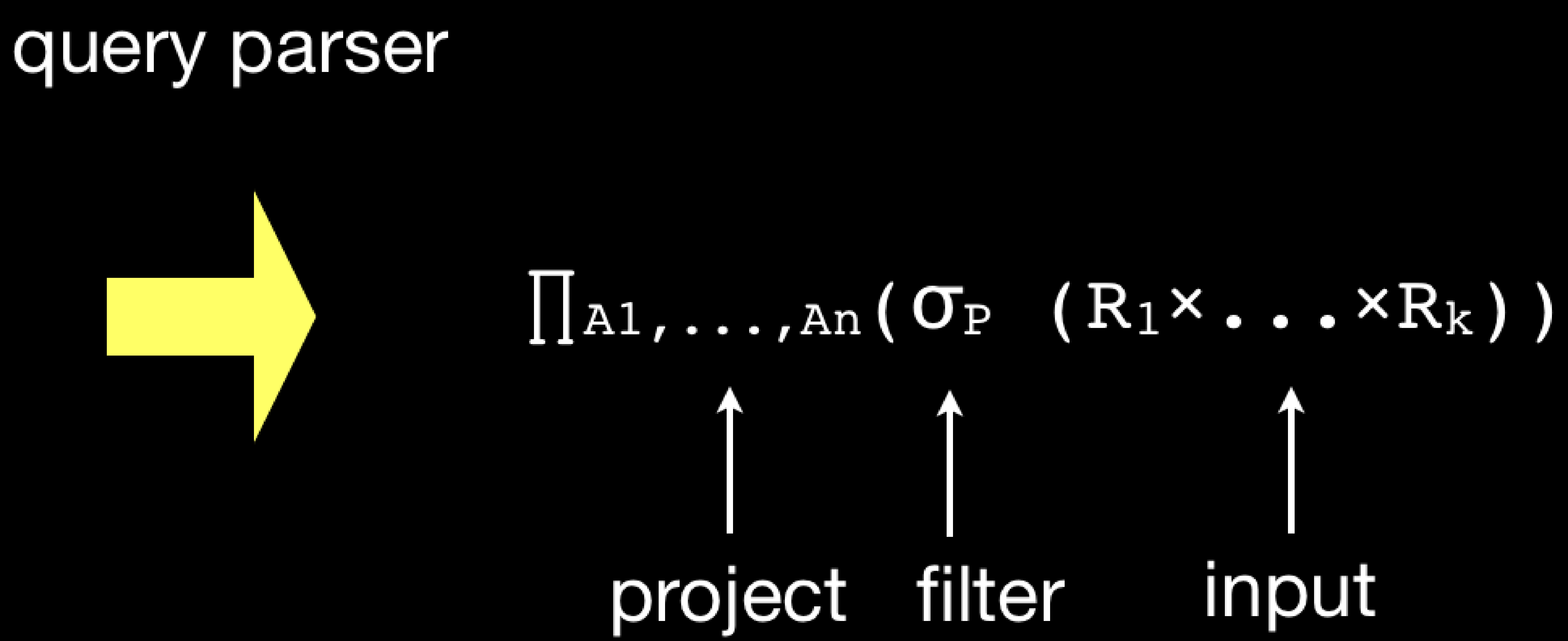
Query Parser

Input:

```
SELECT A1,..,An
FROM R1,..,Rk
WHERE P;
```



