Query decomposition and data localization

Dario Della Monica

These slides are a modified version of the slides provided with the book

Özsu and Valduriez, *Principles of Distributed Database Systems* (3rd Ed.), 2011

The original version of the slides is available at: extras.springer.com

Outline (distributed DB)

- Introduction (Ch. 1) *
- Distributed Database Design (Ch. 3) *
- Distributed Query Processing (Ch. 6-8) *
  - → Overview (Ch. 6) *
  - → **Query decomposition and data localization (Ch. 7)** *
  - → Distributed query optimization (Ch. 8) *
- Distributed Transaction Management (Ch. 10-12) *

* Özsu and Valduriez, *Principles of Distributed Database Systems* (3rd Ed.), 2011

Outline (today)

- Query decomposition and **data localization** (Ch. 7) *
  - → The problem of distributed data localization
  - → A naïve algorithm
  - → Optimization steps (reductions)
    - ✦ PHF (selection, join)
    - ✦ VF (projection)
    - ✦ DHF (selection, join)
    - ✦ Hybrid Fragmentation (selection/join + projection)

* Özsu and Valduriez, *Principles of Distributed Database Systems* (3rd Ed.), 2011

Data Localization

Input: Relational algebra expression on global, distributed relations (distributed query)

Output: Relational algebra expression on fragments (localized query)

- Localization uses global information about distribution of fragments (no optimization, no use of quantitative information, e.g., catalog statistics)
- Recall that fragmentation is obtained by several application of rules expressed by relational algebra …
  - ➠ primary horizontal fragmentation: selection σ
  - ➠ derived horizontal fragmentation: semijoin $\ltimes$
  - ➠ vertical fragmentation: projection Π
- … and that reconstruction (reverse fragmentation) rules are also expressed in relational algebra
  - ➠ horizontal fragmentation: union $\cup$
  - ➠ vertical fragmentation: join $\bowtie$

A naïve algorithm to localize distribute queries

- Localization program: relational algebra expression that reconstructs a global relation from its fragments, by reverting the rules employed for fragmentation
- A localized query is obtained from distributed, global query by replacing leaves (global relations) with (the tree of) its corresponding localization program
  - ➡ Leaves of localized queries are fragments
- This approach to obtain a localized query from a distributed one is inefficient and the result can be improved through several optimizations
  - ➡ During data localization there is a **first optimization phase**
    - ✦ we call it **reduction**
    - ✦ different from the **proper optimization phase** (finding the “best” strategy for executing the query)

Example

PROJ ⋈ (EMP ⋈ ASG)

Example

Assume

$PROJ \bowtie (EMP \bowtie ASG)$

- EMP is fragmented as follows:
  - → $EMP_1 = \sigma_{ENO \leq \text{"E3"}}(EMP)$
  - → $EMP_2 = \sigma_{\text{"E3"} < ENO \leq \text{"E6"}}(EMP)$
  - → $EMP_3 = \sigma_{ENO \geq \text{"E6"}}(EMP)$
- ASG is fragmented as follows:
  - → $ASG_1 = \sigma_{ENO \leq \text{"E3"}}(ASG)$
  - → $ASG_2 = \sigma_{ENO > \text{"E3"}}(ASG)$

Example

Assume

$PROJ \bowtie (EMP \bowtie ASG)$

- EMP is fragmented as follows:
  - $EMP_1 = \sigma_{ENO \leq \text{"E3"}}(EMP)$
  - $EMP_2 = \sigma_{\text{"E3"} < ENO \leq \text{"E6"}}(EMP)$
  - $EMP_3 = \sigma_{ENO \geq \text{"E6"}}(EMP)$
- ASG is fragmented as follows:
  - $ASG_1 = \sigma_{ENO \leq \text{"E3"}}(ASG)$
  - $ASG_2 = \sigma_{ENO > \text{"E3"}}(ASG)$

Replace EMP by $(EMP_1 \cup EMP_2 \cup EMP_3)$
and ASG by $(ASG_1 \cup ASG_2)$ in any query

Example

Assume

- EMP is fragmented as follows:
  - $EMP_1 = \sigma_{ENO \leq "E3"}(EMP)$
  - $EMP_2 = \sigma_{"E3" < ENO \leq "E6"}(EMP)$
  - $EMP_3 = \sigma_{ENO \geq "E6"}(EMP)$
- ASG is fragmented as follows:
  - $ASG_1 = \sigma_{ENO \leq "E3"}(ASG)$
  - $ASG_2 = \sigma_{ENO > "E3"}(ASG)$

Replace EMP by $(EMP_1 \cup EMP_2 \cup EMP_3)$ and ASG by $(ASG_1 \cup ASG_2)$ in any query

$$PROJ \bowtie (EMP \bowtie ASG)$$
$$=$$
$$PROJ \bowtie ((EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2))$$

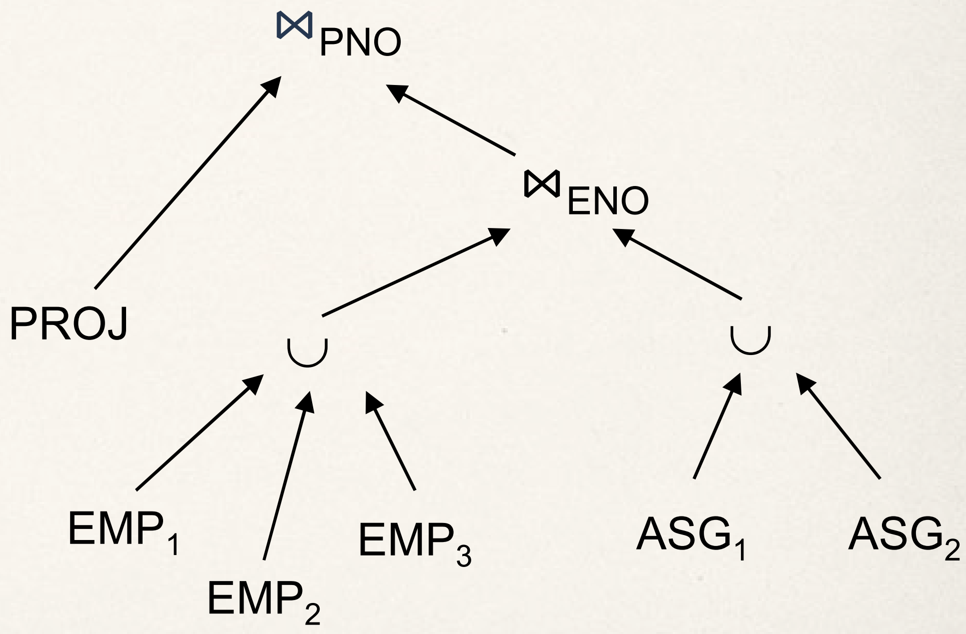
Example

Assume

- EMP is fragmented as follows:
  - $EMP_1 = \sigma_{ENO \leq \text{“E3”}}(EMP)$
  - $EMP_2 = \sigma_{\text{“E3”} < ENO \leq \text{“E6”}}(EMP)$
  - $EMP_3 = \sigma_{ENO \geq \text{“E6”}}(EMP)$
- ASG is fragmented as follows:
  - $ASG_1 = \sigma_{ENO \leq \text{“E3”}}(ASG)$
  - $ASG_2 = \sigma_{ENO > \text{“E3”}}(ASG)$

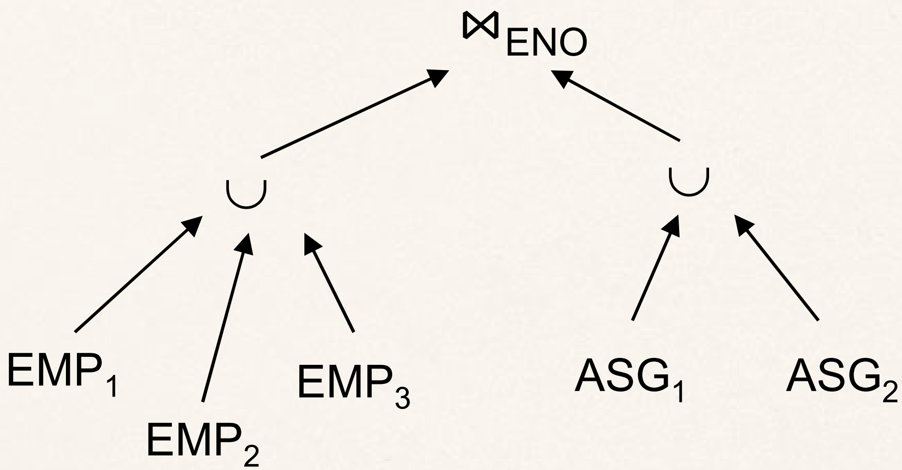
Replace EMP by ($EMP_1 \cup EMP_2 \cup EMP_3$) and ASG by ($ASG_1 \cup ASG_2$) in any query

$$PROJ \bowtie (EMP \bowtie ASG) = PROJ \bowtie ((EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2))$$



