

# Processamento e Otimização de Consultas

Banco de Dados

Profa. Dra. Cristina Dutra de Aguiar Ciferri

# Motivação

- Consulta
  - pode ter sua resposta computada por uma variedade de métodos (geralmente)
- Usuário (programador)
  - sugere uma estratégia para achar a resposta, independentemente de ser a estratégia mais eficiente
- SGBD
  - responsável por transformar a consulta realizada pelo usuário em uma consulta **equivalente mais eficiente**

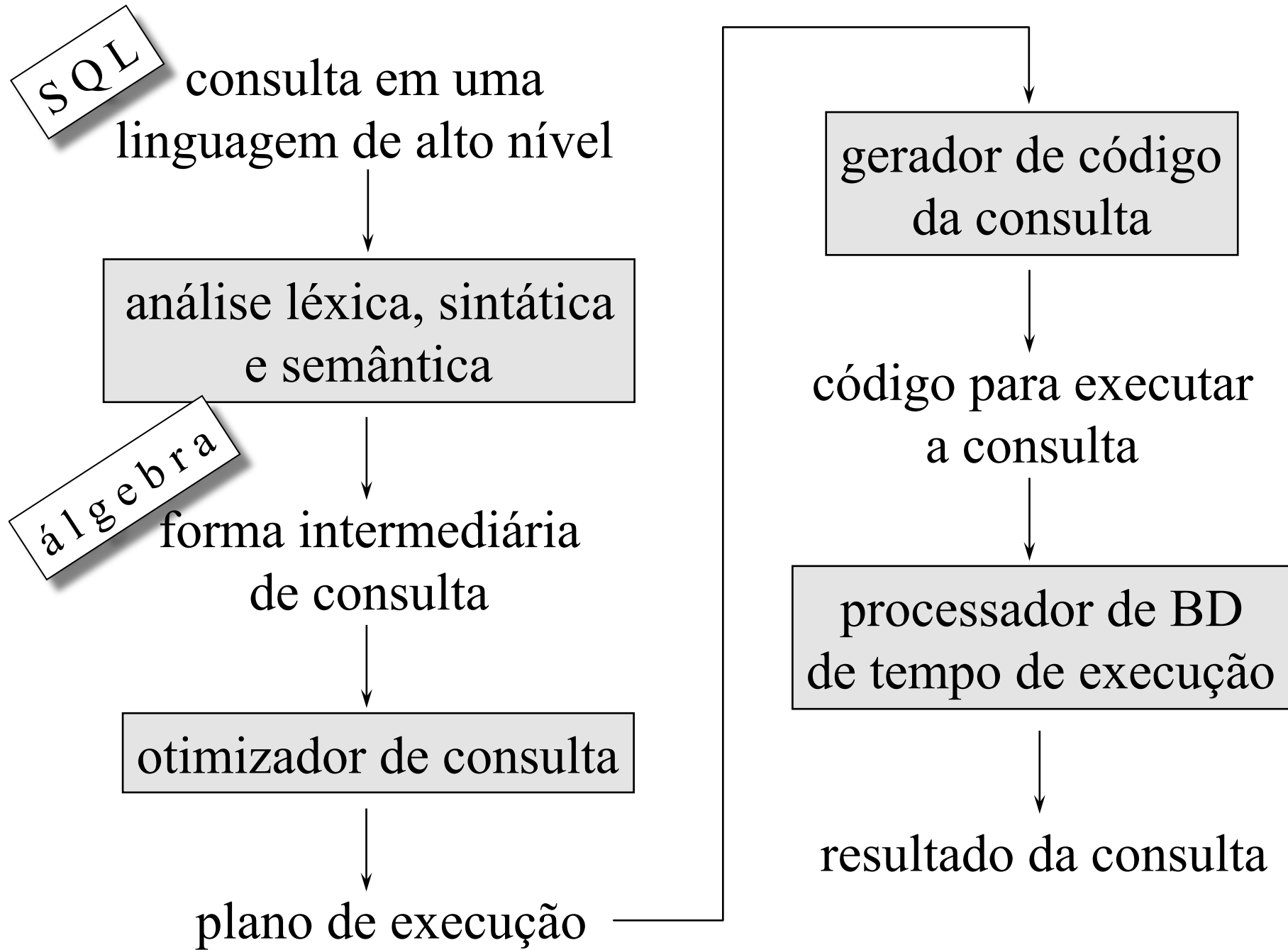
# Processamento de Consultas

- Característica
  - melhoria da estratégia para processamento de uma consulta
  - não apresenta uma estratégia **ótima**
  - porém apresenta uma estratégia **eficiente**
- Modelo relacional
  - facilita a otimização da consulta
  - permite que uma consulta seja expressa inteiramente em uma linguagem de consulta relacional (i.e., SQL) sem o uso de uma linguagem hospedeira

