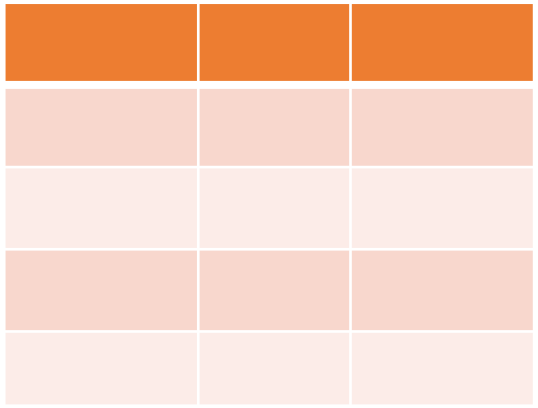
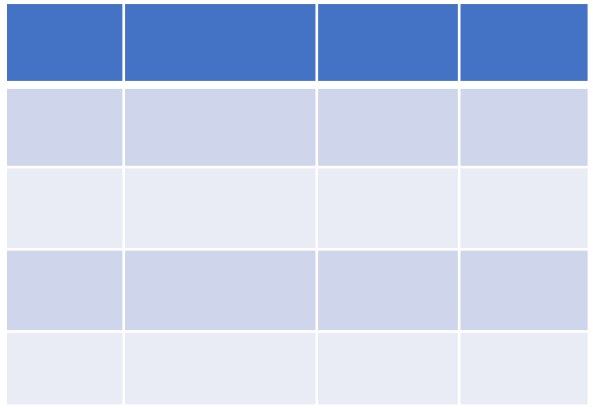
Relational Algebra

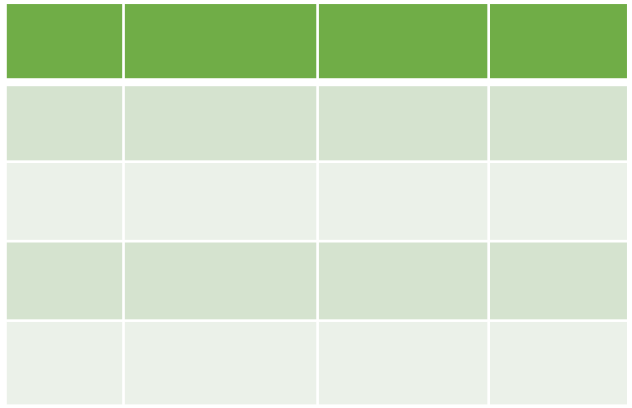
select, project, join, set operators, renaming, notation

Suan Lee

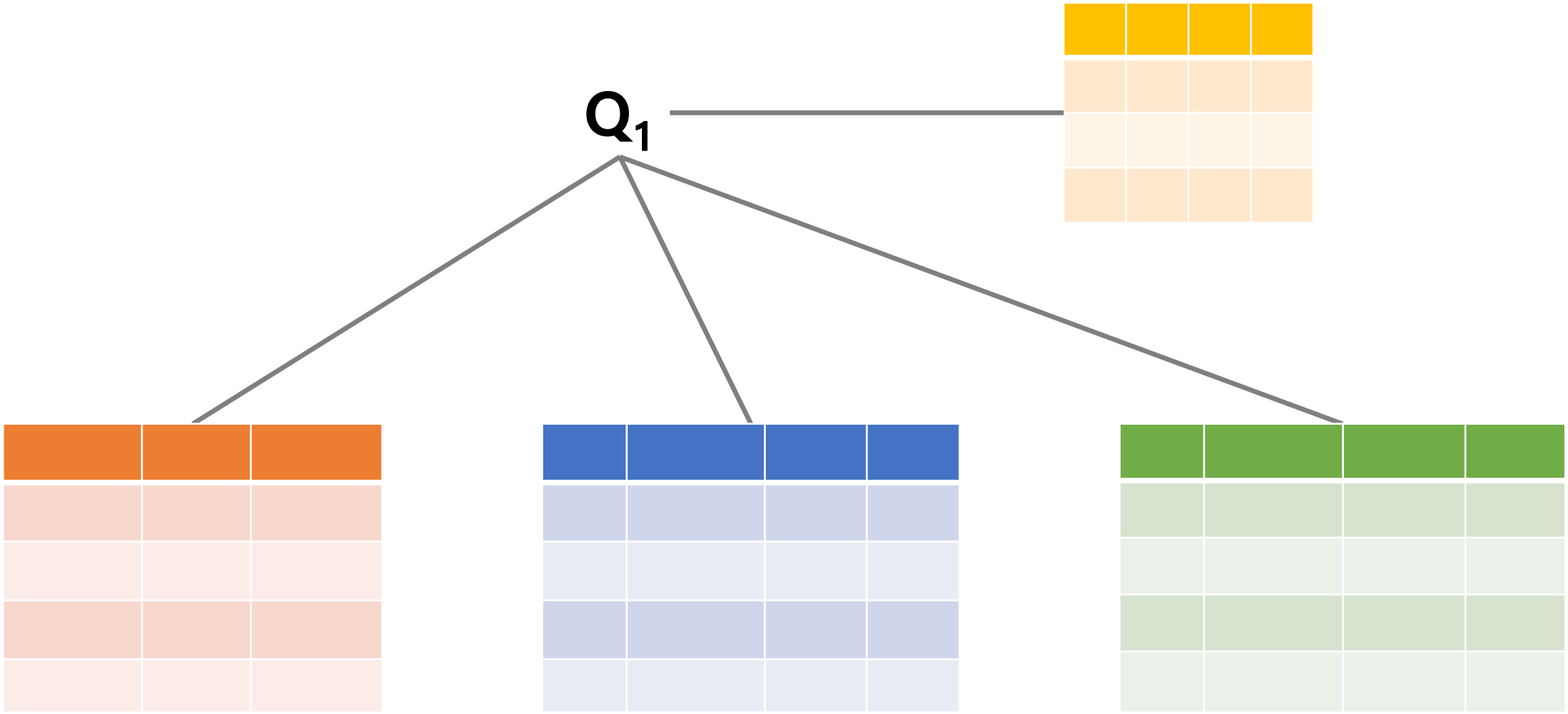
- Query (expression) on set of relations produces relation as a result