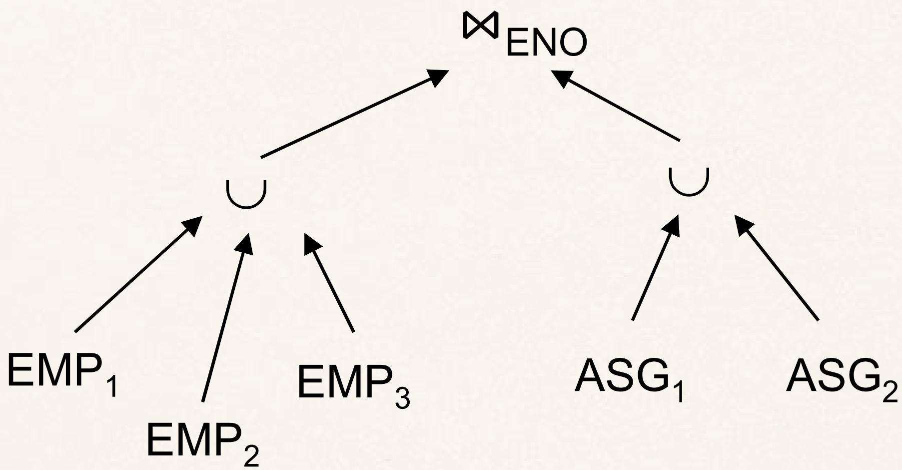
Provides Parallellism

$(EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2)$



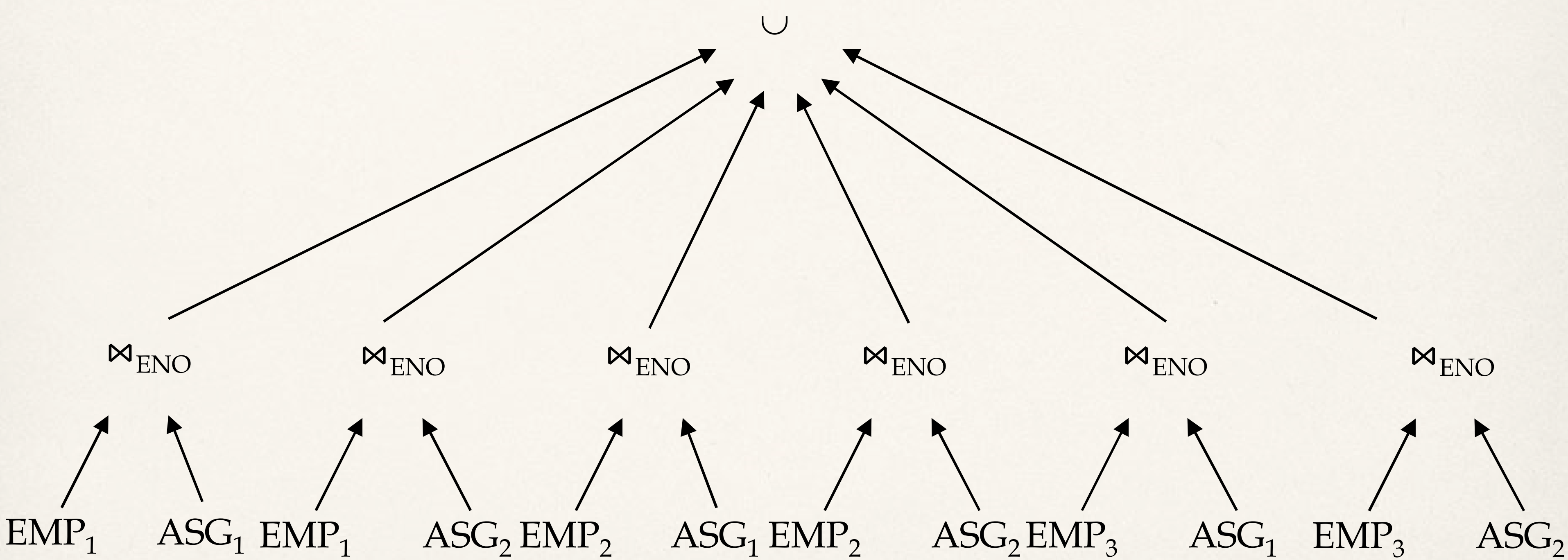
Provides Parallellism

$$(EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2)$$
$$=$$
$$(EMP_1 \bowtie ASG_1) \cup (EMP_1 \bowtie ASG_2) \cup$$
$$(EMP_2 \bowtie ASG_1) \cup (EMP_2 \bowtie ASG_2) \cup$$
$$(EMP_3 \bowtie ASG_1) \cup (EMP_3 \bowtie ASG_2)$$



Provides Parallellism

$$(EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2)$$
$$=$$
$$(EMP_1 \bowtie ASG_1) \cup (EMP_1 \bowtie ASG_2) \cup$$
$$(EMP_2 \bowtie ASG_1) \cup (EMP_2 \bowtie ASG_2) \cup$$
$$(EMP_3 \bowtie ASG_1) \cup (EMP_3 \bowtie ASG_2)$$