# Processamento de Consultas

- Técnica utilizada para processar, otimizar e executar consultas de alto nível
- Objetivo:
  - produzir uma **estratégia de consulta** para recuperar o resultado da mesma

**plano para:** executar a consulta  
acessar os dados  
armazenar resultados intermediários



# Detalhamento

- Análises (léxica, sintática, semântica)
  - verificam a sintaxe da consulta
  - verificam se os nomes das relações da consulta são os mesmos nomes presentes no BD
  - substituem referências ao nome de uma visão por expressões da álgebra, a fim de computar essa visão

# Detalhamento

- Forma intermediária de consulta
  - utiliza estrutura de dados de árvore ou grafo (**árvore de consulta** ou grafo de consulta)
- Módulo de otimização de consulta
  - produz um plano de estratégia de execução
    - ◆ indica qual o **plano de execução**
    - ◆ gera o código da consulta para executar tal plano

# Detalhamento

- Processador do BD de tempo de execução
  - executa o código da consulta (compilado ou interpretado)
  - produz o resultado da consulta



# Fases

- Fase 0
  - nível de SQL
  - consiste em transformar a consulta expressa em SQL em uma árvore de consulta expressa em álgebra relacional

# Fases

- Fase 1
  - nível de álgebra relacional
  - consiste em aplicar heurísticas para converter uma árvore de consulta em uma árvore de consulta equivalente
  - consiste em encontrar uma expressão que seja equivalente à expressão dada, mas que seja mais eficiente na sua execução

# Fases

- Fase 2
  - consiste na seleção de uma estratégia detalhada para o **processamento da consulta**
    - ◆ como a consulta será executada
    - ◆ quais índices serão escolhidos
    - ◆ qual a ordem de processamento das tuplas

# Árvore de Consulta (**Fase 0**)

- Estrutura de árvore que corresponde a uma expressão da álgebra relacional
- Representação
  - nós folhas
    - ◆ **relações** de entrada para a consulta
  - nós internos
    - ◆ **operações** da álgebra relacional
- Indica uma ordem específica das operações durante a execução de uma consulta

# Árvore de Consulta

- Execução
  - efetuar uma operação do nó interno sempre que os seus operandos estiverem disponíveis
  - substituir o nó interno pela relação que resulta da execução da operação
  - final da execução:
    - ◆ o nó raiz é executado
    - ◆ a relação resultado para a consulta é produzida

# Exemplo

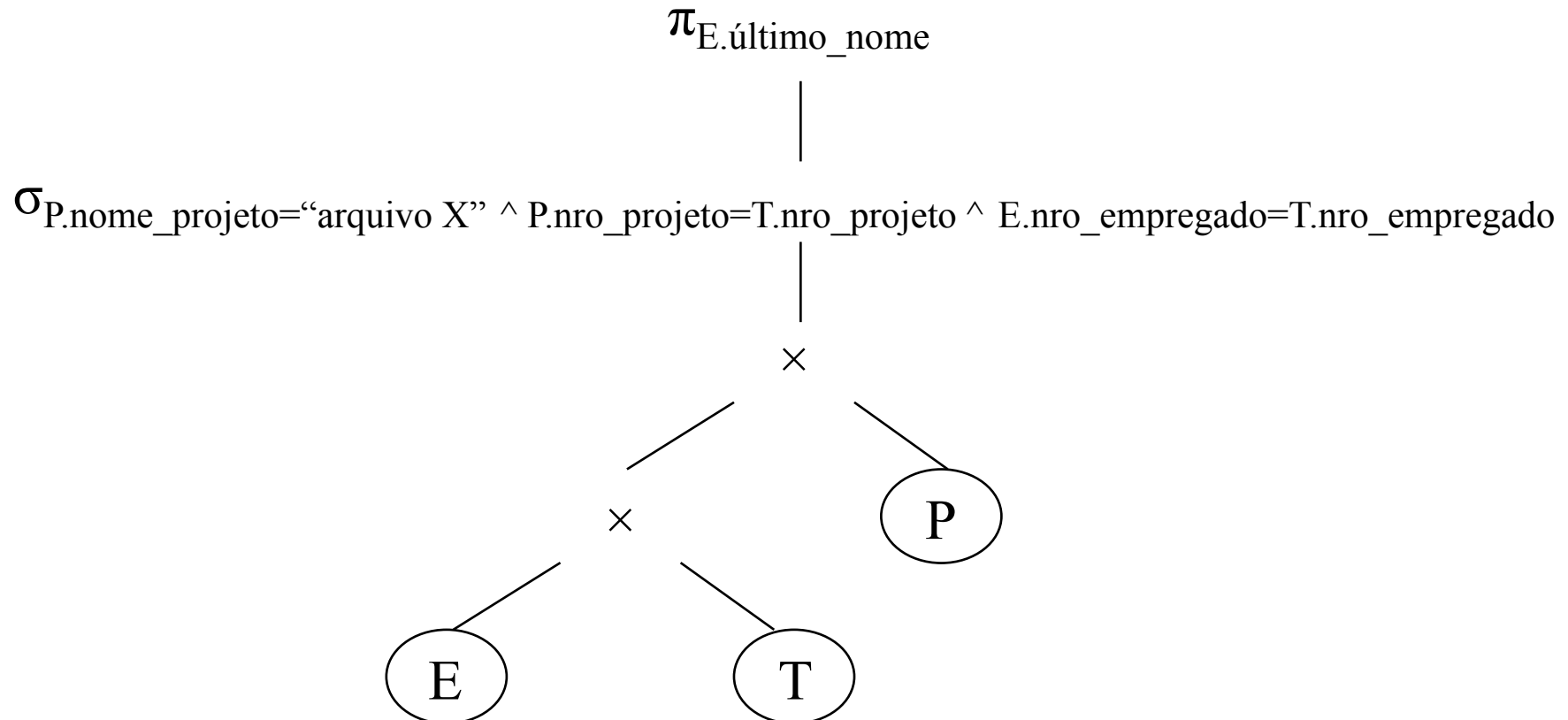
- Dado o seguinte comando SQL

```
SELECT E.último_nome  
FROM empregado E, trabalha T, projeto P  
WHERE P.nome_projeto = "arquivo X"  
      AND P.nro_projeto = T.nro_projeto  
      AND E.nro_empregado = T.nro_empregado
```

