

Sumário

1 Introdução ao Processamento de Consultas

2 Otimização de Consultas

3 Plano de Execução de Consultas

4 Introdução a Transações

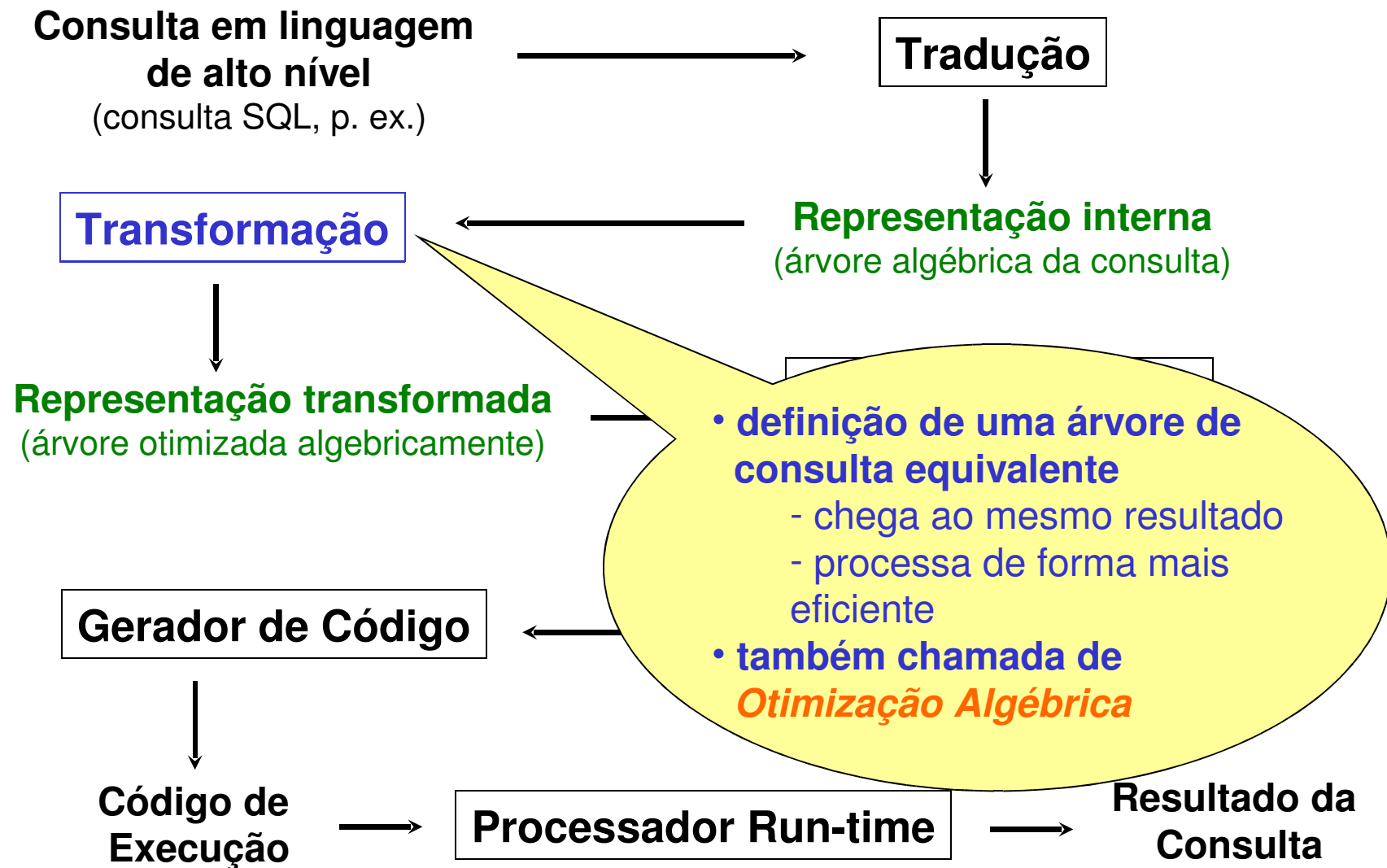
5 Recuperação de Falhas

6 Controle de Concorrência

7 Fundamentos de BDs Distribuídos

8 SQL Embutida

Etapas do Processamento de Consultas



Otimização Algébrica

- Objetivo do passo de Transformação
 - entrada: árvore da consulta inicial
 - saída: árvore da consulta otimizada (pode manter a mesma árvore)
- Base
 - regras de equivalência algébrica
 - devem ser conhecidas pelo otimizador para que possam ser geradas transformações válidas
 - algoritmo de otimização algébrica
 - indica a ordem de aplicação das regras e de outros processamentos de otimização

