

Nome do aluno: _____ **Matrícula do aluno:** _____

1) **(1,5 pontos)** Suponha que se deseja fazer uma **seleção por igualdade** sobre um atributo que é **chave primária** e **chave de ordenação** de um arquivo. Duas das possíveis estratégias de busca são a **busca linear** e a **busca binária**. Considerando que o tempo de seek seja **50 vezes** mais lento que o tempo de transferência, indique para qual **quantidade de blocos** do arquivo passa a **valer a pena** usar a **busca binária**. **Justifique sua resposta.**

- a) 64 blocos
- b) 128 blocos
- c) 256 blocos
- d) 512 blocos
- e) 1024 blocos

2) **(1,5 pontos)** Suponha que se deseja **ordenar** um conjunto de registros divididos em **4096 partições** com **dois registros** cada. Ainda, suponha que, por limitação do espaço em memória, seja usado um mecanismo de **intercalação** que mescle partições, gerando partições maiores e em menor número. O mecanismo escolhido consegue diminuir o número de partições pela **metade** após cada estágio de intercalação. Dadas essas características, **quantos** estágios de intercalação são necessários **ao todo**?

Considere o esquema relacional da Figura 1 e as estatísticas da Figura 2 para responder as questões de 3 até 7.

3) **(1,5 pontos)** Considere que os registros da tabela **aloc** estão espalhados em **500 blocos**, e os registros da tabela **depto** estão espalhados em **5 blocos**. Nesse cenário, indique o **menor** custo para realizar um **block nested loop join** entre essas duas tabelas, **somando** o número de **seeks** e o número de **transferências**. Considere que o seek seja **50 vezes** mais lento que a transferência e que só caiba **um bloco** de cada relação na **memória**.

4) **(1,5 pontos)** Marque V para a consultas que se **beneficiam pela ordenação** dos registros em alguma ordem específica, e F caso contrário? Para acertar a questão, **todas** marcações precisam estar corretas. Responda na folha de respostas.

- a) Select salario from func union select custo from projeto
- b) Select salario from func union all select custo from projeto
- c) Select salario from func where salario < 1000 union select salario from func where salario >2000
- d) Select idFunc from func where salario < 1000 union select idFunc from func where salario >2000

Obs. A cláusula **all** indica que todos registros devem ser considerados, independente da quantidade de vezes que cada um deles aparece.

5) (1,5 pontos) Em álgebra relacional, a operação de **diferença** não é **comutativa**. 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.

6) (1,5 pontos) Transforme a consulta SQL abaixo em uma expressão em álgebra relacional **otimizada**. Use as **três regras** de otimização vistas em aula.

```
select nomeDepto, avg(salario)
from depto join func using(idDepto)
group by nomeDepto
```

7) (1,5 pontos) Considere a consulta abaixo:

```
select *
from func natural join aloc natural join proj
where funcao = 'coordenador' or duracao = 2
```

Suponha que essa consulta foi transformada em um plano de execução usando as **regras de otimização** vistas em aula. Nesse caso, qual seria a **primeira junção** a ser feita? **Quantos registros** seriam retornados por essa junção?

depto (<u>idDepto</u> , nomeDepto, setor)
func (<u>idFunc</u> , idDepto, nomeFunc, salario)
idDepto referencia depto
proj (<u>idProj</u> , nomeProj, custo, duracao)
aloc (<u>idFunc</u> , <u>idProj</u> , <u>função</u> , duracao)
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(função, aloc)	10
V(duração,aloc)	20

Figura 2 – Estatísticas das tabelas