- Query (expression) on set of relations produces relation as a result



Examples: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

Examples: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Simplest query: relation name

Use operators to filter, slice, combine

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Select operator: picks certain rows

Students with GPA>3.7

Students with GPA>3.7 and HS<1000

Applications to Stanford CS major

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Select operator: picks certain rows

Students with GPA>3.7

$\sigma_{GPA>3.7}$ Student

Students with GPA>3.7 and HS<1000

Applications to Stanford CS major

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Select operator: picks certain rows

Students with GPA>3.7

$\sigma_{GPA>3.7}$ Student

Students with GPA>3.7 and HS<1000

$\sigma_{GPA>3.7 \wedge HS<1000}$ Student

Applications to Stanford CS major

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Select operator: picks certain rows

Students with GPA>3.7

$\sigma_{GPA>3.7}$ Student

Students with GPA>3.7 and HS<1000

$\sigma_{GPA>3.7 \wedge HS<1000}$ Student

Applications to Stanford CS major

$\sigma_{cName='Stanford' \wedge major='cs'}$ Apply

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Select operator: picks certain rows

Students with GPA>3.7

$\sigma_{GPA>3.7}$ Student

Students with GPA>3.7 and HS<1000

$\sigma_{GPA>3.7 \wedge HS<1000}$ Student

Applications to Stanford CS major

$\sigma_{cName='Stanford' \wedge major='cs'}$ Apply

σ_{cond} Rel

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Project operator: picks certain columns

ID and decision of all applications

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Project operator: picks certain columns

ID and decision of all applications

$\Pi_{sID,\ dec}Apply$

sID	dec

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Project operator: picks certain columns

ID and decision of all applications

$\Pi_{sID,\ dec}Apply$

$\Pi_{A1,\ A2,\ \ldots,\ An}$

sID	dec

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

To pick both rows and columns…

ID and name of students with GPA>3.7

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

To pick both rows and columns…

ID and name of students with GPA>3.7

$\Pi_{sID,\ sName}(\sigma_{GPA>3.7}Student)$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

To pick both rows and columns…

ID and name of students with GPA>3.7

$$\Pi_{sID,\ sName}(\sigma_{GPA>3.7}Student)$$

$$\sigma_{cond}(Expr)$$

$$\Pi_{A1,\ A2,\ \ldots\ An}(Expr)$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Duplicates

List of application majors and decisions

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Duplicates

List of application majors and decisions

$$\Pi_{major,\ dec}\ Apply$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Duplicates

List of application majors and decisions

$\Pi_{major,\ dec}$ Apply

SQL: multisets, bags
Relational Algebra: sets

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Student × *Apply*

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Student × *Apply*

College

cName	state	enr

Student

sID	sName	GPA	HS

×

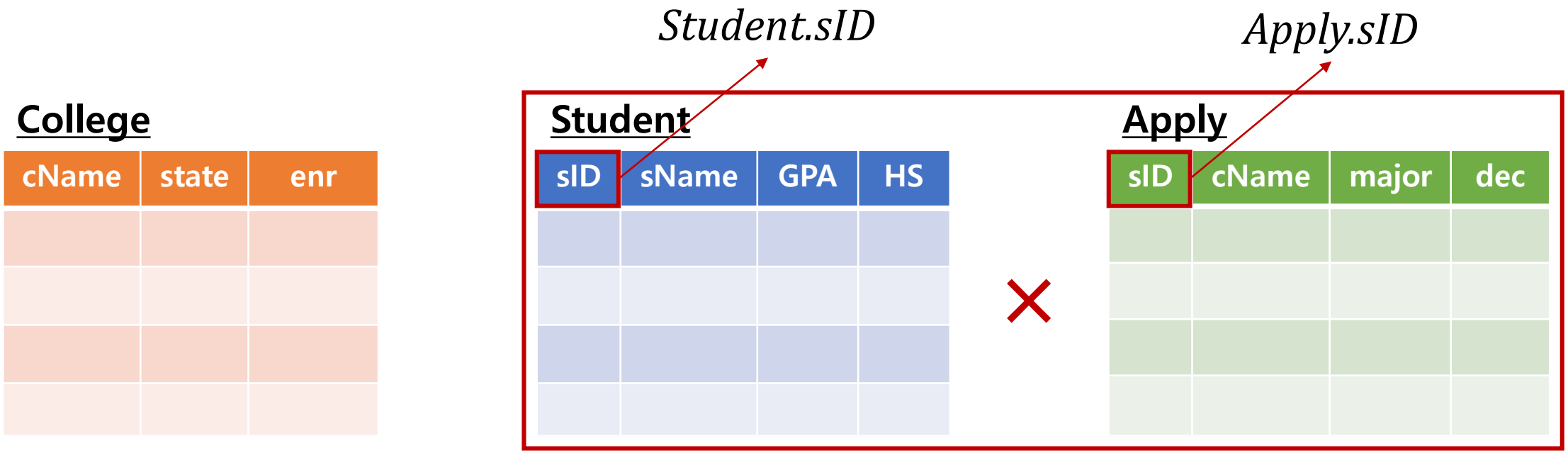
Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

