COMP 330: The Relational Algebra

Chris Jermaine and Kia Teymourian

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

Now let's go thru the operations!

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

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

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’?

- $\pi_{\text{DRINKER}}(\sigma_{\text{BAR='PBR'}}(\text{LIKES}))$

Join: Cartesian Product

Join combines tuples

Simplest join is Cartesian product (aka: cross product)

$R \times S$...

- Returns $r \bullet s$ 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 'Chris' 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 'Chris' likes?

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

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 beet that they like?

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 beet that they like?

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

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'?

- $\text{PBRGOOD} \leftarrow \pi_{\text{DRINKER}}(\sigma_{\text{BAR}=\text{'PBR'}}(\text{LIKES}))$
- $(\pi_{\text{DRINKER}}(\text{LIKES})) - \text{PBRGOOD}$

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'
- $\text{ALLPEPES} \leftarrow \pi_{\text{DRINKER}}(\text{LIKES})$
- 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't get a beer they like'
- $\text{ALLCOMBOS} \leftarrow \pi_{\text{DRINKER}}(\text{LIKES}) \times \pi_{\text{BAR}}(\text{SERVES})$
- $\text{NOGOODBEER} \leftarrow \text{ALLCOMBOS} - \pi_{\text{DRINKER,BAR}}(\text{LIKES} * \text{SERVES})$

Then the answer is

- $\text{ALLPEPES} - (\text{FREQUENTS} - (\text{ALLCOMBOS} - \text{NOGOODBEER}))$

Questions?