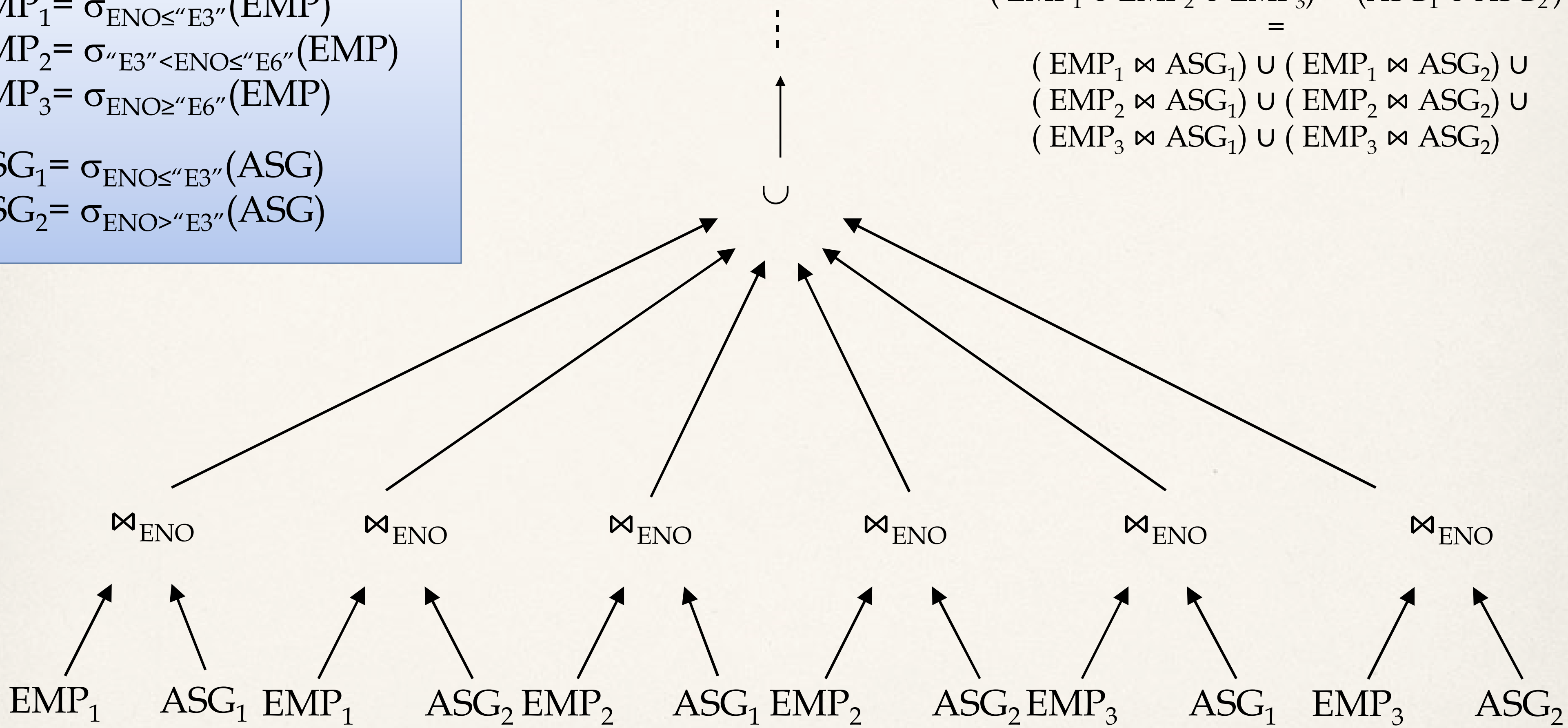


Provides Parallellism

$EMP_1 = \sigma_{ENO \leq \text{“E3”}}(EMP)$
$EMP_2 = \sigma_{\text{“E3”} < ENO \leq \text{“E6”}}(EMP)$
$EMP_3 = \sigma_{ENO \geq \text{“E6”}}(EMP)$

$ASG_1 = \sigma_{ENO \leq \text{“E3”}}(ASG)$
$ASG_2 = \sigma_{ENO > \text{“E3”}}(ASG)$

$$(EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2)$$
$$=$$
$$(EMP_1 \bowtie ASG_1) \cup (EMP_1 \bowtie ASG_2) \cup$$
$$(EMP_2 \bowtie ASG_1) \cup (EMP_2 \bowtie ASG_2) \cup$$
$$(EMP_3 \bowtie ASG_1) \cup (EMP_3 \bowtie ASG_2)$$

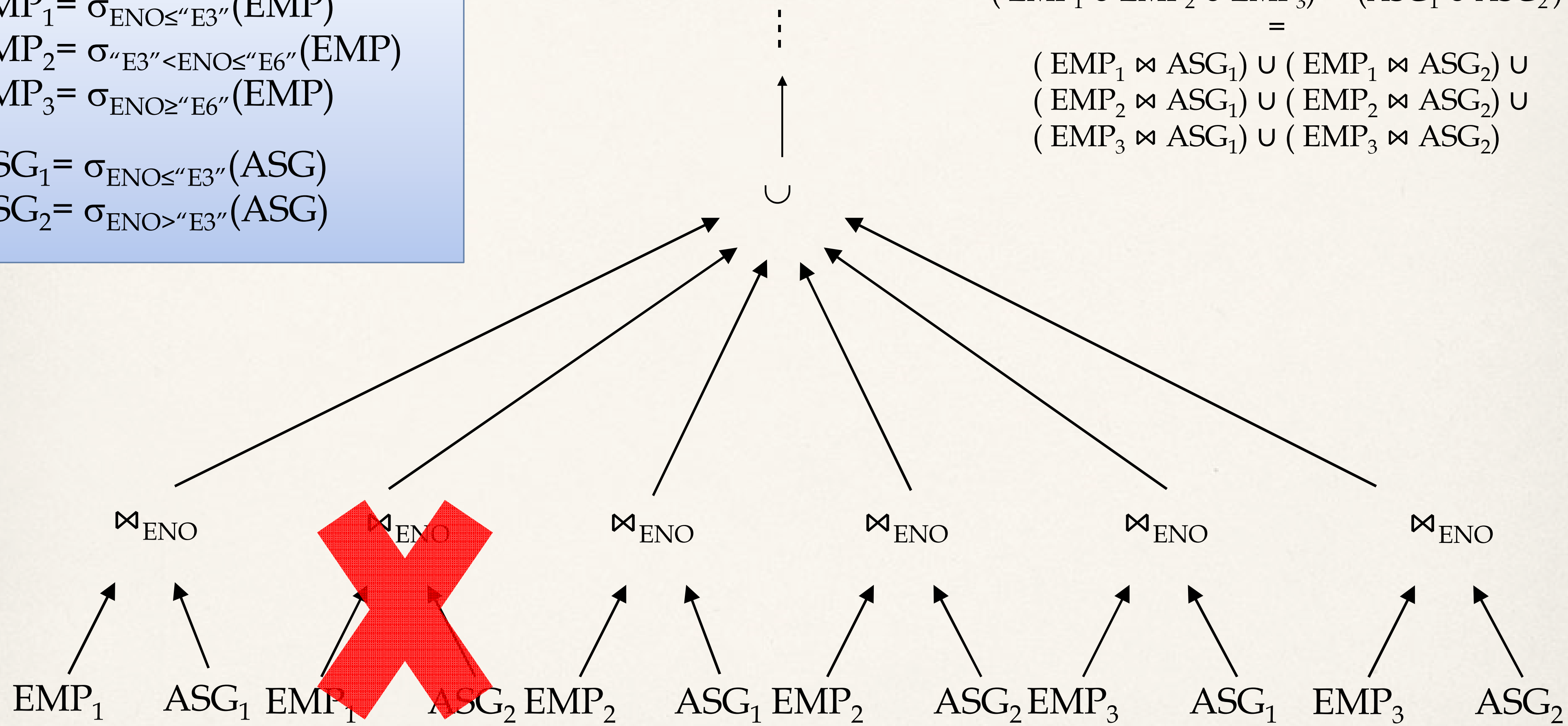


Provides Parallellism

$EMP_1 = \sigma_{ENO \leq \text{"E3"}}(EMP)$
$EMP_2 = \sigma_{\text{"E3"} < ENO \leq \text{"E6"}}(EMP)$
$EMP_3 = \sigma_{ENO \geq \text{"E6"}}(EMP)$

$ASG_1 = \sigma_{ENO \leq \text{"E3"}}(ASG)$
$ASG_2 = \sigma_{ENO > \text{"E3"}}(ASG)$

$$(EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2) \\ = \\ (EMP_1 \bowtie ASG_1) \cup (EMP_1 \bowtie ASG_2) \cup \\ (EMP_2 \bowtie ASG_1) \cup (EMP_2 \bowtie ASG_2) \cup \\ (EMP_3 \bowtie ASG_1) \cup (EMP_3 \bowtie ASG_2)$$