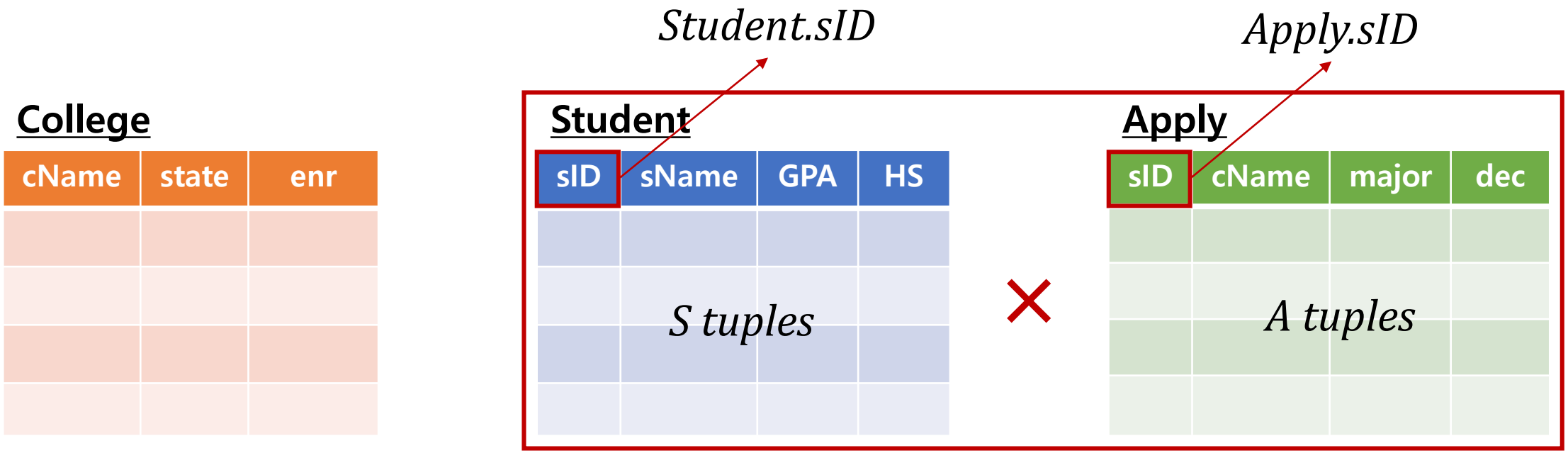
Student × *Apply*



Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Student × *Apply*



Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Names and GPAs of students with HS>1000 who applied to CS and were rejected

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Names and GPAs of students with HS>1000 who applied to CS and were rejected

(Student × Apply)

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Names and GPAs of students with HS>1000 who applied to CS and were rejected

$$\sigma_{\text{Student.sID=Apply.sID}} (\text{Student} \times \text{Apply})$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Names and GPAs of students with HS>1000 who applied to CS and were rejected

$$\sigma_{\text{Student.sID=Apply.sID} \wedge \text{HS>1000} \wedge \text{major='cs'}} (\text{Student} \times \text{Apply})$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Names and GPAs of students with HS>1000 who applied to CS and were rejected

$$\sigma_{\text{Student.sID=Apply.sID} \wedge \text{HS>1000} \wedge \text{major='cs'} \wedge \text{dec='R'}} (\text{Student} \times \text{Apply})$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Cross-product: combine two relations

(a.k.a. **Cartesian product**)

Names and GPAs of students with HS>1000 who applied to CS and were rejected

$$\Pi_{\text{sName, GPA}} (\sigma_{\text{Student.sID=Apply.sID} \wedge \text{HS>1000} \wedge \text{major='cs'} \wedge \text{dec='R'}} (\text{Student} \times \text{Apply}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

- Enforce equality on all attributes with same name
- Eliminate one copy of duplicate attributes

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

- Enforce equality on all attributes with same name
- Eliminate one copy of duplicate attributes

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

- Enforce equality on all attributes with same name
- Eliminate one copy of duplicate attributes

