (2 by 2), this does not have to be the case.

So, we have a more intuitive way of achieving the same effect, the natural join, denoted by the $\bowtie$ symbol

Natural-Join Operation: Intuition

Natural join combines a cross product and a selection into one operation.

It performs a selection forcing equality on ***those attributes that appear in both relation schemes*.**

Duplicates are removed as in all relation operations.

$$NP \bowtie State = \sigma_{NP.state = State.state}(NP \times State)))$$

Natural-Join Operation: Intuition

Natural join combines a cross product and a selection into one operation.

$$NP \bowtie State = \sigma_{NP.state = State.state}(NP \times State)))$$

There are two special cases:

- If the two relations have no attributes in common, then their natural join is simply their **cross product**.
- If the two relations have more than one attribute in common, then the natural join selects only the rows where **all pairs of matching attributes match**. (let's see an example on the next slide).

A

l-name	*f-name*	*age*
Bouvier	Selma	40
Bouvier	Patty	40
Smith	Maggie	2

B

l-name	*f-name*	*ID*
Bouvier	Selma	1232
Smith	Selma	4423

Both the *l-name* and the *f-name* match, so select.

Only the *f-names* match, so don't select.

Only the *l-names* match, so don't select.

l-name	*f-name*	*age*	*l-name*	*f-name*	*ID*
Bouvier	**Selma**	40	**Bouvier**	**Selma**	1232
Bouvier	**Selma**	40	Smith	**Selma**	4423
Bouvier	Patty	2	**Bouvier**	Selma	1232
Bouvier	Patty	40	Smith	Selma	4423
Smith	Maggie	2	Bouvier	Selma	1232
Smith	Maggie	2	**Smith**	Selma	4423

We remove duplicate attributes…

l-name	*f-name*	*age*	*l-name*	*f-name*	*ID*
Bouvier	Selma	40	Bouvier	Selma	1232

Note that this is just a way to visualize the natural join, we don't really have to do the cross product as in this example

$A \bowtie B =$

l-name	*f-name*	*age*	*ID*
Bouvier	Selma	40	1232

Natural-Join Operation

- Notation: $r \bowtie s$
- Let r and s be relation instances on schemas R and S respectively.
  - Output: a relation on schema $R \cup S$
- If t_r and t_s have the same value on each of the attributes in $R \cap S$, a tuple t is added to the result, where
  - t has the same value as t_r on r
  - t has the same value as t_s on s
- Example:

  $R = (A, B, C, D)$

  $S = (E, B, D)$
- Result schema = (A, B, C, D, E)
- $r \bowtie s$ is defined as:

$$\Pi_{r.A,\ r.B,\ r.C,\ r.D,\ s.E} (\sigma_{r.B = s.B \ \wedge\ r.D = s.D} (r \times s))$$

Natural Join Operation – Example

- Relation instances *r*, *s*:

A	*B*	*C*	*D*
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	*D*	*E*
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

