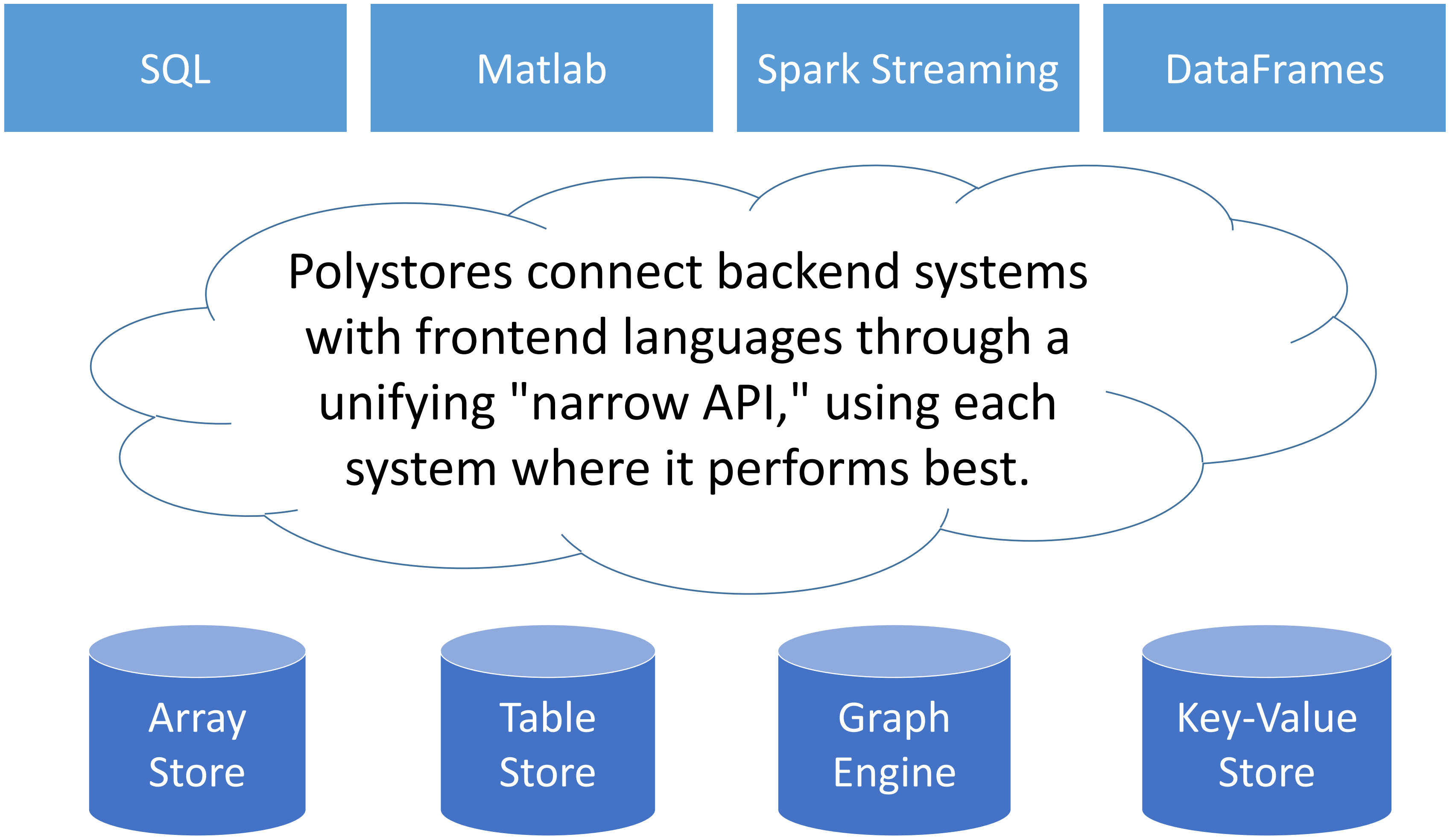
# LARA: A Language of Linear and Relational Algebra for Polystores

