Tools & Models for Data Science

Relational Algebra

Chris Jermaine & Risa Myers

Rice University

Relational Calculus vs. Algebra

- In Relational Calculus
  - You say what you want
  - And not how to compute it
- But obviously...
  - This needs to be compiled into an actual computational plan
  - And in relational DBs, the plan is expressed in relational algebra
- RA is the “abstract machine” of relational databases

What Is An Algebra?

- Many Definitions!
  - Simplest: it is a set (domain) with a number of operations
  - The domain is closed under those operations
- In RA...
  - The domain is the set of all valid relations
  - The set of operations includes $\pi, \sigma, \times, \bowtie, \cup, \cap, -$
- Now let's go through the operations!

Projection

- Projection removes attributes
- $\pi_A(R)$...
  - A is a set of attributes of relation R
  - This simply removes all attributes not in A from R
  - Note: cardinality of output can differ from R
  - Output is a relation

Projection Example

COURSE(<u>CRN</u>, NAME, DOW, STARTTIME, ENDTIME)

1. Return course names
   $\pi_{name}(\text{COURSE})$
   {('Comp430'), ('Comp140'), ...}
2. Return course name and days of the week when courses meet
   $\pi_{name,dow}(\text{COURSE})$
   {('Comp430', 'MWF'), ('Comp140', 'TR'), ...)}

Projection Visualization

COURSE

CRN	NAME	DOW	STARTTIME	ENDTIME
12809	COMP 430	MWF	14:00:00	14:50:00
12810	COMP 533	MWF	14:00:00	14:50:00
10396	COMP 140	TR	10:50:00	12:05:00
13970	COMP 436	WF	14:30:00	15:45:00

$\pi_{name,dow}(\text{COURSE})$

NAME	DOW
COMP 430	MWF
COMP 533	MWF
COMP 140	TR
COMP 436	WF

Selection

- Selection removes tuples
- $\sigma_B(R)$...
  - B is a boolean predicate that can be applied to a single tuple from R
  - This simply removes all tuples not accepted by B
  - Again: output is a relation

FREQUENTS

DRINKER	CAFE
Risa	JL
Risa	BH
Chris	BH
Chris	DT

$\sigma_{\text{DRINKER='Risa'}}(\text{FREQUENTS})$

DRINKER	CAFE
Risa	JL
Risa	BH

Selection Example

COURSE(CRN, NAME, DOW, STARTTIME, ENDTIME)

1. Which courses have name ‘Comp 533’?

   $\sigma_{name='\text{Comp 533}'}(\text{COURSE})$

   {(12810,Comp 533, MWF, 14:00:00, 14:50:00)}

2. Which courses meet for less than an hour at a time?

   $\sigma_{endTime-startTime\leq 1:00}(\text{COURSE})$

   {(12809,Comp 430, MWF, 14:00:00,14:50:00), (12810,Comp 533, MWF, 15:00:00,15:50:00),

   $\ldots$}

Selection Visualization

COURSE

CRN	NAME	DOW	STARTTIME	ENDTIME
12809	COMP 430	MWF	14:00:00	14:50:00
12810	COMP 533	MWF	14:00:00	14:50:00
10396	COMP 140	TR	10:50:00	12:05:00
13970	COMP 436	WF	14:30:00	15:45:00

$\sigma_{endTime - startTime \leq 1:00}(\text{COURSE})$

CRN	NAME	DOW	STARTTIME	ENDTIME
12809	COMP 430	MWF	14:00:00	14:50:00
12810	COMP 533	MWF	14:00:00	14:50:00
[redacted]				

Selection/Projection Example 1

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

? Query: Who likes ‘Cold Brew’ coffee?

Selection/Projection Example 1

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Query: Who likes 'Cold Brew' coffee?
  - $\pi_{\text{DRINKER}}(\sigma_{\text{COFFEE}=\text{'Cold Brew'}}(\text{LIKES}))$

Selection/Projection Example 2

COURSE(<u>CRN</u>, NAME, DOW, STARTTIME, ENDTIME)

- ? What are the name and days of the week for classes that meet at 1 PM?
- ? What are the names of courses that meet for less than an hour at a time?

