Optimization: Rel. Alg. Equivalencies

R&G Chapter 14,15

(slides adapted from content by J.Gehrke, J.Shanmugasundaram, and/or C.Koch)

Last Class for Project Questions!

Project 1 due by 12:01 AM, Monday Morning
(between Sunday and Monday)

(Are there any groups of 2 people looking for a 3rd?)

Recap: Indexes

- Tree Indexes: ISAM, B+ Tree
  - Sort data, track page boundary values.
  - If over 1 page required for boundaries, recur.
- Hash Indexes: Static, Extendible, Linear
  - Use hash-fn to assign data to buckets
  - Buckets overflow, so use resizable hash-fns.

Recap: Indexes

$$\sigma_{(R.A>1)\wedge(R.B<R.C)}$$

$$\texttt{FileScan}(R)$$

$$\sigma_{(R.B<R.C)}$$

$$\texttt{IndexScan}(R, R.A, Range(1, \infty))$$

<u>Using Indexes 1</u>

Selection Predicate + File Scan = *Potential* Index Scan

Recap: Indexes

$\bowtie_{R.A<S.B}$

$\texttt{FileScan}(R)$ $\texttt{FileScan}(S)$

$\Rightarrow$

$\texttt{IndexNLJ}(S, S.B, Range(R.A, \infty))$

$\texttt{FileScan}(R)$

<u>Using Indexes 2</u>

Join + 2xFile Scan = *Potential* Index Nested Loop

Index-Nested Loop

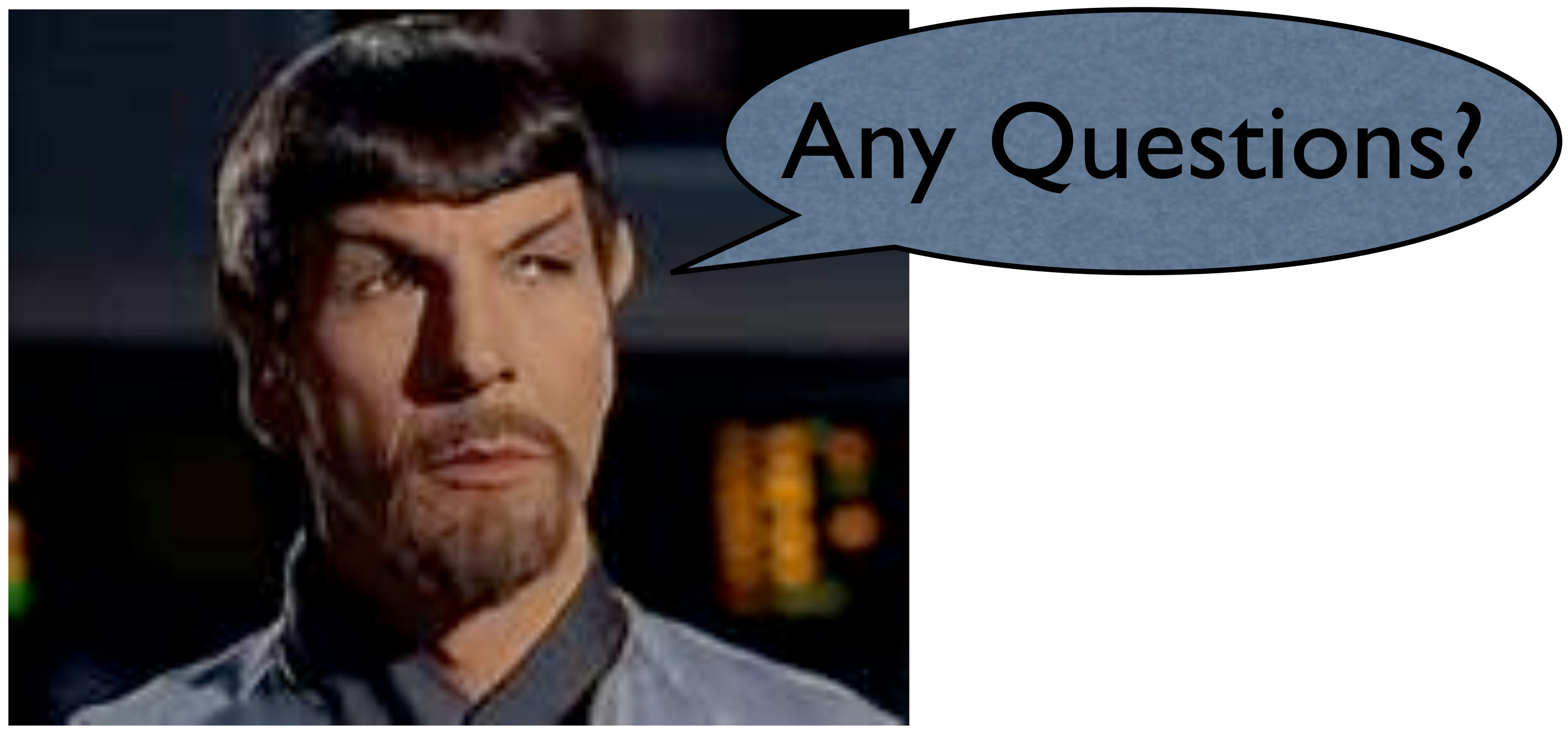
$$\texttt{IndexNLJ}(S, S.B, Range(R.A, \infty))$$