- Construa a árvore de consulta canônica

# Árvore de Consulta Canônica

árvore de consulta 1



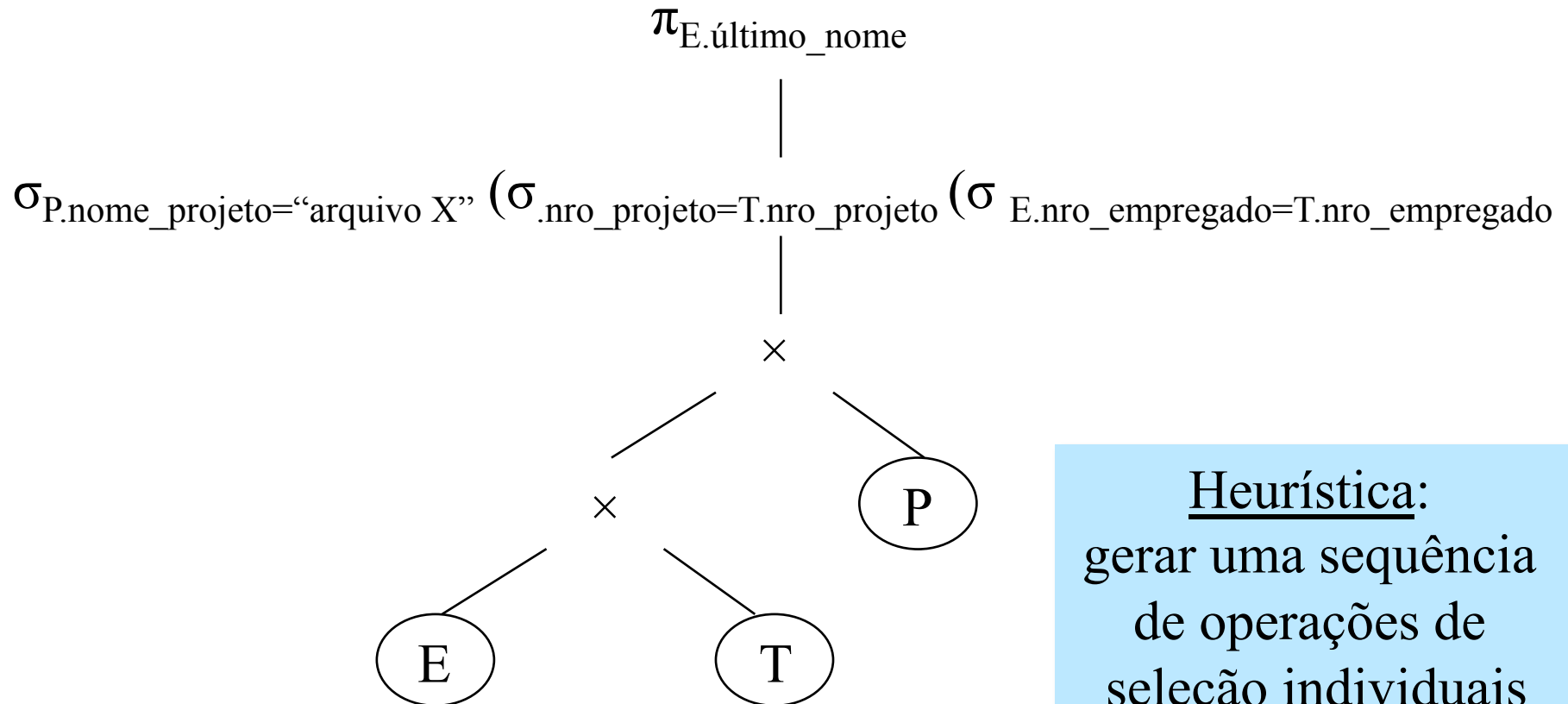
# Conversão da Árvore de Consulta (Fase 1)

