

DI- FCT-NOVA

27 de março de 2014

Sistemas de Bases de Dados

1º teste, 2013/14

Duração: 1 hora + 30 minutos (sem consulta)

Grupo 1

Considere parte duma base de dados de uma cadeia de ginásios, onde se regista informação de sócios da cadeia, dos professores das várias modalidades, as aulas em que os sócios estão inscritos, e os movimentos (que podem ser de tipo entrada ou saída) dos sócios e professores nos ginásios. Essa base de dados inclui as seguintes tabelas (onde os atributos que constituem a chave primária estão sublinhados):

<p> <code>personas({<u>BI</u>,Nome,Idade,Sx,CatRendimentos,HabLiterarias...})</code> <code>aulas({<u>CodAula</u>,Modalidade,DiaSemana,Hora,NumProf,Local})</code> <code>movimentos(<u>BI</u>,Ano,Dia,Hora,Local,Tipo)</code> </p>	<p> <code>socio({<u>BI</u>,DataInicio})</code> <code>inscricoes(<u>BI</u>,<u>CodAula</u>)</code> <code>professores({<u>BI</u>,Categoria})</code> </p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------

Para cada uma destas tabelas existe um índice clustered de árvore B+ sobre o(s) atributo(s) da chave primária. Além disso são definidas na base de dados as seguintes *foreign keys* todas elas não permitindo valores nulos: de BI em `professores`, `socios` e `movimentos` para BI em `personas`; de NumProf em `aulas`, para NumProf em `professores`; de BI em `inscricoes` para BI em `socios`; de `codAula` em `inscricoes` para `codAula` em `aulas`.

Tendo em conta o sistema de gestão de bases de dados usado, tipicamente cabem num bloco 20 tuplos da tabela de `personas` ou da tabela `aulas`, ou 40 tuplos da tabela `movimentos` ou ainda 100 tuplos da tabela `socios`, da tabela `professores` ou da tabela `inscricoes`.

Sabemos ainda que num dado momento a tabela `personas` tem 10.000 tuplos, a de `professores` tem 40 tuplos, a de `socios` 1.000 tuplos, a de `aulas` 500 tuplos, a de `inscricoes` 20.000 tuplos e a de `movimentos` 2.000.000 tuplos.

Nota: Neste grupo, sempre que se solicitarem exemplos, estes devem ser **exclusivamente** sobre esta base de dados. Além disso, **todas** as respostas deverão conter uma **breve justificação**.

- 1 a) Apresente um exemplo de pergunta SQL e respetivo processamento, em que a política LRU de substituição de blocos em *buffer*, usual em sistemas de operação, daria origem a um processamento pouco eficiente e em que, pelo contrário, a política MRU seria muito mais eficiente.
- 1 b) Considere que sobre a relação `personas` estão definidos três índices non-clustered de árvore B+, um sobre o atributo `nome`, outro sobre o atributo `idade`, e ainda outro sobre a concatenação dos atributos `nome` e `idade`. Qual dos três índices lhe parece que tornaria mais eficiente a resposta da pergunta:
select BI from pessoas where nome = 'José Júlio Alves Alferes' and idade = 40
- 1 c) Considere que as aulas se distribuem mais ou menos uniformemente pelas várias modalidade praticadas na cadeia de ginásio. Ou seja, o número de aulas (na relação `aulas`) de uma modalidade é mais ou menos o mesmo que o número de aulas de qualquer outra modalidade. Assumindo isto, acha que valeria a pena criar um índice de árvore B+ para o `aulas(Modalidade)`? Se sim, para que tipo de perguntas SQL esse índice seria útil? Se não, porque é que não?
- 1 d) Suponha agora que o sistema lhe permitia definir índices de bitmap. Apresente um exemplo duma pergunta SQL sem funções de agregação em que a existência dum índice de bitmap tornaria muito mais eficiente a execução.
- 1 e) Considere que está definido na relação `personas` um índice non-clustered de árvore B+ sobre o atributo `idade`, e também um índice de hash dinâmico sobre o mesmo atributo. Para cada uma das perguntas abaixo, diga qual das 3 abordagens lhe parece mais eficiente: usar o índice de B+; usar o índice de hash; não usar nenhum deles.

1. **select * from pessoas where idade > 21 and idade < 46**

2. **select * from pessoas where idade = 70**

3. **select * from pessoas where idade > 81 and idade < 106**

Grupo 2

Nota: Dê respostas **breves**.

2 a) Em geral, os sistemas de bases de dados usam a técnica de *slotted-page* para organizar os tuplos de uma tabela em blocos. (1) Explique as vantagens de organizar os tuplos de uma tabela em blocos, usando *slotted-pages*. (2) Que limitação é que este tipo de técnica coloca ao que cada tuplo pode conter, e como é que usualmente os sistemas de bases de dados contornam essa limitação?

2 b) Suponha que temos um índice de *extendable hashing* numa relação que num dado momento tem tuplos para as seguintes chaves de pesquisa

(2,3,5,11,17,19,23,29)

Apresente, graficamente, a estrutura desse índice, para uma situação em que cada *bucket* armazena no máximo 3 chaves de pesquisa, e a função de *hash* é $h(x) = x \bmod 8$.

2 c) Qual a diferença entre um índice *clustered* e um índice *non-clustered*?