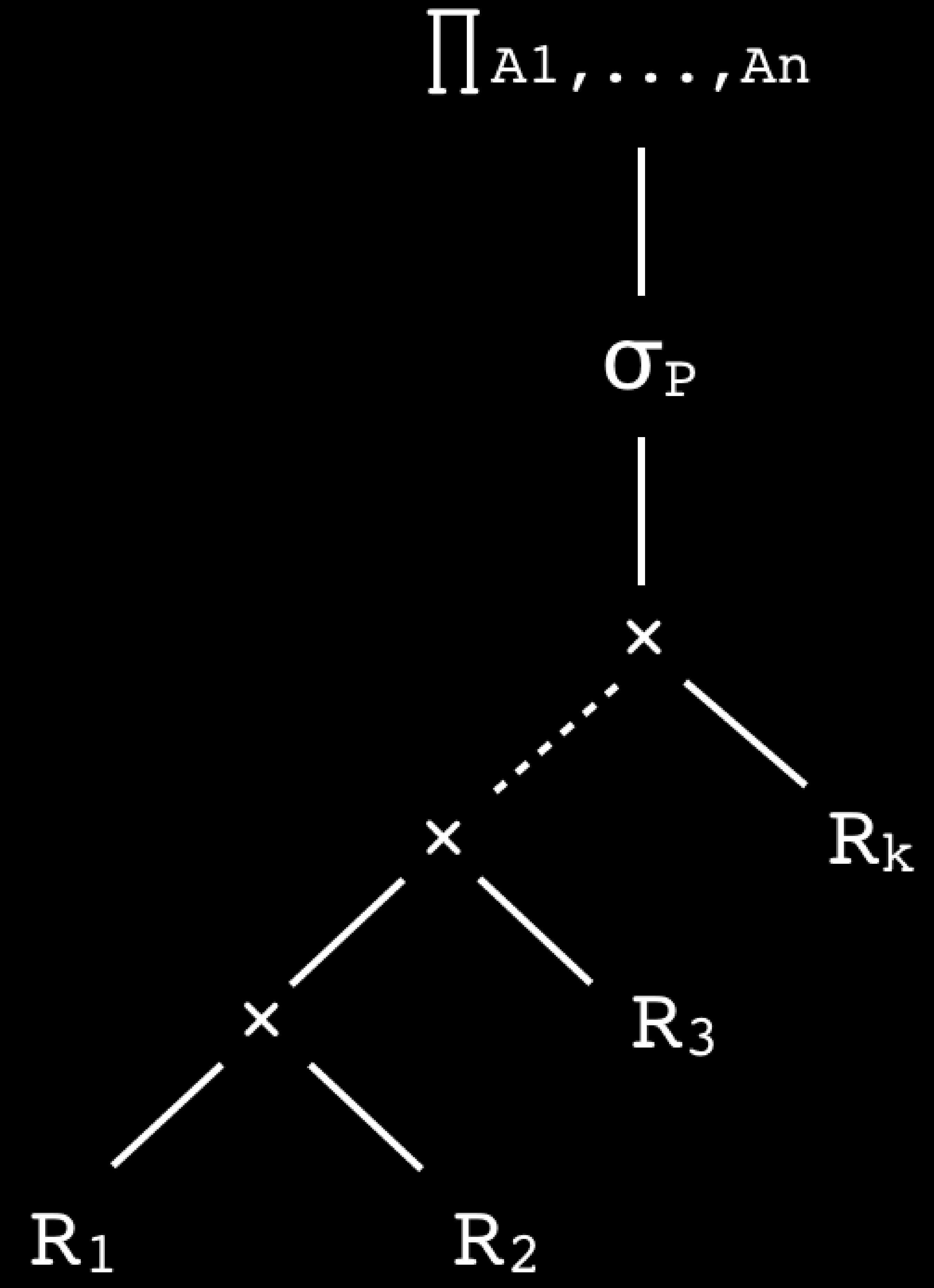
query parser

$$\prod_{A1,...,An}(\sigma_P\ (R_1\times...\times R_k))$$

Canonical Form of a Query

Input:

$\prod_{A1,...,An}(\sigma_P\ (R_1\times...\times R_k))$

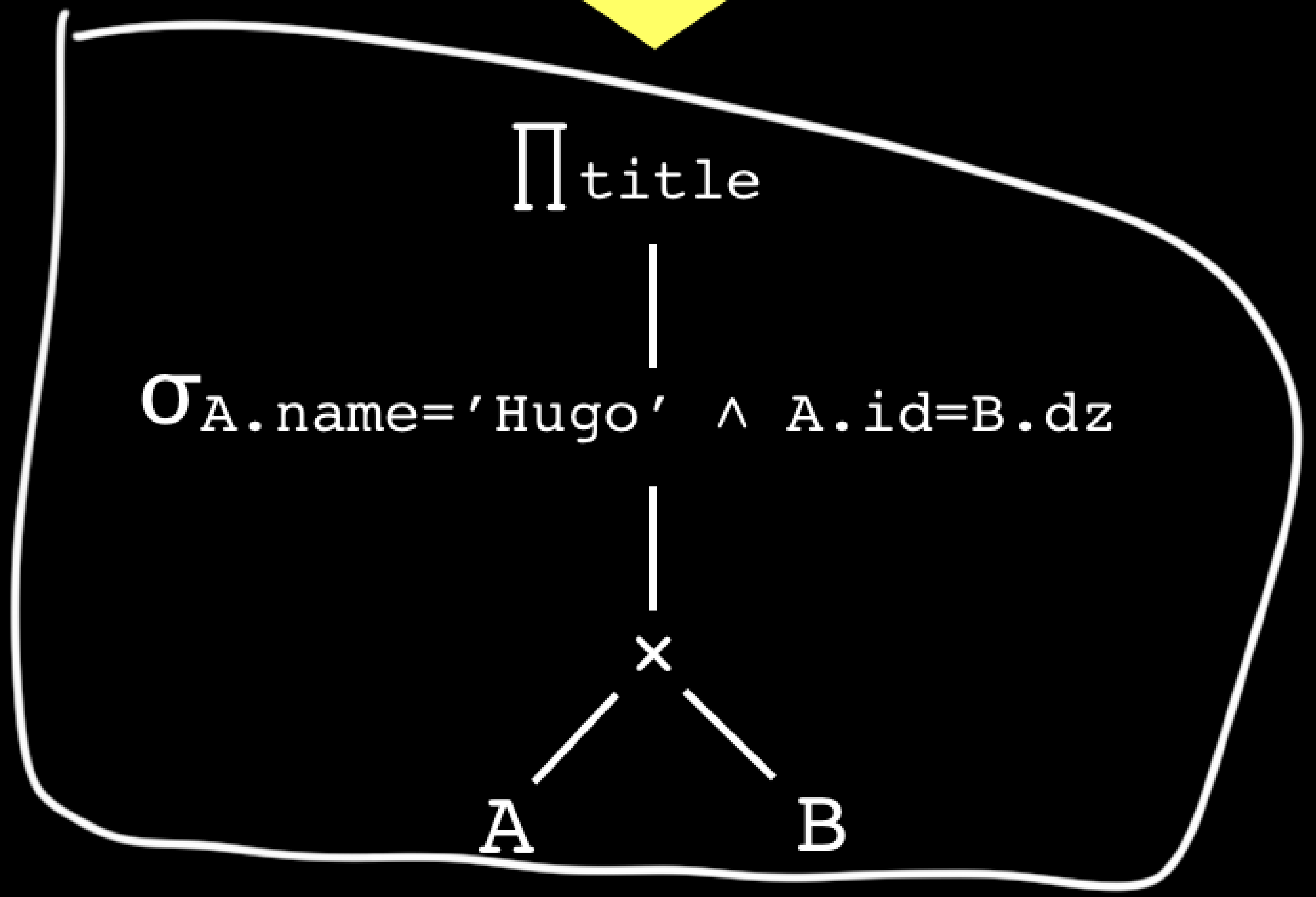


DAG , e.g. self-joins, nested queries

Example for Both Transformation Steps

```
SELECT title
FROM   A,B
WHERE  A.name ='Hugo' AND A.id = B.dz;
```

$$\Pi_{\text{title}}(\sigma_{\text{A.name='Hugo'} \wedge \text{A.id=B.dz}}(A \times B))$$



