

Sistemas de Banco de Dados

Fundamentos em Bancos de Dados Relacionais

Wladimir Cardoso Brandão

www.wladimirbrandao.com

Material distribuído sob licença CC BY-NC-ND 4.0

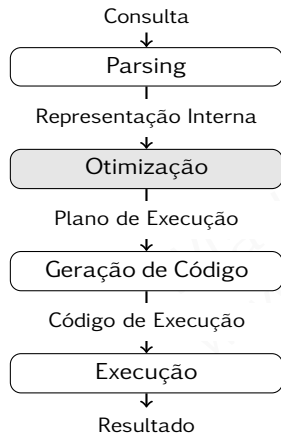
Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International



OTIMIZAÇÃO DE CONSULTA



Escolha de algoritmos e estratégias eficientes para execução de sequências de operações de álgebra relacional



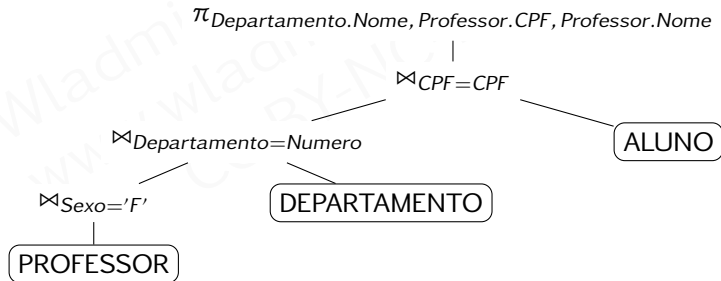
- ▶ HEURÍSTICA → aplicação de regras
 - ▶ CONSULTAS INTERPRETADAS
- ▶ CUSTO → estimativa de custo de execução
 - ▶ CONSULTAS COMPILADAS
- ▶ SEMÂNTICA → compreensão e substituição
 - ▶ Consulta é substituída por outra "melhor"

Execução do plano tipicamente de forma encadeada (*pipeline*) sem geração de arquivos intermediários



Estrutura em árvore que representa uma expressão de álgebra relacional

- ▶ FOLHA → arquivo de entrada
- ▶ INTERNO → operação da álgebra relacional
- ▶ RAÍZ → operação de projeção final da álgebra relacional
- ▶ ORDEM → operações executadas da folha esquerda para a raiz





HEURÍSTICA → método para modificação da representação interna de uma consulta, visando tornar seu plano de execução eficiente

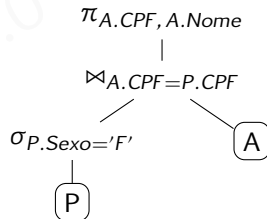
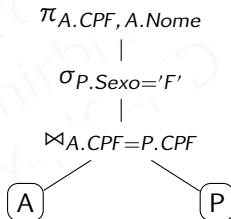
- ▶ Não garante que a melhor representação interna, aquela que gere o plano de execução mais eficiente, seja gerada
- ▶ Aplicação de um conjunto de regras sobre uma árvore de consulta inicial
 - ▶ Árvore inicial obtida a partir do *parsing* da consulta
 - ▶ Operações algébricas que reduzam o tamanho dos resultados intermediários devem ser executadas primeiro
 - ▶ Operações unárias de SELEÇÃO, PROJEÇÃO e AGREGAÇÃO antes de operações binárias de JUNÇÃO, e de CONJUNTO
 - ▶ Operações algébricas de menor custo devem ser executadas primeiro
 - ▶ Operações binárias mais eficientes antes



ÁRVORE INICIAL \rightarrow o *parsing* da consulta pode ser feito no sentido natural, da esquerda para a direita, ou reverso, da direita para a esquerda

- ▶ Arquivos e operações são inseridos na árvore na ordem de *parsing*

```
SELECT  A.CPF, A.Nome
FROM    ALUNO A,
        PROFESSOR P
WHERE   A.CPF = P.CPF
AND     P.Sexo = 'F';
```





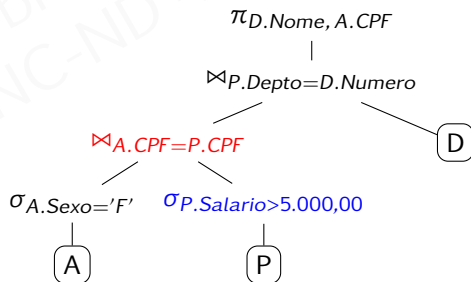
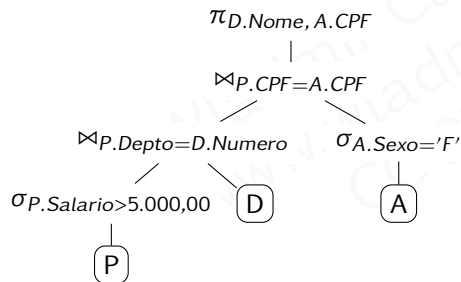
REGRAS → modificam árvore de consulta inicial, baseando-se no princípio *reduzir para combinar*

- ▶ REDUÇÃO → operações unárias precedem operações binárias
 - ▶ Primeira operação em cada nó folha deve ser SELEÇÃO, seguida de PROJEÇÃO, seguida de AGREGAÇÃO
- ▶ COMBINAÇÃO → operações de menor custo precedem as de custo maior
 - ▶ Nós folhas devem ser reordenados de forma que junções mais eficientes sejam executadas primeiro, evitando-se produtos cartesianos