- Observações
  - o produto cartesiano das relações empregado, trabalha e projeto produz uma grande relação, a qual provavelmente precisará ser armazenada em disco
  - acesso a disco
    - ◆ para ler as relações empregado, trabalha e projeto
    - ◆ para ler e escrever resultados intermediários
  - objetivo
    - ◆ reduzir o tamanho dos resultados intermediários



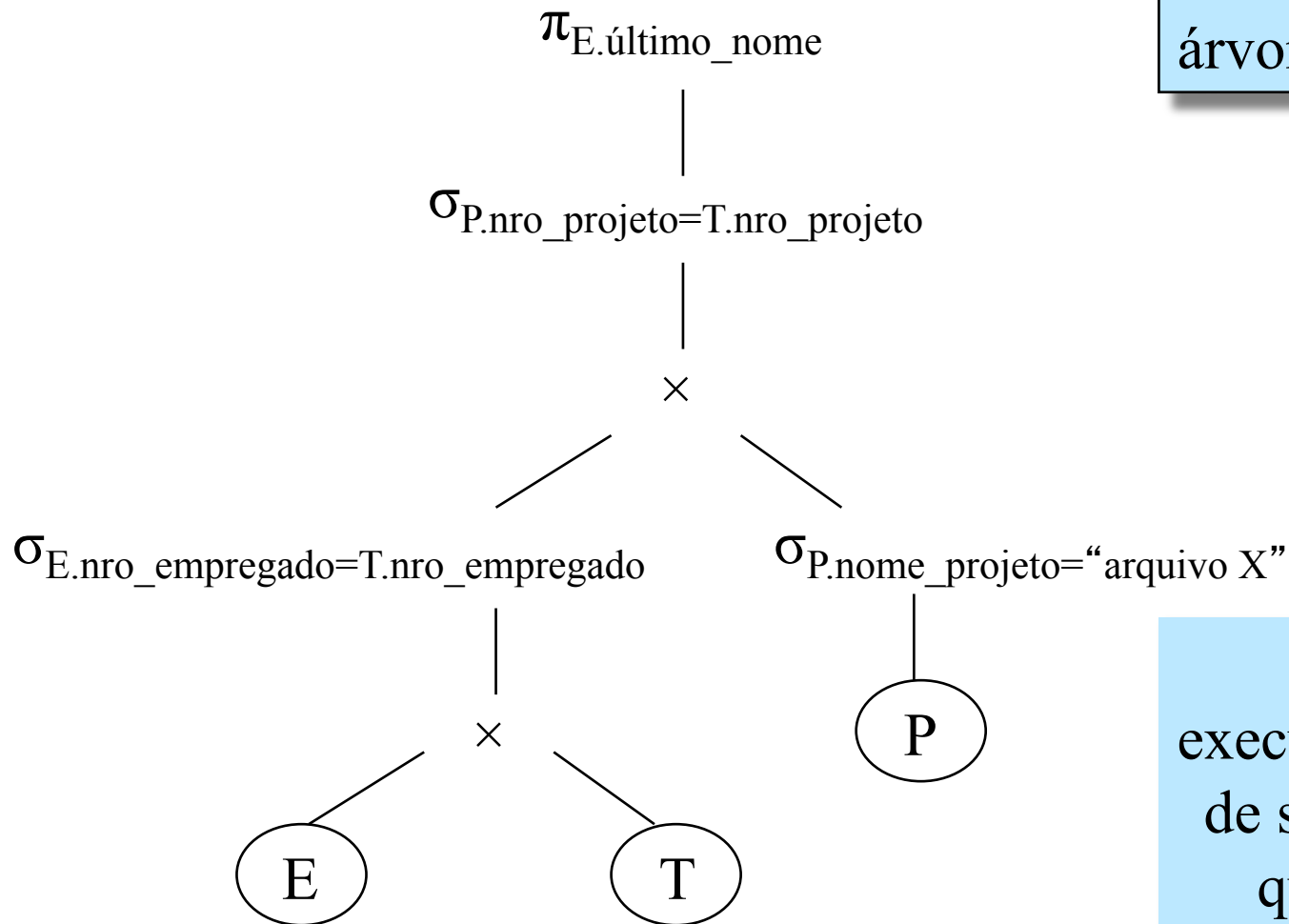
# Conversão da Árvore de Consulta

árvore de consulta 2



Heurística:  
gerar uma sequência  
de operações de  
seleção individuais  
*algoritmo passo 1*

