

Otimização Algébrica

Carina F. Dorneles
dorneles@inf.ufsc.br

Processamento de uma consulta

```
SELECT nome,  
FROM aluno JOIN matricula  
ON codigo=codal  
JOIN disciplina ON codigo=coddisc  
WHERE conceito = 'E'
```

Consulta SQL

Exame, análise e verificação

Representação Interna

Otimizador de consulta

Plano de execução

Gerador do código da consulta

Código de execução

Processador runtime do SGBD

→ *Resultado*



Código	Título	Ano	NrPaginas
LI005	Web e Banco de dados	2013	330
LI670	Introdução a Banco de Dados	2000	500
LI340	Programação C	2012	250
LI003	Algoritmos e Lógica	2000	700
LI888	Computação e Engenharia	20112	1050
LI777	Arquitetura de Computadores	2000	890
LI999	Introdução à Computação	2010	200

Representação Interna

- Estrutura que representa a consulta SQL original, chamada **árvore algébrica canônica**
 - **Árvore algébrica:**
 - nodos **folha**: relações do banco de dados
 - nodos **internos**: **operações** da álgebra
 - **Processamento** da árvore
 - execução é feita de **baixo para cima**
 - **resultados** das operações intermediárias **são relações**
 - a **execução termina** quando o nodo **raiz** é executado
-

Árvore canônica

- Nem sempre é a mais **otimizada**
 - Otimizador considera
 - **Ordem** de execução dos operadores
 - Transformações na árvore original
 - Geração de **várias árvores**, que formam o **planos de execução**
 - **Algoritmos** para implementação dos operadores
 - Existência de **índices**
 - **Estimativas** sobre os dados
 - Recuperadas do catálogo do BD (tamanho das tabelas, seletividade de atributos, etc...)
-

Árvore canônica

- Nem sempre é a mais **otimizada**
- Otimizador considera
 - **Ordem** de execução dos operadores
 - Transformações na árvore original
 - Geração de **várias árvores**, que formam o **planos de execução**
 - **Algoritmos** para implementação dos operadores
 - Existência de **índices**
 - **Estimativas** sobre os dados
 - Recuperadas do catálogo do BD (tamanho das tabelas, seletividade de atributos, etc...)

Chamada de fase de “otimização heurística”



Algoritmo de otimização algébrica

- Executado em 6 passos:

1. Decompor operações de σ mais restritivas
 2. Mover σ para o final da árvore
 3. Arranjar os nodos folhas para aplicar o σ mais restrito primeiro
 4. Formar π a partir de σ e X subsequentes
 5. Decompor π e mover para o mais baixo da árvore possível
 6. Identificar candidatos para operações combinadas, para criar várias árvores
-

Exemplo de otimização

- Relações de um banco de dados acadêmico:

Aluno (codAl, nomeAl,)

Disciplina (codDisc, nomeDisc, ...)

Historico (codAl, codDisc, xxxx, conceito...)

- Obter os **nomes** dos alunos que obtiveram **conceito "E"** em **disciplina** denominada **“Programação I”**
-

Expressão algébrica

π nomeAl

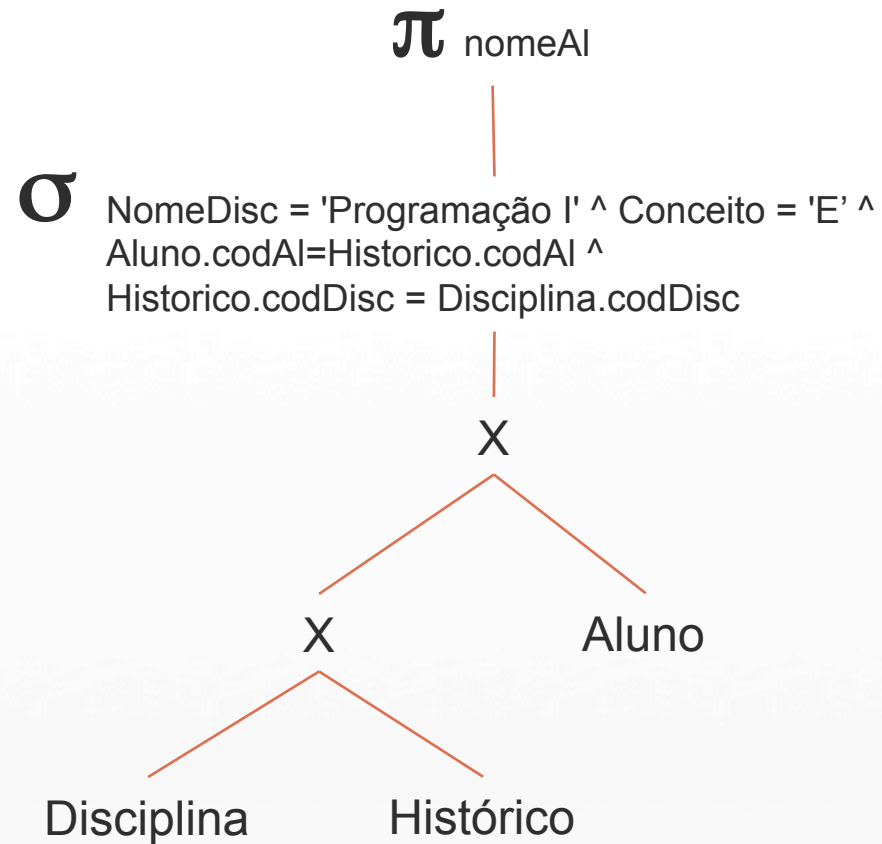
(σ NomeDisc = 'Programação I' ^ Conceito = 'E' ^ Aluno.codAl=Historico.codAl