$$\texttt{FileScan}(R)$$

$\texttt{foreach}\ \langle A, \ldots \rangle \in R$

$\quad \texttt{foreach}\ \langle B, \ldots \rangle \in \texttt{IndexScan}(S, S.B, Range(A, \infty))$

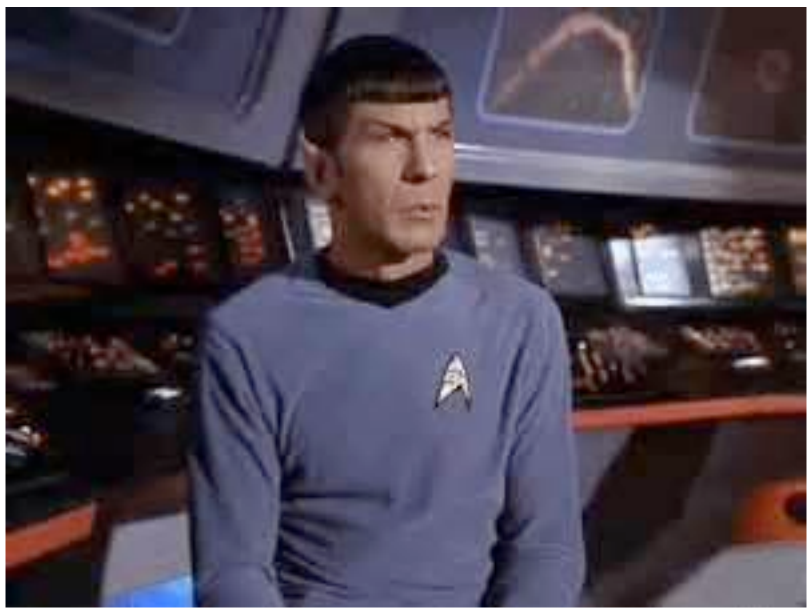
$\quad\quad \texttt{emit}\langle A, \ldots, B, \ldots \rangle$


Any Questions?

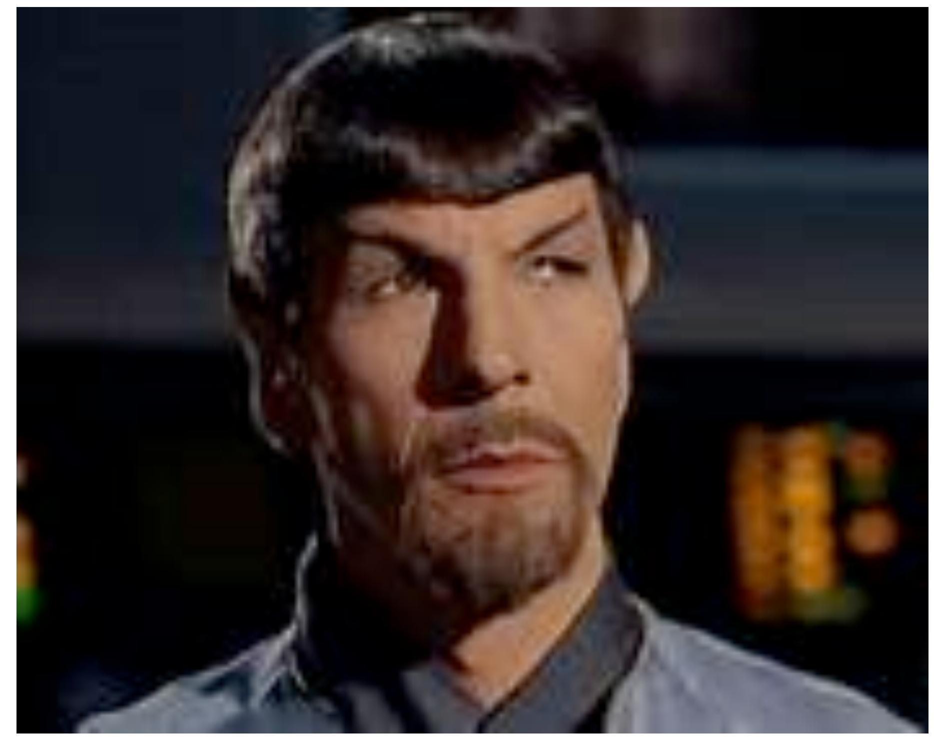
Optimization

- Rewriting Queries into Equivalent Forms
- Some rewrites are **always** good ideas
    - Pushing down selections
    - Pushing down projections
- Some rewrites **might** be good ideas
    - Join reordering
- How do we get to these equivalent forms?

