

# Banco de Dados

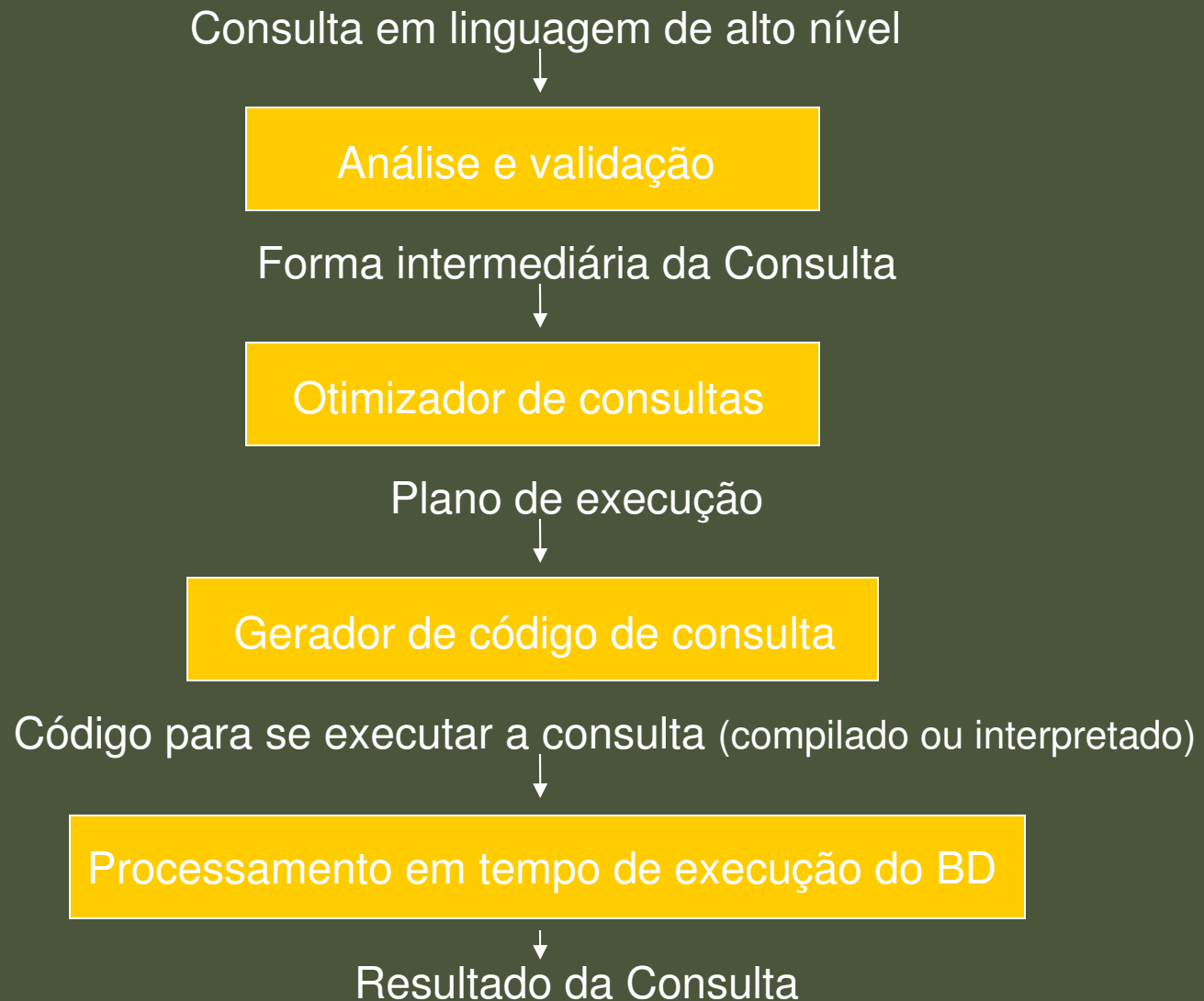


## Otimização de Consultas

Joao Eduardo Ferreira  
Osvaldo Kotaro Takai  
Marcelo Finger

# Introdução

---



# Passos para otimização de consultas

---

- ❑ Tradução de consultas SQL – Álgebra relacional
- ❑ Algoritmos básicos para execução de consultas:
  - ordenação externa
  - seleção
  - junção
  - projeção e conjunto
  - agregação
- ❑ Heurística na otimização de consultas
- ❑ Estimativa de custos