^ Historico.codDisc = Disciplina.codDisc

(Aluno X

(Historico X Disciplina)))

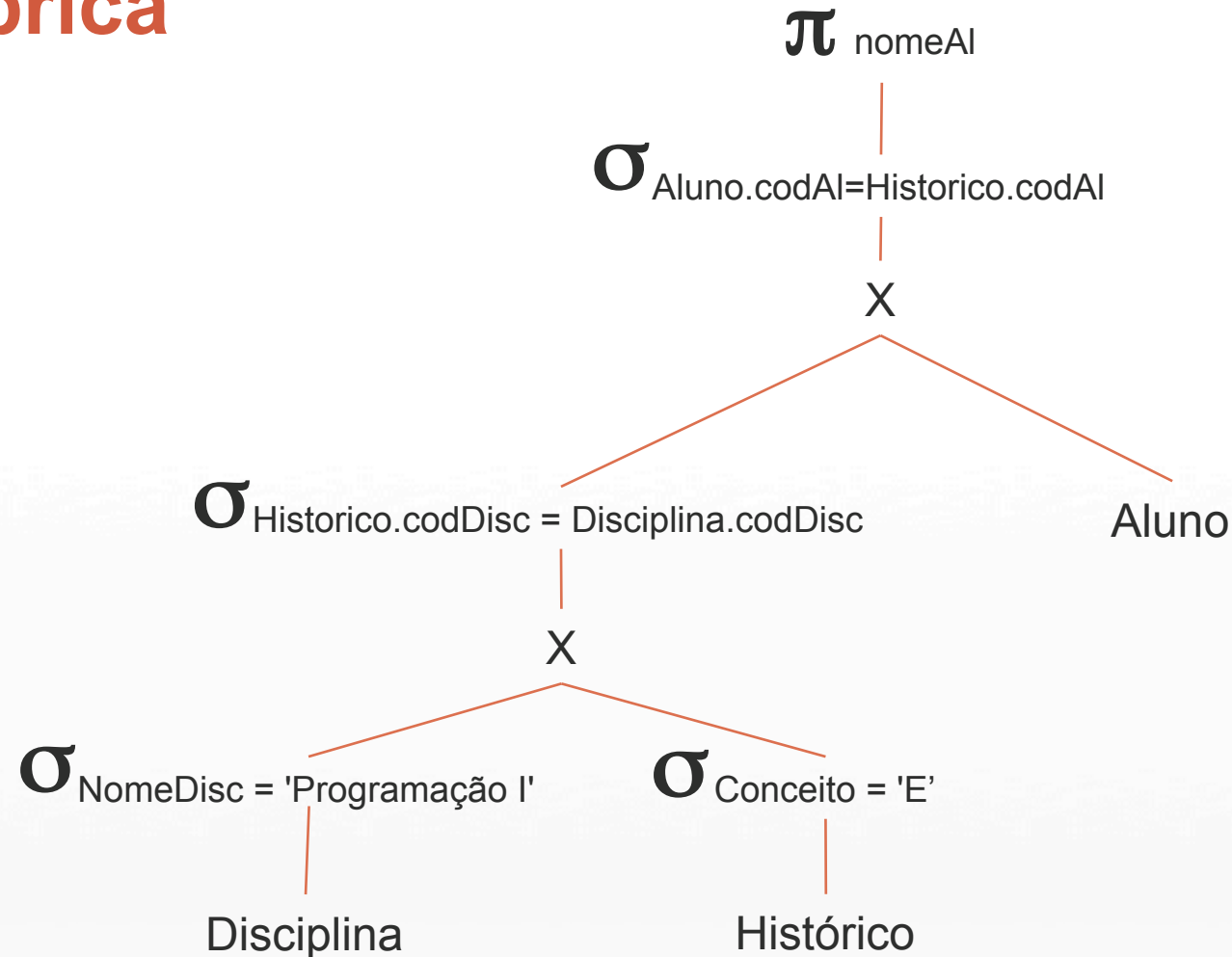
Árvore Algébrica - Canônica



Árvore Algébrica