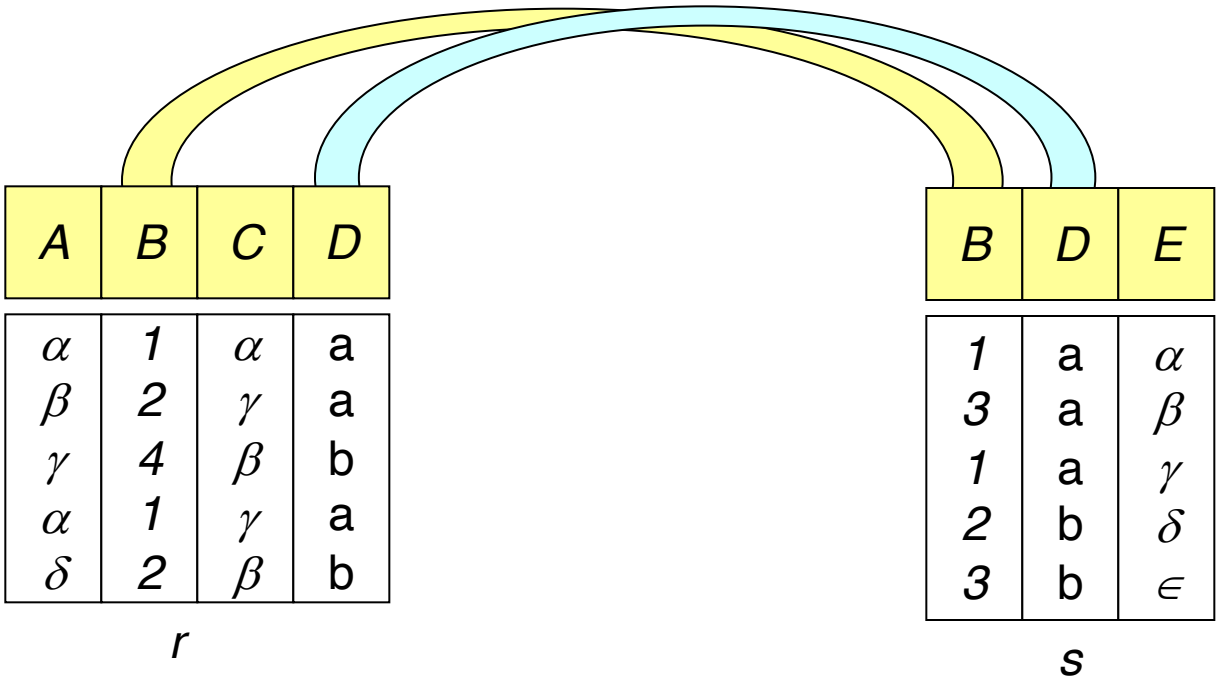
s

$r \bowtie s$

A	*B*	*C*	*D*	*E*
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

How did we get here?

Lets do a trace over the next few slides…



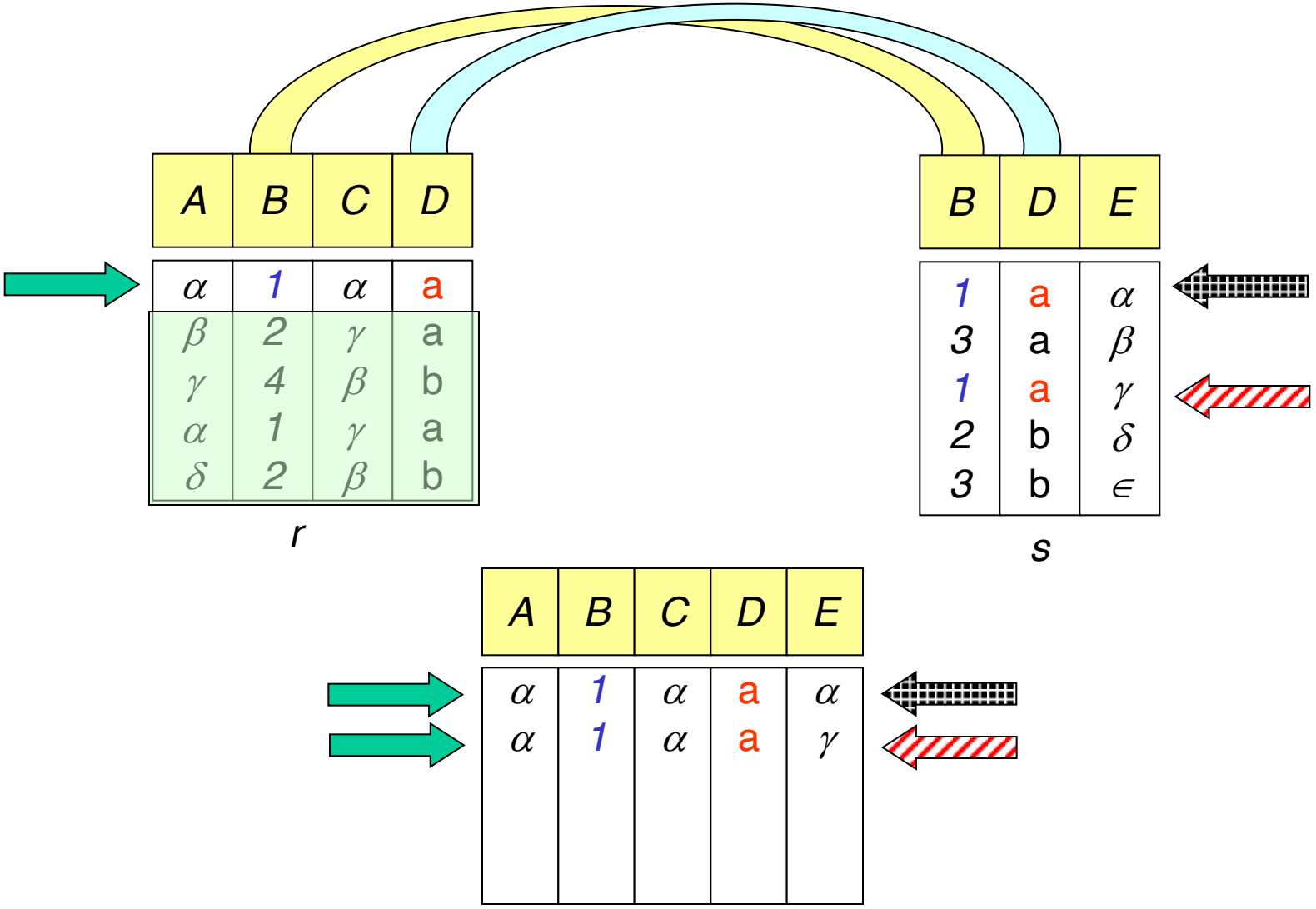
A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