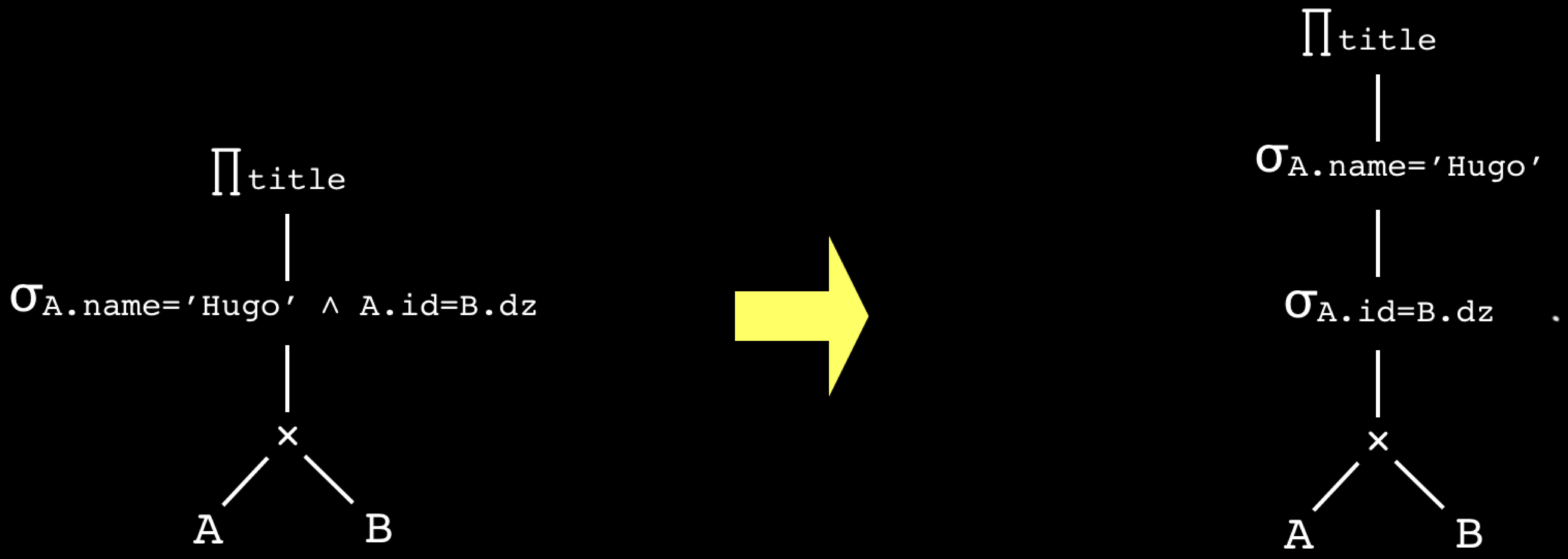
rules

Rule42 (Dag dag)

→ Dag

rule-based optimization
(heuristic)