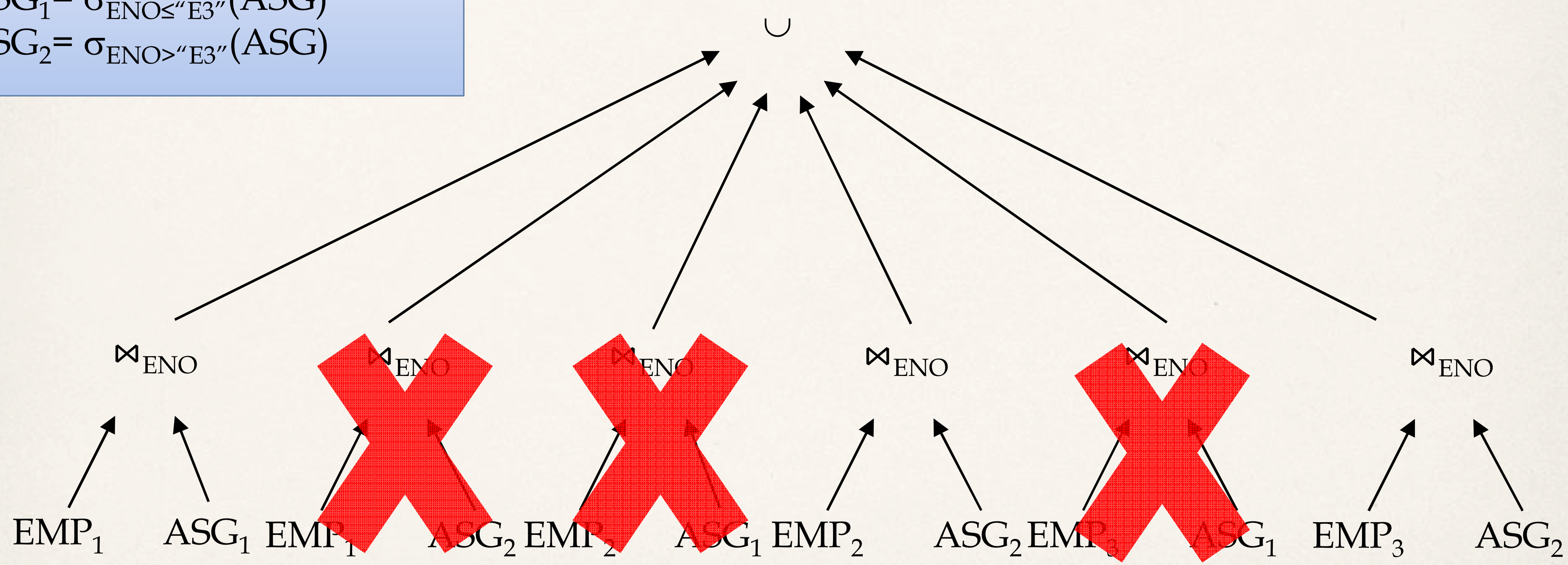


Provides Parallellism

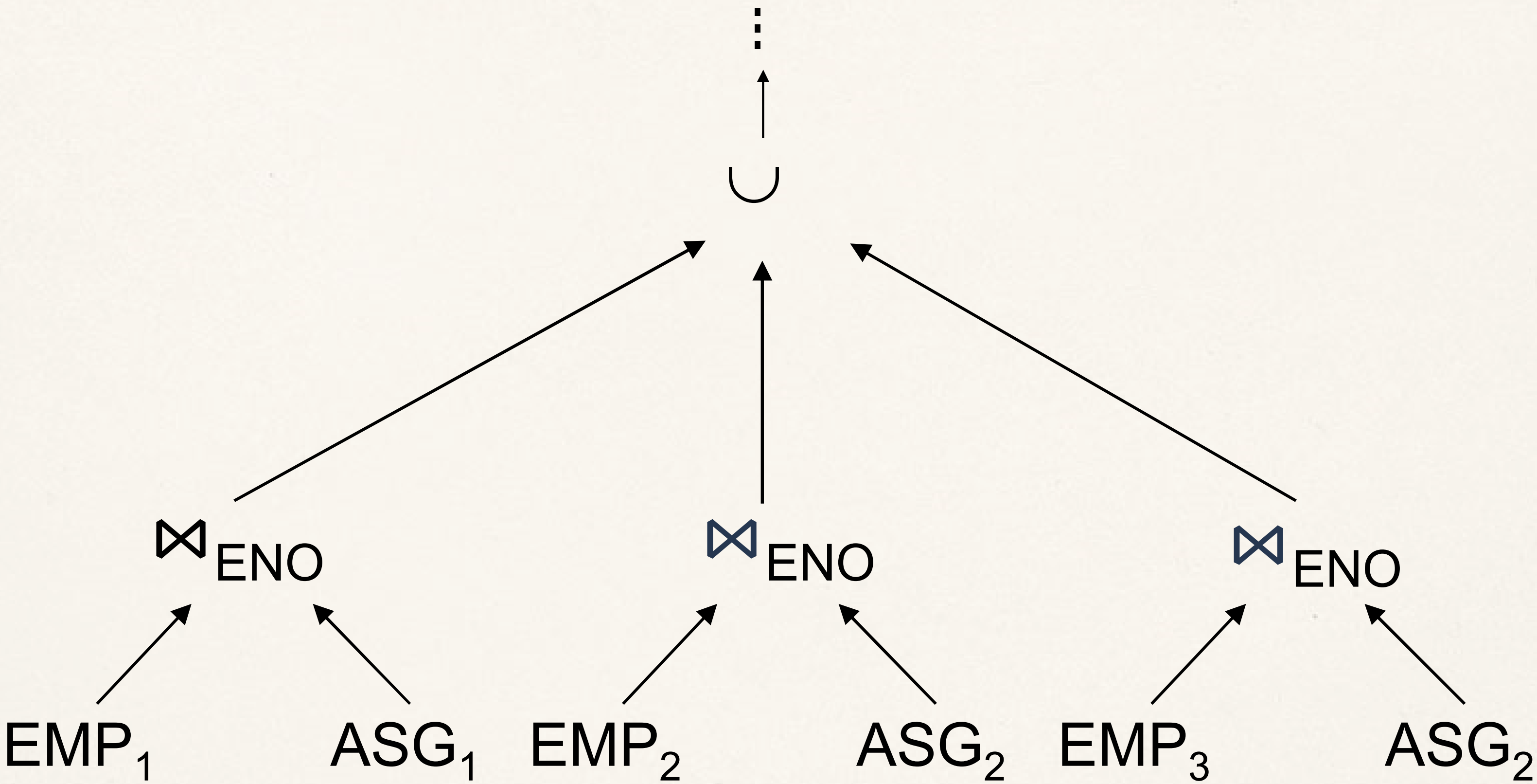
$EMP_1 = \sigma_{ENO \leq \text{“E3”}}(EMP)$
$EMP_2 = \sigma_{\text{“E3”} < ENO \leq \text{“E6”}}(EMP)$
$EMP_3 = \sigma_{ENO \geq \text{“E6”}}(EMP)$

$ASG_1 = \sigma_{ENO \leq \text{“E3”}}(ASG)$
$ASG_2 = \sigma_{ENO > \text{“E3”}}(ASG)$

$$(EMP_1 \cup EMP_2 \cup EMP_3) \bowtie (ASG_1 \cup ASG_2)$$
$$=$$
$$(EMP_1 \bowtie ASG_1) \cup (EMP_1 \bowtie ASG_2) \cup$$
$$(EMP_2 \bowtie ASG_1) \cup (EMP_2 \bowtie ASG_2) \cup$$
$$(EMP_3 \bowtie ASG_1) \cup (EMP_3 \bowtie ASG_2)$$



Eliminates Unnecessary Work



Identify (pairs of) fragments that can be ignored because they produce empty relations (e.g., when a selection or a join is applied to them)

Reduction for PHF – Selection

- Reduction of a selection over a relation fragmented with PHF (ignore a fragment if selection predicate and fragment predicate are contradictory)
  - Consider $\sigma_p(R)$
  - Horizontal fragmentation on R: $F_R=\{R_1, R_2, \ldots, R_w\}$, where $R_j=\sigma_{p_j}(R)$
  - $\sigma_p(R_j)=\emptyset$ if $\forall x$ in R: $\neg(p_i(x) \wedge p_j(x))$ i.e., p and p_j are contradictory

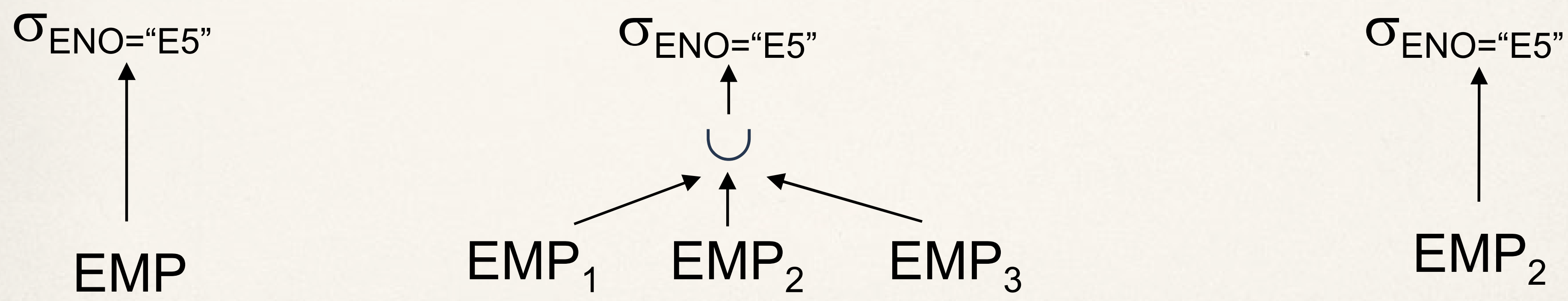
Reduction for PHF – Selection (Example)