Selection/Projection Example 2

COURSE(<u>CRN</u>, NAME, DOW, STARTTIME, ENDTIME)

- What are the name and days of the week for classes that meet at 1 PM?

  $\pi_{name,dow}(\sigma_{startTime=13:00:00}(COURSE))$

- What are the names of courses that meet for less than an hour at a time?

  $\pi_{name}(\sigma_{endTime-startTime<1:00}(COURSE))$

Rename ρ

$\rho_{A/B}(R)$

- Renames attribute B to A in relation R
- Output is a relation

$\rho_{S(A_1 \ldots A_n)}(R)$

- Renames relation R to S and renames all attributes as specified
- Output is a relation

Example

? Rename attribute ‘name’ in COURSE to ‘courseName’

- $\rho_{courseName/name}(\mathrm{COURSE})$

Assignment $\leftarrow$

X $\leftarrow$ (relational algebra statement)

- Assigns the relation to a temporary variable
- For convenience

Example

? Assign the courses that meet on Monday-Wednesday-Friday to the variable MWF

- MWF$\leftarrow (\sigma_{dow='MWF'}(\mathrm{COURSE}))$

Join: Cartesian Product

- Join combines tuples
- Simplest join is Cartesian product (aka: cross product)
- Used to match up tuples from different relations
- $R \times S$ is equivalent to

```
for r in R
  for s in S
    output r • s
```

- "•" is concatenation

What is the output cardinality?

A |R|

B |S|

C $|R| \times |S|$

D |R| + |S|

Cartesian Product Example

on
0
1

×

codeA	codeB
A	X
B	Y

×

color
red
blue
green

=

on	codeA	codeB	color
0	A	X	red
0	A	X	blue
0	A	X	green
0	B	Y	red
0	B	Y	blue
0	B	Y	green
1	A	X	red
1	A	X	blue
1	A	X	green
1	B	Y	red
1	B	Y	blue
1	B	Y	green

What is a Join?

- Concatenates attributes from one relation to another
- Returns a new relation
- Cartesian product/ cross product
  - Every possible pairing
- Natural or Theta joins
  - Based on predicates
- Left / Right Outer joins
  - All tuples from one relation and matching relations from the other

Join: Theta Join

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Often you want $\sigma_B(R \times S)$
- Shorthand for this is $R \bowtie_B S$
- ? Query: Who likes a coffee that ‘Risa’ likes?

Join: Theta Join

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Often you want $\sigma_B(R \times S)$
- Shorthand for this is $R \bowtie_B S$
- Query: Who likes a coffee that 'Risa' likes?
  - TEMP(d_1, c_1, d_2, c_2)
    $\leftarrow \text{LIKES} \bowtie_{\text{COFFEE}=\text{COFFEE}} (\sigma_{\text{DRINKER}=\text{'Risa'}}(\text{LIKES}))$
  - π_{d_1}(TEMP)

Theta Join Example

TEACHES(netId, crn, semester, year)

? What is the netid for people who have taught the same class in more than 1 year?

Theta Join Example

TEACHES(netId, crn, semester, year)

- What is the netid for people who have taught the same class in more than 1 year?
- $\pi_{netid}\left(\rho_{sem1(\ldots)}(\text{TEACHES}) \bowtie_{sem1.netId=sem2.netId \,\wedge\, sem1.crn=sem2.crn \,\wedge\, sem1.year \,<\, sem2.year} \rho_{sem2(\ldots)}(\text{TEACHES})\right)$

Theta Join Example

TEACHES(<u>netId</u>, <u>crn</u>, <u>semester</u>, <u>year</u>)

- What is the netid for people who have taught the same class in more than 1 year?
- $\pi_{netid}\left(\rho_{sem1(...)}(\text{TEACHES}) \bowtie_{sem1.netId=sem2.netId \ \wedge \ sem1.crn=sem2.crn \ \wedge \ sem1.year \ < \ sem2.year} \rho_{sem2(...)}(\text{TEACHES})\right)$
- Why "$sem1.year < sem2.year$"?