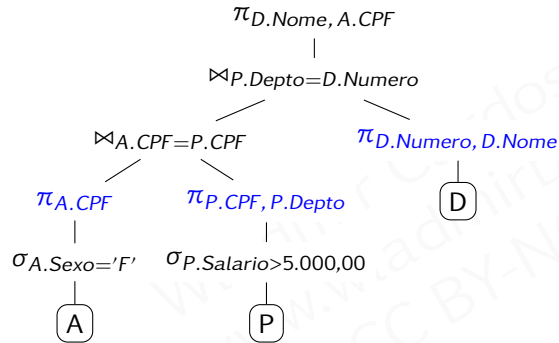
Considerando *parsing* no sentido reverso, e existência de índice multinível em *CPF* de PROFESSOR e ALUNO, e índice primário em *Numero* de DEPARTAMENTO:

```
SELECT D.Nome, A.CPF FROM ALUNO A, DEPARTAMENTO D, PROFESSOR P
WHERE A.CPF = P.CPF AND A.Sexo = 'F'
AND P.Depto = D.Numero AND P.Salario > 5.000,00
```





Projeções internas reduzem ainda mais o tamanho dos resultados intermediários

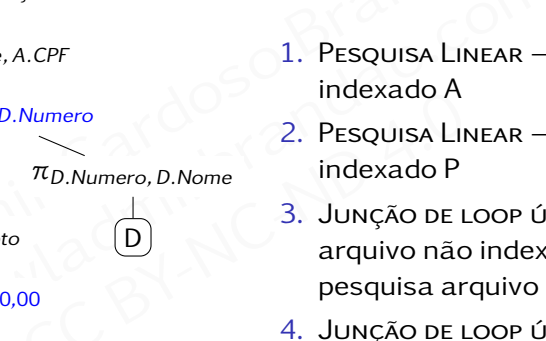


```
SELECT  D.Nome, A.CPF
FROM    DEPARTAMENTO D,
        PROFESSOR P,
        ALUNO A
WHERE   D.Numero = P.Depto
        AND P.CPF = A.CPF
        AND P.Salario > 5.000,00
        AND A.Sexo = 'F'
```

Consulta pode ser reescrita de forma que a árvore inicial resultante do *parsing* se aproxime da árvore de consulta otimizada

étodos de acesso, algoritmos e estratégias
ações na árvore de consulta otimizada

1. PESQUISA LINEAR —



1. PESQUISA LINEAR → arquivo não indexado A
2. PESQUISA LINEAR → arquivo não indexado P
3. JUNCTÃO DE LOOP ÚNICO → varre arquivo não indexado A e pesquisa arquivo de índice de P
4. JUNCTÃO DE LOOP ÚNICO → varre resultado intermediário e pesquisa arquivo de índice de D



FUNÇÃO DE CUSTO → computa estimativa de custo para algumas estratégias de execução de consulta, escolhendo a estratégia de menor estimativa

- ▶ TEMPO DE RESPOSTA → # estratégias de execução avaliadas é limitado
- ▶ CONSULTAS COMPILADAS → otimização feita na compilação
- ▶ EVIDÊNCIAS DE CUSTO → combinadas pela função de custo
 - ▶ COMPUTAÇÃO → custo de processamento (CPU) de dados em memória primária
 - ▶ MEMÓRIA → consumo de *buffers* de memória primária
 - ▶ DISCO → consumo de blocos de disco
 - ▶ I/O → custo com operações de paginação, transferência de dados entre memória primária e secundária
 - ▶ COMUNICAÇÃO → custo de transferência de dados via rede



CATÁLOGO → armazena informação necessária para estimar custo

- ▶ ARQUIVO → organização, tamanho, blocagem, # blocos, tamanho de registro, # de registros
- ▶ ÍNDICE → tipo, # níveis, # blocos em 1º nível
- ▶ REGISTRO → distribuição de valores
 - ▶ SELETIVIDADE → fração de registros que satisfazem uma condição de igualdade em um campo
 - ▶ Exemplo → 50% de registros de PROFESSOR tem Sexo = 'M'
 - ▶ CARDINALIDADE → # médio de registros que satisfazem uma condição de associação por igualdade em um campo
 - ▶ Exemplo → cada registro de DEPARTAMENTO está associado em média a 5 registros de PROFESSOR



COMPREENSÃO → SGBD precisa *compreender* o significado da consulta para reescrevê-la de uma forma *melhor*, que gere um plano de execução mais eficiente

A seguinte consulta demanda FORÇA BRUTA para execução:

```
SELECT CPF, Nome FROM PROFESSOR  
WHERE CPF IN (SELECT CPF FROM ALUNO)
```

Mas pode ser reescrita para:

```
SELECT A.CPF, A.Nome FROM PROFESSOR A, ALUNO B  
WHERE A.CPF = B.CPF
```

Forma que demanda JUNÇÃO POR LOOP ÚNICO se PROFESSOR ou ALUNO tiverem índice em *CPF*



- [1] Elmasri, Ramez; Navathe, Sham. *Fundamentals of Database Systems*. 7ed. Pearson, 2016.
- [2] Silberschatz, Abraham; Korth, Henry F.; Sudarshan, S. *Database System Concepts*. 6ed. McGraw-Hill, 2011.
- [3] Date, Christopher J. *An Introduction to Database Systems*. 8ed. Pearson, 2004.