

# Processamento e Otimização de Consultas

Banco de Dados II  
Prof. Guilherme Tavares de Assis

Universidade Federal de Ouro Preto – UFOP  
Instituto de Ciências Exatas e Biológicas – ICEB  
Departamento de Computação – DECOM

1

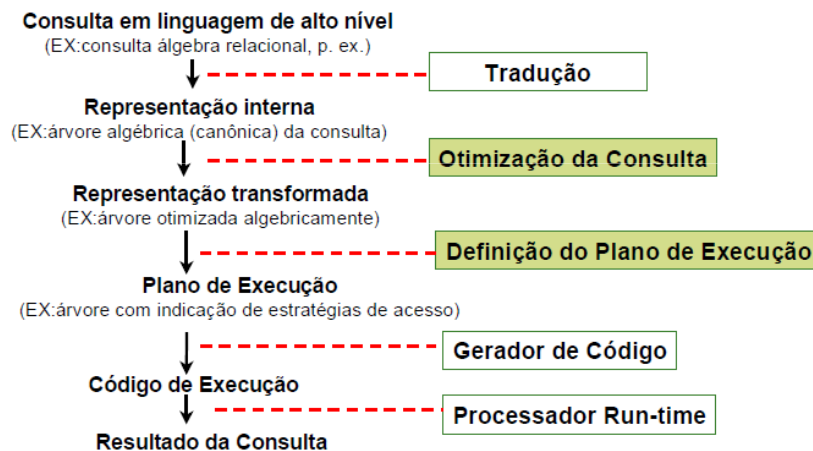
## Introdução

- Processamento de consultas consiste em uma sequência de atividades que devem ser executadas por um SGBD, no momento de se realizar uma consulta no banco de dados.
- Dentre tais atividades, um SGBD necessita idealizar uma estratégia ou um plano de execução eficiente para recuperar os resultados de uma consulta com base nos arquivos de banco de dados.
  - A eficiência de uma determinada estratégia está ligada ao número de acessos necessários ao disco para a realização de uma consulta.
  - O processo de escolha de uma estratégia de execução para o processamento de uma consulta é nomeado otimização de consulta.

2

## Introdução

### Etapas do processamento de consultas



3

## Introdução

- Consultas em SQL, em geral, são traduzidas em consultas da álgebra relacional e, então, otimizadas.
- Encontrar a estratégia de execução ótima é difícil, já que:
  - requer muito tempo;
  - pode requerer informações, como estrutura de implementação e conteúdo dos arquivos, que não estejam disponíveis no catálogo do banco de dados;
  - os SGBDs possuem distintos algoritmos de acesso ao banco de dados para implementar as operações necessárias.
- Em geral, a definição do plano de execução não é pela melhor estratégia, mas sim pela estratégia de razoável eficiência.

4

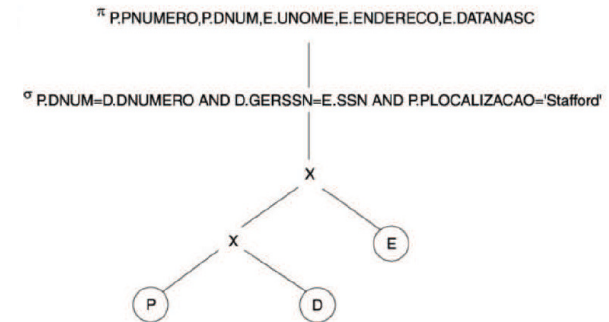
## Tradução de Consultas SQL para Álgebra Relacional

- No processo de tradução de uma consulta SQL para a álgebra relacional, é gerada uma árvore de consulta algébrica (canônica) equivalente à consulta.
  - Os nós externos e internos da árvore representam as relações do banco de dados e as operações da álgebra, respectivamente.
  - Nessa etapa, não há preocupação com a otimização.
- Para se processar uma árvore algébrica, tem-se que:
  - os nós internos (operações da álgebra) são executados quando seus operandos (relações) encontram-se disponíveis, a começar pelos nós internos mais profundos na árvore, sendo substituídos pela relação resultante;
  - consequentemente, o nó raiz é o último a ser executado.

