
Nome do aluno: _____ Matrícula do aluno: _____

1) (2,0 pontos) Suponha que se deseja fazer uma **seleção por igualdade** sobre um atributo que é **chave primária** e **chave de ordenação** de um arquivo que ocupa **4096 blocos**. Indique qual será o custo **médio** em número de **transferências de blocos** necessárias para cada uma das estratégias abaixo:

a) busca linear

Resposta: $4096 / 2 = 2048$ transferências

b) busca binária

Resposta: $\log(4096) = 12$ transferências

c) uso de índice. (considere que todo o índice já esteja na memória)

Resposta: 1 transferência

2) (3,0 pontos) Considere que os **10.000 registros** da tabela **A** estão espalhados em **500 blocos**, e os **1.000 registros** da tabela **B** estão espalhados em **10 blocos**. Nesse cenário, indique o custo em número de **transferências de bloco** para realizar a junção entre essas duas tabelas para cada uma das estratégias abaixo. Considere que caiba apenas um bloco de cada tabela na memória.

a) **nested loop join** usando a tabela **A** no nível externo

Resposta:

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

$$500 + 10.000 * 10$$

$$500 + 100.000$$

$$100.500$$
 transferências

b) **block nested loop join** usando a tabela **A** no nível externo

Resposta:

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

$$500 + 500 * 10$$

$$500 + 5.000$$

$$5.500$$
 transferências

c) **merge join** (ignore o custo para ordenar as tabelas)

Resposta:

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

$$500 + 10 = 510$$
 transferências

3) (1,0 ponto) Marque V para a consultas que se **beneficiam de um índice composto (setor, salario)**, e F caso contrário? Cada item certo vale 0,2. Para pontuar, pelo menos três marcações precisam estar corretas. Responda na folha de respostas.

- a) select distinct salario, setor from func (V)
- b) select avg(salario) from func where setor = sul (V)
- c) select * from func where salario =2000 (F)
- d) select * from func where setor = sul and salario > 1000 (V)
- e) select max(salario) from func group by setor (V)

Resposta: V-V-F-V-V

4) (2,0 ponto) Em álgebra relacional, é possível realizar a seguinte transformação:
de $\sigma\theta(E1 \text{ ó } E2)$ para $\sigma\theta(E1) \cup E2$.

O mesmo não pode ser feito se substituirmos o operador de diferença pelo operador de união. Nesse caso a expressão transformada poderia trazer resultados diferentes. Demonstre um caso que prove essa afirmação usando **consultas SQL** sobre uma tabela qualquer. Para auxiliar na explicação, mostre os **registros da tabela**, bem como os **registros retornados** pelas consultas.

Resposta:

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

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

Consulta 1:

```
select salario from (
    select distinct salario from func where idDept = 1
    union
    select distinct salario from func where idDept = 2
    )as tab
where tab.salario > 3000
```

Resposta:

{2000,3000} \cup {3000,4000}
{2000,3000,4000}
 $>3000 = \{4000\}$

Consulta 2:

```
select distinct salario from func where idDept = 1 where salario > 3000
union
select distinct salario from func where idDept = 2
```

Resposta:

{ } \cup {3000,4000}

{3000,4000}

As duas consultas geram resultados diferentes

5) (2,0 pontos) Transforme a consulta SQL abaixo em uma expressão em álgebra relacional **otimizada**. Use as **três regras** de otimização vistas em aula.

```
select f.nome, p.nome
from func f join aloc a on f.idFunc = a.idFunc
              join proj p on a.idProj = p.idProj
where f.salario >=1000 and p.custo < 10.000
```

Para responder essa questão, considere as estatísticas da Figura 1. Além disso, considere que 5% dos funcionários ganham menos do que 1000 reais e 5% dos projetos custem menos do que 10.000 reais.

Resposta:

Seletividade:

$$\text{Salario} < 1.000 = 5/100$$

$$\text{Salario} \geq 1.000 = 95/100$$

$$\text{Custo} < 10.000 = 5/100$$

Número de registros após filtros:

$$n(\text{Func}) = 1.000 * 95/100 = 950$$

$$n(\text{Proj}) = 10.000 * 5/100 = 50$$

$$n(\text{aloc}) = 100.000$$

Melhor fazer a junção entre proj e aloc primeiro

```
func1 <= idFunc, nome ( salario >=1000 (func))
proj1 <= idProj, nome ( custo <10000 (proj))
juncao1 <= idFunc, nome ( aloc /X/aloc.idProj = proj.idProj proj1)
resp <= proj.nome, func.nome (func1 /X/aloc.idFunc = func.idFunc juncao1)
```

Chave	Valor
N. registros func	1.000
N. registros aloc	100.000
N. registros proj	10.000

Figura 1 ó Estatísticas das tabelas