Regras de Equivalência Algébrica

1. Cascata de Seleções

$$\sigma_{c1 \wedge c2 \wedge \dots \wedge cn} (R) \equiv \sigma_{c1} (\sigma_{c2} (\dots (\sigma_{cn} (R))))$$

2. Comutatividade de Seleções

$$\sigma_{c1} (\sigma_{c2} (R)) \equiv \sigma_{c2} (\sigma_{c1} (R))$$

3. Cascata de Projeções

$$\pi_{\text{listaAtributos1}} (R) \equiv \pi_{\text{listaAtributos1}} (\pi_{\text{listaAtributos2}} (\dots (\pi_{\text{listaAtributosN}} (R))))$$

- válido em que situação?

Regras de Equivalência Algébrica

4. Comutatividade de Seleções e Projeções

$$(a) \pi_{a_1, a_2, \dots, a_n}(\sigma_c(R)) \equiv \sigma_c(\pi_{a_1, a_2, \dots, a_n}(R)) \quad \text{ou}$$

$$(b) \pi_{a_1, a_2, \dots, a_n}(\sigma_c(R)) \equiv \pi_{a_1, a_2, \dots, a_n}(\sigma_c(\pi_{a_1, a_2, \dots, a_n, a_p, \dots, a_t}(R)))$$

- válidas em quais situações?

5. Comutatividade de Operações Produtórias (“X”)

$$R \text{ “X” } S \equiv S \text{ “X” } R$$

- por “X” entenda-se: \bowtie ou $\bowtie \theta$ ou \bowtie

- a ordem dos atributos e tuplas do resultado não é relevante

Regras de Equivalência Algébrica

6. Comutatividade de Seleções e Operações Produtórias

(a) $\sigma_c (R \text{ "X" } S) \equiv (\sigma_c (R)) \text{ "X" } S$ ou

(b) $\sigma_c (R \text{ "X" } S) \equiv (\sigma_{c_1} (R)) \text{ "X" } (\sigma_{c_2} (S))$

(c) $\sigma_c (R \text{ "X" } S) \equiv \sigma_{c_3} ((\sigma_{c_1} (R)) \text{ "X" } (\sigma_{c_2} (S)))$

- válidas em quais situações?

7. Comutatividade de Projeções e Operações Produtórias

(a) $\pi_{\text{listaAtributos1}} (R \text{ "X" } S) \equiv (\pi_{\text{listaAtributos2}} (R)) \text{ "X" } S$ ou

(b) $\pi_{\text{listaAtributos1}} (R \text{ "X" } S) \equiv \pi_{\text{listaAtributos1}} ((\pi_{\text{listaAtributos2}} (R)) \text{ "X" } S)$ ou

(c) $\pi_{\text{listaAtributos1}} (R \text{ "X" } S) \equiv (\pi_{\text{listaAtributos2}} (R)) \text{ "X" } (\pi_{\text{listaAtributos3}} (S))$ ou

(d) $\pi_{\text{listaAtributos1}} (R \text{ "X" } S) \equiv \pi_{\text{listaAtributos1}} ((\pi_{\text{listaAtributos2}} (R)) \text{ "X" }$

- válidas em quais situações? $(\pi_{\text{listaAtributos3}} (S)))$

Regras de Equivalência Algébrica

8. Comutatividade de Operações de Conjunto

$$R \cup S \equiv S \cup R \quad \text{e}$$

$$R \cap S \equiv S \cap R$$

- por quê “—” não é comutativa?

9. Associatividade de Operações Produtórias e de Conjunto (“oX”)

$$(R \text{ “oX” } S) \text{ “oX” } T \equiv R \text{ “oX” } (S \text{ “oX” } T)$$

- por “oX” entenda-se: **X** ou **X** θ ou \bowtie ou \cup ou \cap

- observação: a operação de conjunto deve ser sempre a mesma em cada ocorrência de “oX”; Já a operação produtória pode ser diferente

- por quê “—” não é associativa?