Dylan Hutchison

advised by Bill Howe, Dan Suciu

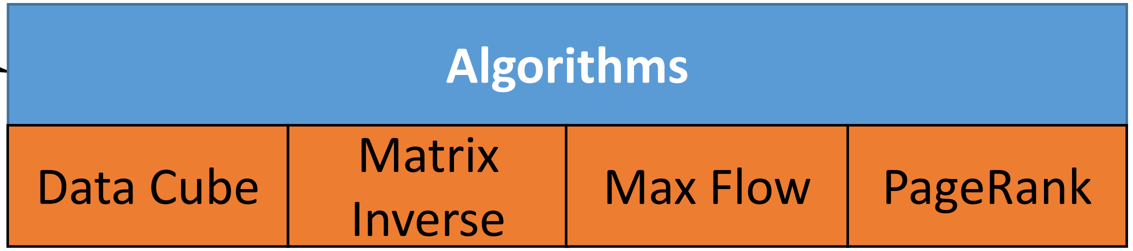
- Work in Progress -

# Polystores

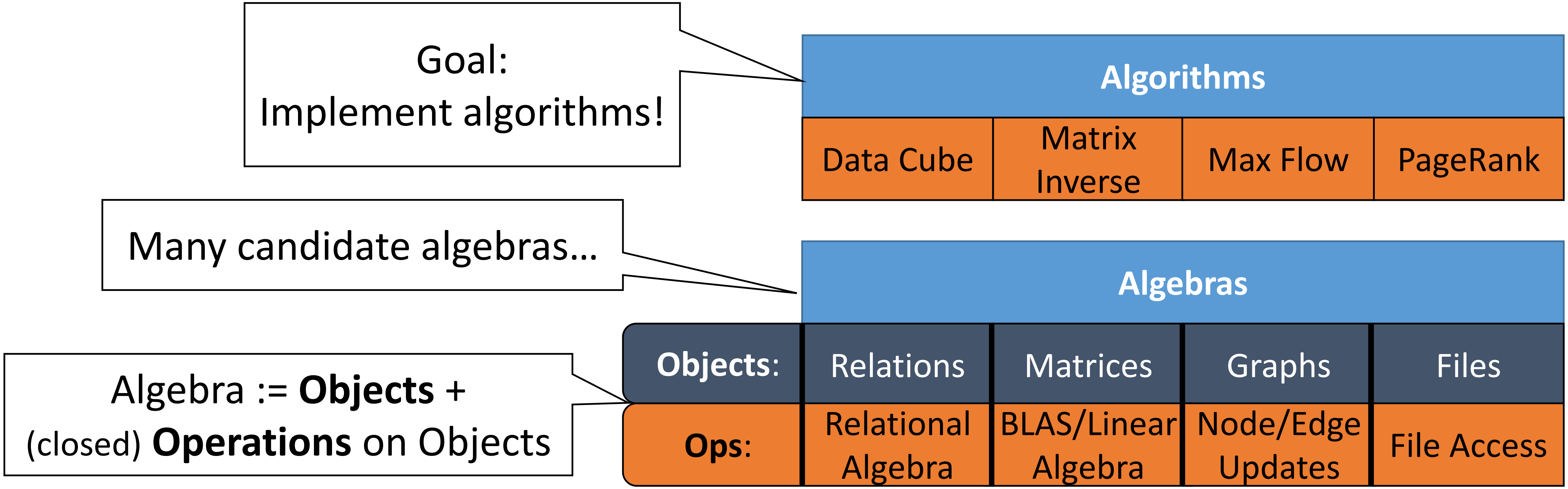

SQL
Matlab
Spark Streaming
DataFrames
Polystores connect backend systems with frontend languages through a unifying "narrow API," using each system where it performs best.
Array Store
Table Store
Graph Engine
Key-Value Store