College

cName	state	enr

Student

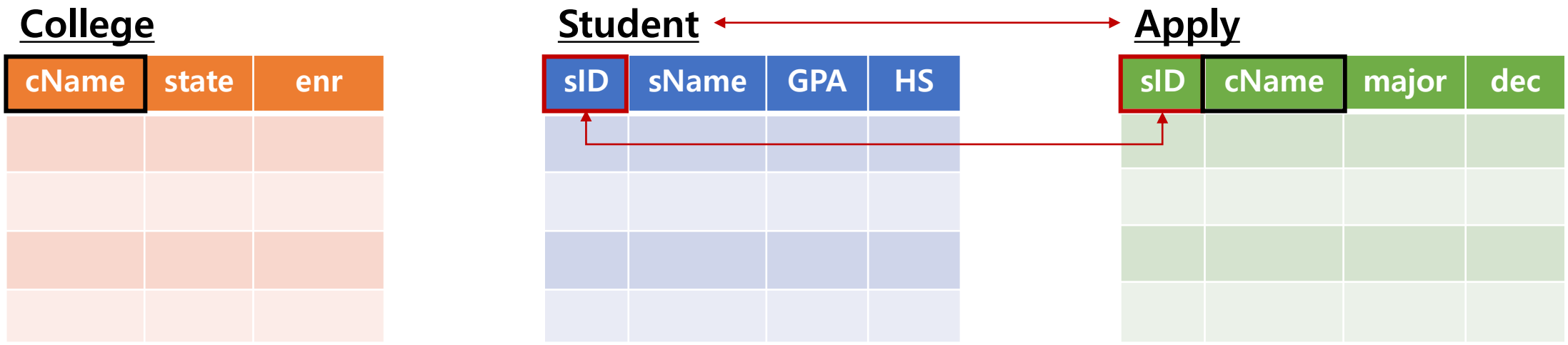
sID	sName	GPA	HS

Apply

sID	cName	major	dec

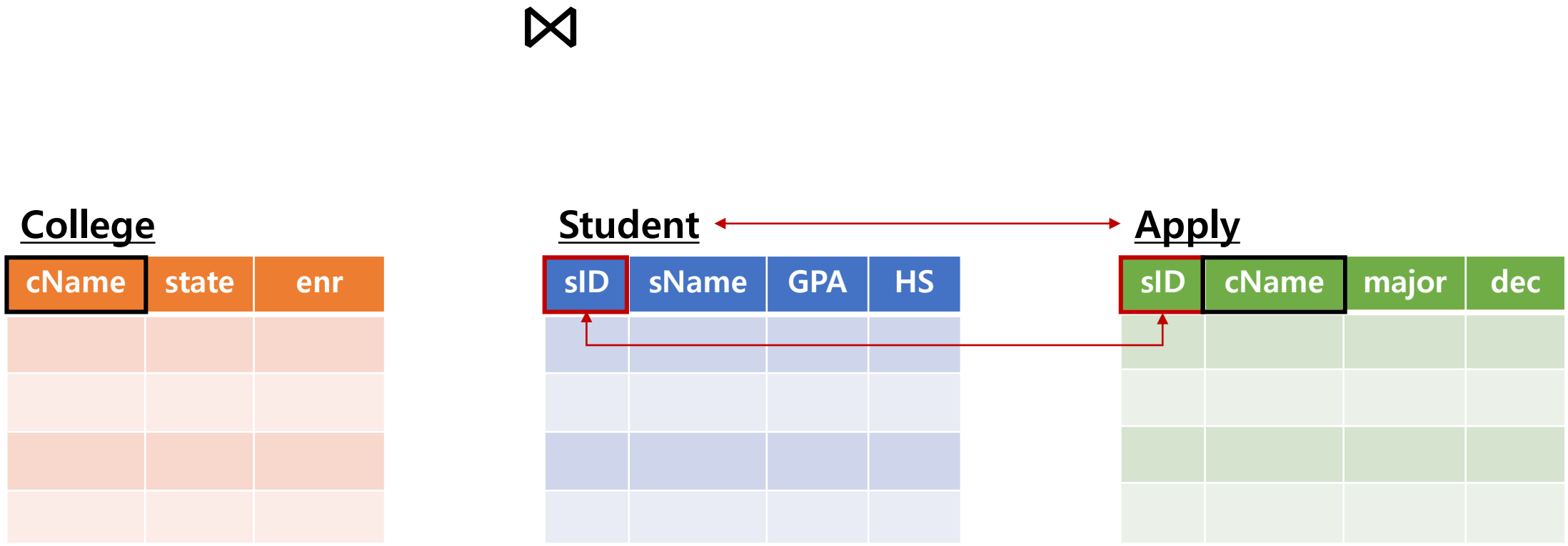
Natural Join

- Enforce equality on all attributes with same name
- Eliminate one copy of duplicate attributes



Natural Join

- Enforce equality on all attributes with same name
- Eliminate one copy of duplicate attributes



Natural Join

Names and GPAs of students with HS>1000 who applied to CS and were rejected

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

Names and GPAs of students with HS>1000 who applied to CS and were rejected

(Student⋈Apply)

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

Names and GPAs of students with HS>1000 who applied to CS and were rejected

$$\Pi_{\text{sName, GPA}}(\sigma_{\text{HS>1000} \wedge \text{major='cs'} \wedge \text{dec='R'}} (\text{Student} \bowtie \text{Apply}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

Names and GPAs of students with HS>1000 who applied to CS at college with enr>20,000 and were rejected

$$\Pi_{\text{sName, GPA}}(\sigma_{\text{HS>1000} \wedge \text{major='cs'} \wedge \text{dec='R'}} (\text{Student} \bowtie \text{Apply}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

Names and GPAs of students with HS>1000 who applied to CS at college with enr>20,000 and were rejected

$$\Pi_{\text{sName, GPA}}(\sigma_{\text{HS>1000} \wedge \text{major='cs'} \wedge \text{dec='R'} \wedge \text{enr>20000}}(\text{Student} \bowtie (\text{Apply} \bowtie \text{College}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Natural Join

$$Exp1 \bowtie Exp2 \equiv$$

$$\Pi_{schema(E1),\ schema(E2)}(\ldots)$$

$$\sigma_{E1A1=E2A1 \wedge E1A2=E2A2 \wedge \cdots}(Exp1 \times Exp2)$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Theta Join

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Theta Join

$$Exp1 \bowtie Exp2 \equiv \sigma_{\Theta}(Exp1 \bowtie Exp2)$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Theta Join

$$Exp1 \bowtie Exp2 \equiv \sigma_{\Theta}(Exp1 \bowtie Exp2)$$

- Basic operation implemented in DBMS
- Term "join" often means theta join

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Query (expression) on set of relations produces relation as a result

- Simplest query: relation name
- Use operators to filter, slice, combine
- Operators so far: select, project, cross-product, natural join, theta join

Relational algebra query (expression) on set of relations produces relation as a result

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

Relational algebra query (expression) on set of relations produces relation as a result

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Union operator

List of college and student names

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Union operator

List of college and student names

Stanford
Susan
Cornell
Mary
John
…

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Union operator

List of college and student names

Stanford
Susan
Cornell
Mary
John
…

$\times$ $\bowtie$ t_1-t_2

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Union operator

List of college and student names

Stanford
Susan
Cornell
Mary
John
...

$\times$ $\bowtie$ t_1 t_2

t_1-t_2

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Union operator

List of college and student names

Stanford
Susan
Cornell
Mary
John
...

$\times$ $\bowtie$ t_1 t_2 $\cup$ t_1-t_2

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Union operator

List of college and student names

Stanford
Susan
Cornell
Mary
John
...