Toy Examples

- Sometimes, you need to try things out
- Pencil & paper
- Relations & data

Join: Natural Join

- Often you want to join two relations
  - Using an equality check on all attributes having the same name
  - Then project away redundant attributes
- Shorthand for this is $R * S$

Join: Natural Join Example

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

? Who goes to a cafe serving a coffee that they like?

Join: Natural Join Example

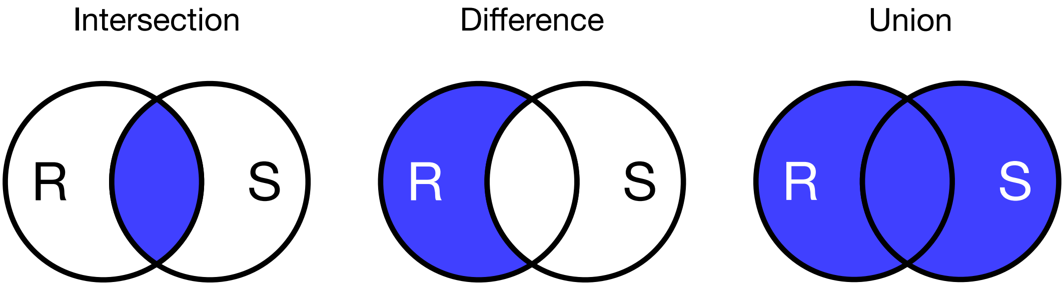
LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Who goes to a cafe serving a coffee that they like?
  - $\pi_{\text{DRINKER}}(\text{LIKES} * \text{FREQUENTS} * \text{SERVES})$

Set-Based Operations

- Can use standard set operations as well: $\cup, \cap, -$
  - Types and numbers of input attributes must match
  - By convention, attribute names come from LHS
  - $R \cup S$: all tuples in R or in S
  - $R \cap S$: all tuples in R and in S
  - $R - S$: all tuples in R and not in S

Set-Based Operations



? What is the maximum number of tuples in $R \cup S$?

- A |R|
- B |S|
- C $|R| \times |S|$
- D $|R| + |S|$

Union Example

FACULTY (NETID, LASTNAME, FIRSTNAME, HIREDATE, TERMDATE)
STUDENT (NETID, LASTNAME, FIRSTNAME)

? What are the names of all the people at Rice?

Union Example

FACULTY (<u>NETID</u>, LASTNAME, FIRSTNAME, HIREDATE, TERMDATE)
STUDENT (<u>NETID</u>, LASTNAME, FIRSTNAME)

- What are the names of all the people at Rice?
- $\pi_{lastname,firstname}(\text{STUDENT}) \cup \pi_{lastname,firstname}(\text{FACULTY})$

Union Example

FACULTY (<u>NETID</u>, LASTNAME, FIRSTNAME, HIREDATE, TERMDATE)
STUDENT (<u>NETID</u>, LASTNAME, FIRSTNAME)

- What are the names of all the people at Rice?
- $\pi_{lastname,firstname}(\text{STUDENT}) \cup \pi_{lastname,firstname}(\text{FACULTY})$

? Why do we project out lastname, firstname?

Intersection Example

FACULTY (<u>NETID</u>, LASTNAME, FIRSTNAME, HIREDATE, TERMDATE)
STUDENT (<u>NETID</u>, LASTNAME, FIRSTNAME)

? Who has been both a student and a faculty member?

Intersection Example

FACULTY (<u>NETID</u>, LASTNAME, FIRSTNAME, HIREDATE, TERMDATE)
STUDENT (<u>NETID</u>, LASTNAME, FIRSTNAME)

- Who has been both a student and a faculty member?
- $\pi_{lastname,firstname}(\text{STUDENT}) \cap \pi_{lastname,firstname}(\text{FACULTY})$

Difference Example

FACULTY (<u>NETID</u>, LASTNAME, FIRSTNAME, HIREDATE, TERMDATE)
STUDENT (<u>NETID</u>, LASTNAME, FIRSTNAME)

- $\pi_{lastname,firstname}(\text{FACULTY}) - \pi_{lastname,firstname}(\text{STUDENT})$

? What does this expression represent (in English)?