# How to choose an algebra?

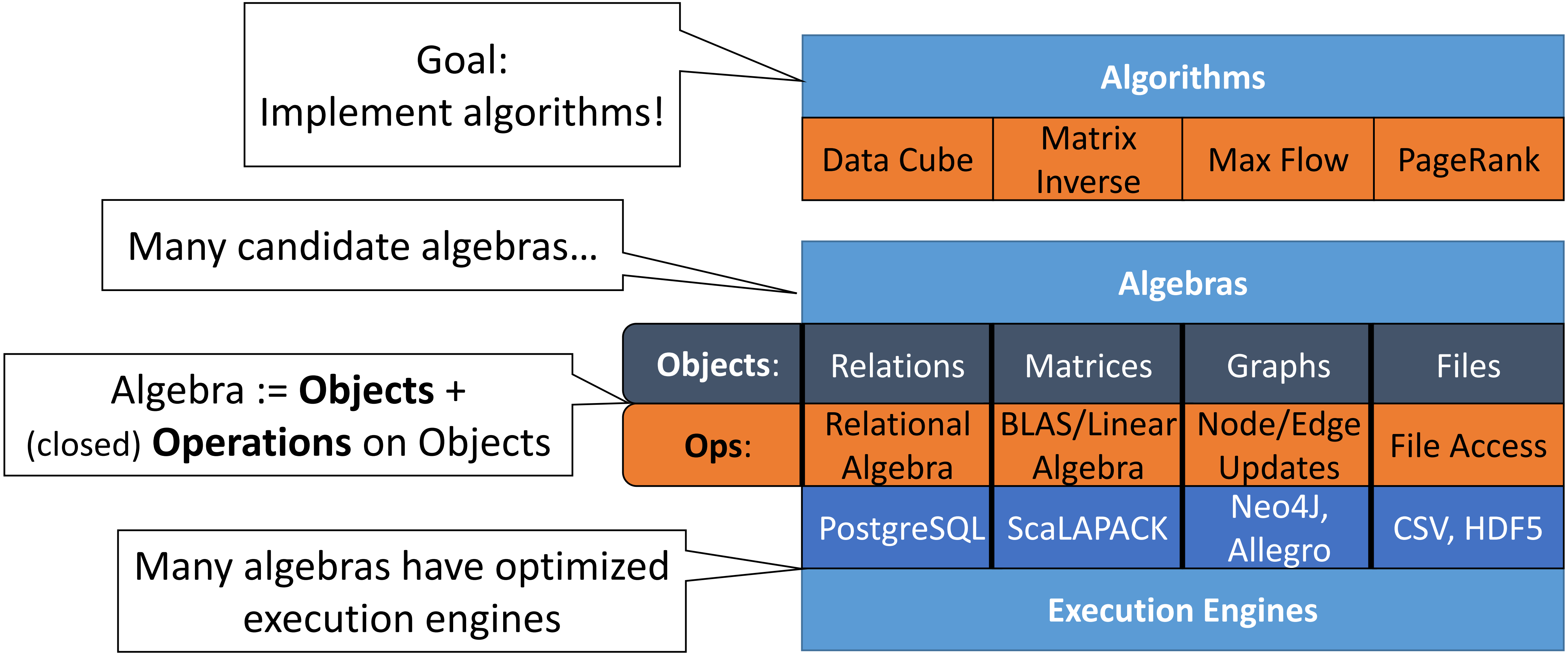
Goal:
Implement algorithms!