- Reduction of a selection over a relation fragmented with PHF (ignore a fragment if selection predicate and fragment predicate are contradictory)
  - → Example

```
SELECT  *
FROM    EMP
WHERE   ENO="E5"
```

$EMP_1 = \sigma_{ENO \leq \text{"E3"}}(EMP)$
$EMP_2 = \sigma_{\text{"E3"} < ENO \leq \text{"E6"}}(EMP)$
$EMP_3 = \sigma_{ENO \geq \text{"E6"}}(EMP)$

$ASG_1 = \sigma_{ENO \leq \text{"E3"}}(ASG)$
$ASG_2 = \sigma_{ENO > \text{"E3"}}(ASG)$



distributed query **localized query** **reduced local query**

Reduction for PHF – Join

- Reduction of a join over relations fragmented with PHF (ignore the join of 2 fragments if their fragment predicates are contradictory over the join attributes)
  - ➡ Possible if fragmentation is done on join attribute
  - ➡ Distribute join over union

$$R \bowtie S \Leftrightarrow (R_1 \cup R_2) \bowtie (S_1 \cup S_2)$$
$$\Leftrightarrow (R_1 \bowtie S_1) \cup (R_1 \bowtie S_2) \cup (R_2 \bowtie S_1) \cup (R_2 \bowtie S_2)$$

  - ➡ Then, join between 2 fragments can be simplified in some cases
    - ✦ Given $R_i = \sigma_{p_i}(R)$ and $S_j = \sigma_{p_j}(S)$ [p_i and p_j defined over join attributes]

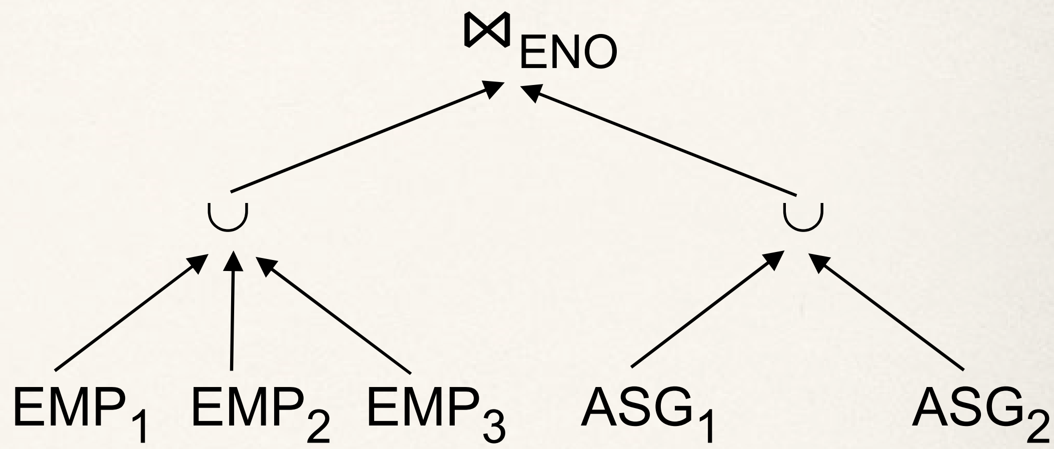
$R_i \bowtie S_j = \varnothing$ if $\forall x$ in $R \cup S : \neg(p_i(x) \wedge p_j(x))$ *[there is a mistake in the textbook]*

i.e., p_i and p_j are contradictory

Reduction for PHF – Join (Example)

$EMP_1 = \sigma_{ENO \leq \text{"E3"}}(EMP)$
$EMP_2 = \sigma_{\text{"E3"} < ENO \leq \text{"E6"}}(EMP)$
$EMP_3 = \sigma_{ENO \geq \text{"E6"}}(EMP)$

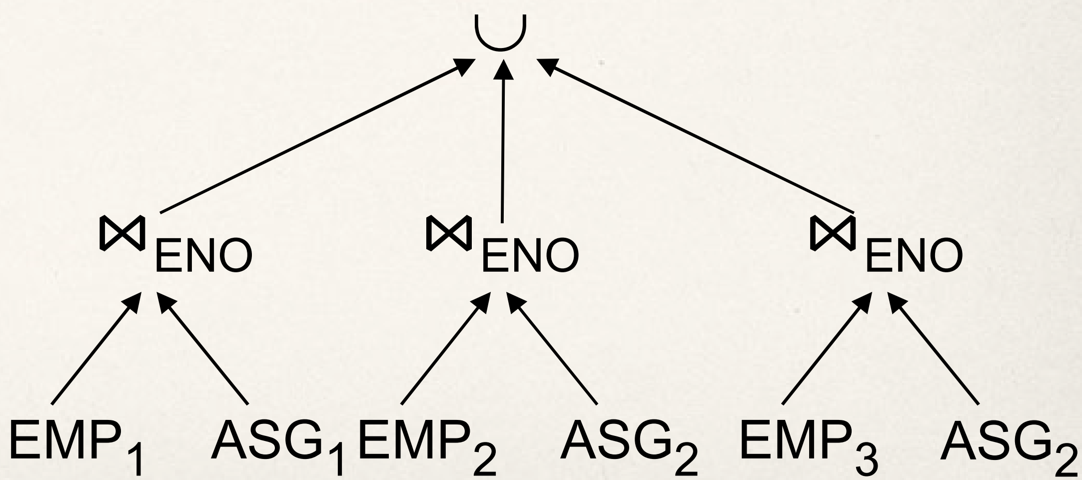
$ASG_1 = \sigma_{ENO \leq \text{"E3"}}(ASG)$
$ASG_2 = \sigma_{ENO > \text{"E3"}}(ASG)$



- Consider the query

```
SELECT  *
FROM    EMP,ASG
WHERE   EMP.ENO=ASG.ENO
```

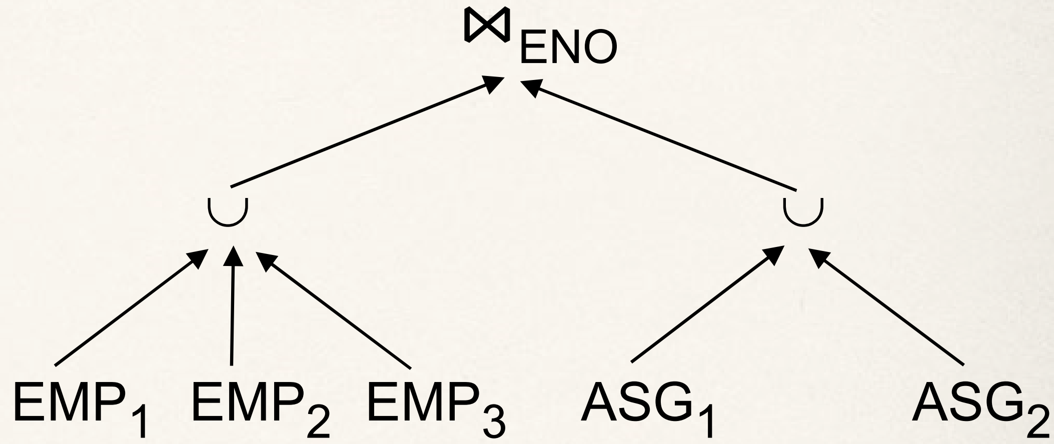
- Distribute join over unions
- Apply the reduction rule



Reduction for PHF – Join (Example)

$EMP_1 = \sigma_{ENO \leq \text{"E3"}}(EMP)$
$EMP_2 = \sigma_{\text{"E3"} < ENO \leq \text{"E6"}}(EMP)$
$EMP_3 = \sigma_{ENO \geq \text{"E6"}}(EMP)$

$ASG_1 = \sigma_{ENO \leq \text{"E3"}}(ASG)$
$ASG_2 = \sigma_{ENO > \text{"E3"}}(ASG)$