× ⋈ t_1-t_2

t_1
t_2

∪

$$\Pi_{cName}\text{College} \cup \Pi_{sName}\text{Student}$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Difference operator

IDs of students who didn't apply anywhere

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Difference operator

IDs of students who didn't apply anywhere

Π_{sID}Student - Π_{sID}Apply

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Difference operator

IDs and names of students who didn't apply anywhere

Π_{sID}Student - Π_{sID}Apply

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Difference operator

IDs and names of students who didn't apply anywhere

$\Pi_{sID,\ sName}Student - \Pi_{sID,\ ?}Apply$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Difference operator

IDs and names of students who didn't apply anywhere

Π_{sID}Student - Π_{sID}Apply

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Difference operator

IDs and names of students who didn't apply anywhere

$(\Pi_{sID}\text{Student} - \Pi_{sID}\text{Apply})$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Difference operator

IDs and names of students who didn't apply anywhere

$$\Pi_{sID,\ sName}((\Pi_{sID}Student - \Pi_{sID}Apply) \bowtie Student)$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Intersection operator

Names that are both a college name and a student name

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Intersection operator

Names that are both a college name and a student name

$\Pi_{cName}College \cap \Pi_{sName}Student$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

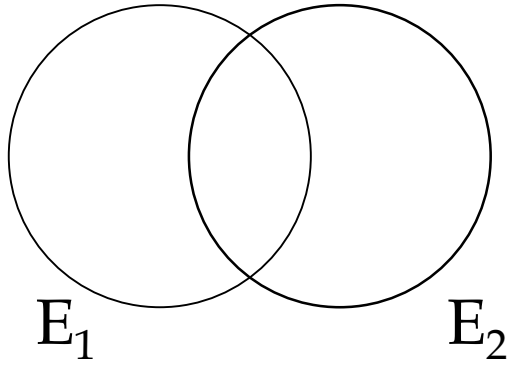
Intersection doesn’t add expressive power (1)

Intersection doesn't add expressive power (1)

$$E_1 \cap E_2 \equiv E_1 - (E_1 - E_2)$$

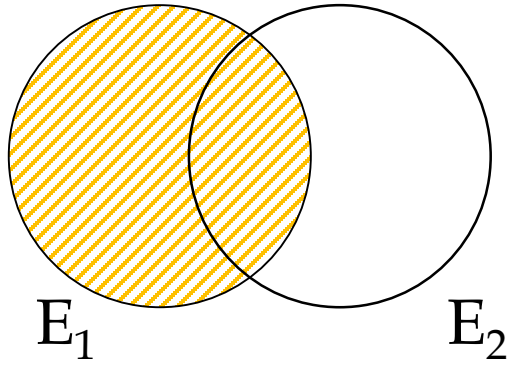
Intersection doesn't add expressive power (1)

$$E_1 \cap E_2 \equiv E_1 - (E_1 - E_2)$$



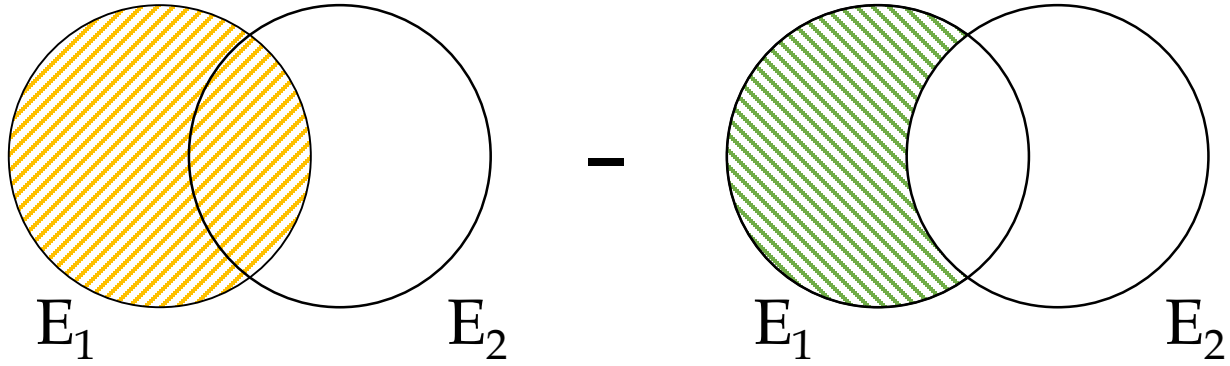
Intersection doesn't add expressive power (1)

$$E_1 \cap E_2 \equiv E_1 - (E_1 - E_2)$$



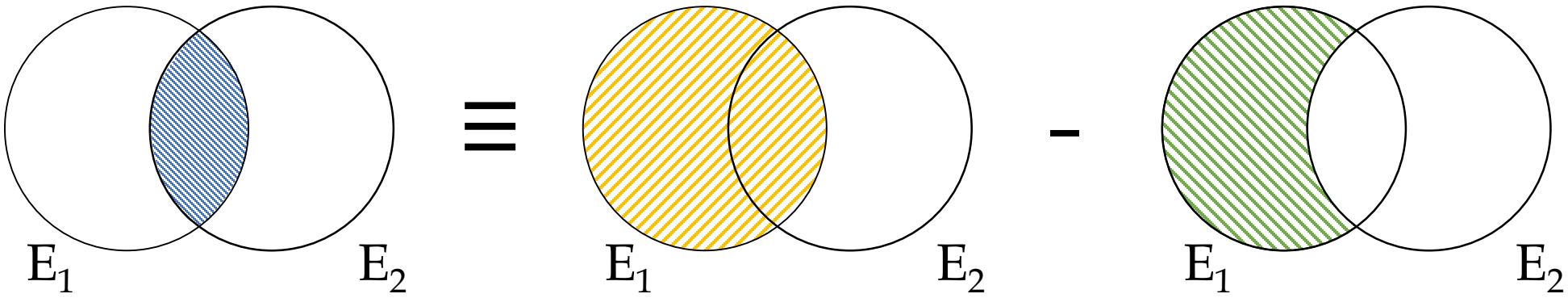
Intersection doesn't add expressive power (1)

$$E_1 \cap E_2 \equiv E_1 - (E_1 - E_2)$$



Intersection doesn't add expressive power (1)

$E_1 \cap E_2 \equiv E_1 - (E_1 - E_2)$



Intersection doesn't add expressive power (2)

$$E_1 \cap E_2 \equiv E_1 \bowtie E_2$$

schema=

Rename operator

1.