# Tradução SQL – Álgebra relacional

## Em SQL:

```
SELECT SOBRENOME, NOME  
FROM EMPREGADO  
WHERE SALARIO > (SELECT MAX (SALÁRIO)  
                  FROM EMPREGADO  
                  WHERE NUD=5);
```

Consulta aninhada não correlacionada

## Em Álgebra Relacional:

$$\rho_{\text{MAX SALARIO}} (\sigma_{\text{NUD}=5} (\text{EMPREGADO}))$$
$$\pi_{\text{SOBRENOME, NOME}} (\sigma_{\text{Salário} > c} (\text{EMPREGADO}))$$

*obs.: consulta aninhada correlacionada: variável tupla do bloco externo aparece na cláusula WHERE do bloco interno.*

# Ordenação externa

---

- ❑ cláusula ORDER BY
- ❑ arquivos grandes
- ❑ FASE 1: Quebra-Ordenação interna (sort)
- ❑ gera “runs” (fragmentos ordenados do arquivo original)
- ❑ ordena runs
- ❑ grava runs
- ❑  $b = \text{numero de blocos do arquivo}$
- ❑  $n_b = \text{espaço disponível de buffer}$
- ❑  $n_r = \text{número de runs iniciais}$
- ❑  $n_r = b/n_b$
- ❑ Complexidade ordem =  $2 * b$  (leitura e escrita)

# Ordenação externa

---

- FASE 2: Intercalação (merge)
- $n$  passagens pela intercalação;
- grau de intercalação  $g_m$   
(numero que runs intercaladas em cada passagem)
- $(n_b - 1) < g_m < n_r$
- número de passagens pelos blocos =  $\log_{g_m}(n_r)$
- Complexidade todo processo:
- $(2 * b) + (\log_{g_m}(b / n_b))$

# Operação Select – Algoritmos básicos

---

- ❑ Pesquisa linear
- ❑ Pesquisa binária
- ❑ Utilização de índice primário (ou chave hash)
- ❑ Utilização de índice primário:  $<$ ,  $<=$ ,  $>$ ,  $>=$
- ❑ Utilização de índice cluster
- ❑ Utilização de índice secundário (arvore B<sup>+</sup>)

# Operação Select – Algoritmos complexos

---

- ❑ Seleção conjuntiva usando índice individual
- ❑ Seleção conjuntiva usando índice composto
- ❑ Seleção conjuntiva por meio da intersecção de ponteiros de registro

- conceito de **seletividade** = tuplas selecionadas/existentes

- o otimizador deve escolher o caminho de acesso que recupere o menor número de registros de modo mais eficiente → quantificar custos



# Operação Join

---

- ❑ Junção de loop aninhado
- ❑ Junção de loop único
- ❑ Junção merge-sort
- ❑ Junção hash

# Operação Project e de Conjunto

---

- ❑ Se incluir a chave, a implementação é trivial, pois as tuplas sempre serão distintas;
- ❑ Caso contrário, tuplas replicadas deverão ser eliminadas usando ordenação, com uso do operador DISTINCT;
- ❑ Produto cartesiano: deve-se evitar;
- ❑ UNION, INTERSECTION, DIFFERENCE

*Várias das técnicas merge-sort (união compatível de conjuntos ordenados)*

# Operação Agregação

---

- ❑ MIN, MAX utilizam índices;
- ❑ COUNT, AVERAGE e SUM utilizam índices densos (uma entrada de índice para todo registro no arquivo original)
- ❑ GROUP BY: utiliza índice cluster

# Heurística para otimização de consultas

---

- Árvores de consulta;
- Otimização de árvores de consulta;
- Árvores de consulta e planos de execução;

# Árvores de consulta

- Para todo projeto localizado em 'Stafford', listar o número do projeto, o número do departamento responsável, o sobrenome, endereço e data de nascimento do gerente responsável pelo departamento.

