

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**

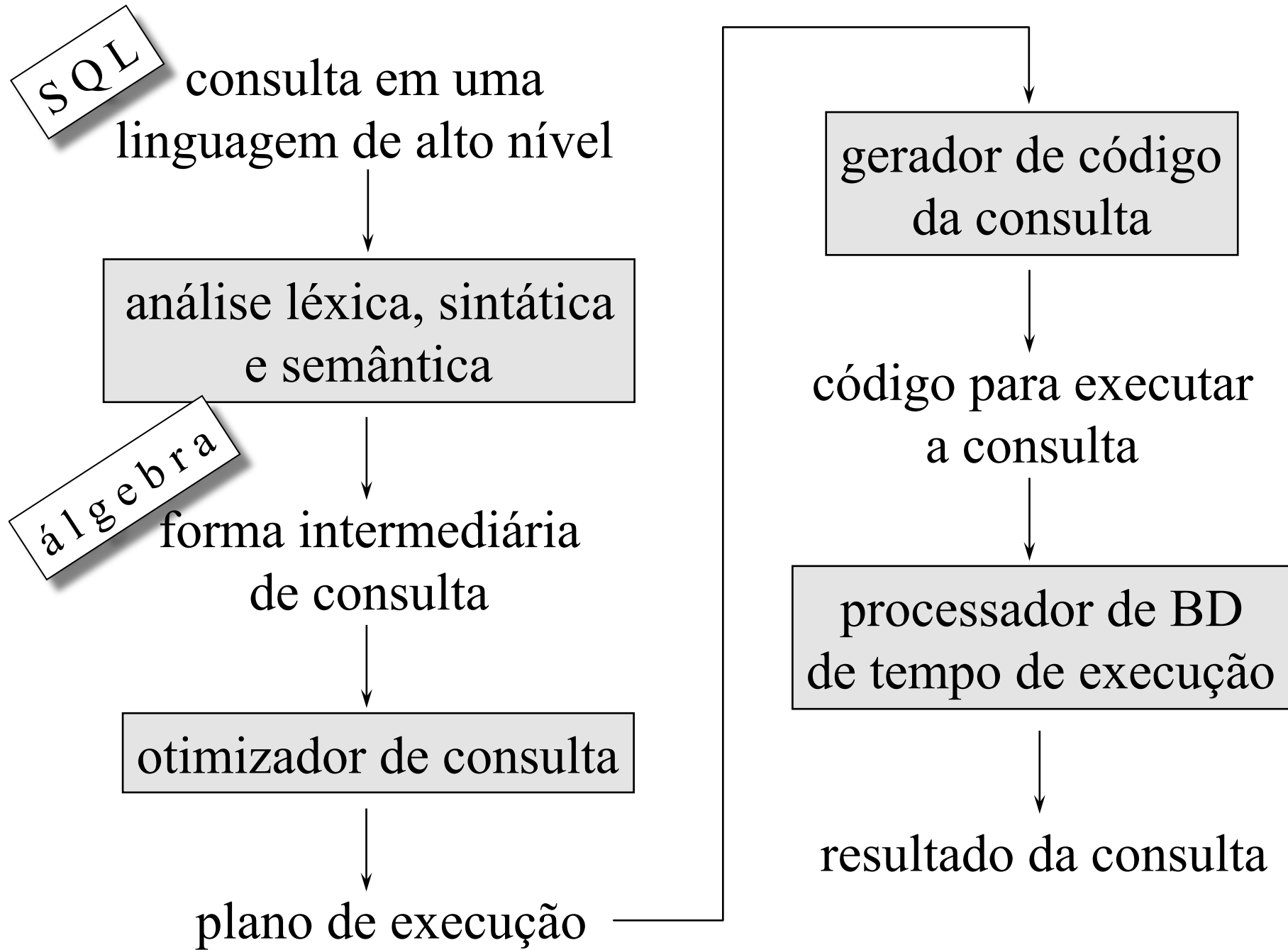
Processamento de Consultas

- 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

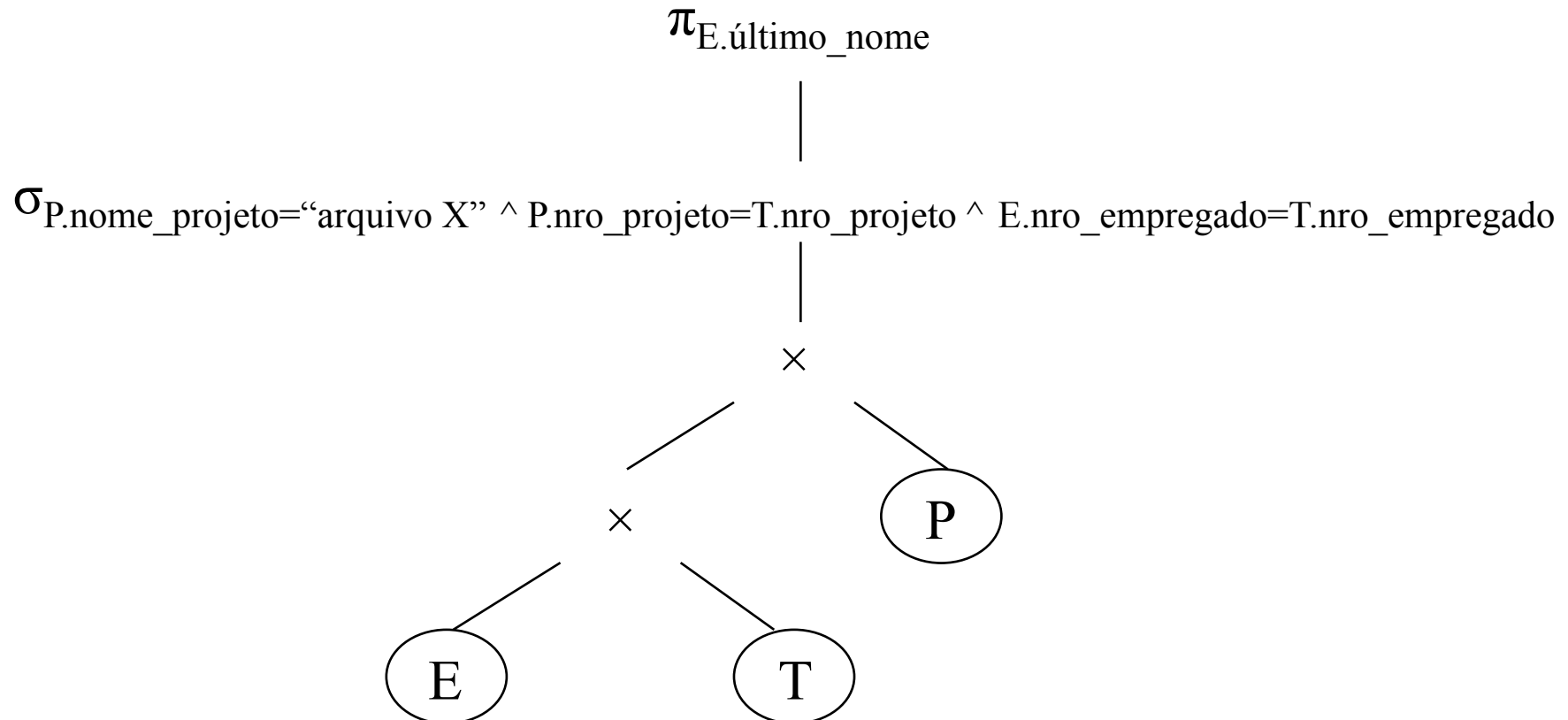
- Dado o seguinte comando SQL

```
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

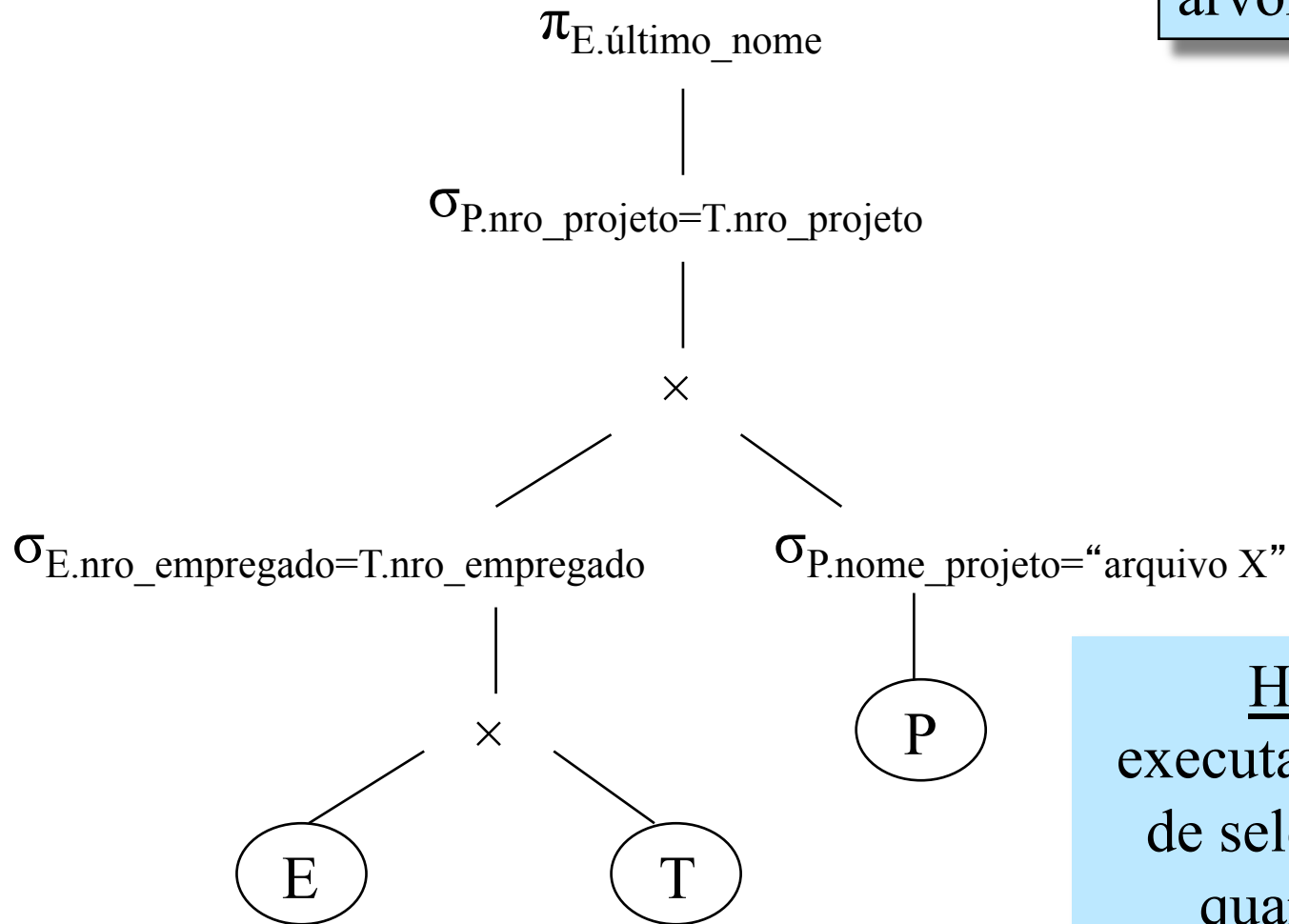


Conversão da Árvore de Consulta (Fase 1)

- 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