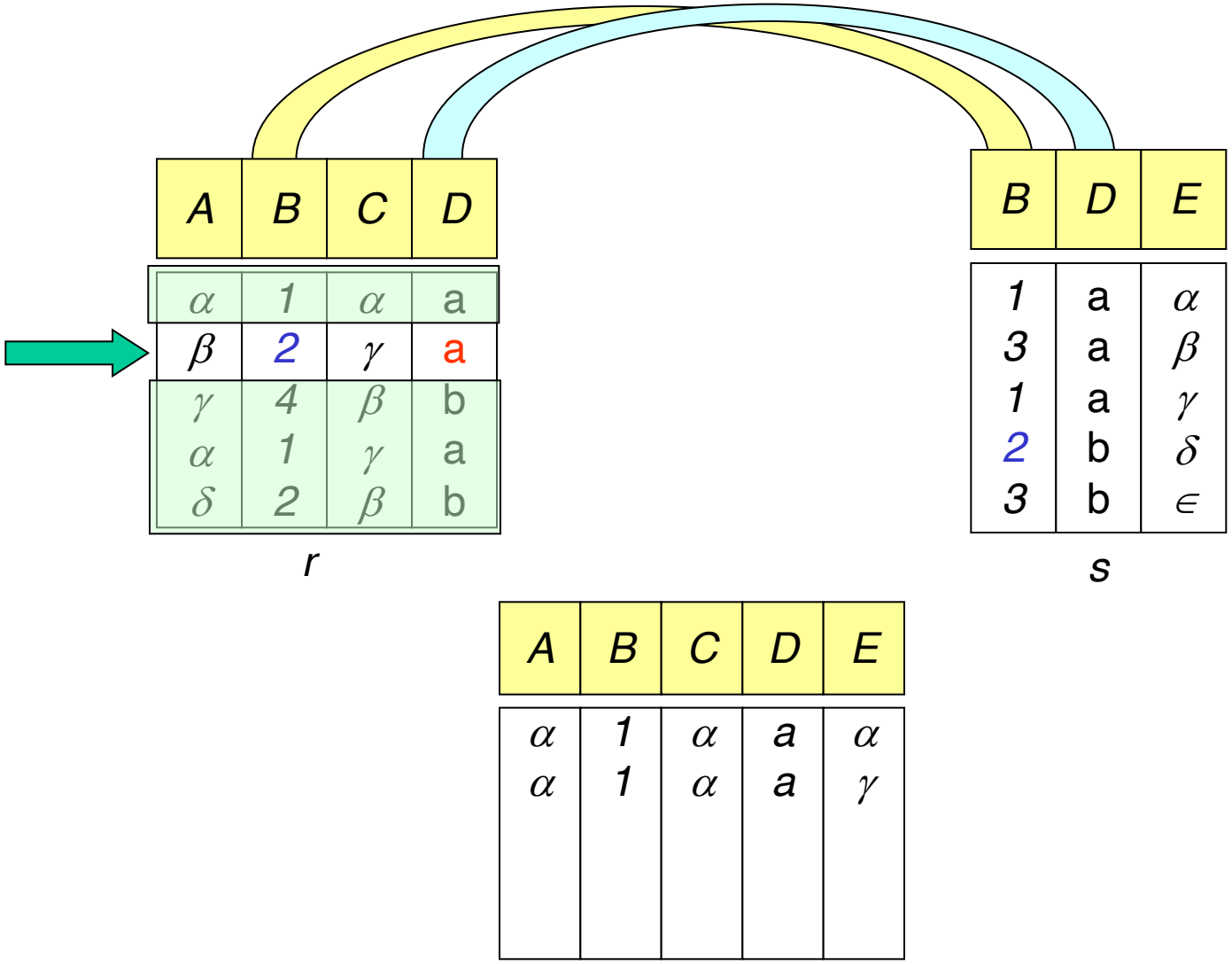
B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