2.

3.

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

1. $\rho_{R(A_1, ..., A_n)}(E)$

2.

3.

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

1. $\rho_{R(A_1, \ldots, A_n)}(E)$

2. $\rho_{R}(E)$

3. $\rho_{A_1, \ldots, A_n}(E)$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

To unify schemas for set operators

List of college and student names

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

To unify schemas for set operators

List of college and student names

$$\Pi_{cName}\text{College} \cup \Pi_{sName}\text{Student}$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

To unify schemas for set operators

List of college and student names

$$\rho_{C(name)}\,(\Pi_{cName}\text{College}) \cup \rho_{C(name)}\,(\Pi_{sName}\text{Student})$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

Stanford	Berkeley
Berkeley	UCLA

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$\text{College} \times \text{College}$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$\sigma_{state=state}(College \times College)$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$\rho_{c1(n1, s1, e1)}(\text{College})$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$\rho_{c1(n1, s1, e1)}(\text{College})$ $\rho_{c2(n2, s2, e2)}(\text{College})$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$(\rho_{c1(n1,\ s1,\ e1)}(\text{College})) \times (\rho_{c2(n2,\ s2,\ e2)}(\text{College}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$\sigma_{s1=s2}((\rho_{c1(n1,\ s1,\ e1)}(\text{College})) \times (\rho_{c2(n2,\ s2,\ e2)}(\text{College})))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$(\rho_{c1(n1,\ s1,\ e1)}(\text{College})) \bowtie (\rho_{c2(n2,\ s2,\ e2)}(\text{College}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$(\rho_{c1(n1,\, s,\, e1)}(\text{College})) \bowtie (\rho_{c2(n2,\, s,\, e2)}(\text{College}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$(\rho_{c1(n1,\, s,\, e1)}(\text{College})) \bowtie (\rho_{c2(n2,\, s,\, e2)}(\text{College}))$$

Stanford	Stanford
Berkeley	Berkeley
Stanford	Berkeley

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$\sigma_{n1 \neq n2}(\rho_{c1(n1,\, s,\, e1)}(\text{College})) \bowtie (\rho_{c2(n2,\, s,\, e2)}(\text{College}))$$

Stanford	Stanford
Berkeley	Berkeley
Stanford	Berkeley

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

Stanford Berkeley

$$\sigma_{n1 \neq n2}(\rho_{c1(n1,\, s,\, e1)}(\text{College})) \bowtie (\rho_{c2(n2,\, s,\, e2)}(\text{College}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

$$\sigma_{n1 \neq n2}(\rho_{c1(n1,\, s,\, e1)}(\text{College})) \bowtie (\rho_{c2(n2,\, s,\, e2)}(\text{College}))$$

Stanford	Berkeley
Berkeley	Stanford

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Rename operator

For disambiguation in "self-joins"

Pairs of colleges in same state

Berkeley Stanford

$$\sigma_{n1<n2}(\rho_{c1(n1,\ s,\ e1)}(\text{College})) \bowtie (\rho_{c2(n2,\ s,\ e2)}(\text{College}))$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Alternate notation (1)

Assignment statements – Pairs of colleges in same state

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Alternate notation (1)

Assignment statements – Pairs of colleges in same state

$C1 := \rho_{c1,\ s,\ e1}College$

$C2 := \rho_{c2,\ s,\ e2}College$

$Cp := C1 \bowtie C2$

$Ans := \sigma_{n1<n2}CP$

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Alternate notation (2)

