

Disciplina: Banco de Dados II
Semestre: 2008/2
Professor: Ronaldo S. Mello

Lista de Exercícios 2 – Processamento de Consultas

Considere as seguintes tabelas em um BD relacional:

Médicos (codm, nome, cidade, idade, especialidade)
Pacientes (codp, nome, cidade, idade, doença)
Consultas (codm, codp, data, hora)

e as seguintes estimativas no catálogo do BD:

Médicos:

$n_{Med} = 100$ tuplas; $t_{Med} = 50$ bytes; $t_{Med}(codm) = 5$ bytes; $t_{Med}(nome) = 15$ bytes; $t_{Med}(cidade) = 15$ bytes; $t_{Med}(idade) = 5$ bytes; $t_{Med}(especialidade) = 10$ bytes; $V_{Med}(nome) = 100$; $V_{Med}(cidade) = 50$; $V_{Med}(idade) = 40$; $V_{Cons}(especialidade) = 10$; um índice primário árvore-B para *codm* (I1) com $N = 10$ e $f_{I1} = 10$; um índice secundário *hash* para *especialidade* (I2); um índice secundário árvore-B para *cidade* (I3) com $N = 5$ e $f_{I3} = 5$. A tabela está ordenada por *codm*.

Pacientes:

$n_{Pac} = 500$ tuplas; $t_{Pac} = 55$ bytes; $t_{Pac}(codp) = 10$ bytes; $t_{Pac}(nome) = 15$ bytes; $t_{Pac}(cidade) = 15$ bytes; $t_{Pac}(idade) = 5$ bytes; $t_{Pac}(doença) = 10$ bytes; $V_{Pac}(nome) = 500$; $V_{Pac}(cidade) = 150$; $V_{Pac}(idade) = 200$; $V_{Pac}(doença) = 100$; um índice primário árvore-B para *codp* (I4) com $N = 7$ e $f_{I4} = 7$; um índice secundário árvore-B para *cidade* (I5) com $N = 5$ e $f_{I5} = 5$; um índice secundário *hash* para *doença* (I6). A tabela está ordenada por *codp*.

Consultas:

$n_{Cons} = 1000$ tuplas; $t_{Cons} = 30$ bytes; $t_{Cons}(codm) = 5$ bytes; $t_{Cons}(codp) = 10$ bytes; $t_{Cons}(data) = 10$ bytes; $t_{Cons}(hora) = 5$ bytes; $V_{Cons}(data) = 300$; $V_{Cons}(codp) = 450$; $V_{Cons}(codm) = 80$; $V_{Cons}(hora) = 15$; um índice secundário árvore-B para (*codm, codp, data*) (I7) com $N = 5$ e $f_{I7} = 5$; um índice secundário *hash* para *codm* (I8); um índice secundário *hash* para *codp* (I9); um índice secundário árvore-B para *data* (I10) com $N = 5$ e $f_{I10} = 5$. A tabela está ordenada por *data*.

1 bloco = 1 kb e $n_{buf} = 5$

1) Mostre a aplicação passo a passo do algoritmo de otimização algébrica para as consultas a seguir. Observação “< > ALL” é idêntico a “NOT IN”.

- a)

```
SELECT m.codm, m.nome, p.codp, p.nome, c.data, c.hora
FROM Médicos m, Pacientes p, Consultas c
WHERE m.codm = c.codm
AND p.codp = c.codp
AND c.data >= '01/03/2008'
AND c.data <= '31/03/2008'
```
- b)

```
SELECT m.codm, m.nome
FROM Médicos m
WHERE m.especialidade = 'ortopedia'
AND m.cidade < > 'Florianópolis'
AND m.codm IN ( SELECT codm
FROM Consultas
WHERE data = '15/03/2008')
```
- c)

```
SELECT p.codp, p.nome, c.data
FROM Pacientes p, Consultas c
WHERE p.codp = c.codp
AND p.idade <= 50
AND p.cidade < > ALL ( SELECT cidade
FROM Médicos
WHERE codm = 4760 OR codm = 5118)
```

2) Apresente as estimativas de custo para as árvores algébricas otimizadas que você obteve no exercício anterior e indique o melhor plano de execução para cada operador algébrico existente nelas, bem como o tamanho estimado da relação resultante.

3) Considere duas relações $R1(x,y)$ e $R2(x,y)$. Apresente um algoritmo de alto nível para a execução da operação de **união** de $R1$ e $R2$ ($R1 \cup R2$). Considere que $R1$ e $R2$ estão ordenadas de forma ascendente pelos pares de valores dos atributos x e y .