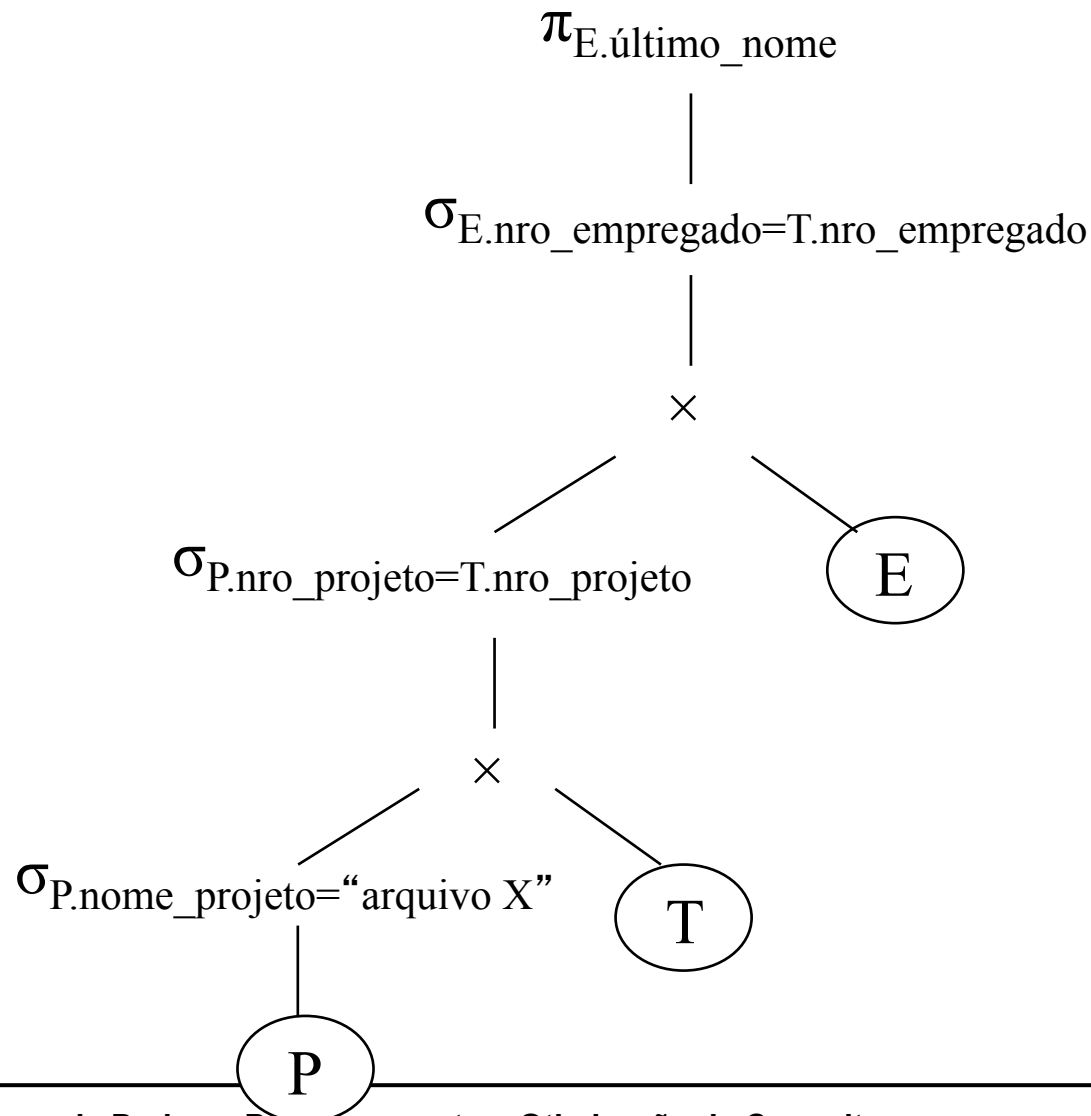
# Conversão da Árvore de Consulta



árvore de consulta 3

Heurística:  
executar as operações  
de seleção tão cedo  
quanto possível  
*algoritmo passo 2*