Break Up Predicates



$\prod_{\texttt{title}}$

$\sigma_{\texttt{A.name='Hugo'} \wedge \texttt{A.id=B.dz}}$

×

A

B

$\prod_{\texttt{title}}$

$\sigma_{\texttt{A.name='Hugo'}}$

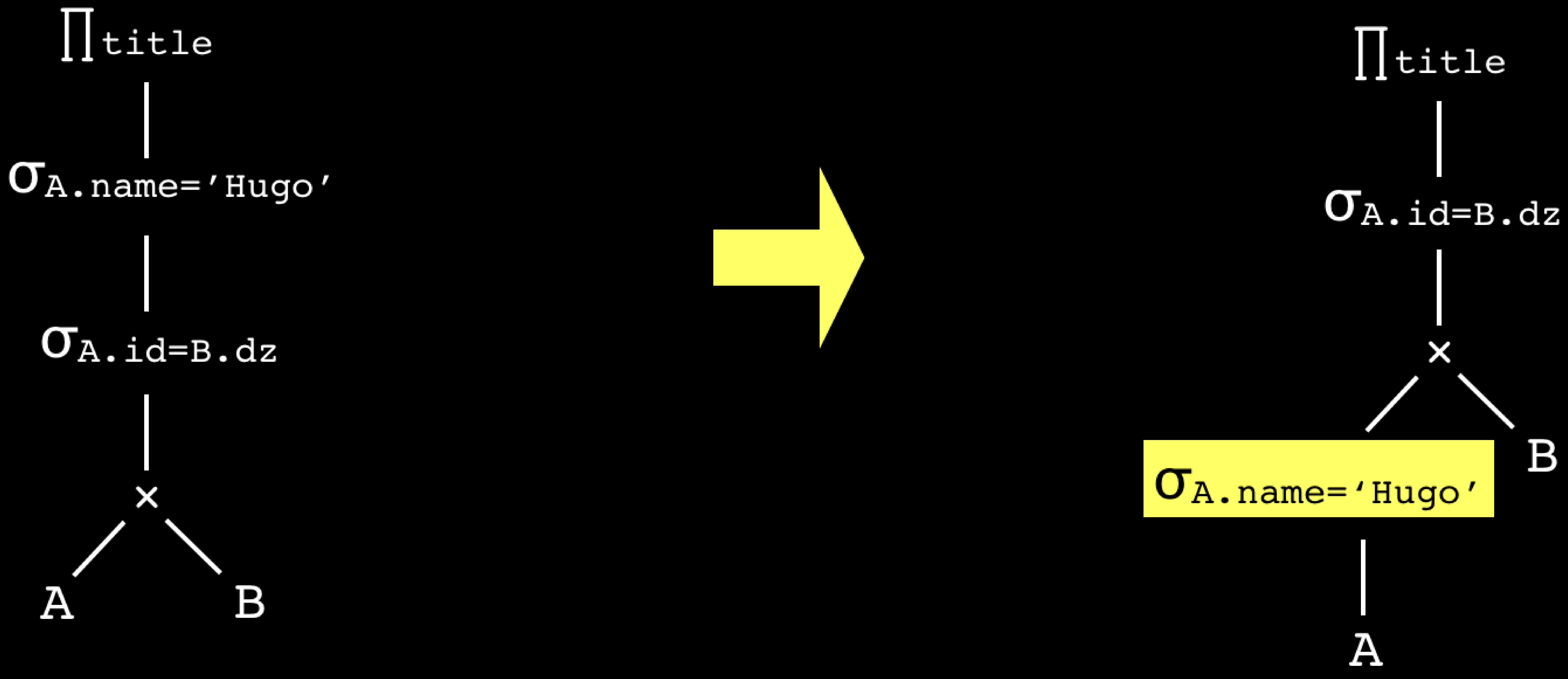
$\sigma_{\texttt{A.id=B.dz}}$

×

A

B

Push Down Selections



$\prod_{\texttt{title}}$

$\sigma_{\texttt{A.name='Hugo'}}$

$\sigma_{\texttt{A.id=B.dz}}$

×

A

B

$\prod_{\texttt{title}}$

$\sigma_{\texttt{A.id=B.dz}}$

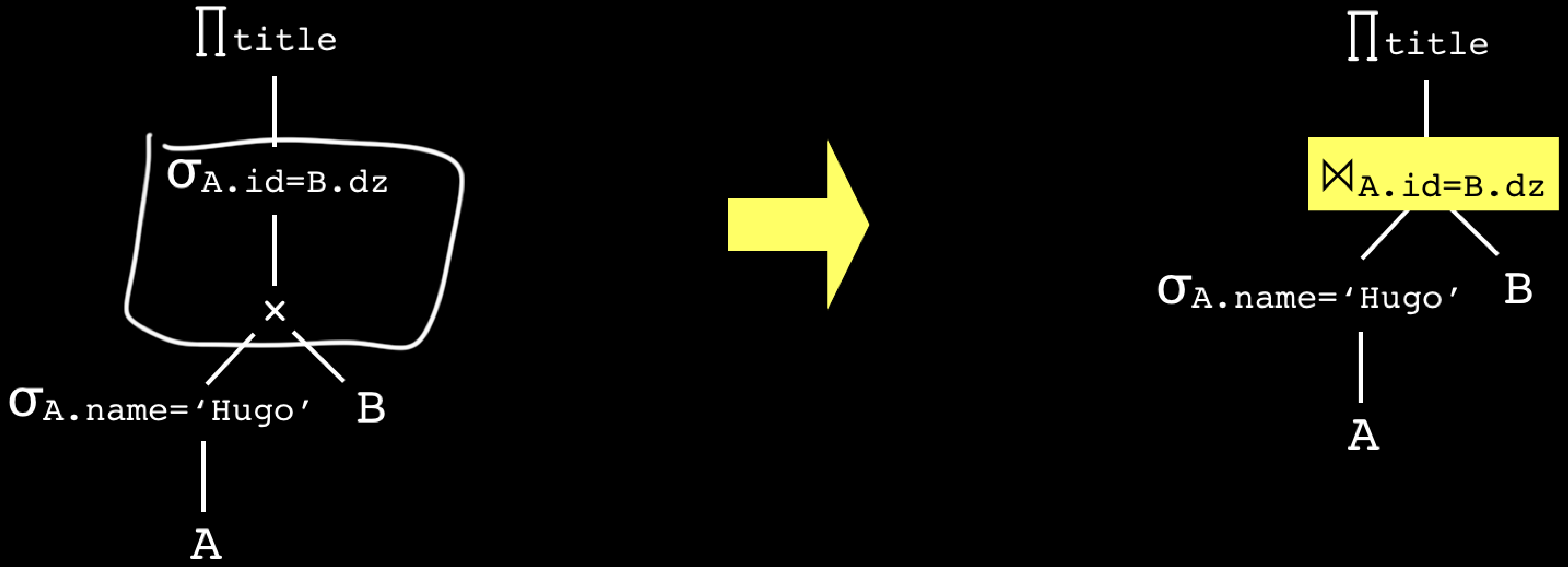
×