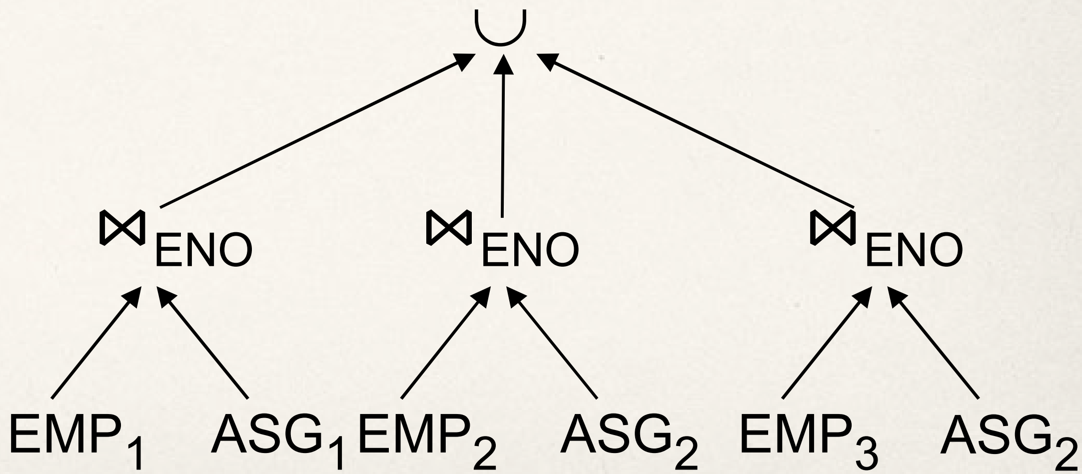


- Consider the query

```
SELECT  *
FROM    EMP,ASG
WHERE   EMP.ENO=ASG.ENO
```

- Distribute join over unions
- Apply the reduction rule

Not always convenient

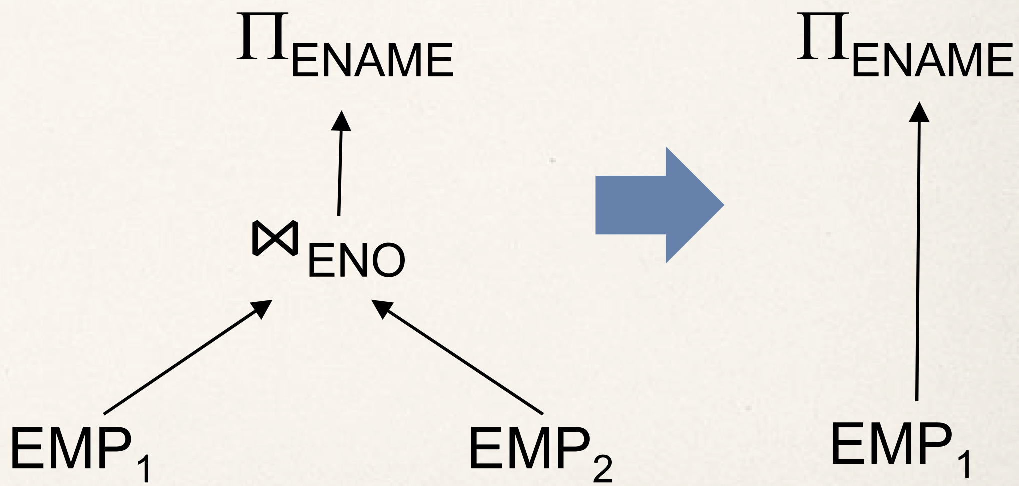


Reduction for VF

- Reduction of a projection over a relation fragmented with VF (ignore the fragment for which the set of projection attributes intersected with set of fragmentation attributes is contained in the primary key)
- Recall that the localization program consists in joins over key attributes
- Let R_1 be a fragment of R obtained as $R_1 = \Pi_{A'}(R)$ where $A' \subseteq attr(R)$:
  - Reduction of a projection $\Pi_{A''}$ over R_1 is possible when $A'' \cap A' \subseteq key(R)$

Ex.: $EMP_1 = \Pi_{ENO,ENAME}(EMP)$
$EMP_2 = \Pi_{ENO,TITLE}(EMP)$

```
SELECT   ENAME
FROM     EMP
```



Reduction for DHF

- Similar to the case PHF
- DHF: 2 relations *S* (owner) and *R* (member) in association one-to-many
  - ➡ *S* participates with cardinality N , *R* participates with cardinality 1
  - ➡ Fragmentation propagate from *S* to *R*
  - ➡ Localization program: union
  - ➡ **Fragments that agree on the values of join attributes are placed at the same site**
- Rule :
  - ➡ Distribute joins over unions
  - ➡ Apply the join reduction for horizontal fragmentation

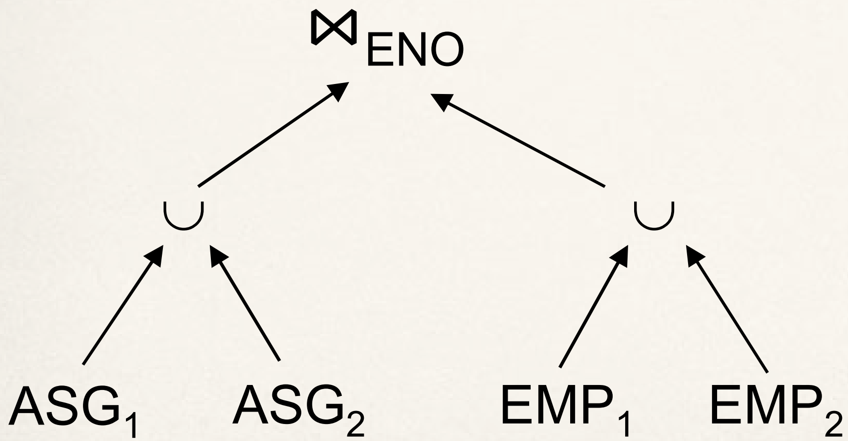
Reduction for DHF – Example

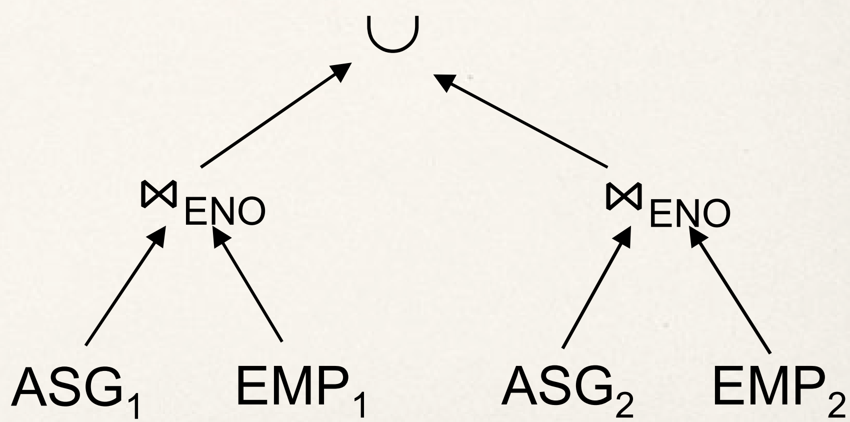
- Example [EMP is owner , ASG is member]

  EMP_1: $\sigma_{TITLE=\text{"Programmer"}}$ (EMP)
  EMP_2: $\sigma_{TITLE\neq\text{"Programmer"}}$ (EMP)
  ASG_1: ASG $\ltimes_{ENO}$ EMP_1
  ASG_2: ASG $\ltimes_{ENO}$ EMP_2

- Query

```
SELECT  *
FROM    EMP, ASG
WHERE   ASG.ENO = EMP.ENO
```





Reduction for DHF – Example

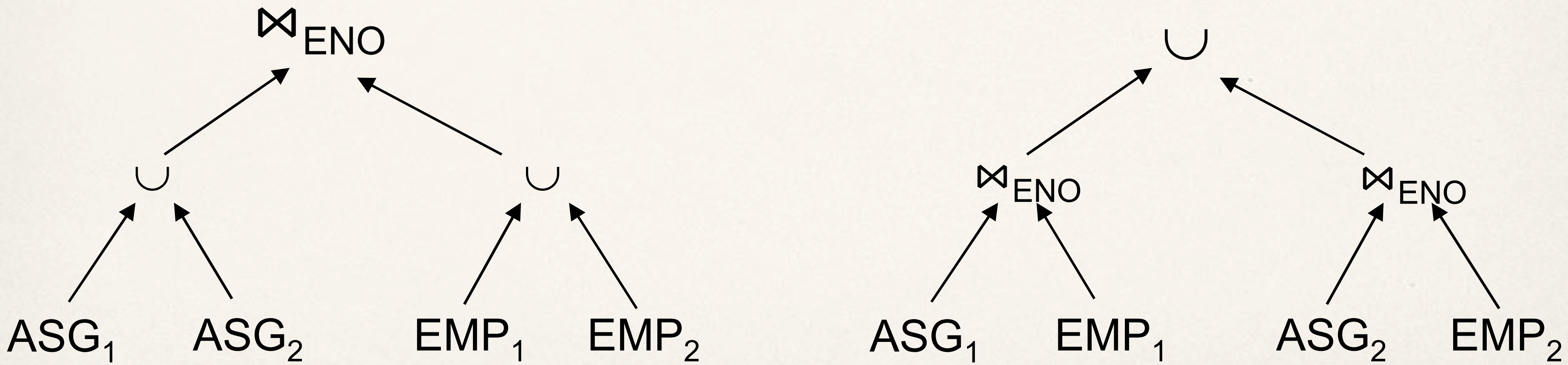
- Example [EMP is owner , ASG is member]