# How to choose an algebra?

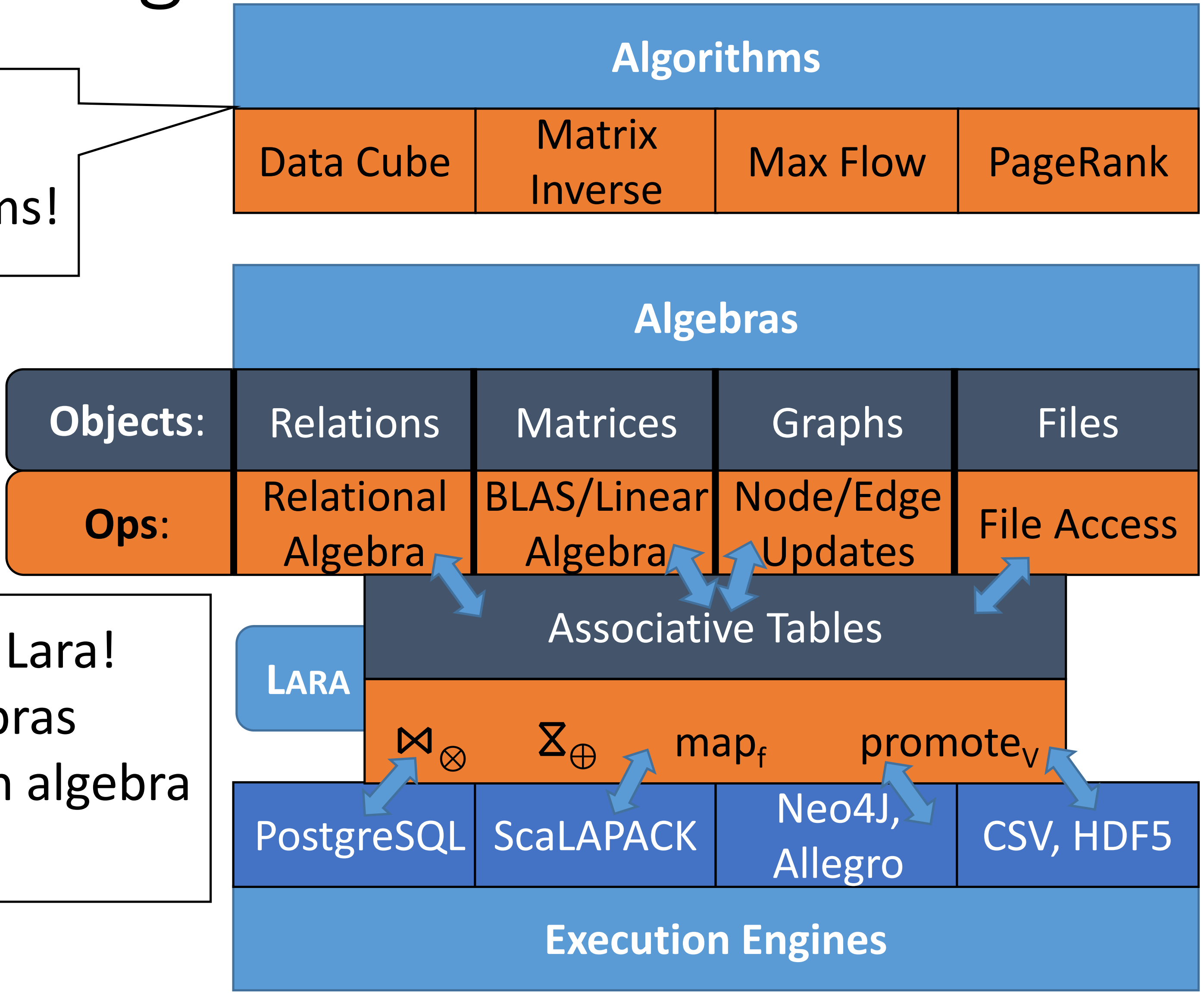


# How to choose an algebra?



# How to choose an algebra?

Goal:
Implement algorithms!




Answer: No choice necessary. Use Lara!