RA Equivalencies



$\neq$



(No Beard, Good) (Beard, Evil)

RA Equivalencies

Selection

$$\sigma_{c_1 \wedge \ldots \wedge c_n}(R) \equiv \sigma_{c_1}(\ldots(\sigma_{c_n}(R))) \quad \text{(Combinable)}$$

$$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R)) \quad \text{(Commutative)}$$

Projection

$$\pi_{a_1} \equiv \pi_{a_1}(\ldots(\pi_{a_n}(R))) \quad \text{(Combinable)}$$

Cross Product (and Join)

$$R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T \quad \text{(Associative)}$$

$$(R \bowtie S) \equiv (S \bowtie R) \quad \text{(Commutative)}$$

RA Equivalencies

Selection

$$\sigma_{c_1 \wedge \ldots \wedge c_n}(R) \equiv \sigma_{c_1}(\ldots(\sigma_{c_n}(R)))$$ (Combinable)

$$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R))$$ (Commutative)

Projection

$$\pi_{a_1} \equiv \pi_{a_1}(\ldots(\pi_{a_n}(R)))$$ (Combinable)

Cross Product (and Join)

$$R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T$$ (Associative)

$$(R \bowtie S) \equiv (S \bowtie R)$$ (Commutative)

Try It: Show that $R \bowtie (S \bowtie T) \equiv (T \bowtie R) \bowtie S$

RA Equivalencies

Selection

$$\sigma_{c_1 \wedge \ldots \wedge c_n}(R) \equiv \sigma_{c_1}(\ldots(\sigma_{c_n}(R))) \quad \text{(Combinable)}$$

$$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R)) \quad \text{(Commutative)}$$

Projection

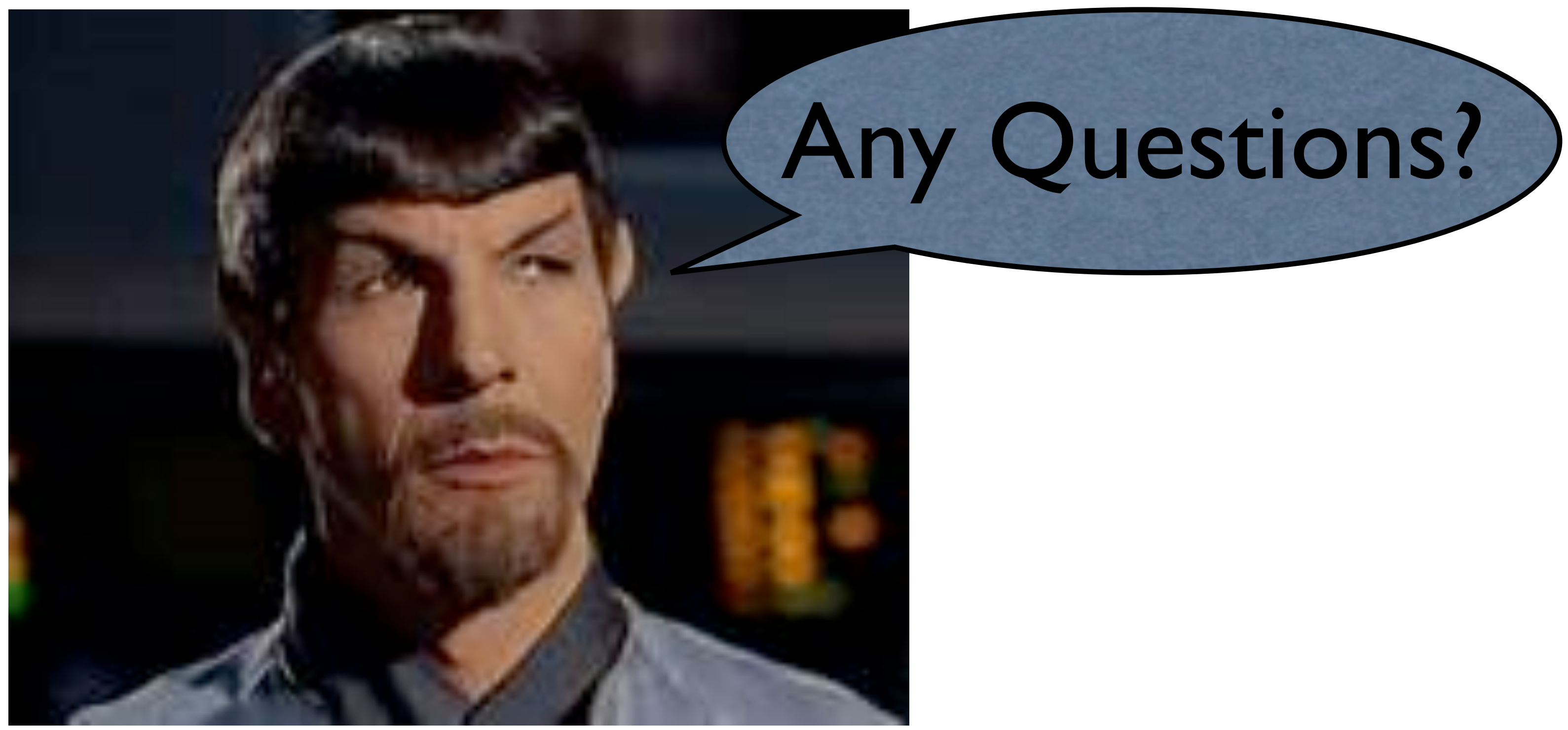
$$\pi_{a_1} \equiv \pi_{a_1}(\ldots(\pi_{a_n}(R))) \quad \text{(Combinable)}$$