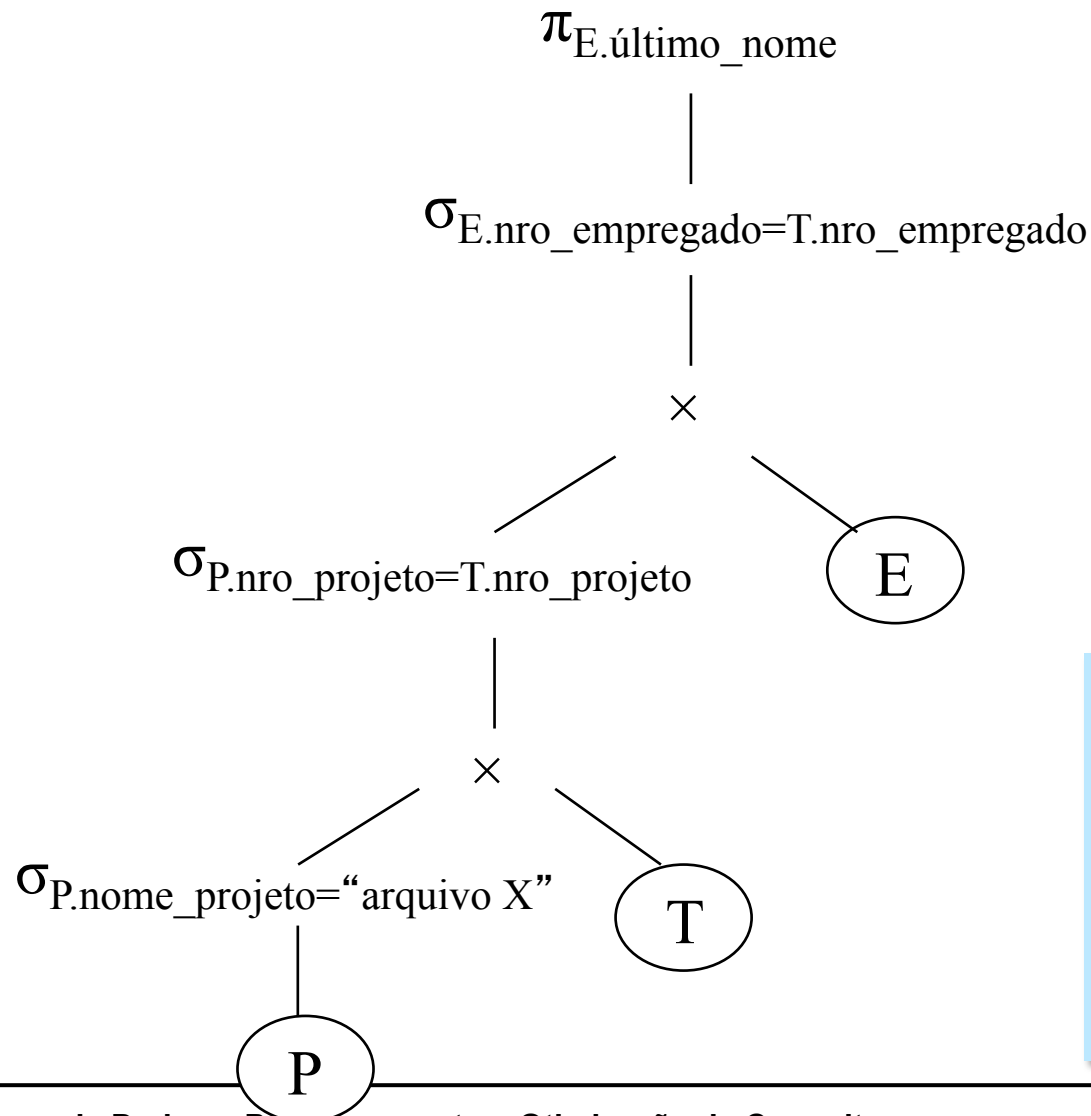
Conversão da Árvore de Consulta

árvore de consulta 2



Heurística:
executar as operações
de seleção tão cedo
quanto possível

Conversão da Árvore de Consulta

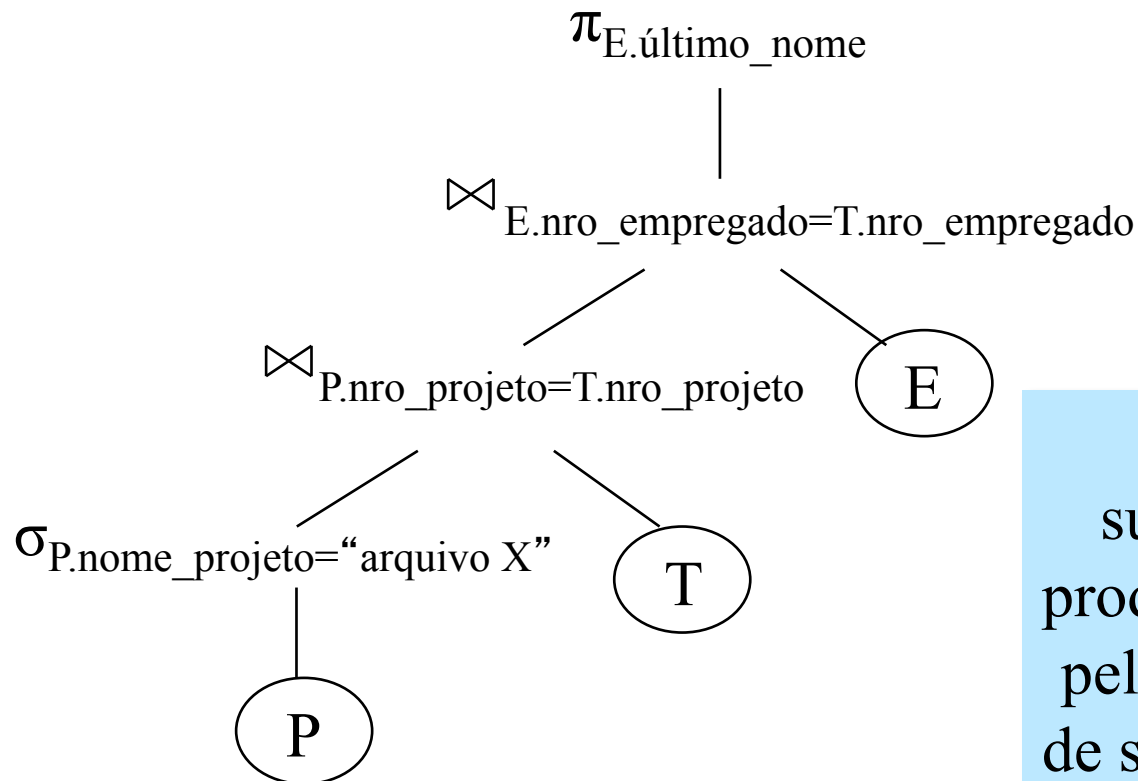


árvore de consulta 3

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

Conversão da Árvore de Consulta

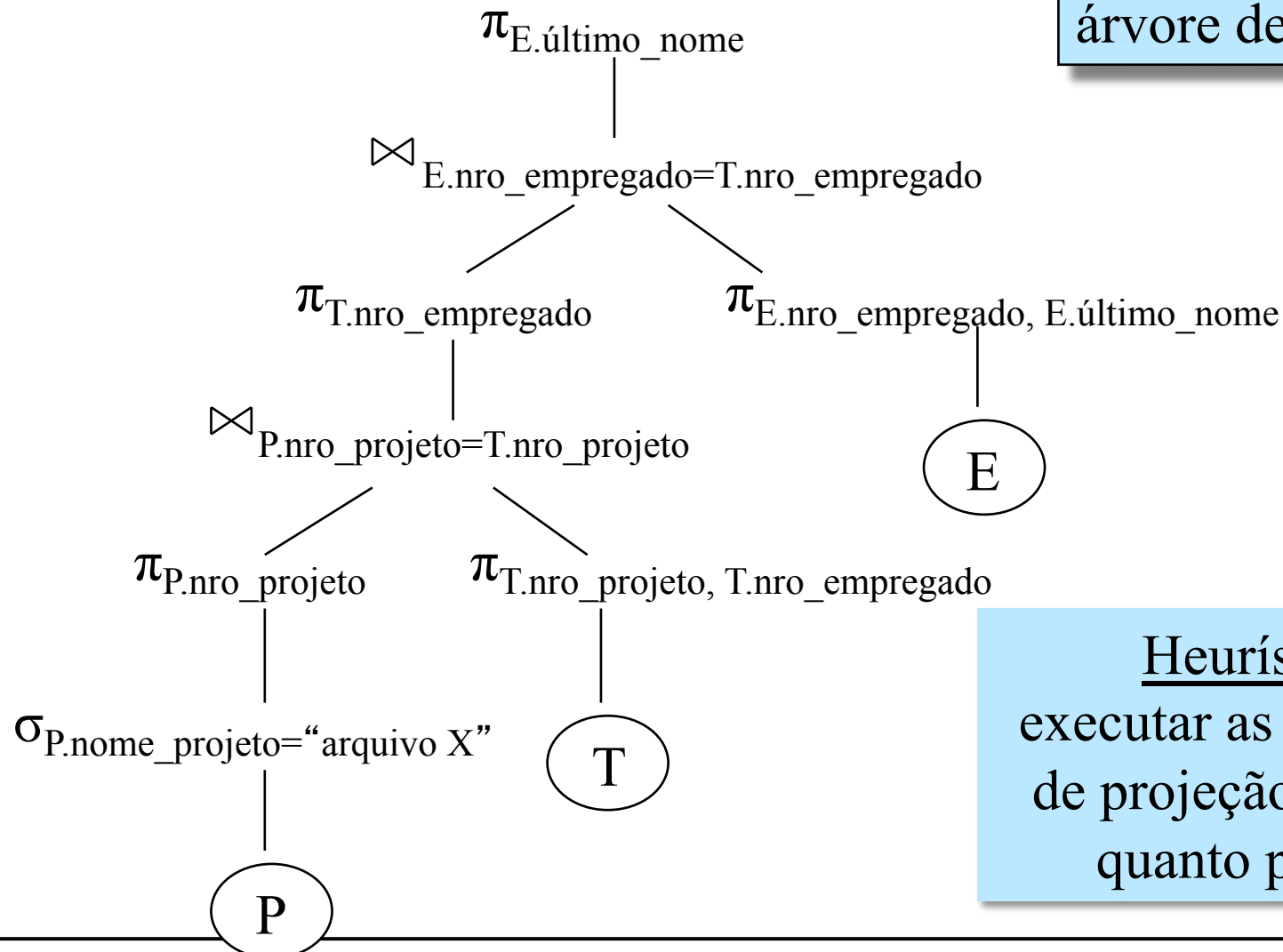
árvore de consulta 4



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