# Conversão da Árvore de Consulta



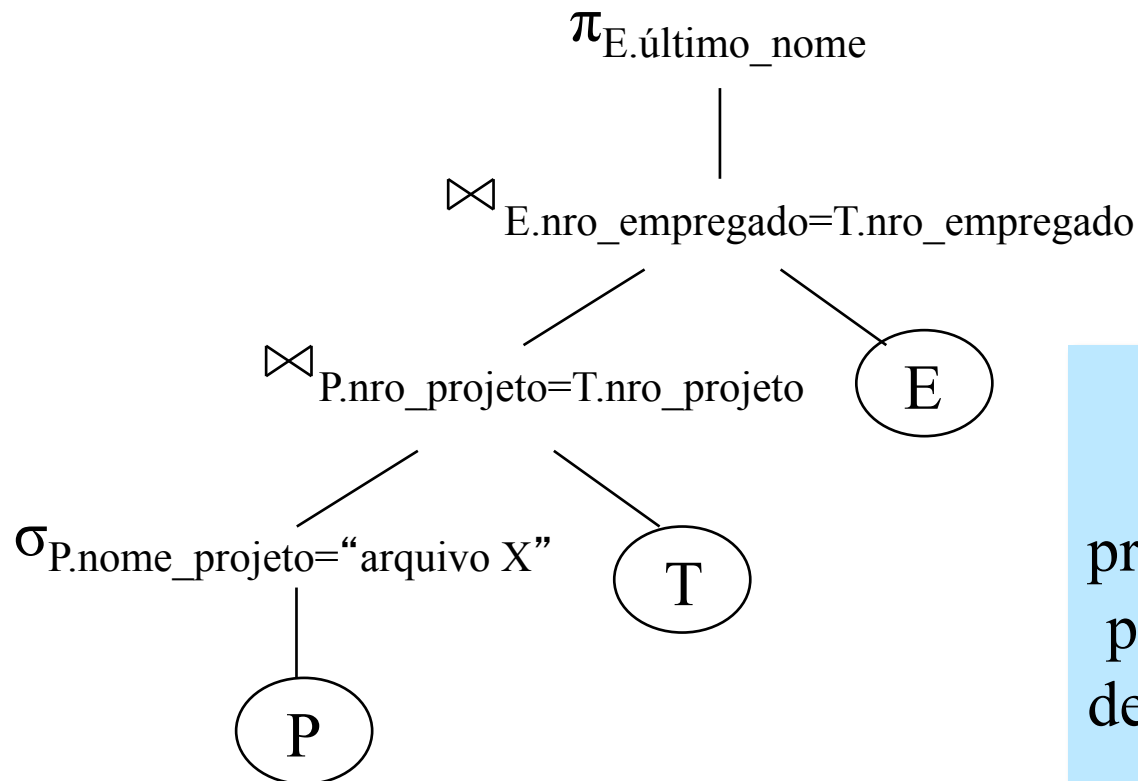
árvore de consulta 4

Heurística:  
diminuir os tamanhos  
das relações a serem  
utilizadas no produto  
cartesiano

