
Nome do aluno: _____ Matrícula do aluno: _____

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 1.000 blocos e os registros que atendem à seleção estão espalhados em N blocos. Indique a partir de qual valor de N passa a valer a pena usar a estratégia de busca linear. 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:

$$1 \text{ ts} + 1.000 \text{ tt} = 1050$$

Busca com índice:

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

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

$$204 + 51N$$

$$1050 < 204 + 51N$$

$$876 < 51N$$

$$N > 16,58$$

$$N > 17$$

2) (1,5 pontos) Suponha que se deseja usar a ordenação externa por intercalação para ordenar um conjunto de registros divididos em 1024 partições com dois registros cada. A intercalação utiliza um buffer de entrada com capacidade para N partições de entrada. Qual deve ser o tamanho mínimo de N para que a ordenação seja concluída sem que sejam necessárias mais do que cinco passadas.

Resposta:

$$\log_2(1024) = 5$$

$$1024 = x^5$$

$$2^10 = x^5$$

$$2^5 \cdot (2^5) = x^5$$

$$4^5 = x^5$$

$$x = 4$$

Tamanho mínimo deve ser igual a quatro.

3) (1,5 pontos) Considere que se deseja fazer uma junção entre as tabelas A e B. A tabela A ocupa 200 blocos e a tabela B ocupa 500 blocos. Duas estratégias de junção possíveis são *block nested loop join* e *indexed nested loop join*. Supondo que a tabela A seja usada no nível externo, indique para qual número de registros de A passa a valer a pena usar *block nested loop join*. Considere o custo como o tempo necessário para realizar os seeks e transferências, sendo que um *seek* é 50 vezes mais lento que uma transferência e o custo do índice é de 5 blocos.

Resposta:

Block Nested Loop Join

Transferências

$$b(A) + b(A) * b(B)$$

$$200 + 200 * 500 = 100.200$$

Seeks:

$$b(A) + b(A)$$

$$200 + 200 = 400$$

$$400 * 50 = 20.000$$

$$100.200 + 20.000 = 120.200$$

Indexed Nested Loop Join

$$b(A)*(tt + ts) + n(A) * c * (tt + ts)$$

$$200 * 51 + n(A) * 5 * 51$$

$$10.200 + n(A) * 5 * 51$$

$$10.200 + 255 n(A)$$

$$120.000 < 10.200 + 255 n(A)$$

$$n(A) > 430,58$$

$$n(A) \geq 431$$

Considere o esquema relacional da Figura 1 e as estatísticas da Figura 2 para responder as questões de 4 até 7.

4) (1,0 pontos) Marque V para as consultas que precisam se preocupar com remoção de duplicatas e F caso contrário? Para acertar a questão, todas marcações precisam estar corretas. Responda na folha de respostas.

- a) Select distinct idFunc from func natural join depto
- b) Select distinct idDept from func natural join depto
- c) Select idFunc from func natural join depto
- d) Select idDept from func natural join depto
- e) Select distinct idFunc, idDept from func natural join depto

Resposta: F-V-F-F-F

Das que possuem distinct, apenas na letra b os registros de fato podem ser duplicados.

5) (1,5 pontos) Em álgebra relacional, a expressão $\Pi_L (E1 \cap E2)$ e a expressão $(\Pi_L E1) \cap (\Pi_L E2)$ não são equivalentes. Demonstre um caso que prove essa afirmação usando consultas SQL sobre a tabela func. Para auxiliar na explicação, mostre os registros da tabela, bem como os registros retornados pelas consultas.

Resposta:

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

(1, 1, 'Joao', 2000)
(2, 1, 'Ana', 3000)
(3, 2, 'Pedro', 3000)
(4, 2, 'Cesar', 4000)

Consulta 1:

```
select distinct salario from (
    select idDepto,salario from func where idDepto = 1
    Intersect
    select idDepto, salario from func where idDepto = 2
)as tab
```

Resposta da consulta:

```
{(1,2000),(1,3000)} ∩ {(2,3000),(2,4000)}
{}
```

Consulta 2:

```
select distinct salario from (
    select idDepto,salario from func where idDepto = 1) as tab1
    Intersect
    select distinct salario from (
        select idDepto,salario from func where idDepto = 2) as tab2
```

Resposta da consulta:

```
{2000,3000} ∩ {3000,4000}
{3000}
```

As duas consultas geram resultados diferentes

6) (1 ponto) O processamento de consultas compostas por subconsultas pode ser valer de uma técnica conhecida como descorrelação, onde tabelas temporárias são criadas para que o resultado possa ser encontrado mais rapidamente. Ao aplicar essa técnica, qual script SQL seria criado para responder à consulta abaixo? Considere como parte do script tanto a criação quanto o uso da tabela temporária.

```
SELECT * FROM func  
WHERE salario IN (  
    SELECT custo  
    FROM projeto )
```

Resposta:

(em pseudo-linguagem):

```
tabela temp = (SELECT custo FROM projeto);  
select distinct func.* from func join temp on func.salario = temp.custo;
```

7) (2 pontos) Considere a consulta abaixo:

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

Com base nas estatísticas disponíveis, estime quantos registros seriam retornados por essa consulta.

Resposta:

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 satisfazem ambas condições

$10.000 * 1/5 * 1/5$

$10.000 * 1/25 = 400$

```
depto (idDept, nomeDept, setor)  
func (idFunc, idDept, nomeFunc, salario)  
    idDept referencia depto  
proj (idProj, nomeProj, custo, duracao)  
aloc (idFunc, idProj, função)  
    idFunc referencia func  
    idProj referencia proj
```

Figura 1 – Esquema Relacional

Chave	Valor
N. registros func	1.000
N. registros aloc	10.000
N. registros proj	50
V(salario, func)	5
V(custo, proj)	5

Figura 2 – Estatísticas das tabelas