Difference Example

FACULTY (<u>NETID</u>, LASTNAME, FIRSTNAME, HIREDATE, TERMDATE)
STUDENT (<u>NETID</u>, LASTNAME, FIRSTNAME)

- $\pi_{lastname,firstname}(\text{FACULTY}) - \pi_{lastname,firstname}(\text{STUDENT})$
- Faculty who have never been students

Set-Based Operations

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

? Who does not like ‘Cold Brew’ coffee?

Set-Based Operations

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Who does not like 'Cold Brew' coffee?
  - $\mathrm{CBGOOD} \leftarrow \pi_{\mathrm{DRINKER}}(\sigma_{\mathrm{COFFEE}=\text{'Cold Brew'}}(\mathrm{LIKES}))$
  - $(\pi_{\mathrm{DRINKER}}(\mathrm{FREQUENTS})) - \mathrm{CBGOOD}$

? Why use FREQUENTS instead of LIKES?

More Set-Based Examples

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

1. Which cafes serve Cold Brew?
2. Who goes to cafes serving Cold Brew?
3. Who goes to a cafe that both Risa and Chris go to?
4. Who avoids Risa at all costs?

More Set-Based Examples

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

1. Which cafes serve Cold Brew?

   $\pi_{CAFE}(\sigma_{COFFEE='Cold\ Brew'}(\text{SERVES}))$

More Set-Based Examples

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

2 Who goes to cafes serving Cold Brew?

$$\pi_{DRINKER}(\sigma_{COFFEE='Cold\ Brew'}(\text{SERVES} * \text{FREQUENTS}))$$

More Set-Based Examples

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

3. Who goes to a cafe that both Risa and Chris go to?

$$C \leftarrow \pi_{CAFE}(\sigma_{DRINKER='Risa'}(\text{FREQUENTS})) \\ \cap\ \pi_{CAFE}(\sigma_{DRINKER='Chris'}(\text{FREQUENTS}))$$

$$\pi_{DRINKER}(\text{FREQUENTS} * C)$$

More Set-Based Examples

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

4 Who avoids Risa at all costs?

Want: Cafes each person goes to - cafes Risa goes to

$RCAFES \leftarrow \pi_{CAFE}(\sigma_{DRINKER='Risa'}(\text{FREQUENTS}))$

$\pi_{DRINKER}(\text{FREQUENTS}) - \pi_{DRINKER}(\text{FREQUENTS} * RCAFES)$

Complicated Set-Based Example

LIKES (DRINKER, COFFEE)
FREQUENTS (DRINKER, CAFE)
SERVES (CAFE, COFFEE)

- Who only goes to cafes where they can get a coffee they like?
  - Use ‘all people’ $-$ ‘those who go to a cafe where they can’t get a coffee they like’
  - $\text{ALLPEEPS} \leftarrow \pi_{\text{DRINKER}}(\text{FREQUENTS})$
  - How about ‘those who go to a cafe where they can’t get a coffee they like’?
  - Use FREQUENTS $-$ ‘DRINKER, CAFE combos where the person can get a coffee they like’
  - $\text{GOODCOFFEE} \leftarrow \pi_{\text{DRINKER,CAFE}}(\text{LIKES} * \text{SERVES})$
- Then the answer is
  - $\text{ALLPEEPS} - \pi_{\text{DRINKER}}(\text{FREQUENTS} - \text{GOODCOFFEE})$

Review

1. If R has 3 tuples and S has 2 tuples, R $\times$ S returns how many tuples?
   - A 2
   - B 3
   - C 6
   - D None of the above

True / False

2. The most commonly used type of join is Cartesian product
3. Attribute names must match to join relations on them
4. When you perform a join, the join column is always duplicated
5. Each relation may be used at most 1 time per query
6. Union and Intersection can operate on relations with different numbers or types of attributes

Wrap up

1. What is Relational Algebra?
2. Why does it matter?

? How can we use what we learned today?

? What do we know now that we didn't know before?