1. Write algorithm in any/all algebras
2. Translate to/from Lara common algebra
3. Use any/all execution engines

# Operations of Lara

- $\bowtie_{\otimes}$ – Join: horizontally merge columns,
  select equal colliding keys, multiply colliding values
- $\Join_{\oplus}$ – Union: vertically merge columns,
  group by colliding keys, sum colliding values
- $\text{map}_f$ – Map keys and old values to new values
- $\text{promote}_V$ – Promote values to keys

# Example: Ranking a Search

Suppose a user enters the search term "green delicious", as in input Q.
Database D scoring sites with search term relevance. Table W weighs words by importance.
Goal: Compute ranks of sites in D for search query Q, weighing by W

D

| site | word | score |
|---|---|---|
| pizzanow.com | pizza | 6 |
| pizzanow.com | delicious | 5 |
| allrecipes.com | delicious | 2 |
| allrecipes.com | green | 2 |
| allrecipes.com | potatoes | 5 |
| recycle.org | green | 2 |
| (others) | | 0 |

Q

| word | score |
|---|---|
| delicious | 1 |
| green | 1 |
| (others) | 0 |

W

| word | score |
|---|---|
| delicious | 1 |
| pizza | 1 |
| potatoes | 3 |
| green | 2 |
| (others) | 0 |

<u>Desired Output</u>

| site | score |
|---|---|
| pizzanow.com | 1*5*1 = 5 |
| allrecipes.com | 1*2*1+1*2*2 = 6 |
| recycle.org | 1*2*2 = 4 |
| (others) | 0 |

# Example: Ranking a Search

Suppose a user enters the search term "green delicious", as in input Q.
Database D scoring sites with search term relevance.  Table W weighs words by importance.
Goal: Compute ranks of sites in D for search query Q, weighing by W

RA: $\gamma_{site,\ +(score)}(\pi_{site,\ word,\ (score*score')\ as\ score}(\pi_{word}(Q) \bowtie D \bowtie \rho_{score \rightarrow score'}(W)))$

D

| site | word | score |
|---|---|---|
| pizzanow.com | pizza | 6 |
| pizzanow.com | delicious | 5 |
| allrecipes.com | delicious | 2 |
| allrecipes.com | green | 2 |
| allrecipes.com | potatoes | 5 |
| recycle.org | green | 2 |
| (others) | | 0 |

Q

| word | score |
|---|---|
| delicious | 1 |
| green | 1 |
| (others) | 0 |

W

| word | score |
|---|---|
| delicious | 1 |
| pizza | 1 |
| potatoes | 3 |
| green | 2 |
| (others) | 0 |

<u>Desired Output</u>

| site | score |
|---|---|
| pizzanow.com | 1*5*1 = 5 |
| allrecipes.com | 1*2*1+1*2*2 = 6 |
| recycle.org | 1*2*2 = 4 |
| (others) | 0 |

# Example: Ranking a Search

Suppose a user enters the search term "green delicious", as in input Q.
Database D scoring sites with search term relevance. Table W weighs words by importance.
Goal: Compute ranks of sites in D for search query Q, weighing by W

RA: $\gamma_{site,\ +(score)}(\pi_{site,\ word,\ (score*score')\ as\ score}(\pi_{word}(Q) \bowtie D \bowtie \rho_{score \rightarrow score'}(W)))$

*(Matlab)* LA: diag(Q) +.* D +.* W

D

| site | word | score |
|---|---|---|
| pizzanow.com | pizza | 6 |
| pizzanow.com | delicious | 5 |
| allrecipes.com | delicious | 2 |
| allrecipes.com | green | 2 |
| allrecipes.com | potatoes | 5 |
| recycle.org | green | 2 |
| (others) | | 0 |

Q

| word | score |
|---|---|
| delicious | 1 |
| green | 1 |
| (others) | 0 |

W

| word | score |
|---|---|
| delicious | 1 |
| pizza | 1 |
| potatoes | 3 |
| green | 2 |
| (others) | 0 |

<u>Desired Output</u>

| site | score |
|---|---|
| pizzanow.com | 1*5*1 = 5 |
| allrecipes.com | 1*2*1+1*2*2 = 6 |
| recycle.org | 1*2*2 = 4 |
| (others) | 0 |

# Example: Ranking a Search

Suppose a user enters the search term "green delicious", as in input Q.
Database D scoring sites with search term relevance. Table W weighs words by importance.
Goal: Compute ranks of sites in D for search query Q, weighing by W