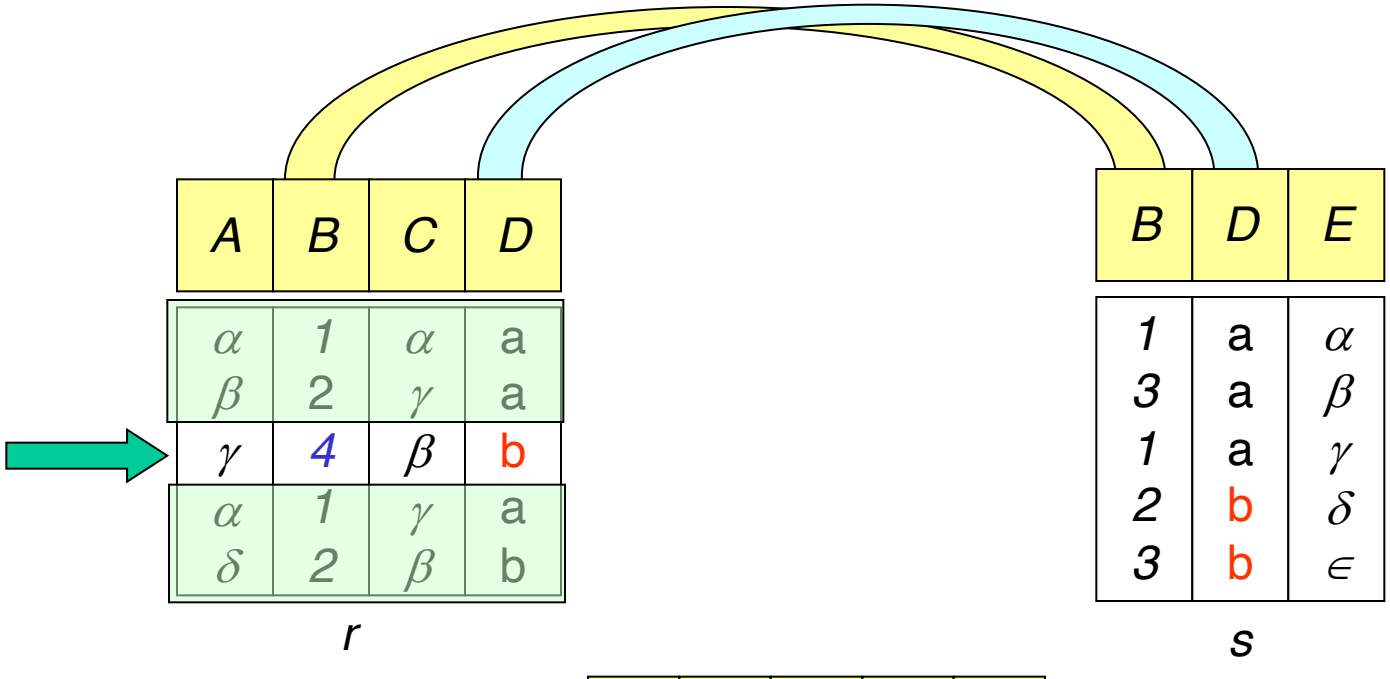
First we note which attributes the two relations have in common…