5

## Tradução de Consultas SQL para Álgebra Relacional

```
SELECT P.Pnumero, P.Dnum, E.Unome, E.Endereco, E.DataNasc
FROM   P, D, E
WHERE  (P.Dnum=D.Dnumero) and (D.GerSSN=E.SSN) and
       (P.Plocalizacao='Stafford')
```



6

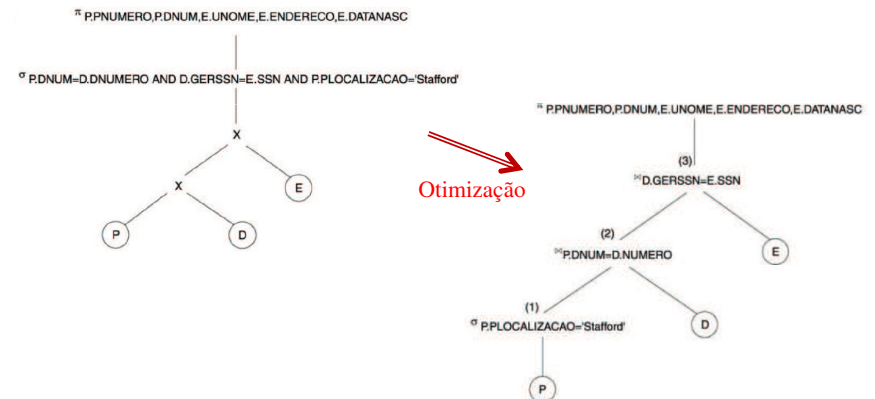
## Otimização de Consultas

- Para uma determinada consulta SQL, uma vez gerada a árvore de consulta algébrica equivalente, a próxima etapa consiste em otimizar a mesma.
  - A árvore de consulta otimizada deve ser eficiente na realização da consulta SQL.
  - A partir da árvore de consulta inicial, pode-se gerar diferentes expressões em álgebra relacional, o que leva à possibilidade de distintas árvores de consulta otimizadas.

7

## Otimização de Consultas

```
π PNUMERO, DNUM, UNOME, ENDereco, DATANASC (
  ((σ PLOCALIZACAO='Stafford' (PROJETO))
   ⋈ DNUM=DNUMERO (DEPARTAMENTO))
   ⋈ GERSSN=SSN (EMPREGADO))
```



8

## Otimização de Consultas

- O processo de otimização de consultas baseia-se em:
  - regras heurísticas: envolvem a reordenação das operações da álgebra relacional em uma árvore de consulta algébrica;
  - estimativas sistemáticas do custo de diferentes planos de execução.

9

## Otimização de Consultas - Regras Heurísticas

- O processo de otimização de consultas, considerando regras heurísticas, consiste nos seguintes passos gerais:
  - O analisador sintático do SGBD gera a árvore de consulta algébrica equivalente à consulta SQL.
  - A árvore de consulta é, então, otimizada de acordo com regras heurísticas.
    - Um exemplo de regra heurística é aplicar as operações de seleção e projeção antes de aplicar operações binárias como a junção.
    - Outro exemplo é reordenar as operações da álgebra relacional a partir das regras de equivalência algébrica.
  - Um plano de execução da consulta é gerado no intuito de que as operações da álgebra sejam executadas, na ordem em que foram estabelecidas na árvore otimizada, considerando os caminhos de acesso disponíveis.

10

## Regras de Equivalência Algébrica

- Cascata de seleções:
 
$$\sigma_{\langle c1 \rangle E \langle c2 \rangle E \langle c3 \rangle}(\mathbf{R}) = \sigma_{\langle c1 \rangle}(\sigma_{\langle c2 \rangle}(\sigma_{\langle c3 \rangle}(\mathbf{R})))$$

- Comutatividade de seleções:
 
$$\sigma_{\langle c1 \rangle}(\sigma_{\langle c2 \rangle}(\mathbf{R})) = \sigma_{\langle c2 \rangle}(\sigma_{\langle c1 \rangle}(\mathbf{R}))$$

- Cascata de projeções:
 
$$\pi_{\langle a1 \rangle}(\pi_{\langle a2 \rangle}(\pi_{\langle a3 \rangle}(\mathbf{R}))) = \pi_{\langle a1 \rangle}(\mathbf{R})$$

- Comutatividade de seleções e projeções:
 
$$\pi_{\langle a1, a2, \dots, an \rangle}(\sigma_{\langle cond \rangle}(\mathbf{R})) = \sigma_{\langle cond \rangle}(\pi_{\langle a1, a2, \dots, an \rangle}(\mathbf{R}))$$

Não se aplica caso a condição de seleção envolva atributos que não estão na lista de projeção

11

## Regras de Equivalência Algébrica

- Comutatividade de produtos cartesianos ou junções:
 
$$\mathbf{R} \times \mathbf{S} = \mathbf{S} \times \mathbf{R} \quad \mathbf{R} \bowtie_{\langle c \rangle} \mathbf{S} = \mathbf{S} \bowtie_{\langle c \rangle} \mathbf{R}$$

