COMP 330/543: The Relational Algebra

Sinan Kockara

Luis Guzman

Chris Jermaine

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 π, σ, $\times$, $\bowtie$, $\cup$, $\cap$, $-$

Projection

Projection removes attributes

$\pi_A(R)$...

- A is a set of attributes of relation R
- This simply removes all atts not in A from R
- Note: cardinality of output can differ from R
- Output is a relation

COURSE

CRN	NAME	DOW	STARTTIME	ENDTIME
12809	COMP 430	MWF	14:00:00	14:50:00
16954	COMP 543	MWF	10:00:00	10:50:00
10396	COMP 140	TTH	10:50:00	12:05:00
13970	COMP 436	TTH	14:30:00	15:45:00

$\pi_{NAME,DOW}(COURSE)$

Answers the query
"what are the names and week days of the courses ?"

Projection

Projection removes attributes

$\pi_A(R)$...

- A is a set of attributes of relation R
- This simply removes all atts not in A from R
- Note: cardinality of output can differ from R
- Output is a relation

COURSE

CRN	NAME	DOW	STARTTIME	ENDTIME
12809	COMP 430	MWF	14:00:00	14:50:00
16954	COMP 543	MWF	10:00:00	10:50:00
10396	COMP 140	TTH	10:50:00	12:05:00
13970	COMP 436	TTH	14:30:00	15:45:00

$\pi_{NAME,DOW}(COURSE)$

Answers the query
"what are the names and week days
of the courses ?"

NAME	DOW
COMP 430	MWF
COMP 543	MWF
COMP 140	TTH
COMP 436	TTH

Selection

Selection removes tuples

$\sigma_B(R)$...

- B is a boolean pred that can be applied to a single tuple from R
- This simply removes all tuples not accepted by B
- Again: output is a relation

COURSE

CRN	NAME	DOW	STARTTIME	ENDTIME
12809	COMP 430	MWF	14:00:00	14:50:00
16954	COMP 543	MWF	10:00:00	10:50:00
10396	COMP 140	TTH	10:50:00	12:05:00
13970	COMP 436	TTH	14:30:00	15:45:00

$\sigma_{NAME="COMP\ 543"}(COURSE)$

Answers the query "which courses have name COMP 543?"

CRN	NAME	DOW	STARTTIME	ENDTIME
16954	COMP 543	MWF	10:00:00	10:50:00

Question

What is the query for:

“which courses meet for less than an hour at a time?”

Question

What is the query for:

"which courses meet for less than an hour at a time?"

$$\sigma_{ENDTIME-STARTTIME<1:00:00}(COURSE)$$

CRN	NAME	DOW	STARTTIME	ENDTIME
12809	COMP 430	MWF	14:00:00	14:50:00
16954	COMP 543	MWF	10:00:00	10:50:00

Selection/Projection Example

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Query: Who likes the beer ‘PBR’?

Selection/Projection Example

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Query: Who likes the beer ‘PBR’?

- $\pi_{\text{DRINKER}}(\sigma_{\text{BEER='PBR'}}(\text{LIKES}))$
- What is equivalent RC?

Selection/Projection Example

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Query: Who likes the beer ‘PBR’?

- $\pi_{\mathrm{DRINKER}}(\sigma_{\mathrm{BEER}=\text{‘PBR’}}(\mathrm{LIKES}))$
- What is equivalent RC?

  $\{ l.\mathrm{DRINKER} \mid l \in \mathrm{LIKES} \wedge l.\mathrm{BEER} = \text{‘PBR’} \}$

Rename

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

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

- $\rho_{S(A_1,A_2,\ldots,A_n)}(R)$

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

Assignment

- $X \leftarrow$ (RA statement)
- Assigns the output of an RA statement, which is a relation, to a <u>temporary</u> relation
  - For convenience

Example – assign the courses that meet on Monday-Wednesday-Friday to a temporary relation "MWF"

$$MWF \leftarrow \sigma_{dow="MWF"}(COURSE)$$

Join: Cartesian Product

Join combines tuples

Simplest join is Cartesian product (aka: cross product)

$R \times S$...

- Returns $r \cdot s$ forall $r \in R$, $s \in S$
- What is the output cardinality?

Join: Theta Join

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Often you want $\sigma_B(R \times S)$

Shorthand for this is $R \bowtie_B S$

Query: Who likes a beer that 'Luis' likes?

Join: Theta Join

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Often you want $\sigma_B(R \times S)$

Shorthand for this is $R \bowtie_B S$

Query: Who likes a beer that 'Luis' likes?