*algoritmo passo 3*

# Conversão da Árvore de Consulta

árvore de consulta 5

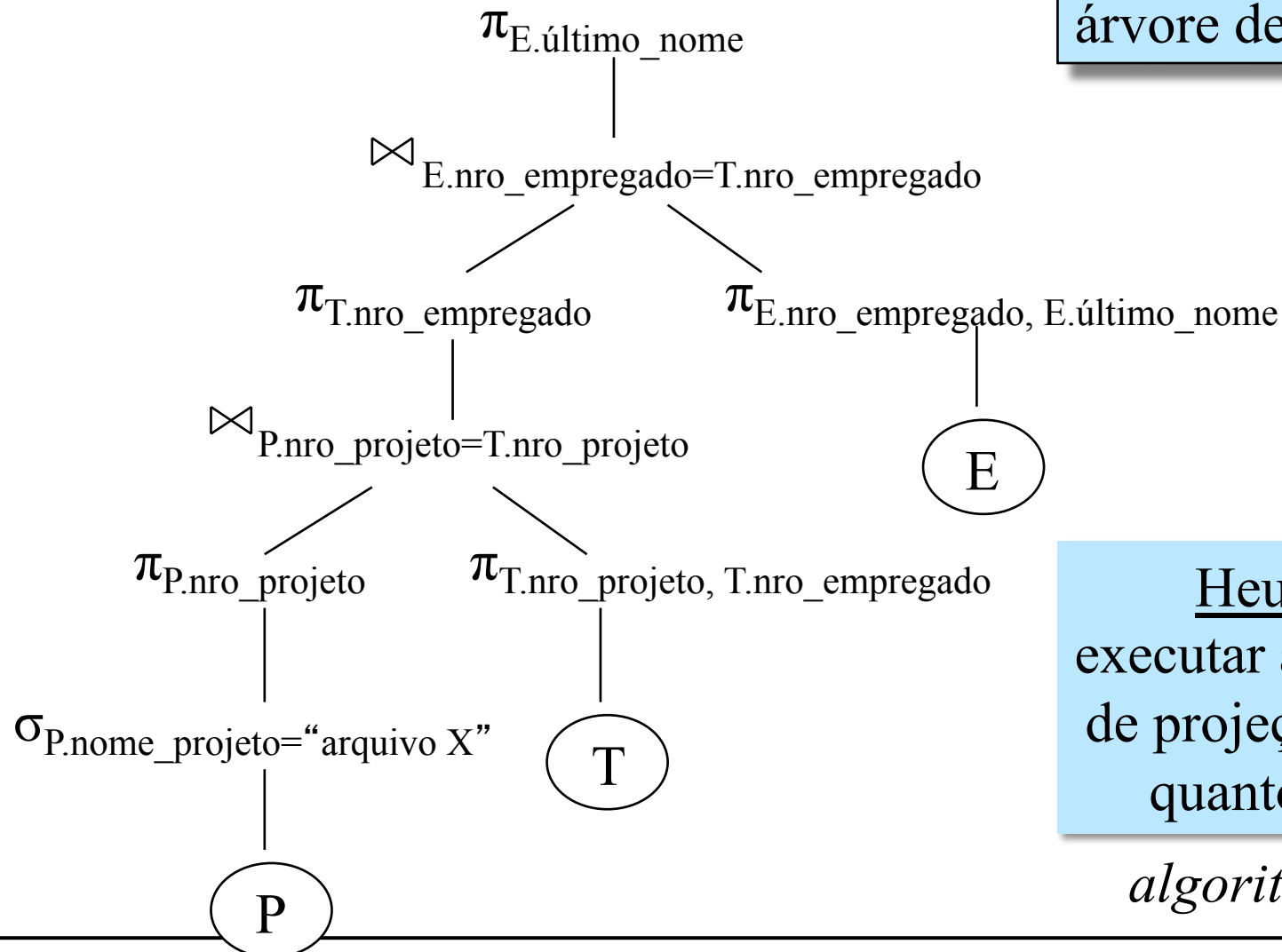


Heurística:  
substituir operações de produto cartesiano seguidas pelos respectivos critérios de seleção por operações de junção

*algoritmo passo 4*

# Conversão da Árvore de Consulta

árvore de consulta 6



Heurística:  
executar as operações  
de projeção tão cedo  
quanto possível

*algoritmo passo 5*

# Observação

- Diversos atributos de um esquema podem ser geralmente eliminados
- Atributos que não podem ser eliminados
  - aqueles que aparecem no resultado de uma consulta
  - aqueles que são necessários para processar operações subsequentes

# Heurísticas Básicas

- Aplicar primeiro as operações que reduzem o tamanho dos resultados intermediários
  - operações de seleção
    - ◆ reduzem o número de tuplas
  - operações de projeção
    - ◆ reduzem o número de atributos
- Aplicar primeiro as operações de seleção e de junção mais restritivas
  - reordenar os nós folha da árvore de consulta
  - evitar a operação de produto cartesiano
  - ajustar o restante da árvore de forma apropriada

# Custo de Consultas (Fase 2)

- Estratégia a ser escolhida depende
  - do tamanho de cada relação
  - da distribuição de valores dentro de colunas
  - custo
- Objetivo
  - estimar o **tamanho do resultado** (número de tuplas a serem retornadas) e o **custo da consulta**



# Otimização de Consultas baseada no Custo

- Características
  - utiliza técnicas de otimização tradicionais que percorrem o **espaço solução** para um problema
  - identifica uma solução que minimiza uma determinada **função custo**
  - depende de **estatísticas** armazenadas pelo SGBD sobre as relações e sobre os índices

# Componentes da Função Custo

- Custo de acesso à **memória secundária**
  - custo para buscar, ler e escrever blocos de dados que residem em disco
- Custo de **armazenamento**
  - custo para armazenar quaisquer arquivos intermediários gerados pela estratégia de execução da consulta
- Custo de **computação**
  - custo para a realização de operações em memória principal (i.e., *buffers*)

# Componentes da Função Custo

- Custo do **uso da memória**
  - custo relacionado ao número de *buffers* de memória principal necessários durante a execução da consulta
- Custo de **comunicação**
  - custo de transmitir uma consulta e os seus resultados do *site* do banco de dados até o *site* ou terminar na qual a consulta foi originada

# Componentes da Função Custo

Característica	Ênfase
BD volumosos	minimizar o custo de acesso à memória secundária
BD pequenos	minimizar o custo de computação
BD distribuídos	minimizar também o custo de comunicação

- Problema NP-Completo
  - a grande maioria das funções custo enfoca apenas um único fator: **custo de acessos à memória secundária** (i.e., disco)

# Como Processar Operações

- Seleção
  - busca linear (busca sequencial)
  - busca binária
  - índice primário
  - índice secundário
  - índice de agrupamento (cluster)
  - função *hash*
  - árvore B e suas variações

# Como Processar Operações

- **Projeção**
  - recupera os registros e considera somente os atributos desejados
- **Ordenação**
  - sort-merge externo
- **Produto Cartesiano**
  - recupera e processa todos os registros