- Comutatividade de seleções e junções:
  - Se todos os atributos da condição de seleção envolverem apenas uma das relações da junção:

$$\sigma_{\langle c1 \rangle}(\mathbf{R} \bowtie_{\langle c2 \rangle} \mathbf{S}) = (\sigma_{\langle c1 \rangle}(\mathbf{R})) \bowtie_{\langle c2 \rangle} \mathbf{S}$$

- Se a condição de seleção puder ser escrita como uma expressão conjuntiva onde cada subexpressão envolva atributos de apenas uma relação da junção:

$$\sigma_{\langle c1 \rangle}(\mathbf{R} \bowtie_{\langle c2 \rangle} \mathbf{S}) = (\sigma_{\langle c1\_1 \rangle}(\mathbf{R})) \bowtie_{\langle c2 \rangle} (\sigma_{\langle c1\_2 \rangle}(\mathbf{S}))$$

12

## Regras de Equivalência Algébrica

- Comutatividade de projeções e junções:

- Se a condição de junção envolve apenas atributos da projeção:

$$\pi_{\langle a1, a2, a3 \rangle} (R \bowtie_{\langle c \rangle} S) = (\pi_{\langle a1, a2 \rangle} (R)) \bowtie_{\langle c \rangle} (\pi_{\langle a3 \rangle} (S))$$

No caso,  $(a1 \in R)$ ,  $(a2 \in R)$ ,  $(a3 \in S)$  e a condição  $c$  envolve apenas os atributos  $a1$ ,  $a2$  e  $a3$

13

## Regras de Equivalência Algébrica

- Comutatividade de operações de união e interseção:

$$R \cup S = S \cup R \text{ e } R \cap S = S \cap R$$

- Associativividade das operações junção, produto cartesiano, união e interseção:

$$R \theta (S \theta T) = (R \theta S) \theta T, \text{ onde } \theta \text{ é a operação}$$

- Comutatividade de seleções e operações de conjunto:

$$\sigma_{\langle c \rangle} (R \theta S) = \sigma_{\langle c \rangle} (R) \theta \sigma_{\langle c \rangle} (S)$$

- Comutatividade da projeção com a união:

$$\pi_{\langle a1, a2 \rangle} (R \cup S) = \pi_{\langle a1, a2 \rangle} (R) \cup \pi_{\langle a1, a2 \rangle} (S)$$

14

## Otimização de Consultas - Regras Heurísticas

- Algoritmo básico (regras heurísticas):

1. Quebrar as seleções com condições conjuntivas em uma cascata de seleções.
2. Mover cada operação de seleção o mais profundo possível na árvore de consulta.
3. Reordenar os nós externos de tal forma que as seleções mais restritivas sejam executadas primeiro.
4. Combinar as operações de produto cartesiano com as subsequentes seleções para formar as junções equivalentes.
5. Quebrar as projeções, quando possível.
6. Mover as projeções quebradas o mais profundo possível na árvore de consulta, criando novas projeções quando for necessário.

15

## Otimização de Consultas - Regras Heurísticas

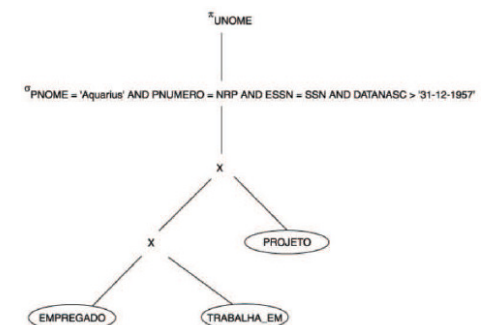
- Exemplo:

- Consulta: recuperar o sobrenome dos empregados nascidos após 1957 que trabalham no projeto 'Aquarius'.

SQL: 

```
select UNOME
from EMPREGADO, TRABALHA_EM, PROJETO
where PNAME='Aquarius' and PNUMERO=NRP and ESSN=SSN and DATANASC > '31-12-1957'
```

