Query Processing: Goal

- Go through tables to find the right tuples
  - Efficiently
- Challenges
  - Selection
    - Use of indices
  - Projection
    - Duplicate elimination
- Cartesian Product
  - Ouch
    - $|R1| \times |R2|$ tuples...
- Join processing
  - Combining Cartesian product and selection can be much more efficient
- Set operations
  - Union, Intersection, Difference
Example

Select B, D
From R, S
Where R.A = “c” \( \land \) S.E = 2 \( \land \) R.C = S.C

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td></td>
<td>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>20</td>
<td></td>
<td>20</td>
<td>y</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>2</td>
<td>10</td>
<td></td>
<td>30</td>
<td>z</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>35</td>
<td></td>
<td>40</td>
<td>x</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>3</td>
<td>45</td>
<td></td>
<td>50</td>
<td>y</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Answer | B | D |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
How do we execute query?

- Do Cartesian product
- Select tuples
- Do projection

One idea

<table>
<thead>
<tr>
<th>RXS</th>
<th>R.A</th>
<th>R.B</th>
<th>R.C</th>
<th>S.C</th>
<th>S.D</th>
<th>S.E</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>10</td>
<td>20</td>
<td>y</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>10</td>
<td>10</td>
<td>x</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Bingo! Got one...
Relational Algebra - can be used to describe plans...

Ex: Plan I

\[ \Pi_{B,D} \]

\[ \sigma_{R.A=“c” \land S.E=2 \land R.C=S.C} \]

\[ X \]

\[ R \times S \]

OR: \[ \Pi_{B,D} [\sigma_{R.A=“c” \land S.E=2 \land R.C=S.C} (RXS)] \]

Another idea:

Plan II

\[ B,D \]

\[ \sigma_{R.A = “c”} \]

\[ \sigma_{S.E = 2} \]

\[ R \times S \]

natural join
Plan III

Use R.A and S.C Indexes

(1) Use R.A index to select R tuples with R.A = “c”
(2) For each R.C value found, use S.C index to find matching tuples
(3) Eliminate S tuples S.E ≠ 2
(4) Join matching R,S tuples, project B,D attributes and place in result
### Chapter 15: Query Processing

- Overview
- Measures of Query Cost
- Selection Operation
- Sorting
- Join Operation
- Other Operations
- Evaluation of Expressions
Basic Steps in Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation

Parsing and translation
- translate the query into its internal form. This is then translated into relational algebra.
- Parser checks syntax, verifies relations

Evaluation
- The query-execution engine takes a query-evaluation plan, executes that plan, and returns the answers to the query.
Basic Steps in Query Processing: Optimization

- A relational algebra expression may have many equivalent expressions
  - E.g., \( \sigma_{\text{salary} < 75000}(\Pi_{\text{salary}}(\text{instructor})) \) is equivalent to \( \Pi_{\text{salary}}(\sigma_{\text{salary} < 75000}(\text{instructor})) \)

- Each relational algebra operation can be evaluated using one of several different algorithms
  - Correspondingly, a relational-algebra expression can be evaluated in many ways.

- Annotated expression specifying detailed evaluation strategy is called an evaluation-plan. E.g.,:
  - Use an index on \( \text{salary} \) to find instructors with \( \text{salary} < 75000 \),
  - Or perform complete relation scan and discard instructors with \( \text{salary} \geq 75000 \)

Basic Steps: Optimization (Cont.)

- **Query Optimization**: Amongst all equivalent evaluation plans choose the one with lowest cost.
  - Cost is estimated using statistical information from the database catalog
    - e.g., number of tuples in each relation, size of tuples, etc.

- In this chapter we study
  - How to measure query costs
  - Algorithms for evaluating relational algebra operations
  - How to combine algorithms for individual operations in order to evaluate a complete expression

- In Chapter 16
  - We study how to optimize queries, that is, how to find an evaluation plan with lowest estimated cost