There are two rows in s that match our first row in r, (in the relevant attributes) so both are joined to our first row…

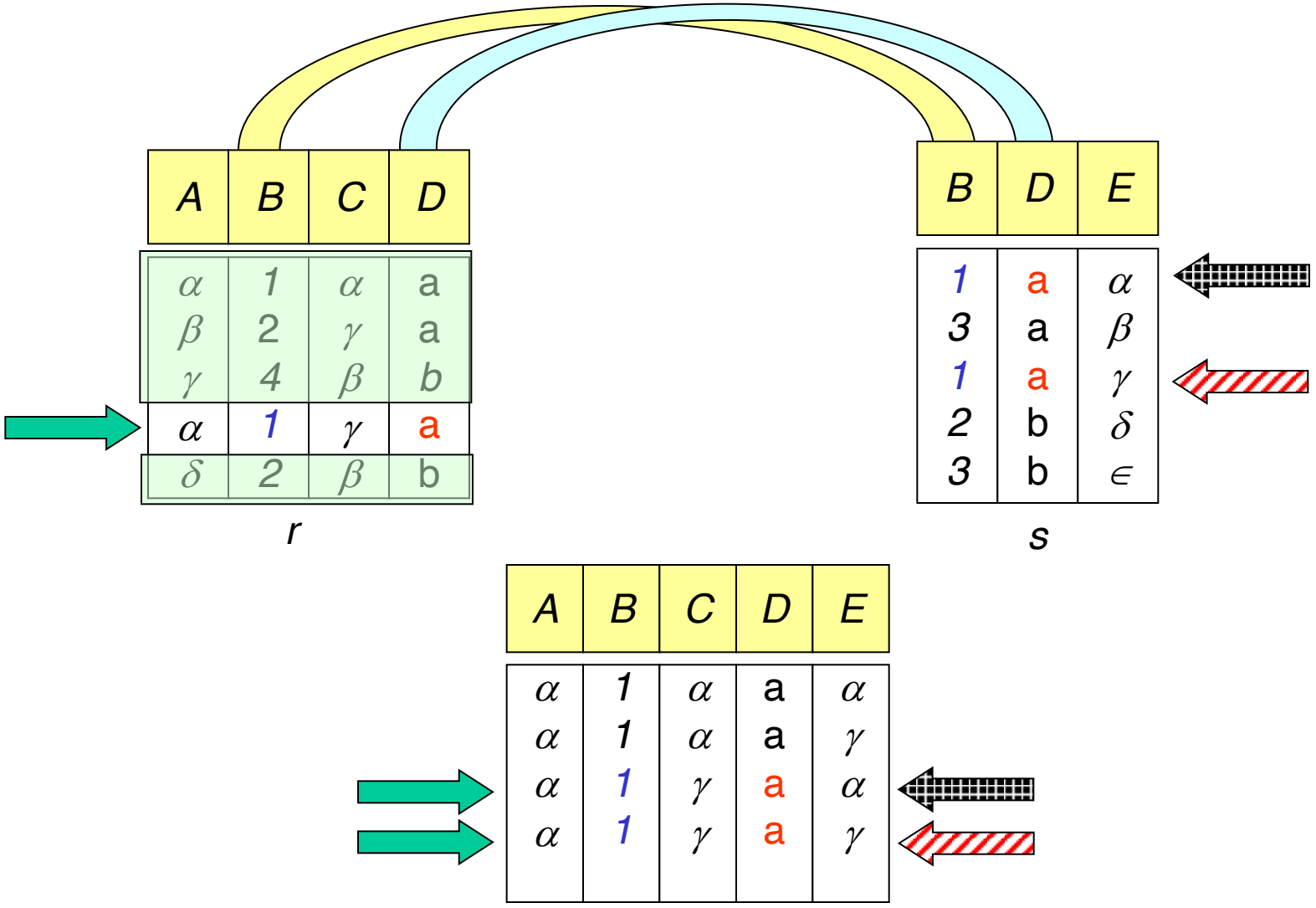


…there are no rows in s that match our second row in r, so do nothing…

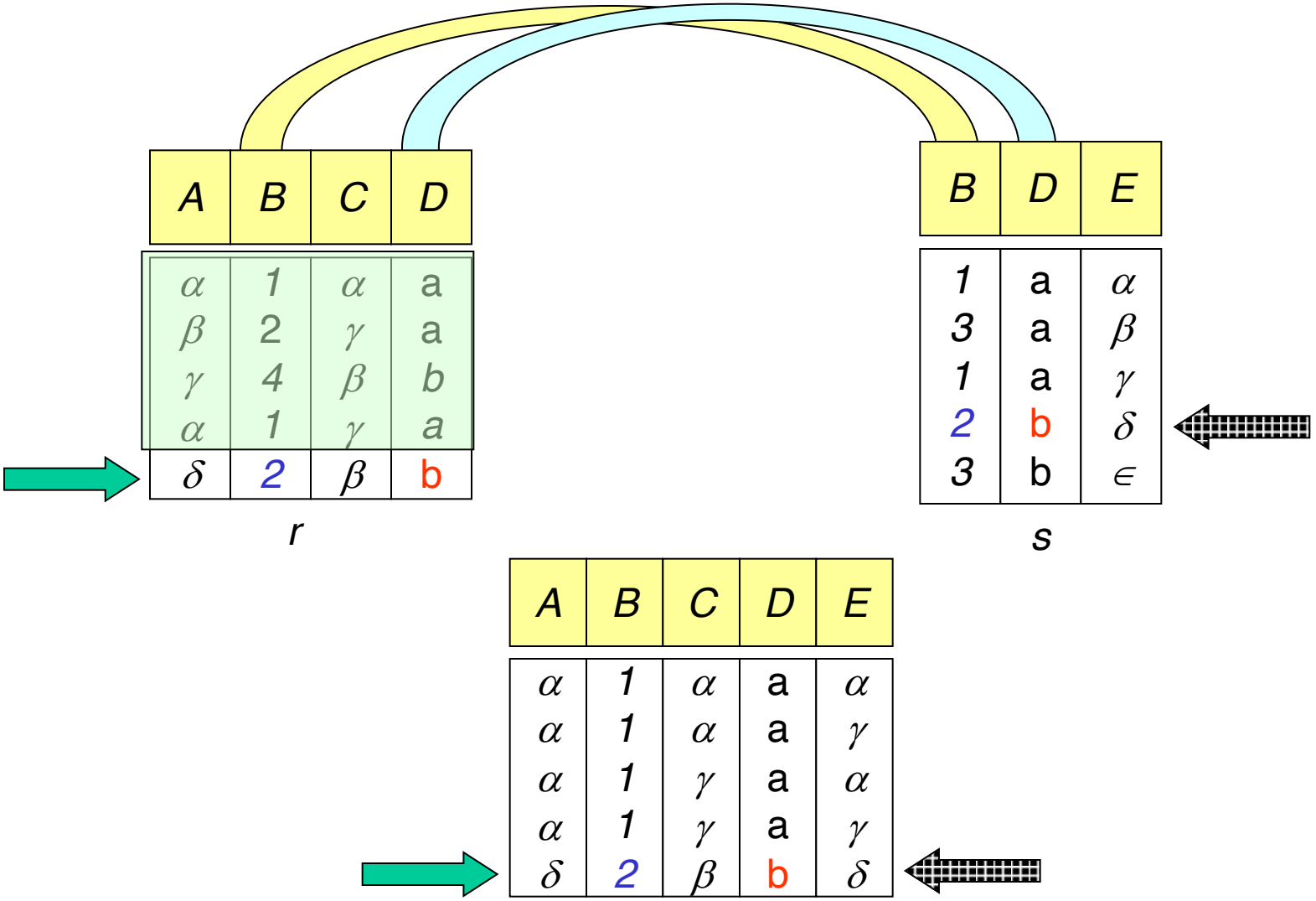


A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ

…there are no rows in s that match our third row in r, so do nothing…



There are two rows in s that match our fourth row in r, so both are joined to our fourth row…



There is one row that matches our fifth row in r,.. so it is joined to our fifth row and we are done!

