Early Materialization

\[ \Pi_{B,C} \]
\[ \sigma_{B=5} \]
\[ \sigma_{A=8} \]
\[ T \]

\[ \Pi_{B,C} \]
\[ \sigma_{B=5} \]
\[ \sigma_{A=8} \]
\[ \mathcal{R}_{\text{RID}} \]
\[ T.A \]
\[ T.B \]
\[ T.C \]

SELECT B,C
FROM T
WHERE A=8 AND B=5
(Partially) Late Materialization

\[
\begin{align*}
&\Pi_{B,C} \\
&\quad \sigma_{B=5} \\
&\quad \sigma_{A=8} \\
&\quad T
\end{align*}
\]

\[
\begin{align*}
&\prod_{B,C} \\
&\nearrow \quad \text{Potential few results} \\
&\land_{\text{RID}} = \prod_{C} \\
&\quad \exists \text{ Li Projection} \\
&\quad \exists \text{ Inj} \\
&\quad \exists \text{ min} \\
&\quad \exists \text{ min}
\end{align*}
\]

\[
\begin{align*}
&\prod_{\text{RID}} \quad \prod_{B,\text{RID}} \\
&\quad \sigma_{A=8} \\
&\quad \sigma_{B=5} \\
&\quad T.A \\
&\quad T.B \\
&\quad T.C
\end{align*}
\]

```
SELECT B,C
FROM T
WHERE A=8 AND B=5
```
(Really) Late Materialization

\[ \Pi_{B,C} \]

\[ \sigma_{B=5} \]

\[ \sigma_{A=8} \]

\[ T \]

\[ \pi_{B,C} \]

\[ \sigma_{A=8} \land \sigma_{B=5} \]

\[ \bigcap_{\text{RID}} \]

\[ \bigcup_{\text{B,C}} \]

SELECT B, C
FROM T
WHERE A=8 AND B=5
Projection vs “Anti-Projection“

When to narrow tuples?

\[ \Pi_{A,B} \]

\[ \bigcup_c \]

When to widen tuples?
Example: Tuple Reconstruction in Column Stores

```sql
SELECT * FROM T
```

Column Store

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

Row-wise output

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tuples 0, 1, 2
Implementing Early Materialization

```sql
SELECT B, C
FROM T
WHERE A = 8 AND B = 5
```

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>
```

(early) materialization

```
\sigma_{A=8 \ AND \ B=5}
```

```
\Pi_{B, C}
```

Output

```
<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>
```
Implementing Late Materialization (1)

```
SELECT B, C
FROM T
WHERE A=8 AND B=5
```

query request

```
A
3
8
8

B
2
5
2

C
7
3
9
```

column store

```
σA=8
0
1

σB=5
0
1
```

bitvector marking qualifying entries

```
σA=8 AND B=5
0
1
```

conjunction bitvector
Implementing Late Materialization (2)

```
SELECT B, C
FROM T
WHERE A=8 AND B=5
```
Joins using Early Materialization

Input to join has been materialized already!

$$\text{SELECT T.B, S.C}
\text{FROM T, S}
\text{WHERE T.A = S.A}$$

\[
\begin{array}{|c|c|}
\hline
\text{T} & \text{S} \\
\hline
\text{B} & \text{A} & \text{A} & \text{C} \\
\hline
\text{7} & \text{3} & \text{2} & \text{3} \\
\text{5} & \text{4} & \text{3} & \text{4} \\
\text{6} & \text{7} & \text{1} & \text{7} \\
\text{4} & \text{4} & \text{1} & \text{4} \\
\text{3} & \text{2} & \text{9} & \text{2} \\
\text{1} & \text{2} & \text{8} & \text{2} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|c|}
\hline
\text{A} & \text{B} & \text{C} \\
\hline
\text{3} & \text{7} & \text{4} \\
\text{2} & \text{3} & \text{3} \\
\text{2} & \text{1} & \text{3} \\
\hline
\end{array}
\]

\[
\begin{array}{|c|c|}
\hline
\text{B} & \text{C} \\
\hline
\text{7} & \text{4} \\
\text{3} & \text{3} \\
\text{1} & \text{3} \\
\hline
\end{array}
\]

$$\Pi_{B, C}$$
Joins using Late Materialization

```
SELECT T.B, S.C
FROM T, S
WHERE T.A = S.A
```
Early Materialization

Advantages:

no re-access of columns necessary
Early Materialization

Advantages:

no re-access of columns necessary

easier planning

Disadvantages:

possible generation of wide intermediate results
Late Materialization

Advantages:

- constructing tuples only when necessary
- slightly more complex planning (actually a disadvantage)

Diagram:

- New layout
- symbol indicating operation

Arrow pointing to:

- Tuple vectorization
Late Materialization

Advantages:

- constructing tuples only when necessary
- slightly more complex planning

Disadvantages:

- re-access of columns possible
<table>
<thead>
<tr>
<th></th>
<th>Early Materialization</th>
<th>Late Materialization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selectivity</strong></td>
<td>Low (many entries selected)</td>
<td>High (few entries selected)</td>
</tr>
<tr>
<td><strong>Compression</strong></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Aggregation</strong></td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>