## Em SQL:

```
SELECT PNUMERO, DNUM, SNAME, DATANASC, ENDERECO  
FROM PROJETO AS P, DEPARTAMENTO AS D, EMPREGADO AS E  
WHERE P.DNUM=D.DNUMERO AND D.GERSS=E.SSN AND  
P.LOCALIZACAO='Stafford':
```

## Em Álgebra Relacional:

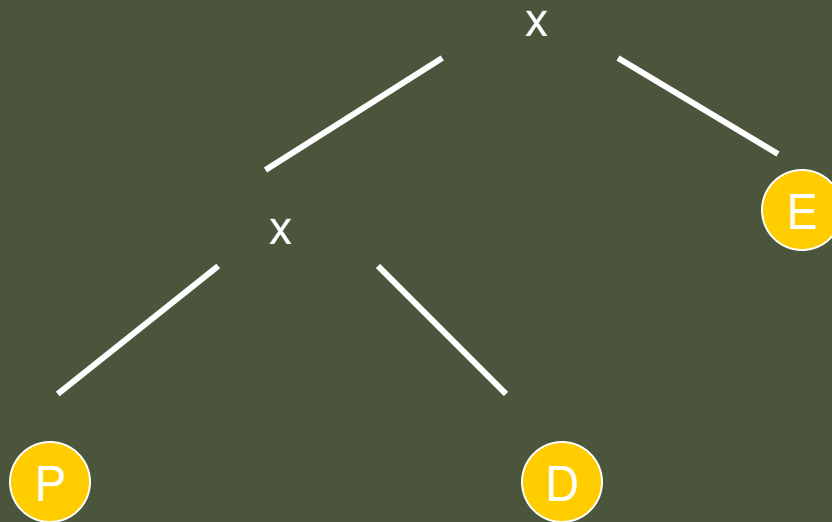
```
STAFFORD_PROJS  $\leftarrow \sigma_{P.LOCALIZAC\tilde{A}O = 'Stafford'}$  (PROJETO)  
CONTR_DEPT  $\leftarrow$  (STAFFORD_PROJS  $\times_{P.DNUM = D.DNUMERO}$  DEPARTAMENTO)  
PROJ_DEPT_MGR  $\leftarrow$  (CONTR_DEPT  $\times_{D.GERSS = E.SSN}$  EMPREGADO)  
RESULT  $\leftarrow \pi_{P.NUMERO, P.DNUM, E.SNAME, E.ENDERECO, E.DATANASC}$  (PROJ_DEPT_MGR)
```