Expression tree – GPAs of students applying to CS in CA

College

cName	state	enr

Student

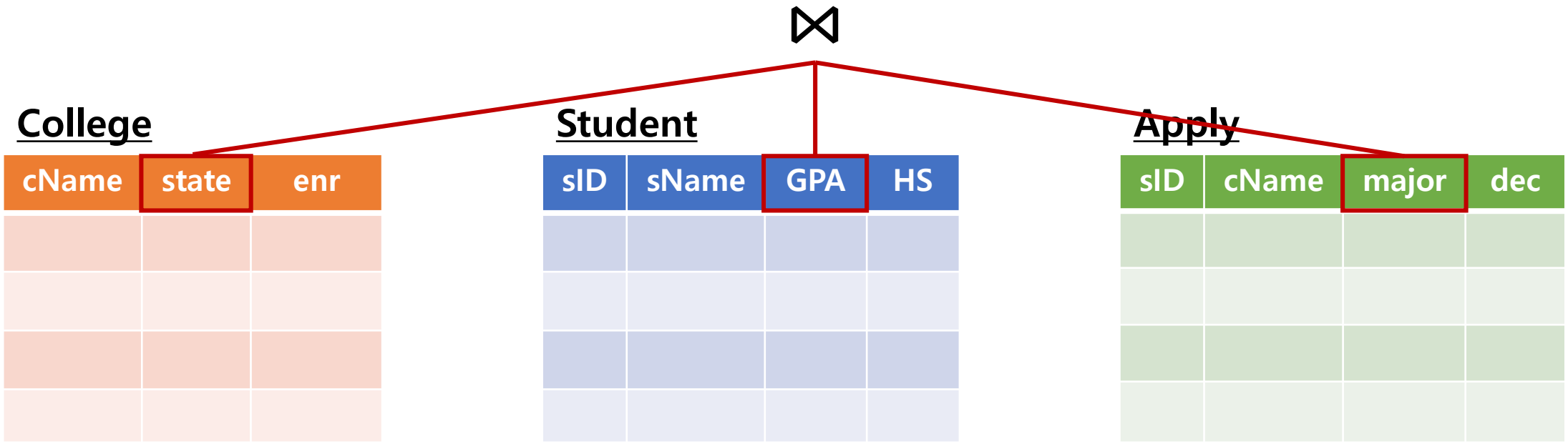
sID	sName	GPA	HS

Apply

sID	cName	major	dec

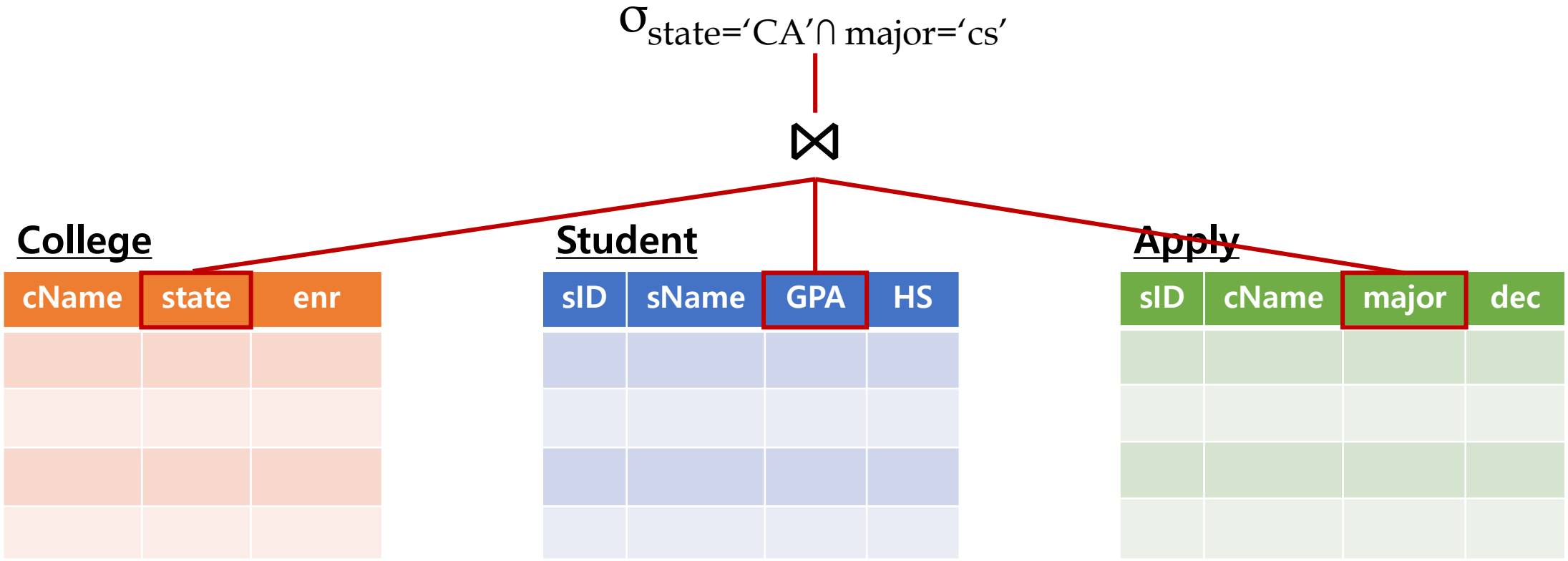
Alternate notation (2)

Expression tree – GPAs of students applying to CS in CA



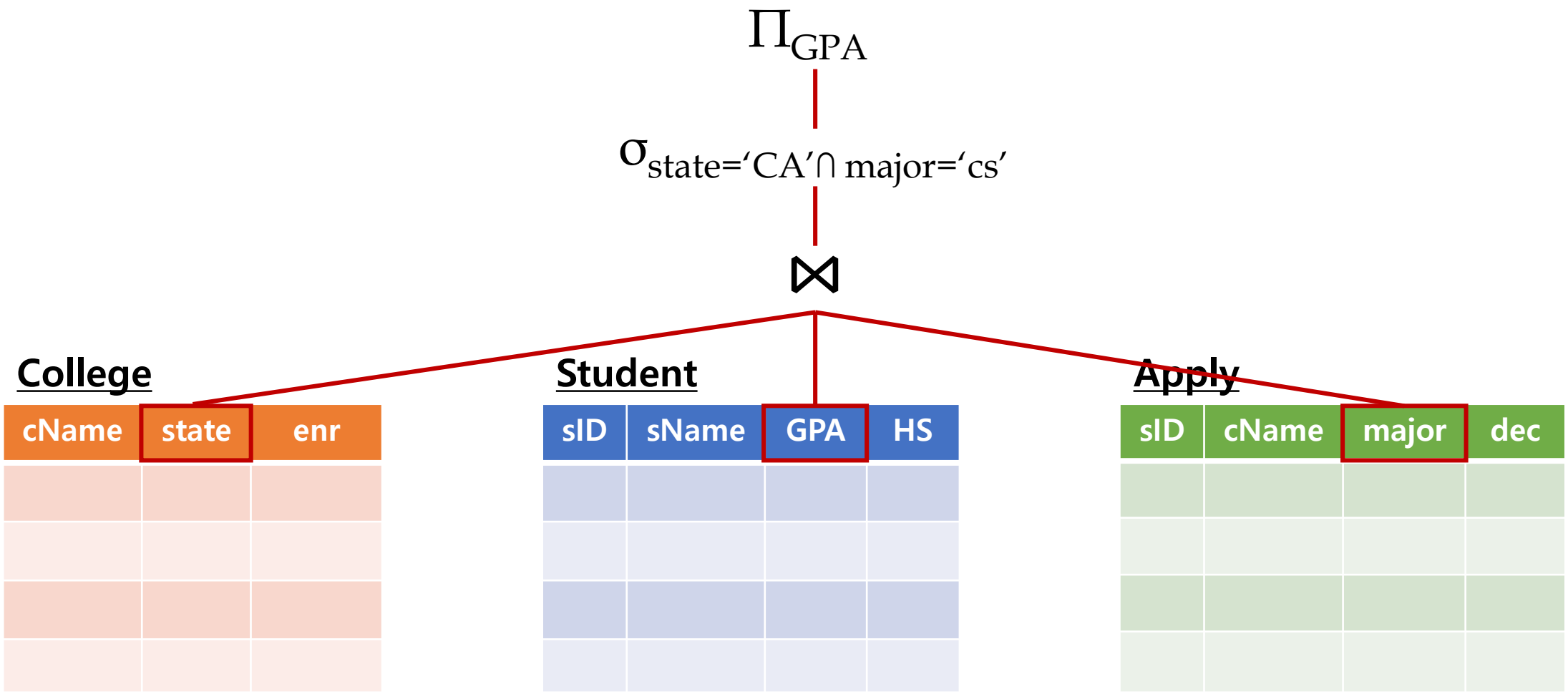
Alternate notation (2)