Cross Product (and Join)

$$R \bowtie (S \bowtie T) \equiv (R \bowtie S) \bowtie T \quad \text{(Associative)}$$

$$(R \bowtie S) \equiv (S \bowtie R) \quad \text{(Commutative)}$$

These are general equivalencies.
We will use them to prove that our rewrites are correct.


Any Questions?

RA Equivalencies

$$\pi_a(\sigma_c(R)) \equiv \sigma_c(\pi_a(R))$$

Selection <u>commutes</u> with Projection

(but only if attribute set **a** and condition **c** are *compatible*)

a must include all columns referenced by **c**

RA Equivalencies

$$\pi_a(\sigma_c(R)) \equiv \sigma_c(\pi_a(R))$$

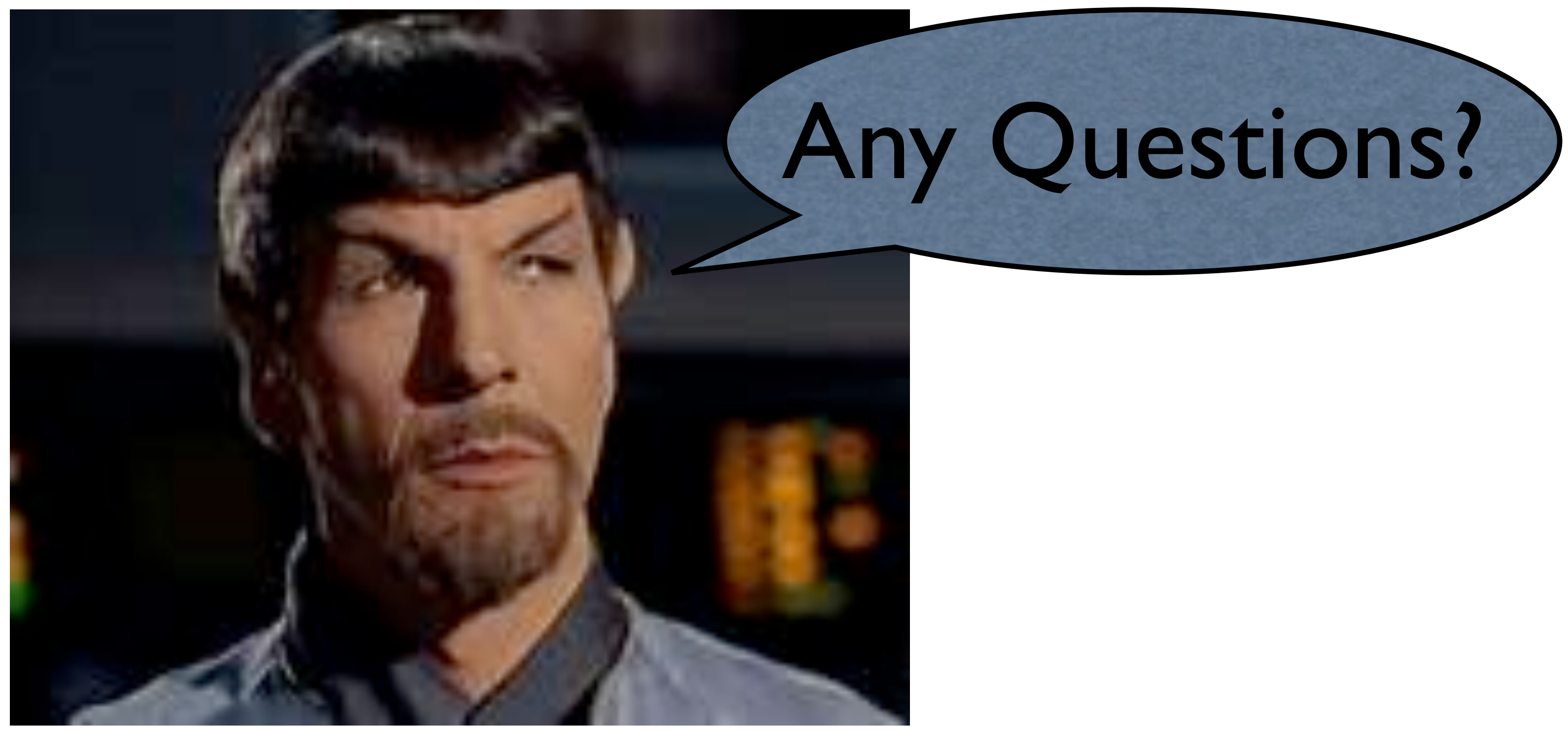
Selection <u>commutes</u> with Projection