Conditional-Join Operation:

The conditional join is actually the most general type of join. I introduced the natural join first only because it is more intuitive and... natural!

Just like natural join, conditional join combines a cross product and a selection into one operation. However instead of only selecting rows that have equality on those attributes that appear in both relation schemes, we allow selection based on any predicate.

$$r \bowtie_c s = \sigma_c(r \times s)$$

Where c is any predicate the attributes of r and/or s

Duplicate rows are removed as always, but duplicate columns are not removed!

Conditional-Join Example:

We want to find all women that are younger than their husbands…

r

l-name	*f-name*	*marr-Lic*	*age*
Simpson	Marge	777	35
Lovejoy	Helen	234	38
Flanders	Maude	555	24
Krabappel	Edna	978	40

s

l-name	*f-name*	*marr-Lic*	*age*
Simpson	Homer	777	36
Lovejoy	Timothy	234	36
Simpson	Bart	*null*	9

$r \bowtie_{r.age < s.age \text{ AND } r.Marr\text{-}Lic = s.Marr\text{-}Lic} s$

What is the result?

Hint: $r \bowtie_c s = \sigma_c(r \times s)$

Conditional-Join Example:

We want to find all women that are younger than their husbands…

r

l-name	*f-name*	*marr-Lic*	*age*
Simpson	Marge	777	35
Lovejoy	Helen	234	38
Flanders	Maude	555	24
Krabappel	Edna	978	40

s

l-name	*f-name*	*marr-Lic*	*age*
Simpson	Homer	777	36
Lovejoy	Timothy	234	36
Simpson	Bart	*null*	9

$r \bowtie_{r.age < s.age \text{ AND } r.Marr\text{-}Lic = s.Marr\text{-}Lic} s$

r.l-name	*r.f-name*	*r.Marr-Lic*	*r.age*	*s.l-name*	*s.f-name*	*s.marr-Lic*	*s.age*
Simpson	Marge	777	35	Simpson	Homer	777	36

Note we have removed ambiguity of attribute names by using "dot" notation
Also note the redundant information in the *marr-lic* attributes

Equi-Join

- Equi-Join: Special case of conditional join where the conditions consist only of equalities.
- Natural Join: Special case of equi-join in which equalities are specified on ALL fields having the same names in both relations.

Equi-Join

r

l-name	*f-name*	*marr-Lic*	*age*
Simpson	Marge	777	35
Lovejoy	Helen	234	38
Flanders	Maude	555	24
Krabappel	Edna	978	40

s

l-name	*f-name*	*marr-Lic*	*age*
Simpson	Homer	777	36
Lovejoy	Timothy	234	36
Simpson	Bart	*null*	9

$r \bowtie_{r.Marr\text{-}Lic = s.Marr\text{-}Lic} s$

r.l-name	*r.f-name*	*Marr-Lic*	*r.age*	*s.l-name*	*s.f-name*	*s.age*
Simpson	Marge	777	35	Simpson	Homer	36
Lovejoy	Helen	234	38	Lovejoy	Timothy	36

Review on Joins

- All joins combine a cross product and a selection into one operation.
- Conditional Join
  - the selection condition can be of any predicate (e.g. rating1 > rating2)
- Equi-Join:
  - Special case of conditional join where the conditions consist only of equalities.
- Natural Join
  - Special case of equi-join in which equalities are specified on ALL fields having the same names in both relations.

Note of Current Schedule

- This Week:
  - Tue: Additional Operations
  - Thu: Advance Usage of Relational Algebra
- Next Week:
  - Tue: Review Class
  - Thu: **Midterm Exam**