

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 o **uso de índice**. Indique para **qual quantidade de blocos** do arquivo passa a **valer a pena** usar o **índice**. O cálculo deve levar em consideração a soma dos custos de *seek* e transferência, sendo que um *seek* é 50 vezes mais lento que uma transferência. Para o uso de índice, considere que a árvore do índice tenha altura igual a **três**, para qualquer um dos tamanhos de arquivo considerados, e que também é necessário acessar o arquivo de dados. **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 **1024 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. Ainda, em **cada estágio** de intercalação, **cada registro** sofre **uma operação de entrada e uma de saída**. Dadas essas características, **quantas operações** de entrada e saída são necessárias **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 **proj** estão concentrados em um único bloco. Nesse cenário, indique o custo para realizar um **merge join** entre essas duas tabelas, em **número de seeks** e **número de transferências**, separadamente. **Ignore** o custo para ordenar as relações, e considere o pior caso, em que só cabe **um bloco** de cada relação na **memória**.

4) (1,5 pontos) Marque V para as 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 distinct idFunc, salario from func
- b) Select salario, count(*) from func group by salario
- c) Select * from func order by salario
- d) Select salario from func

5) (1,5 pontos) Em álgebra relacional, a operação de **diferença** não é **associativa**. Demonstre um caso que prove essa afirmação usando consultas em **álgebra relacional** 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 f.*
from depto natural join func
where setor = 'sul' and salario > 2.000
```

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

```
Select *
from func natural join alloc natural join proj
where salario = 2.000 and custo = 2000
```

Existem **duas** possíveis ordens de junção que respondem essa consulta usando as relações de chave estrangeira. Das duas, a **mais eficiente** envolve combinar primeiro **alloc** e **proj**. Qual **deveria ser** o valor de **V(salario, func)** para que **valesse a pena** realizar primeiro a junção entre **alloc** e **func**?

depto (<u>idDepto</u> , nomeDepto, setor)
func (<u>idFunc</u> , idDepto, nomeFunc, salario)
idDepto referencia depto
proj (<u>idProj</u> , nomeProj, custo, duracao)
alloc (<u>idFunc</u> , <u>idProj</u> , função)
idFunc referencia func
idProj referencia proj

Figura 1 ó Esquema Relacional

Chave	Valor
N. registros func	1.000
N. registros alloc	10.000
N. registros proj	50
V(salario, func)	3
V(custo, proj)	5

Figura 2 ó Estatísticas das tabelas