  EMP_1: $\sigma_{TITLE=\text{“Programmer”}}(EMP)$
  EMP_2: $\sigma_{TITLE\neq\text{“Programmer”}}(EMP)$
  ASG_1: $ASG \ltimes_{ENO} EMP_1$
  ASG_2: $ASG \ltimes_{ENO} EMP_2$

Always convenient
- the number of joins is always equal to the number of fragments
- all joins can be performed in parallel (are disjoint)

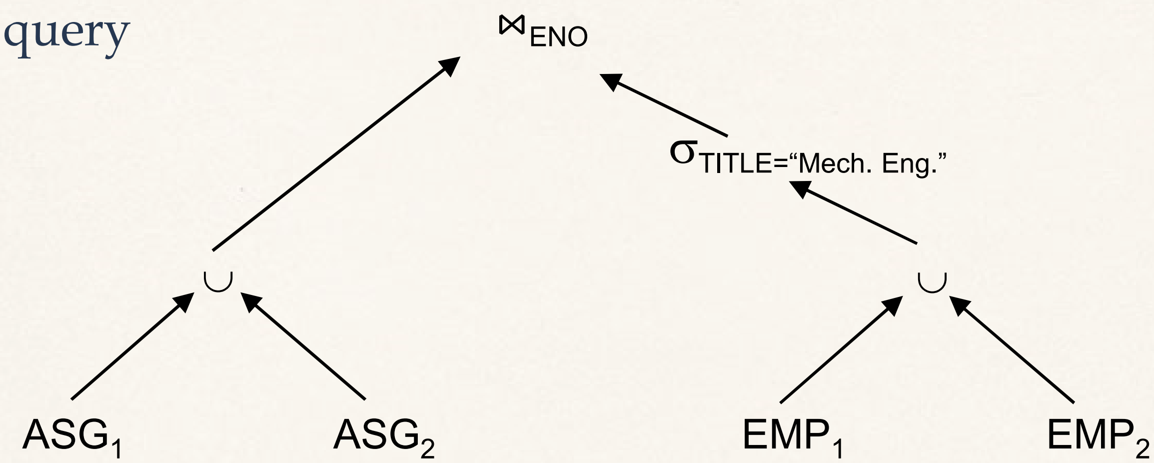
- Query

```
SELECT    *
FROM      EMP, ASG
WHERE     ASG.ENO = EMP.ENO
```



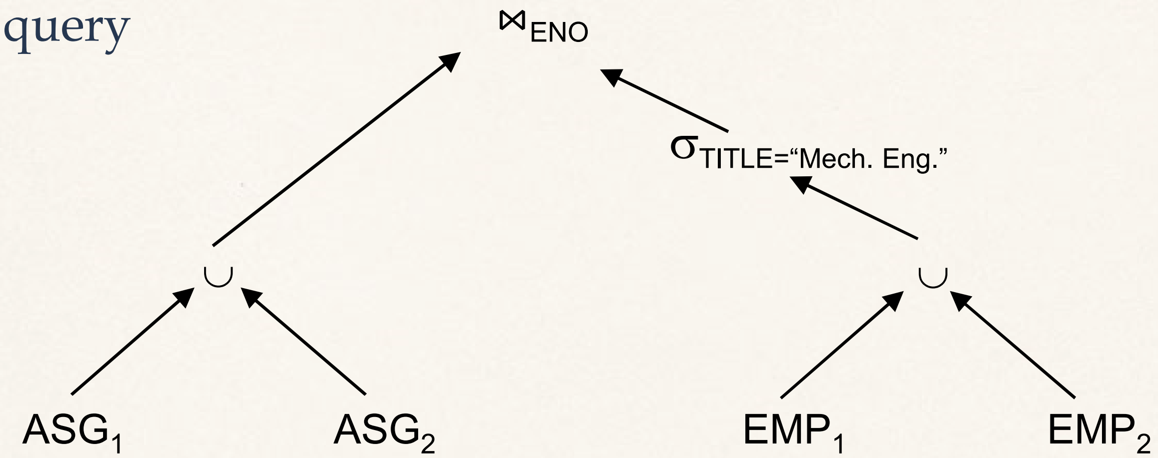
Complex reduction for PHF and DHF

1. Generic query



Complex reduction for PHF and DHF

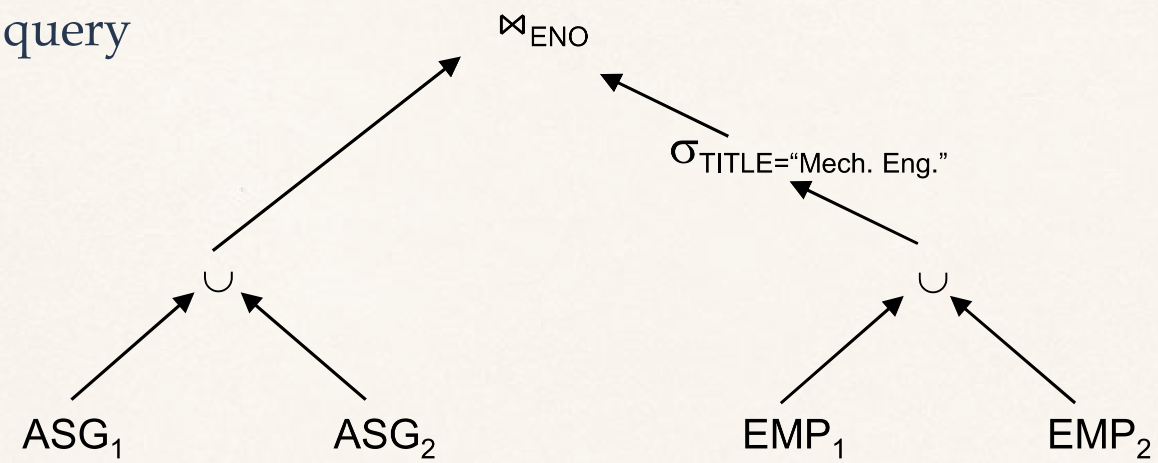
1. Generic query



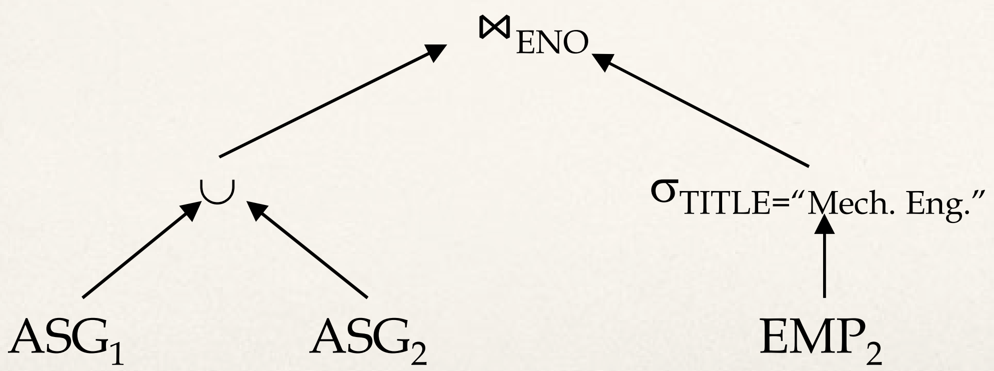
2. Reduction of selection over a relation fragmented with HF