$\sigma_{\texttt{A.name='Hugo'}}$

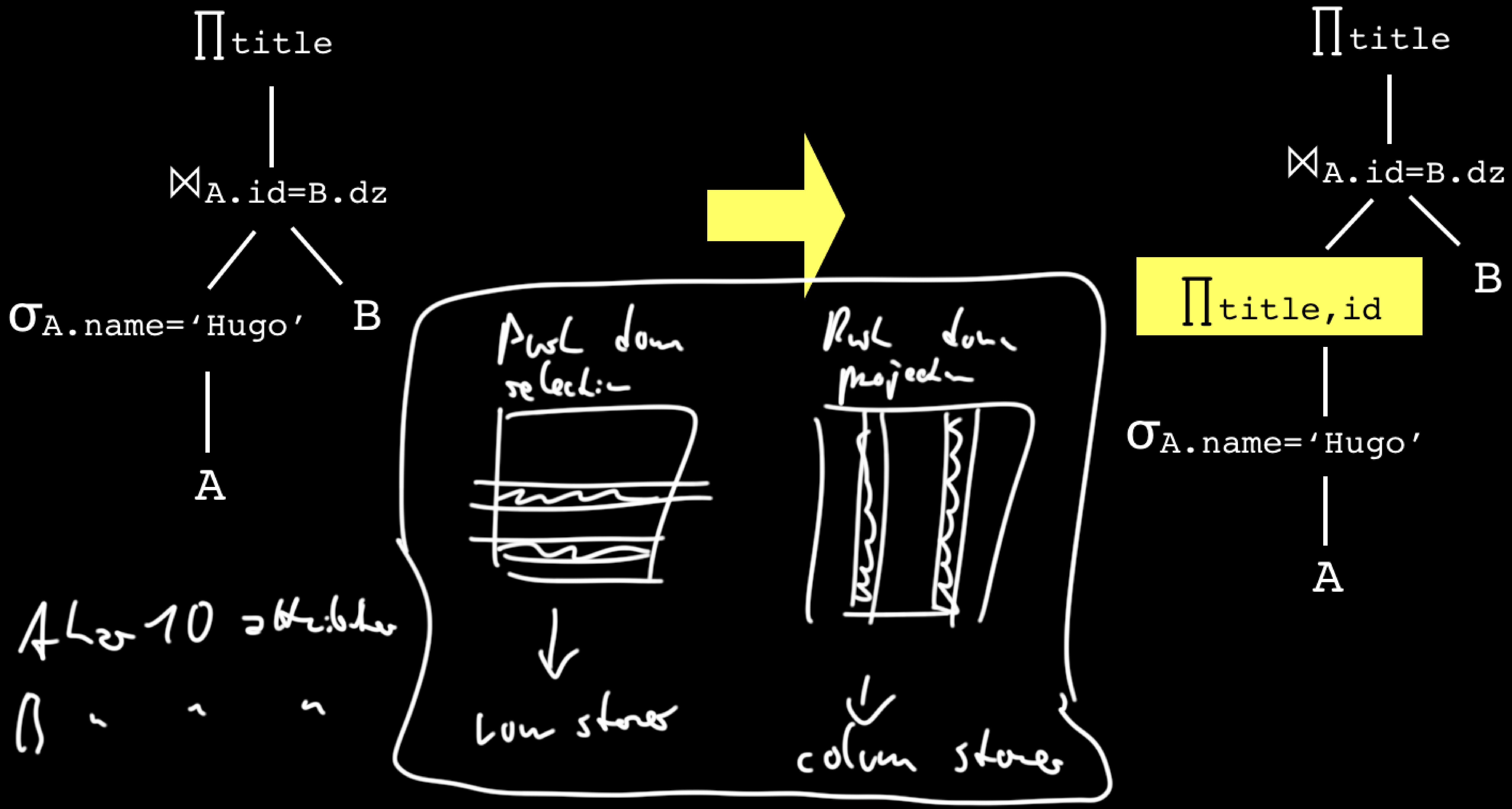
B

A

Build Joins



Push Down Projections



Π_{title}

$\bowtie_{\text{A.id=B.dz}}$

$\sigma_{\text{A.name='Hugo'}}$

B

A

Π_{title}

$\bowtie_{\text{A.id=B.dz}}$

$\Pi_{\text{title,id}}$

B

$\sigma_{\text{A.name='Hugo'}}$

A

A has 10 attributes

B " " "

Most Important Rules

1. push down selections and projections

2. combine selections and cross products into joins

3. insert additional projections