
Nome do aluno: _____ Matrícula do aluno: _____

Quando necessário, considere o esquema relacional da Figura 1 e as estatísticas da Figura 2 para responder as questões.

1) (1,5 pontos) Suponha que se deseja fazer uma seleção por igualdade sobre um atributo, e existe um índice secundário sobre esse atributo. Duas das possíveis estratégias de busca são a busca linear e o uso de índice. Suponha que a tabela ocupe 1.276 blocos. Indique o quão seletivo esse atributo deve ser para valer a pena usar a estratégia de índice. A seletividade deve ser um valor percentual, medindo a razão entre o número de blocos de interesse e número total de blocos. O cálculo do custo deve levar em consideração a soma dos tempos de *seek* e transferência, sendo que um *seek* é 50 vezes mais lento que uma transferência. Considere que a árvore do índice tenha altura igual a quatro.

Resposta

Busca Linear:

$$1\ ts + 1.276\ tt = 1326$$

Busca com índice:

$$(h+N) * (tt+ts)$$

$$(4 + N) * (51)$$

$$204 + 51N$$

$$1326 > 204 + 51N$$

$$1122 > 51N$$

$$N < 22$$

$$\text{seletividade} = N/1276 = 1,7 \%$$

2) (1,5 pontos) Considere que se deseja responder a consulta abaixo. A tabela *func* ocupa 100 blocos e a tabela *aloc* ocupa 200 blocos. Duas estratégias de junção possíveis são *block nested loop join* e *indexed nested loop join*. Supondo que a tabela *func* seja usada no nível externo, indique para qual $V(\text{salario}, \text{func})$ passa a ser vantajoso usar o algoritmo de junção com índice. Considere o custo como o tempo necessário para realizar os *seeks* e transferências, sendo que um *seek* é quatro vezes mais lento que uma transferência e que o custo do índice seja de quatro blocos. *Suponha que salários iguais a 2.000 existam em todos os blocos de func.*

```
SELECT * FROM func NATURAL JOIN aloc  
WHERE func.salario = 2.000
```

Resposta:

Block Nested Loop Join: o filtro não interfere na equação geral, uma vez que todos os blocos de func serão acessados.

*$b(func) + b(func) * b(alloc)$ transferências*

$$100 + 100 * 200 = 20.100$$

$b(func) + b(func)$ seeks

$$100 + 100 = 200$$

$$200 * 4 = 800$$

$$20.100 + 800 = 20.900$$

Com índice: Para os funcionários que não ganhem 2.000, não será necessário usar o índice para buscar suas alocações. Vamos chamar os funcionários que ganham 2.000 reais de $n(func)$

*$b(func) + n(func) * h$ (seeks e transferências)*

*$(100 + n(func) * 4) * 5$ {5 = tempo de realizar 1 seek e uma transferência}*

$$500 + n(func) * 20$$

$$500 + n(func) * 20 < 20.900$$

$$n(func) = 1020$$

com até 1020 funcionários ganhando 2.000 reais é vantajoso usar o índice. Isso corresponde a quantos valores distintos de salário?

$$n(func) = total(func) / V(salario, func)$$

$$1020 = 15.675 / V(salario, func)$$

$$V(salario, func) = 15.675 / 1020 = 15,36$$

A partir de $V(salario, func) = 16$ passa a valer a pena usar o índice

Como essa questão foi de alta complexidade, foram aceitas resoluções alternativas.

3) (1,5 pontos) Uma das estratégias quando se executa um plano de execução é o *pipeline*. Explique de forma clara e abrangente as duas formas básicas de realizar *pipeline*.

Resposta:

Na estratégia de pipeline, uma operação consome as tuplas que são produzidas pelas operações abaixo delas, à medida que essas tuplas vão sendo geradas.

Existem duas variações: Na push, as operações de nível inferior geram tuplas e as coloca em um buffer. A operação de nível superior devem retirar essas tuplas do buffer. Essa comunicação entre as operações pode ser realizada através do paradigma de programação paralela produtor/consumidor.

Na estratégia pull, a operação de nível superior deve solicitar tuplas para a operação que está abaixo. Essa comunicação entre as operações pode ser realizada através de funções de iteração, como open, next e close.

4) (1,5 pontos) Em álgebra relacional, as operações de seleção são comutativas. Exemplifique essa propriedade usando duas consultas equivalentes em álgebra relacional sobre a tabela func. Para auxiliar na explicação, mostre que o resultado gerado pelas consultas é o mesmo.

Resposta:

Registros de func (idFunc, idDepto, nomeFunc, salario):

*(1, 1, 'Joao', 2000)
(2, 1, 'Ana', 3000)
(3, 2, 'Pedro', 3000)
(4, 2, 'Cesar', 4000)
(5, 3, 'Carla', 4000)
(6, 3, 'Paulo', 2000)*

Consulta 1:

salario > 2.000 (idDepto = 3 (func))

Consulta 2:

idDepto = 3 (salario > 2.000 (func))

Nos dois casos, a resposta é a mesma: (5, 3, 'Carla', 4000)

5) (1 ponto) Considere a visão materializada que faça a interseção entre tuplas da relação R e da relação S. De que forma o SGBD poderia realizar a manutenção incremental da visão caso uma tupla seja inserida em R? E no caso de remoção de uma tupla em R?

Resposta:

No caso de inserção, o SGBD pode verificar se a tupla inserida existe em S. Caso exista, a tupla é adicionada ao resultado da visão.

No caso de remoção, o SGBD pode verificar se essa tupla existe na visão. Caso exista, a tupla é removida do resultado da visão.

6) Com base na consulta abaixo:

```
SELECT *  
FROM func NATURAL JOIN aloc NATURAL JOIN proj  
WHERE salario = 2.000 OR custo = 4.000
```

responda as seguintes questões:

- a) (1,5 pontos) Crie uma expressão em álgebra relacional usando as regras de otimização discutidas em aula.
- b) (1,5 pontos) Com base nas estatísticas disponíveis, estime quantos registros seriam retornados por essa consulta.

Resposta:

Letra a)

rel1 proj /X/ aloc
rel2 func /X/ rel1
resp salario = 2.000 v custo = 4.000 (rel2)

Letra b)

Existem 10.000 alocações
1/5 devem ser de funcionários que ganham 2.000
1/5 devem ser de projetos com custo igual a 4.000

Número de registros que não satisfazem nenhuma das condições
*100.000 * 4/5 * 4/5*
*100.000 * 16/25 = 64.000*
Número de registros que satisfazem pelo menos uma das condições
100.000 - 64.000 = 36.000

func (<u>idFunc</u> , nomeFunc, salario)
proj (<u>idProj</u> , nomeProj, custo, duracao)
aloc (<u>idFunc</u> , <u>idProj</u> , função)
idFunc referencia func
idProj referencia proj

Figura 1 – Esquema Relacional

Chave	Valor
N. registros func	15.675
N. registros aloc	100.000
N. registros proj	50
V(salario, func)	5
V(custo, proj)	5

Figura 2 – Estatísticas das tabelas