Conversão da Árvore de Consulta

árvore de consulta 5



Heurística:
executar as operações
de projeção tão cedo
quanto possível

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)

Estatísticas

- Para cada relação
 - r : número de registros
 - b : número de blocos que contêm os registros
 - R : tamanho de um registro em *bytes*
 - bfr : número de registros que cabe em um bloco
 - d : número de valores distintos para um atributo
 - sl : fração de registros que satisfazem à condição de igualdade no atributo

Estatísticas

- Para cada relação
 - s : número médio de registros que satisfazem à condição de igualdade no atributo
 - ♦ se o atributo é chave primária
então $s = 1$
senão $s = (r/d)$ // d valores distintos distribuídos
// uniformemente

s : cardinalidade de seleção de um atributo

Estatísticas

- Para cada índice
 - f : *fan-out* médio dos nós internos para índices estruturados em árvore
 - x : número de níveis
 - ◆ para uma árvore balanceada: $x = \lceil \log_f d \rceil$
 - ◆ para um índice *hash*: $x = 1$
 - b_{l_1} : número de blocos no nível de folha

Estatísticas Armazenadas

- Atualização
 - não é realizada a cada atualização do item de dado em questão
 - geralmente é realizada somente em baixas cargas
- Conseqüências
 - as estatísticas podem não ser completamente precisas
 - o otimizador pode não escolher a melhor estratégia para a otimização da consulta