(but only if attribute set **a** and condition **c** are *compatible*)

a must include all columns referenced by **c**

<u>Show that</u>

$$\pi_a(\sigma_c(R)) \equiv \pi_a(\sigma_c(\pi_{a \cup \texttt{cols(c)}}(R)))$$


Any Questions?

RA Equivalencies

$$\sigma_c(R \times S) \equiv R \bowtie_c S$$

Selection combines with Cross Product
to form a Join as per the definition of Join
(Note: This only helps if we have a join algorithm for conditions like **c**)

RA Equivalencies

$$\sigma_c(R \times S) \equiv R \bowtie_c S$$

Selection <u>combines</u> with Cross Product
to form a Join as per the definition of Join
(Note: This only helps if we have a join algorithm for conditions like **c**)

<u>Show that</u>

$$\sigma_{R.B=S.B \wedge R.A>3}(R \times S) \equiv \sigma_{R.A>3}(R \bowtie_{R.B=S.B} S)$$

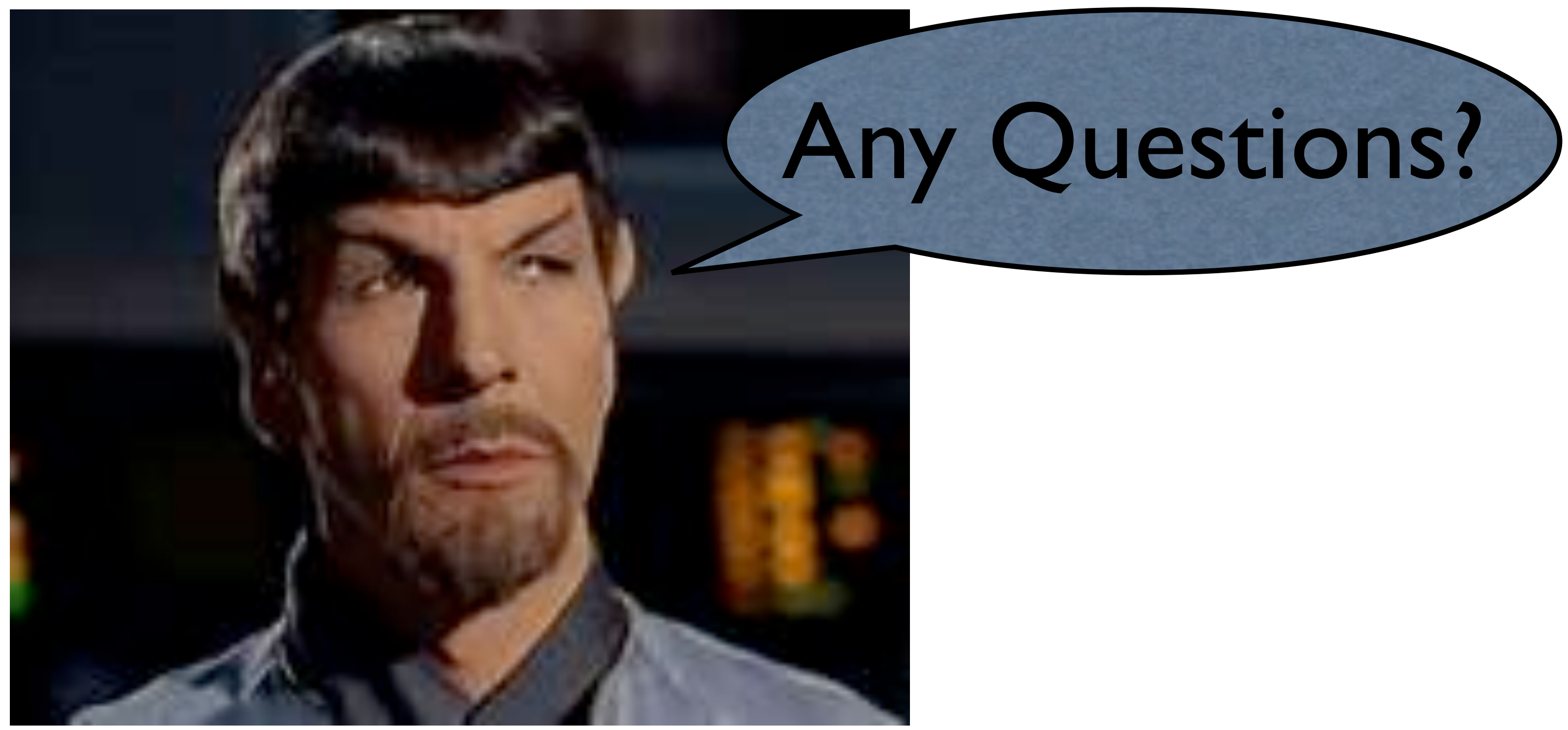


Image copyright: Paramount Pictures

RA Equivalencies

$$\sigma_c(R \times S) \equiv (\sigma_c(R)) \times S$$

Selection commutes with Cross Product

(but only if condition **c** references attributes of R exclusively)

The last example demonstrates pushing down selection predicates. This is something that we ALWAYS want to do wherever possible.

RA Equivalencies

$$\sigma_c(R \times S) \equiv (\sigma_c(R)) \times S$$

Selection commutes with Cross Product

(but only if condition **c** references attributes of R exclusively)

Show that

$$\sigma_c(R \bowtie_{c'} S) \equiv (\sigma_c(R)) \bowtie_{c'} S$$

The last example demonstrates pushing down selection predicates. This is something that we ALWAYS want to do wherever possible.