Regras de Equivalência Algébrica

9. Associatividade de Operações Produtórias e de Conjunto (“oX”)

$$(R \text{ “oX” } S) \text{ “oX” } T \equiv R \text{ “oX” } (S \text{ “oX” } T)$$

Observação: predicados de junção devem ser devidamente ajustados na associatividade de operações produtórias

Exemplo: seja θ_1 um predicado sobre atributos de R e S, θ_2 um predicado sobre atributos de S e T, e θ_3 um predicado sobre atributos de R e T. Então:

$$(R \text{ “X” }_{\theta_1} S) \text{ “X” }_{\theta_2 \wedge \theta_3} T \equiv R \text{ “X” }_{\theta_1 \wedge \theta_3} (S \text{ “X” }_{\theta_2} T)$$

Regras de Equivalência Algébrica

10. Comutatividade de Seleção e Operações de Conjunto (“o”)

$$\sigma_c(R \text{ “o” } S) \equiv (\sigma_c(R)) \text{ “o” } (\sigma_c(S))$$

- atributos a filtrar devem existir em ambas as relações (renomeações de atributos podem ser realizadas)
- por “o” entenda-se: \cup ou \cap ou $—$

11. Comutatividade de Projeção e União

$$\pi_{\text{listaAtributos}}(R \cup S) \equiv (\pi_{\text{listaAtributos}}(R)) \cup (\pi_{\text{listaAtributos}}(S))$$

- atributos a projetar devem existir em ambas as relações (renomeações de atributos podem ser realizadas)
- por quê “ $—$ ” e “ \cap ” não são comutativas?

Regras de Equivalência Algébrica

12. Fusão de Seleções e Operações Produtórias

(a) $\sigma_c(R \bowtie S) \equiv R \bowtie \sigma_c S$ ou

(b) $\sigma_c(R \bowtie S) \equiv R \bowtie_c S$ ou

(c) $R \bowtie \sigma_c S \equiv R \bowtie_c S$

- válidas em quais situações?

Algoritmo de Otimização Algébrica

- Algoritmo de alto (altíssimo!) nível
 - Tenta gerar a árvore mais otimizada possível
- Exemplo de Consulta (BD Clínica)

```
SELECT p.codp, p.nome, c.data
FROM Pacientes p, Consultas c, Medicos m
WHERE m.especialidade = 'ortopedia'
AND m.codm = c.codm
AND p.codp = c.codp
AND c.hora >= '18:00'
AND c.hora <= '22:00'
AND p.idade > 60
```

Médicos(<u>codm</u> , nome, idade, especialidade)
Pacientes(<u>codp</u> , nome, doença, idade)
Consultas(<u>codm</u> , <u>codp</u> , <u>data</u> , hora)

Algoritmo de Otimização Algébrica

- Composto de 6 grandes passos
 - cada passo pode aplicar uma mesma regra várias vezes
- Passo 1
 - aplicar a regra 1
 - desmembrar operações de seleção
 - maior flexibilidade para mover seleções
- Passo 2
 - aplicar as regras 2, 4, 6 e 10 (e regra 1 ao final)
 - objetivo
 - mover seleções para níveis inferiores da árvore o máximo possível (e fundi-las novamente, se possível)

Algoritmo de Otimização Algébrica

- Passo 3

- aplicar regra 9

- mudar de posição sub-árvores envolvidas em operações produtórias
 - objetivos
 - combinar prioritariamente sub-árvores com menor número de dados
 - » investigar sub-árvores com seleções mais **restritivas**
 - evitar produtos cartesianos
 - » combinações sem atributos de junção
 - como saber quais as seleções mais **restritivas**?
 - análise do **grau de seletividade** de um predicado
 - » estatística geralmente mantida no DD

- regra 5

- útil apenas para processadores que executam junções cujo laço externo se aplica à relação da esquerda (deve ser a de menor tamanho)

Grau de Seletividade ($GS_{ai}(R)$)

- Definido pela seguinte razão
 - $GS_{ai}(R) = t_p(R) / |R|$, onde $t_p(R)$ é o número de tuplas que satisfazem o predicado aplicado sobre um atributo ai em uma relação R e $|R|$ é o número de tuplas em R ($G_s \in [0,1]$)
- $GS_{ai}(R)$ pequeno (≈ 0) \Rightarrow seleção mais restritiva
- Um atributo chave a_c possui baixo G_s em predicados de igualdade
 - $GS_{ac}(R) = 1 / |R|$
- Simplificação: mantém-se uma estimativa de distribuição uniforme de valores de atributos
 - $GS_{ai}(R) = (|R| / V(a_i)) / |R| = 1 / V(a_i)$, onde $V(a_i)$ é o número de valores distintos de a_i

Algoritmo de Otimização Algébrica

- Passo 4

- aplicar a regra 12
 - otimizar operações produtórias

- Passo 5

- aplicar as regras 3, 4, 7 e 11
 - desmembrar e mover projeções para níveis inferiores da árvore, tanto quanto possível, definindo novas projeções conforme se faça necessário

- Passo 6

- identificar sub-árvores que representem grupos de operações que possam ser executados por um único algoritmo
 - defina-os uma única vez (uma única sub-árvore) na “árvore”

Passo 6 - Exemplo