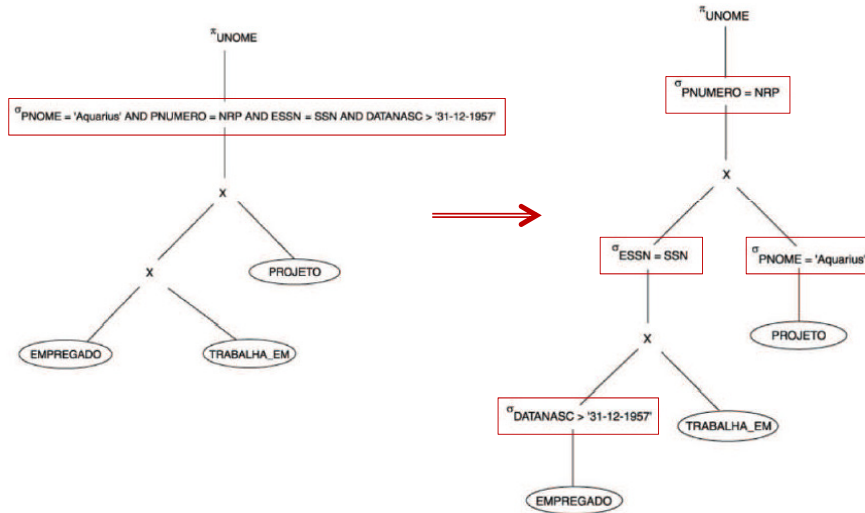
- Árvore de consulta algébrica inicial:



16

## Otimização de Consultas - Regras Heurísticas

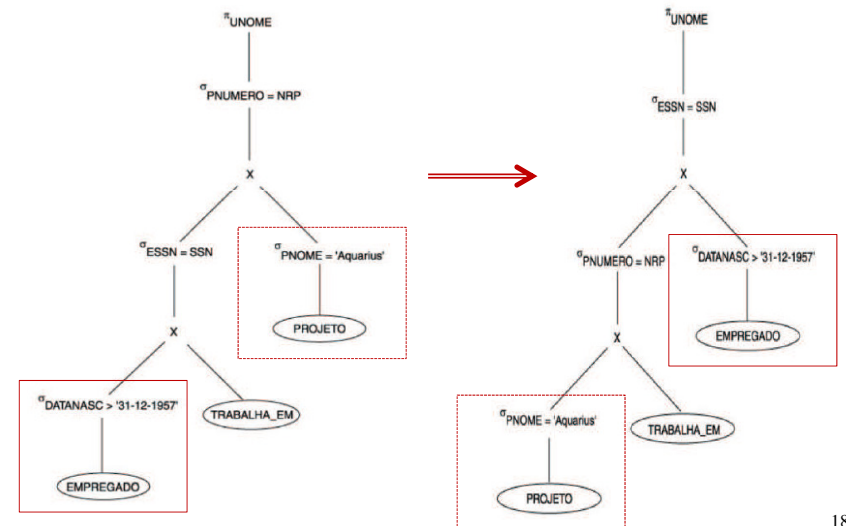
- Aplicação dos passos 1 e 2 do algoritmo:



17

## Otimização de Consultas - Regras Heurísticas

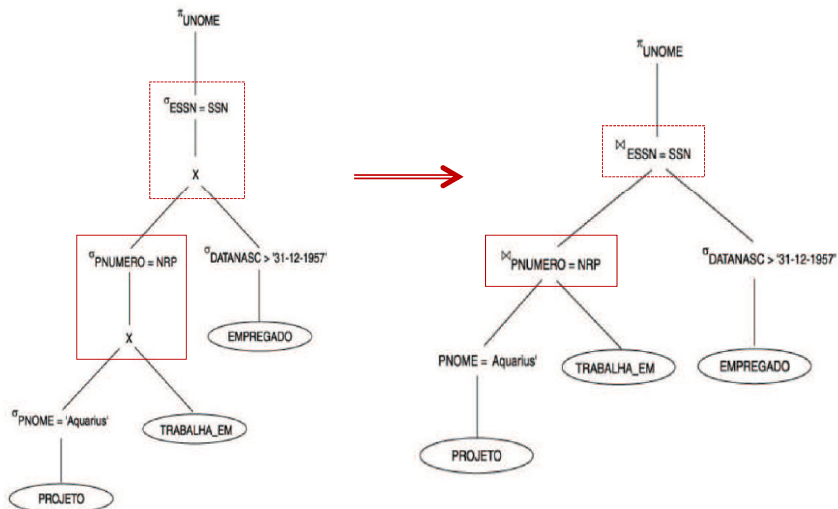
- Aplicação do passo 3 do algoritmo:



18

## Otimização de Consultas - Regras Heurísticas

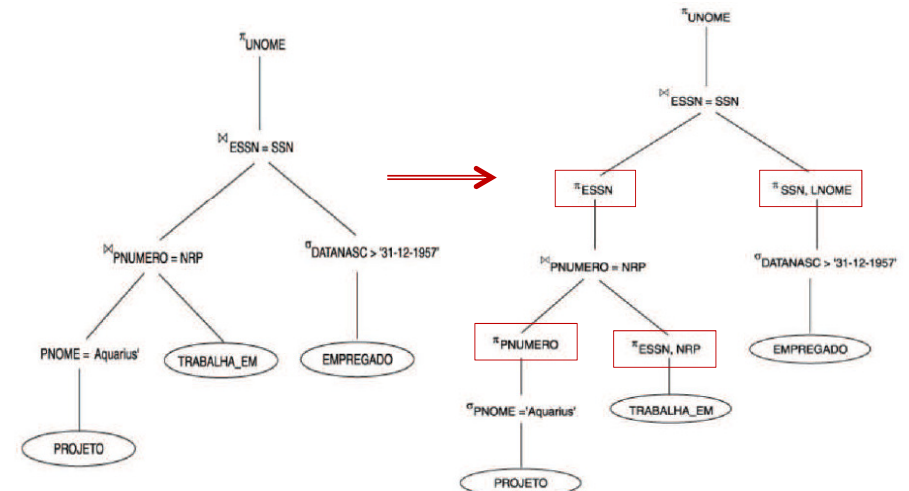
- Aplicação do passo 4 do algoritmo:



19

## Otimização de Consultas - Regras Heurísticas

- Aplicação dos passos 5 e 6 do algoritmo:



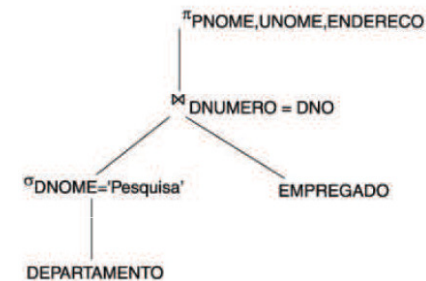
20

## Definição do Plano de Execução

- Uma vez gerada a árvore de consulta otimizada a partir da árvore de consulta inicial, a próxima etapa consiste em definir o plano de execução da consulta que inclui informações como, por exemplo:
  - métodos de acesso disponíveis para cada relação envolvida;
  - utilização de índices existentes;
  - algoritmos para computação das operações algébricas.

21

## Definição do Plano de Execução



- Para a árvore de consulta acima, o otimizador do SGBD poderia definir, por exemplo:
  - um índice de busca para a seleção referente à relação "Departamento";
  - uma varredura ordenada como método de acesso para a relação "Empregado", baseada no departamento dos empregados.

22

## Definição do Plano de Execução

- O plano de execução da consulta também pode conter a especificação de como o resultado de uma operação algébrica é tratado.
  - Avaliação materializada: o resultado de uma operação é armazenado em uma relação temporária.
  - Avaliação por *pipeline*: o resultado de uma operação é encaminhado, diretamente, para a próxima operação algébrica.
    - Nesse caso, há um menor custo de processamento já que, na execução da consulta, não haverá necessidade de leitura e escrita de resultados intermediários no disco.

23

## Otimização de Consultas - Custo

- O otimizador de consultas não deve depender apenas das regras heurísticas.
  - Os custos de diferentes planos de execução de consulta devem ser estimados e comparados, escolhendo aquele com menor estimativa de custo.
  - O número de planos a serem avaliados deve ser limitado para que o tempo gasto na avaliação seja aceitável.
- A otimização baseada no custo usa técnicas tradicionais de otimização que buscam, no espaço de soluções, uma solução que minimize a função objetivo de custo.

24

## Otimização de Consultas - Custo

- O custo da execução de uma consulta refere-se a:
  - custo de acesso ao dispositivo de armazenamento secundário: busca, leitura e escrita de blocos de dados;
  - custo de armazenamento de arquivos temporários;
  - custo de computação das operações algébricas da consulta;
  - custo de uso de memória referente ao número de *buffers* de memória necessários para a execução da consulta;
  - custo de comunicação referente ao transporte da consulta e dos resultados pelos componentes do SGBD.
- Para grandes bancos de dados, o foco está na minimização do custo de acesso ao dispositivo de armazenamento secundário, enquanto que, para bancos de dados menores (quando os dados cabem em memória), o foco está no custo de computação.

## Otimização de Consultas - Custo

- Alguns metadados armazenados no catálogo do SGBD são utilizados pelo otimizador de consultas, a saber:
  - número de tuplas de uma relação;
  - tamanho da tupla de uma relação;
  - número de blocos que contêm as tuplas de uma relação;
  - número de valores distintos de um determinado atributo de uma relação;
  - número médio de tuplas que satisfazem uma condição de igualdade em um determinado atributo de uma relação;
  - índices existentes a partir de atributos de uma relação.