RA Equivalencies

$$\sigma_c(R \times S) \equiv (\sigma_c(R)) \times S$$

Selection <u>commutes</u> with Cross Product

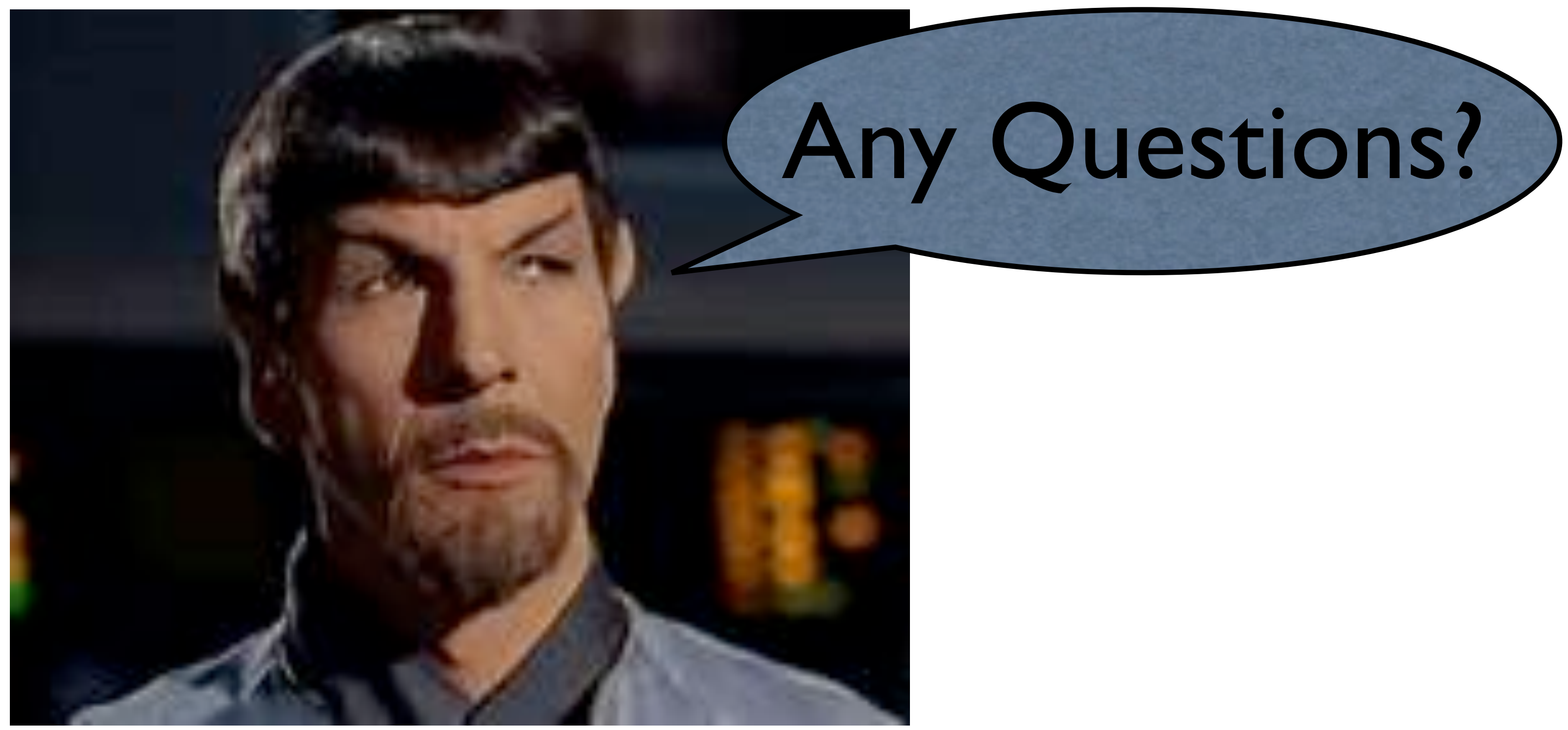
(but only if condition **c** references attributes of R exclusively)

<u>Show that</u>

$$\sigma_c(R \bowtie_{c'} S) \equiv (\sigma_c(R)) \bowtie_{c'} S$$

$$\sigma_{R.B=S.B \wedge R.A>3}(R \times S) \equiv (\sigma_{R.A>3}(R)) \bowtie_{R.B=S.B} S$$

The last example demonstrates pushing down selection predicates. This is something that we ALWAYS want to do wherever possible.


Any Questions?

RA Equivalencies

$$\pi_a(R \times S) \equiv (\pi_{a_1}(R)) \times (\pi_{a_2}(S))$$

Projection commutes (distributes) over Cross Product

(where $\mathbf{a_1}$ and $\mathbf{a_2}$ are the attributes in $\mathbf{a}$ from R and S respectively)

This example illustrates pushing down selections. This is something we ALWAYS want to do, wherever possible (unless the inner projections don't actually project anything away)

RA Equivalencies

$$\pi_a(R \times S) \equiv (\pi_{a_1}(R)) \times (\pi_{a_2}(S))$$

Projection <u>commutes</u> (distributes) over Cross Product

(where $\mathbf{a_1}$ and $\mathbf{a_2}$ are the attributes in $\mathbf{a}$ from R and S respectively)

<u>Show that</u>

$$\pi_a(R \bowtie_c S) \equiv (\pi_{a_1}(R)) \bowtie_c (\pi_{a_2}(S))$$

This example illustrates pushing down selections. This is something we ALWAYS want to do, wherever possible (unless the inner projections don't actually project anything away)

RA Equivalencies

$$\pi_a(R \times S) \equiv (\pi_{a_1}(R)) \times (\pi_{a_2}(S))$$

Projection <u>commutes</u> (distributes) over Cross Product

(where $\mathbf{a_1}$ and $\mathbf{a_2}$ are the attributes in $\mathbf{a}$ from R and S respectively)

<u>Show that</u>

$$\pi_a(R \bowtie_c S) \equiv (\pi_{a_1}(R)) \bowtie_c (\pi_{a_2}(S))$$

(under what condition)

How can we work around this limitation?