Passos 1, 2 e 3:

Quebrar seleções e mover para baixo. Aplicar seleções mais restritas primeiro, se houver



Árvore Algébrica

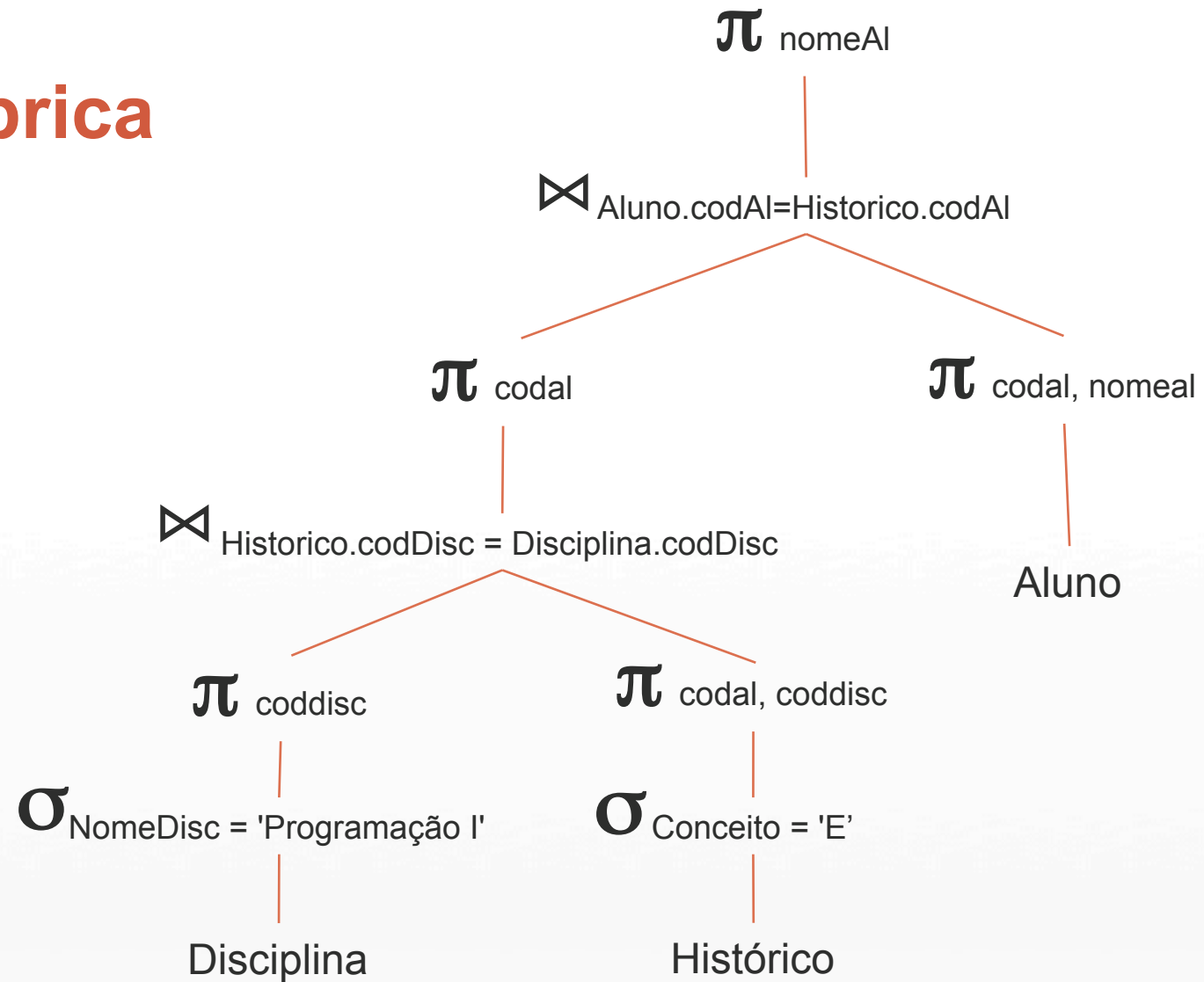
Passo 4:

Seleções são combinadas com produtos cartesianos, formando junções



Árvore Algébrica

Passo 5:
desmembrar e
descer projeções

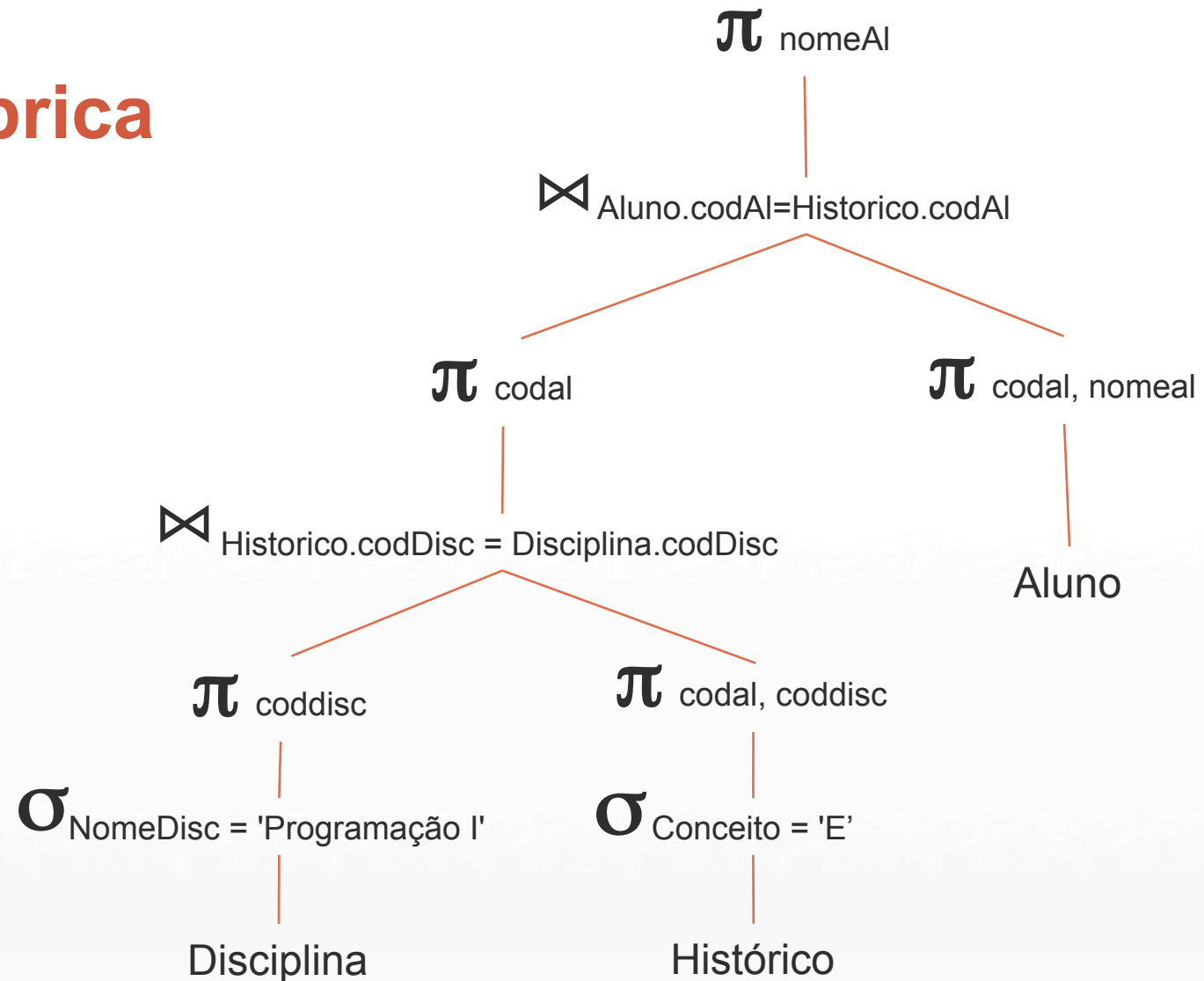


Árvore Algébrica

Passo 6:

Identificar
candidatos para
operações
combinadas, para
criar várias árvores

Conteúdo de BD II



Árvore Otimizada!!

Gerada no Passo 5