RA: $\gamma_{site,\ +(score)}(\pi_{site,\ word,\ (score*score')\ as\ score}(\pi_{word}(Q) \bowtie D \bowtie \rho_{score \rightarrow score'}(W)))$

*(Matlab)* LA: diag(Q) +.* D +.* W

Hybrid: $\pi_{word}(Q) \bowtie$ D +.* W

D

| site | word | score |
|---|---|---|
| pizzanow.com | pizza | 6 |
| pizzanow.com | delicious | 5 |
| allrecipes.com | delicious | 2 |
| allrecipes.com | green | 2 |
| allrecipes.com | potatoes | 5 |
| recycle.org | green | 2 |
| (others) | | 0 |

Q

| word | score |
|---|---|
| delicious | 1 |
| green | 1 |
| (others) | 0 |

W

| word | score |
|---|---|
| delicious | 1 |
| pizza | 1 |
| potatoes | 3 |
| green | 2 |
| (others) | 0 |

Desired Output

| site | score |
|---|---|
| pizzanow.com | 1*5*1 = 5 |
| allrecipes.com | 1*2*1+1*2*2 = 6 |
| recycle.org | 1*2*2 = 4 |
| (others) | 0 |

# Example: Ranking a Search

Suppose a user enters the search term "green delicious", as in input Q.
Database D scoring sites with search term relevance. Table W weighs words by importance.
Goal: Compute ranks of sites in D for search query Q, weighing by W

| | | |
|---|---|---|
| RA: | | $\gamma_{\text{site}, +(\text{score})}(\pi_{\text{site, word, (score*score') as score}}(\pi_{\text{word}}(Q) \bowtie D \bowtie \rho_{\text{score}\rightarrow\text{score'}}(W)))$ |
| *(Matlab)* | LA: | diag(Q) +.* D +.* W |
| | Hybrid: | $\pi_{\text{word}}(Q) \bowtie$ D +.* W |
| | LARA: | $(Q \bowtie_* D \bowtie_* W)\ ⧖_+ E_{\text{site}}$ |

D

| site | word | score |
|---|---|---|
| pizzanow.com | pizza | 6 |
| pizzanow.com | delicious | 5 |
| allrecipes.com | delicious | 2 |
| allrecipes.com | green | 2 |
| allrecipes.com | potatoes | 5 |
| recycle.org | green | 2 |
| (others) | | 0 |

Q

| word | score |
|---|---|
| delicious | 1 |
| green | 1 |
| (others) | 0 |

W

| word | score |
|---|---|
| delicious | 1 |
| pizza | 1 |
| potatoes | 3 |
| green | 2 |
| (others) | 0 |

<u>Desired Output</u>

| site | score |
|---|---|
| pizzanow.com | 1*5*1 = 5 |
| allrecipes.com | 1*2*1+1*2*2 = 6 |
| recycle.org | 1*2*2 = 4 |
| (others) | 0 |

# Example: Ranking a Search

Executes on both RDBMS and BLAS, depending on cost model

Suppose a user enters the search term "green delicious", as in input Q.
Database D scoring sites with search term relevance. Table W weighs words by importance.
Goal: Compute ranks of sites in D for search query Q, weighing by W

| | |
|---|---|
| RA: | $\gamma_{\text{site, +(score)}}(\pi_{\text{site, word, (score*score') as score}}(\pi_{\text{word}}(Q) \bowtie D \bowtie \rho_{\text{score}\rightarrow\text{score'}}(W)))$ |
| LA: *(Matlab)* | diag(Q) +.* D +.* W |
| Hybrid: | $\pi_{\text{word}}(Q) \bowtie$ D +.* W |
| LARA: | $(Q \bowtie_* D \bowtie_* W)$ ⧖$_+$ $E_{\text{site}}$ |

Many ways to express algorithms. Lara presents an economical algebra preserving
- LA's familiar math, numerical prowess
- RA's flexibility, scale-out optimization