This example illustrates pushing down selections. This is something we ALWAYS want to do, wherever possible (unless the inner projections don't actually project anything away)

RA Equivalencies

$$\pi_a(R \times S) \equiv (\pi_{a_1}(R)) \times (\pi_{a_2}(S))$$

Projection <u>commutes</u> (distributes) over Cross Product

(where $\mathbf{a_1}$ and $\mathbf{a_2}$ are the attributes in $\mathbf{a}$ from R and S respectively)

<u>Show that</u>

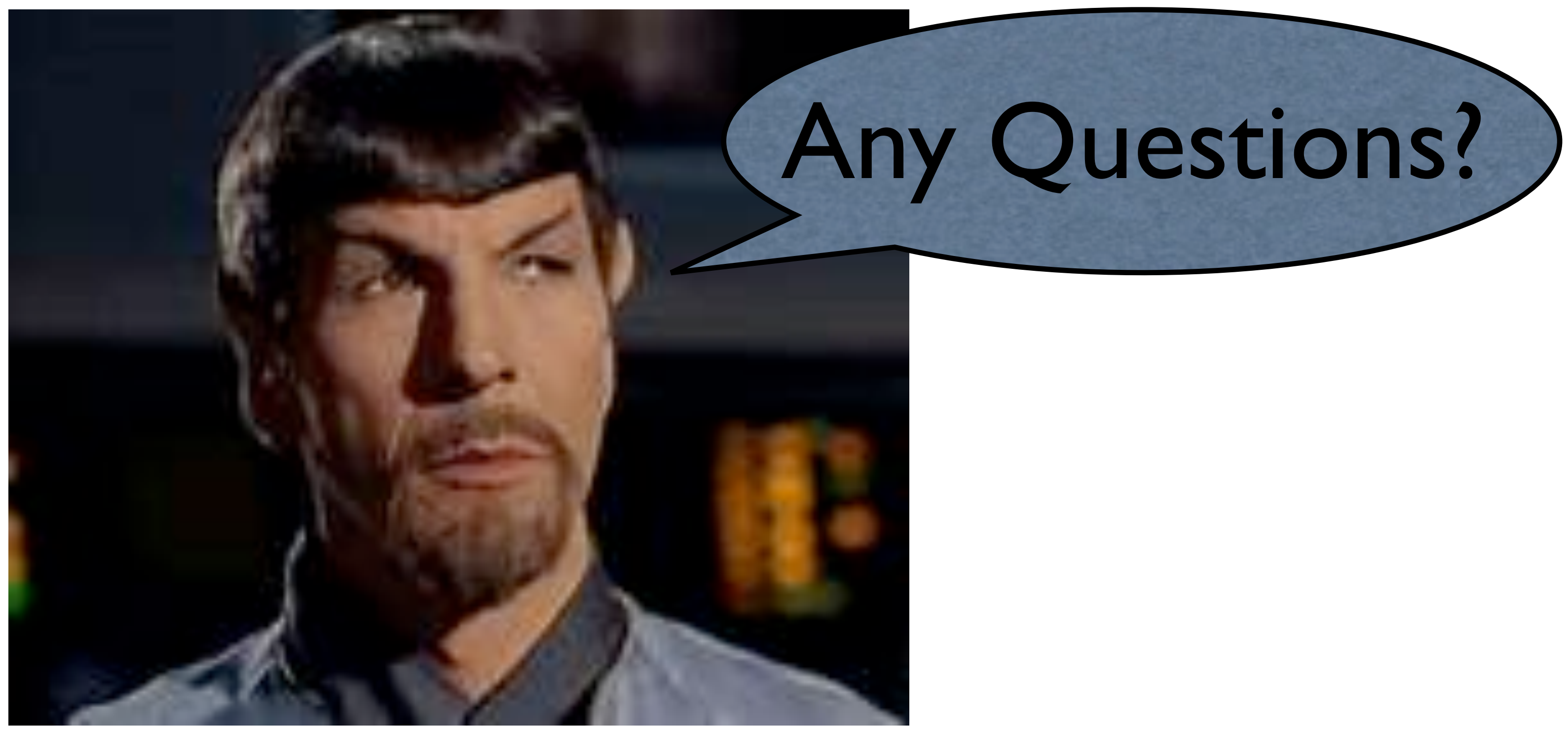
$$\pi_a(R \bowtie_c S) \equiv (\pi_{a_1}(R)) \bowtie_c (\pi_{a_2}(S))$$

(under what condition)

How can we work around this limitation?

$$\pi_a((\pi_{a_1 \cup (\mathtt{cols}(c) \cap \mathtt{cols}(R))}(R)) \bowtie_c (\pi_{a_2 \cup (\mathtt{cols}(c) \cap \mathtt{cols}(S))}(S)))$$

This example illustrates pushing down selections. This is something we ALWAYS want to do, wherever possible (unless the inner projections don't actually project anything away)


Any Questions?

RA Equivalencies

Union and Intersections are <u>Commutative</u> and <u>Associative</u>

Selection and Projection both commute
with both Union and Intersection

Enumerating Possible Plans

- There are many algorithms suitable for processing a specific data set:

  1. How do we identify feasible algorithms.
  2. How do we choose between them