Expression tree – GPAs of students applying to CS in CA



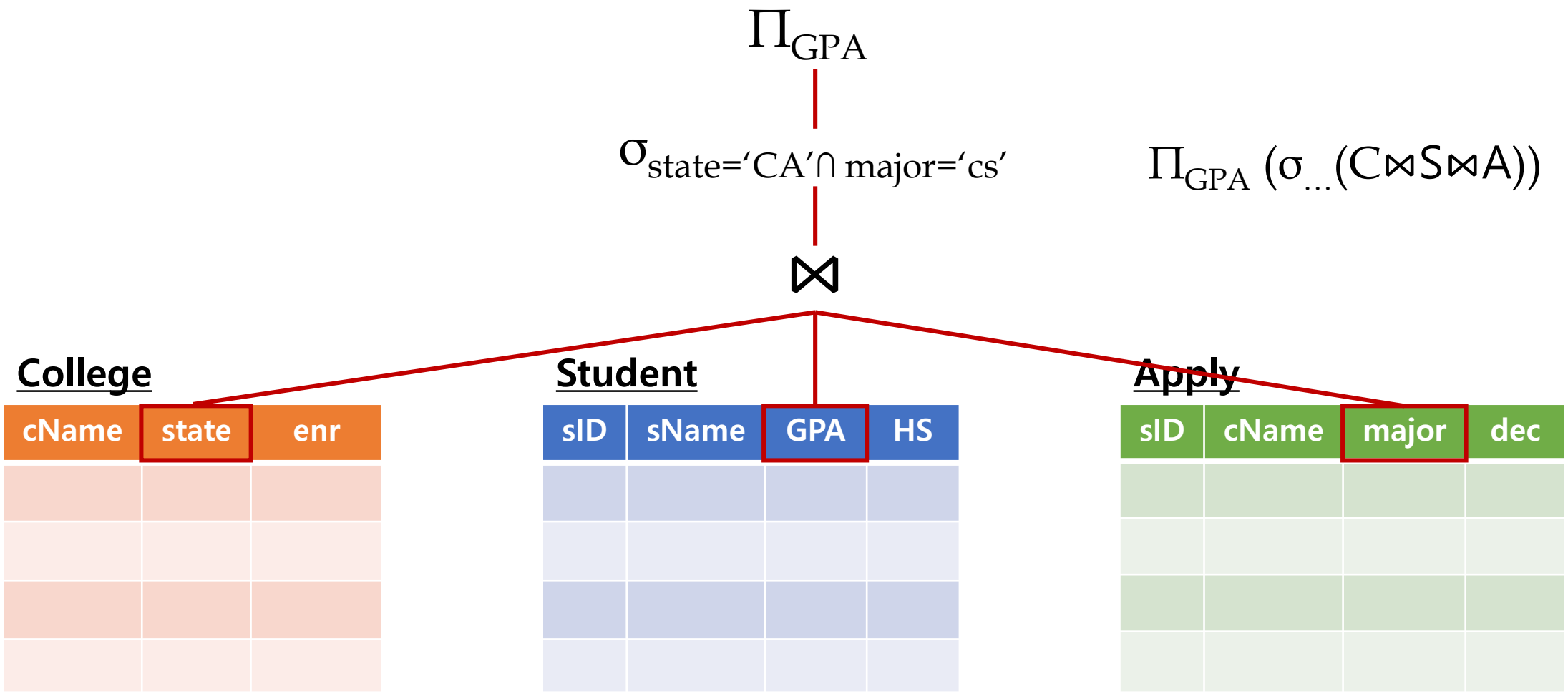
Alternate notation (2)

Expression tree – GPAs of students applying to CS in CA



Alternate notation (2)

Expression tree – GPAs of students applying to CS in CA



$\Pi_{GPA}\ (\sigma_{...}(C \bowtie S \bowtie A))$

Relational Algebra summary

Relational Algebra summary

Core

R

Relational Algebra summary

Core

- R
- $\sigma_c(E)$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Abbreviations

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Abbreviations

- $E_1 \bowtie E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Abbreviations

- $E_1 \bowtie E_2$
- $E_1 \bowtie_\Theta E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Abbreviations

- $E_1 \bowtie E_2$
- $E_1 \bowtie_\Theta E_2$
- $E_1 \cap E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Abbreviations

- $E_1 \bowtie E_2$
- $E_1 \bowtie_\Theta E_2$
- $E_1 \cap E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $(E_1) \times (E_2)$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Abbreviations

- $E_1 \bowtie E_2$
- $E_1 \bowtie_\Theta E_2$
- $E_1 \cap E_2$

Relational Algebra summary

Core

- R
- $\sigma_c(E)$
- $\Pi_{A1, \ldots, An}(E)$
- $(E_1) \times (E_2)$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A1, \ldots, An)}(E)$

Abbreviations

- $E_1 \bowtie E_2$
- $E_1 \bowtie_\Theta E_2$
- $E_1 \cap E_2$