D

| site | word | score |
|---|---|---|
| pizzanow.com | pizza | 6 |
| pizzanow.com | delicious | 5 |
| allrecipes.com | delicious | 2 |
| allrecipes.com | green | 2 |
| allrecipes.com | potatoes | 5 |
| recycle.org | green | 2 |
| (others) | | 0 |

Q

| word | score |
|---|---|
| delicious | 1 |
| green | 1 |
| (others) | 0 |

W

| word | score |
|---|---|
| delicious | 1 |
| pizza | 1 |
| potatoes | 3 |
| green | 2 |
| (others) | 0 |

<u>Desired Output</u>

| site | score |
|---|---|
| pizzanow.com | 1*5*1 = 5 |
| allrecipes.com | 1*2*1+1*2*2 = 6 |
| recycle.org | 1*2*2 = 4 |
| (others) | 0 |

# LARA: A Unifying Algebra

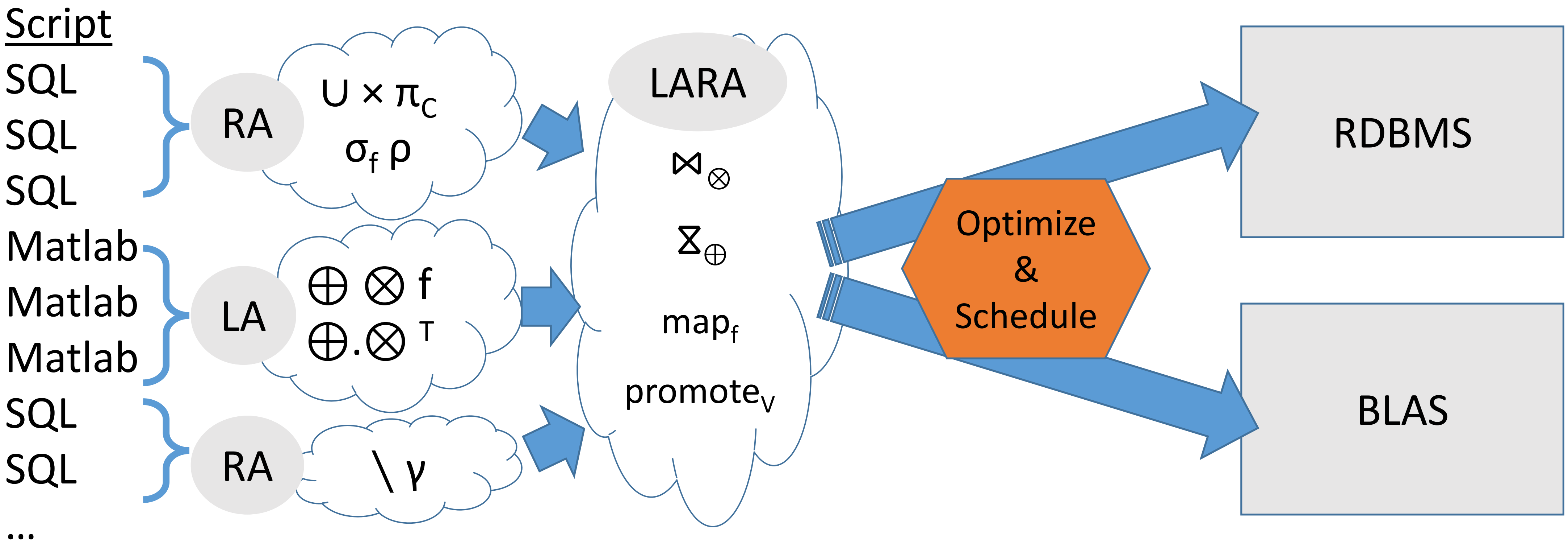
Do you have an application more easily expressed in several algebras?

Do you seek multi-system optimizations?

Let's discuss!