# Como Processar Operações

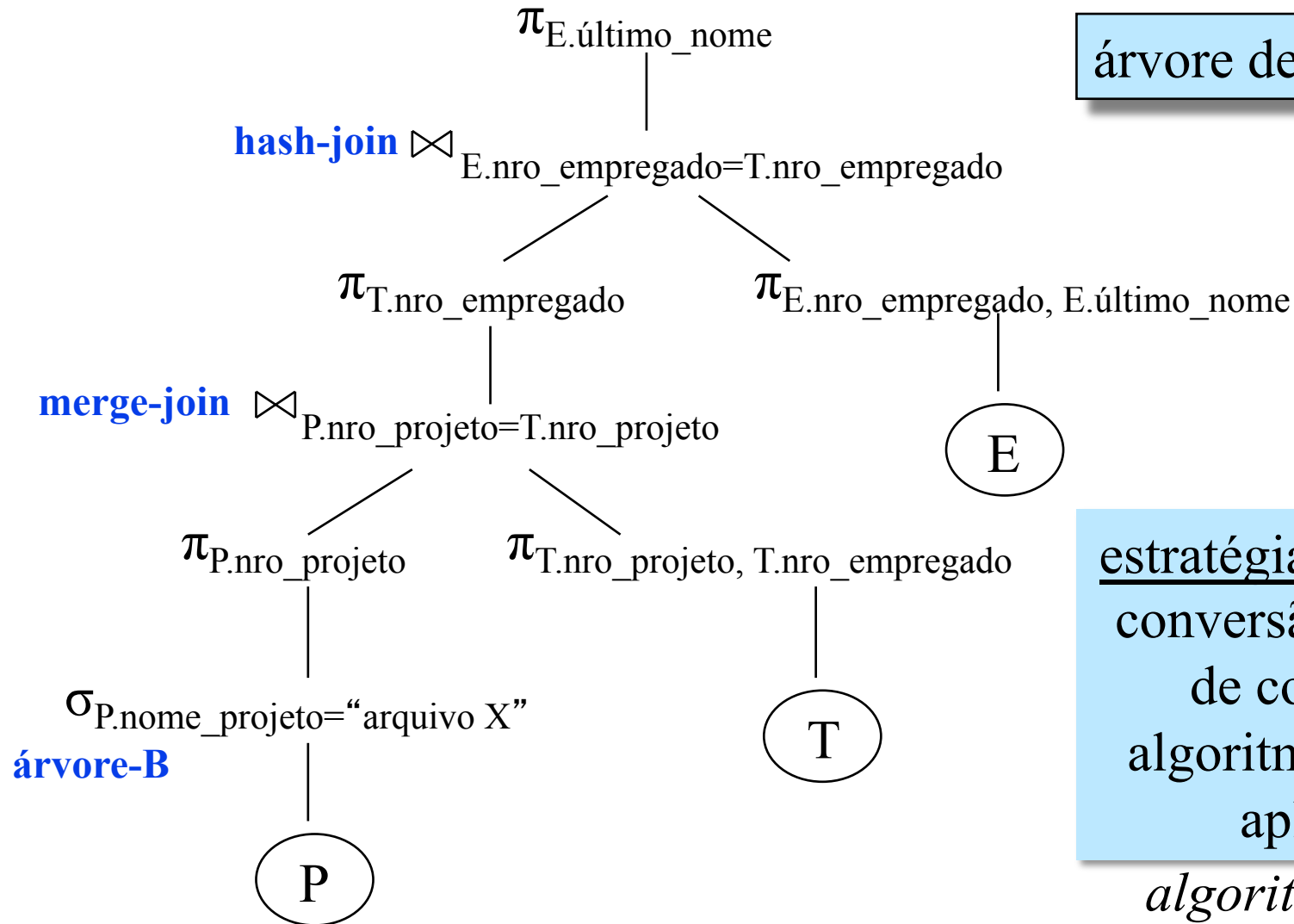
- **Junção**
  - junção de laço aninhado simples
  - junção de laço aninhado de blocos
  - junção de laço aninhado indexada
  - *sort-merge* junção
  - *hash* junção
- **União, Intersecção e Diferença**
  - ordenação
  - função *hash*

A geração de expressões é apenas parte do processo de otimização de consultas. Cada operação na expressão pode ser implementada com **diferentes algoritmos**. Uma estratégia de consulta define exatamente que algoritmo é utilizado para cada operação e como a execução das operações é coordenada.



# Exemplo

árvore de consulta 7



estratégia de consulta:  
conversão da árvore  
de consulta +  
algoritmos a serem  
aplicados

*algoritmo passo 6*

# Materialização *versus* Pipelining

- Materialização
  - o resultado de uma operação é armazenado em disco como uma relação temporária
  - a relação temporária é lida do disco para ser utilizada como entrada para a próxima operação
- Pipelining
  - combina diversos algoritmos que correspondem às operações individuais
  - utiliza como entrada de uma operação as tuplas que foram geradas pela operação anterior