Complex reduction for PHF and DHF

1. Generic query

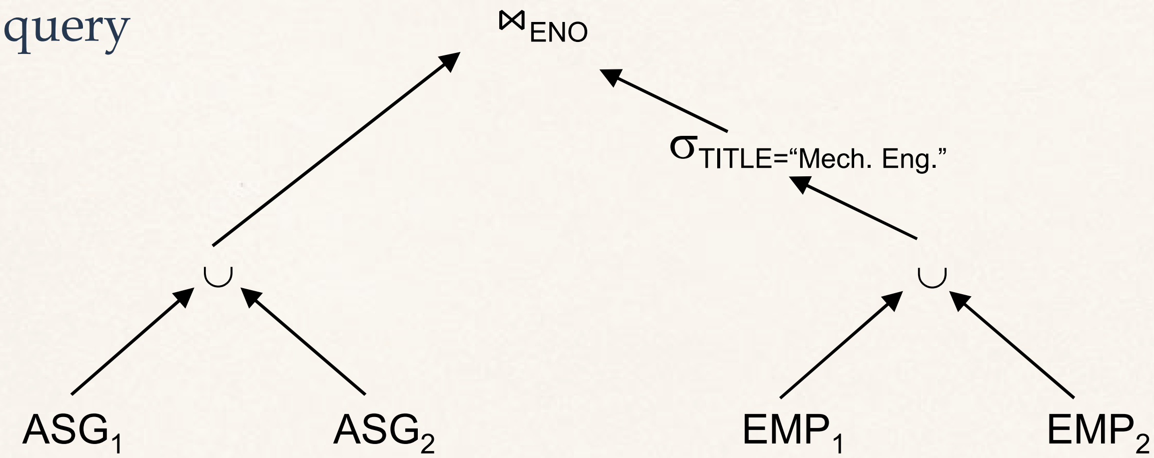


2. Reduction of selection over a relation fragmented with HF

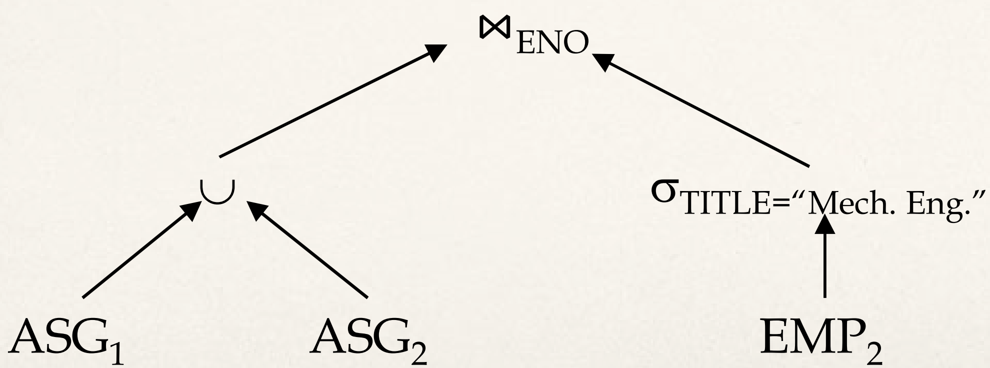


Complex reduction for PHF and DHF

1. Generic query



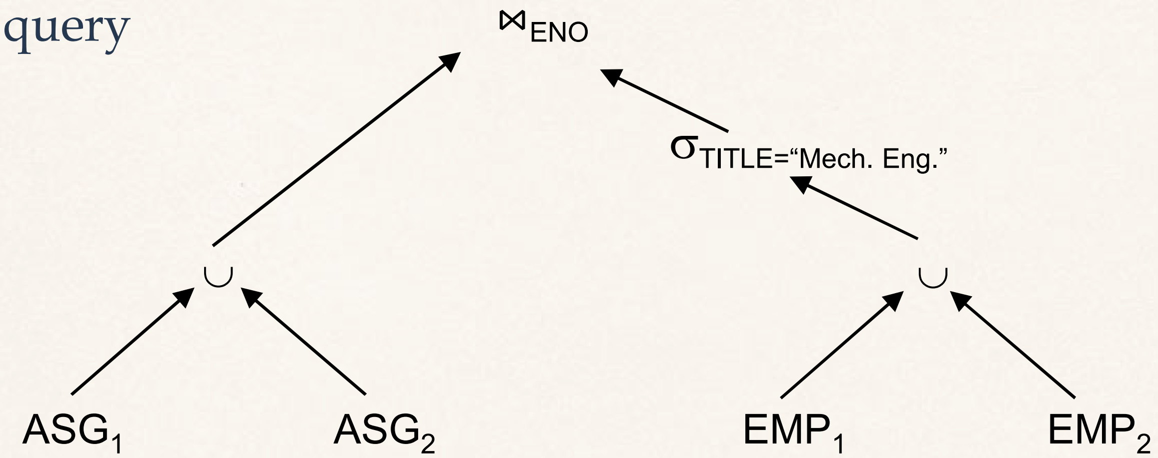
2. Reduction of selection over a relation fragmented with HF



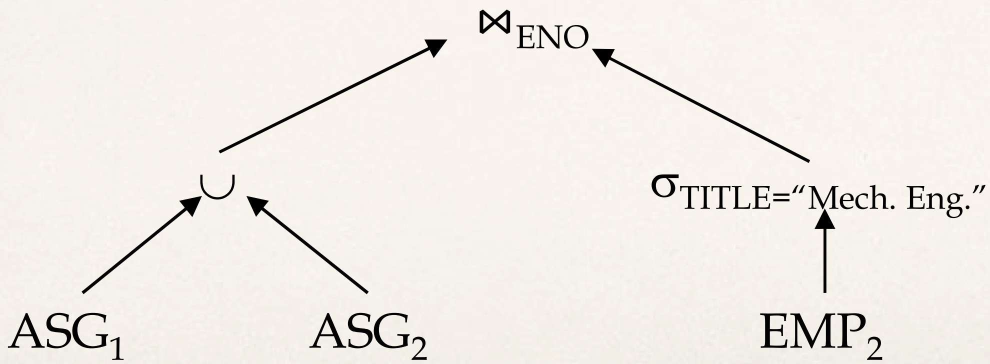
3. Reduction of join over a relation fragmented with DHF

Complex reduction for PHF and DHF

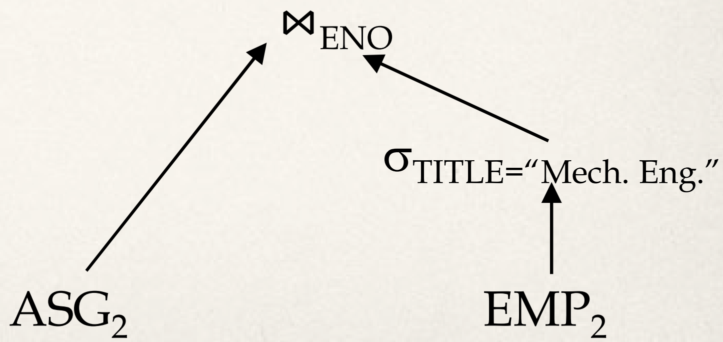
1. Generic query



2. Reduction of selection over a relation fragmented with HF



3. Reduction of join over a relation fragmented with DHF



Reduction for Hybrid Fragmentation

- Combine the rules already specified
  - Remove empty relations generated by contradicting predicates (inside selections or joins) on horizontal fragments
  - Remove useless relations generated by projections on vertical fragments
  - Distribute joins/selections/projections over unions in order to isolate and remove useless operands

Reduction for Hybrid Fragmentation

Example

Consider the following hybrid fragmentation:

$EMP_1 = \sigma_{ENO \leq "E4"} (\Pi_{ENO,ENAME} (EMP))$

$EMP_2 = \sigma_{ENO > "E4"} (\Pi_{ENO,ENAME} (EMP))$

$EMP_3 = \Pi_{ENO,TITLE} (EMP)$

Thus, the localization program for EMP is:

$EMP = (EMP_1 \cup EMP_2) \bowtie EMP_3$

Consider also the query:

```
SELECT ENAME
FROM   EMP
WHERE  ENO="E5"
```