☺

# Vision for Polystore Systems



# APIs of RA and LA

Relational Algebra

Object: **Relation**

- $\cup$ – Union
- $\times$ – Cartesian Product
- $\pi_C$ – (Extended) Projection
- $\sigma_f$ – Select
- $\rho$ – Rename
- $\setminus$ – Difference
- $\gamma$ – Aggregate

Linear Algebra

Object: **N-D Matrix**

- $\oplus$ – Element-wise add
- $\otimes$ – Element-wise multiply
- $\oplus.\otimes$ – Matrix multiply
- Reduce – Sum along a dimension
- Apply function to each element
- $^T$ – Transpose
- (Construction & De-construction)

# Objects of Lara

**Associative Tables**. Several interpretations:

- Relational table with key columns & value columns with default values
- Total function from key-space to value-space
- Sparse tensor

| | [white] | [0] |
|---|---|---|
| pid | color | wgt |
| p01 | blue | 3 |
| p02 | red | 4 |
| p04 | blue | 2 |

(a) 'Part' table

| | [unknown] | [WA] |
|---|---|---|
| sid | fav | state |
| s01 | blue | WA |
| s02 | red | NJ |
| s04 | blue | NJ |

(b) 'Supplier' table

| | | [0] | [n] |
|---|---|---|---|
| sid | pid | qty | urgent |
| s01 | p02 | 3 | n |
| s02 | p03 | 1 | n |

(c) 'Request' table

# Lara -> RA & LA

| Lara | RA | LA |
| --- | --- | --- |
| $\bowtie_{\otimes}$ | $\bowtie$, $\pi_{\otimes}$, $\rho$ | Tensor product |
| $⧖_{\oplus}$ | $\gamma_{\oplus}$, $\cup$ | Reduce, e-wise sum |
| $map_f$ | $\pi_f$ | Apply |
| $promote_V$ | Re-index | Re-key |

# Example derived operation: Outer Join

| cid | pid | [white] color |
|---|---|---|
| M | p01 | blue |
| T | p01 | red |
| M | p02 | green |
| W | p01 | yellow |

(a) $P$

| cid | pid | sid | [white] color |
|---|---|---|---|
| M | p01 | s01 | blue |
| M | p01 | s02 | blue |
| M | p02 | s01 | green |
| M | p02 | s02 | green |
| T | p01 | s01 | red |
| T | p01 | s02 | red |
| W | p01 | s01 | yellow |
| W | p01 | s02 | yellow |

(b) $P \bowtie (\pi_{()}(S) \times E_{K_S \setminus K_P})$

| cid | sid | [GA] state |
|---|---|---|
| M | s01 | WA |
| M | s02 | NJ |
| T | s02 | DE |
| F | s01 | CA |

(c) $S$

| cid | pid | sid | [GA] state |
|---|---|---|---|
| M | p01 | s01 | WA |
| M | p01 | s02 | NJ |
| M | p02 | s01 | WA |
| M | p02 | s02 | NJ |
| T | p01 | s02 | DE |
| T | p02 | s02 | DE |
| F | p01 | s01 | CA |
| F | p02 | s01 | CA |

(d) $S \bowtie (\pi_{()}(P) \times E_{K_P \setminus K_S})$

Inner Join
$P \bowtie S$

$P ⟗ S$

| cid | pid | sid | [white] color | [GA] state |
|---|---|---|---|---|
| M | p01 | s01 | blue | WA |
| M | p01 | s02 | blue | NJ |
| M | p02 | s01 | green | WA |
| M | p02 | s02 | green | NJ |
| T | p01 | s02 | red | DE |
| T | p01 | s01 | red | (GA) |
| T | p02 | s02 | (white) | DE |
| W | p01 | s01 | yellow | (GA) |
| W | p01 | s02 | yellow | (GA) |
| F | p01 | s01 | (white) | CA |
| F | p02 | s01 | (white) | CA |

*(formulas out of date)* → (e) $(P \bowtie (\pi_{()}(S) \times E_{K_S \setminus K_P})) \times (S \bowtie (\pi_{()}(P) \times E_{K_P \setminus K_S}))$

Figure 5: Example of outer join of $P$ with $S$