Estruturas de Acesso

- Têm importância significativa na escolha de uma estratégia de processamento de consulta
- Exemplo: **operação de seleção**
 - pesquisa sequencial
 - pesquisa binária
 - função *hash*
 - árvore B⁺

índices sobre atributos-chave
e
índices sobre atributos não-chave

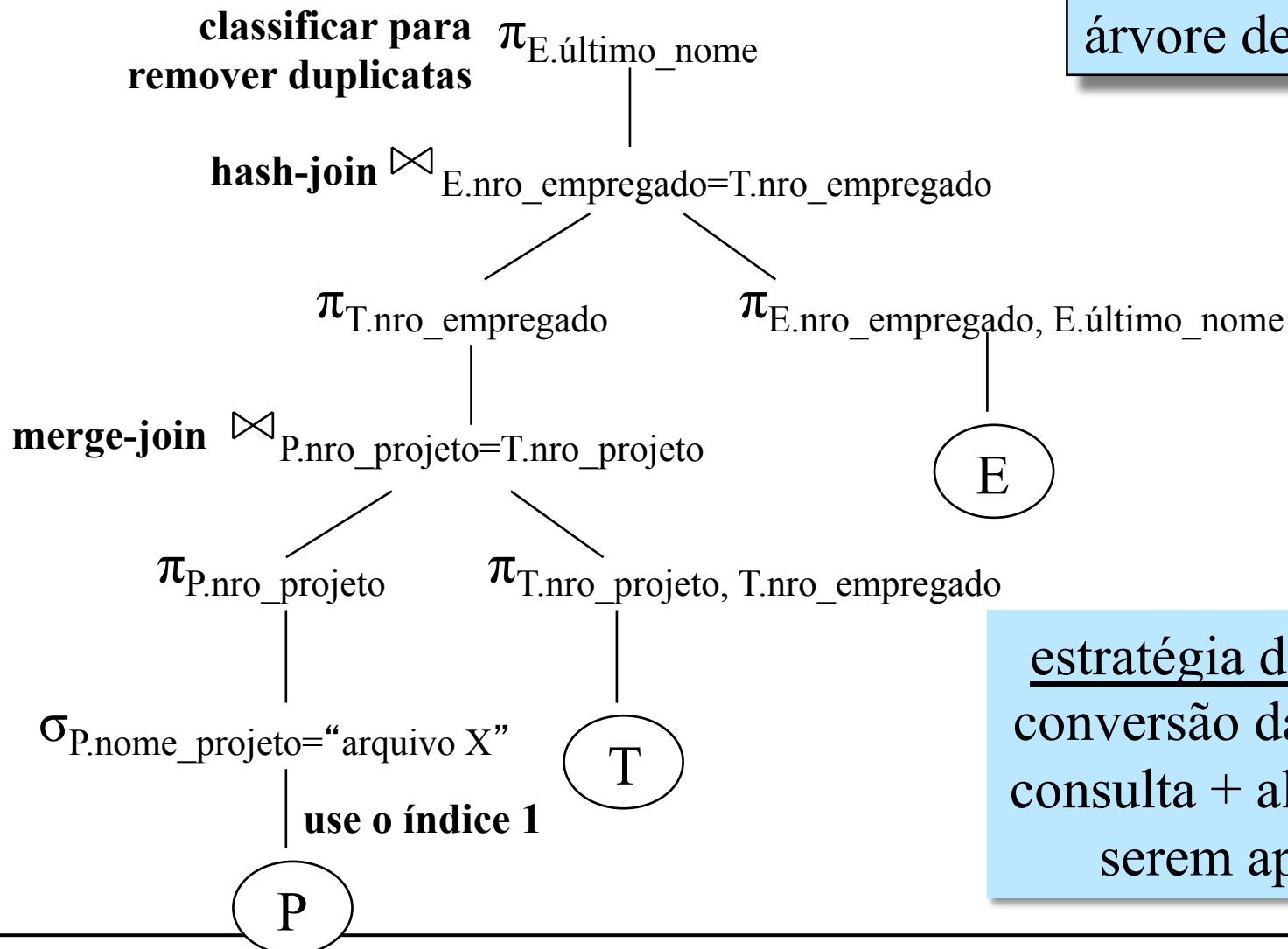
Estruturas de Acesso

- Exemplo: **operação de junção**
 - junção de laço aninhado
 - junção de laço aninhado em blocos
 - junção de laço aninhado indexado
 - *merge-join*
 - *hash-join*

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 6



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

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

Materialização *versus* Pipelining

- Pipelining
 - combina diversos algoritmos que correspondem a operações individuais
 - utiliza como entrada de uma operação as tuplas que foram geradas pela operação anterior