- $\text{TEMP}(d_1, b_1, d_2, b_2) \leftarrow \text{LIKES} \bowtie_{\text{BEER=BEER}} (\sigma_{\text{DRINKER='Luis'}}(\text{LIKES}))$
- $\pi_{d_1}(\text{TEMP})$

Example Walkthrough

LIKES X $\sigma_{DRINKER='Luis'}$(LIKES) =

Drinker	Beer
Luis	PBR
Chris	Corona
Chris	Budlight
Luis	Modelo
Chris	PBR

X

Drinker	Beer
Luis	PBR
Luis	Modelo

=

Drinker	Beer	Drinker	Beer
Luis	PBR	Luis	PBR
Luis	PBR	Luis	Modelo
Chris	Corona	Luis	PBR
Chris	Corona	Luis	Modelo
Chris	Budlight	Luis	PBR
Chris	Budlight	Luis	Modelo
Luis	Modelo	Luis	PBR
Luis	Modelo	Luis	Modelo
Chris	PBR	Luis	PBR
Chris	PBR	Luis	Modelo

Example Walkthrough

TEMP(d_1, b_1, d_2, b_2) $\leftarrow$ LIKES X $\sigma_{DRINKER='Luis'}$(LIKES)

TEMP

d1	b1	d2	b2
Luis	PBR	Luis	PBR
Luis	PBR	Luis	Modelo
Chris	Corona	Luis	PBR
Chris	Corona	Luis	Modelo
Chris	Budlight	Luis	PBR
Chris	Budlight	Luis	Modelo
Luis	Modelo	Luis	PBR
Luis	Modelo	Luis	Modelo
Chris	PBR	Luis	PBR
Chris	PBR	Luis	Modelo

$\sigma_{b1=b2'}$(TEMP)

d1	b1	d2	b2
Luis	PBR	Luis	PBR
Luis	Modelo	Luis	Modelo
Chris	PBR	Luis	PBR

Π_{d1}($\sigma_{b1=b2'}$(TEMP))

d1
Luis
Chris

$\sigma_{D1 != 'Luis'}$ (π_{d1}(TEMP))

d1
Chris

Join: Natural Join

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Often you want to join two relations

- Using an equality check on all atts having the same name
- Then project away redundant attributes

Shorthand for this $R * S$

Query: Who goes to a bar serving a beer that they like?

Natural Join Example

STUDENT

StudentID	CourseID	Instructor
S1	C101	Dr. Smith
S2	C102	Dr. Johnson
S3	C103	Dr. Brown

COURSE

CourseID	Instructor	CourseName
C101	Dr. Smith	Math
C102	Dr. Johnson	Physics

STUDENT * COURSE

StudentID	CourseID	Instructor	CourseName
S1	C101	Dr. Smith	Math
S2	C102	Dr. Johnson	Physics

Join: Natural Join

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Often you want to join two relations

- Using an equality check on all atts having the same name
- Then project away redundant attributes

Shorthand for this $R * S$

Query: Who goes to a bar serving a beer that they like?

- LIKES $*$ SERVES * FREQUENTS

Join: Natural Join

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Often you want to join two relations

- Using an equality check on all atts having the same name
- Then project away redundant attributes

Shorthand for this $R * S$

Query: Who goes to a bar serving a beer that they like?

- LIKES $*$ SERVES * FREQUENTS
- π_{DRINKER}(LIKES $*$ SERVES * FREQUENTS)

Natural join example

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

Query – who goes to a cafe serving cold brew ?
RC

RA

Natural join example

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

Query – who goes to a cafe serving cold brew ?
RC

$$\{\, f.DRINKER \mid f \in FREQ \wedge \exists(s)\ (s \in SERVES \wedge s.CAFE = f.CAFE \wedge s.COFFEE = \text{'Cold Brew'})\,\}$$

RA

Natural join example

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

Query – who goes to a cafe serving cold brew ?

RC

$$\{ f.DRINKER \mid f \in FREQ \wedge \exists(s)\ (s \in SERVES \wedge s.CAFE = f.CAFE \wedge s.COFFEE = \text{'Cold Brew'}) \}$$

RA

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

Set-Based Operations

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Can use standard set ops as well: $\cup$, $\cap$, $-$

- To use, 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 nor in S

Query: Who does not like the beer ‘PBR’?

Set-Based Operations

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Can use standard set ops as well: $\cup, \cap, -$

- To use, 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 nor in S

Query: Who does not like the beer ‘PBR’?

- PBRGOOD $\leftarrow \pi_{\text{DRINKER}}(\sigma_{\text{BEER}=\text{‘PBR’}}(\text{LIKES}))$
- $(\pi_{\text{DRINKER}}(\text{LIKES})) - \text{PBRGOOD}$

Complicated Set-Based Example

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Who only goes to bars where they can get a beer they like?

Complicated Set-Based Example

LIKES (DRINKER, BEER)

FREQUENTS (DRINKER, BAR)

SERVES (BAR, BEER)

Who only goes to bars where they can get a beer they like?

- Use 'all people' − 'those who go to a bar where they can't get a beer they like'
- ALLPEPES $\leftarrow \pi_{\text{DRINKER}}(\text{LIKES})$
- How about 'those who go to a bar where they can't get a beer they like'?
- Use FREQUENTS − 'DRINKER, BAR combos where the person can get a beer they like'
- GOODBEER $\leftarrow \pi_{\text{DRINKER,BAR}}(\text{LIKES} * \text{SERVES})$

Then the answer is

- ALLPEPES $- \pi_{\text{DRINKER}}(\text{FREQUENTS} - \text{GOODBEER})$

Questions?