# Árvore de consulta - inicial

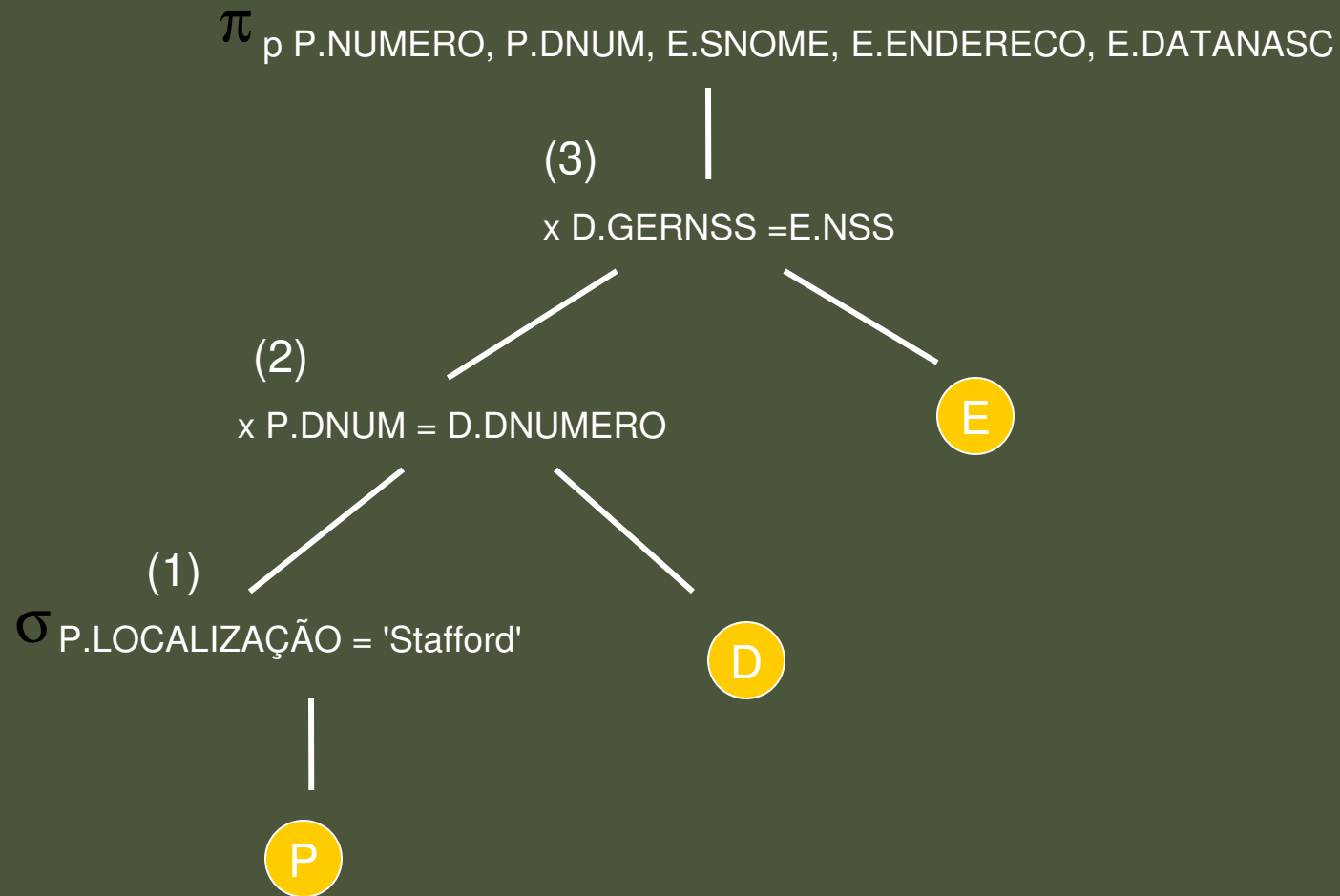
---

$\pi_{\text{p P.NUMERO, P.DNUM, E.SNOME, E.ENDERECO, E.DATANASC}}$

$\sigma_{\text{P.DNUM = D.DNUMERO AND D.GERNSS = E.NSS AND P.LOCALIZAÇÃO = 'Stafford'}}$



# Árvore de consulta - otimizada



# Regras Gerais

---



# Regras Gerais

---

- ❑ Cascata de seleção
- ❑ Cascata de projeção
- ❑ Comutatividade de join
- ❑ Comutatividade projeção e join
- ❑ Comutatividade operações de conjunto
- ❑ Associatividade

# Estimativa de custos

---

- ❑ Componentes de Custo;
- ❑ Informações do Catálogo em função de Custo
- ❑ Exemplo de consulta baseada na otimização de custos.

# Noções de Tuning

---

- ❑ Índices; produto cartesiano; cache.
- ❑ Estatísticas armazenamento: tabelas, índice, log, cache;
- ❑ Estatísticas E/S: atividade leitura-gravação(paginação)
- ❑ Estatísticas: processamento de consultas/transações
- ❑ Estatísticas de índices: níveis de índice x páginas folhas

# Noções de Tuning

---

- Evitar excessiva contenção de registros;
- Minimizar custos adicionais (cópia/logging);
- Escalonamento de processos;
- Relação disco x memória RAM

# FIM

---

## ▣ Referencia bibliográfica:

- Elmasri, R.; Navathe, S.B. Fundamentals of Database Systems, 4th ed. Addison-Wesley, Reading, Mass., 2003.
- Ramakrishnan, R.; Gehrke, J., Database Management Systems, 2 nd ed., McGraw-Hill, 2000.