

DI- FCT/UNL

24 de junho de 2016

Sistemas de Bases de Dados**Exame Recurso 2015/16****Duração: 3 horas (sem consulta)****Grupo 1**

O Euro 2016 exige uma base de dados para registar as vendas de bilhetes nos 48 jogos do evento desportivo. O esquema da base de dados para o sistema de bilhética tem as seguintes 6 tabelas (onde os atributos que constituem a chave primária estão sublinhados, sendo todos os códigos numerados sequencial e consecutivamente a partir de 1):

Seleções(<u>codPaís</u> , nomePaís, selecionador, grupo,...)	Estádios(<u>numEstádio</u> , nomeEst, localEst,...)
Jogadores(<u>codPaís</u> , <u>numJogador</u> , nomeJogador,...)	Lugares(<u>numEstádio</u> , <u>numLugar</u> , bancada,...)
Jogos(<u>codJogo</u> , paísA, paísB, Fase, Data, numEstádio)	Bilhetes(<u>codJogo</u> , <u>numLugar</u> , ocupado, NIF)

Para cada uma destas tabelas existe um índice secundário de árvore B+ sobre o(s) atributo(s) da chave primária, criado com a ordem das colunas indicada nas tabelas. A tabela bilhetes indica para todos os jogos se cada lugar se está **ocupado** (ou não) através de uma coluna booleana, e caso esteja ocupado, o NIF do comprador (NULL caso contrário). As colunas **paísA** e **paísB** são chaves externas para seleções, sendo as demais chaves externas evidentes (atenção que **numLugar** em **Bilhetes** não é chave externa para **Lugares**). A tabela **seleções** contém 32 tuplos (ocupando 3 blocos de disco), a tabela **jogadores** 704 tuplos (118 blocos), a tabela **jogos** 48 tuplos (1 bloco) e a tabela **estádios** 10 tuplos (5 blocos). A tabela **lugares** tem 450.000 tuplos (necessitando de 10.000 blocos) e a tabela **bilhetes** tem 2.160.000 de registos (necessitando de 40.000 blocos). Assuma que no SGBD escolhido um nó da árvore B+ pode conter cerca de 100 chaves de pesquisa e sabe-se que o tempo de um seek é dez vezes superior ao da transferência de um bloco ($t_s = 10 * t_T$), e a memória comporta apenas 100 blocos de 8192 Byte.

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 o melhor plano que consiga para avaliar a seguinte consulta, justificando porque o considera um bom plano.

```
SELECT nomePaís, nomeJogador FROM Seleções NATURAL INNER JOIN jogadores
ORDER BY nomePaís, nomeJogador
```

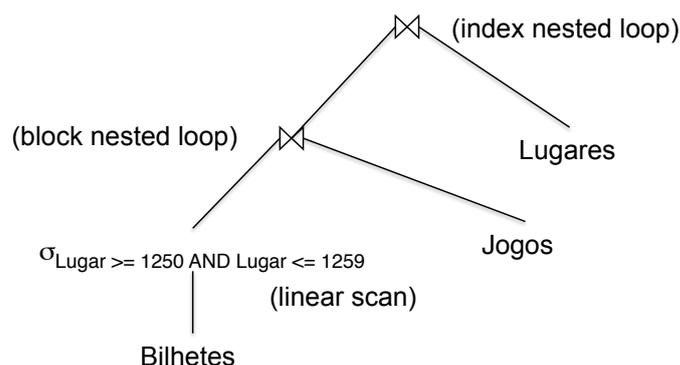
- 1 b) Indique se os índices existentes poderão responder eficientemente à seguinte consulta (e como), ou se seria preferível optar por outro tipo de índices justificando rigorosamente.

```
SELECT COUNT(*)
FROM Bilhetes WHERE codJogo = 10 AND ocupado = TRUE();
```

- 1 c) Determine o número estimado de tuplos devolvidos para a pergunta

```
SELECT Lugares.*
FROM Jogos NATURAL INNER JOIN Lugares WHERE codJogo = 10;
```

- 1 d) Considere o plano de avaliação apresentado na figura ao lado. Estime o menor custo de execução desse plano, indicando explicitamente qual a relação interior e exterior para cada operação de junção.



- 1 e) Os implementadores do sistema obrigam a que locks na tabela **Bilhetes** sejam efetuados por ordem de jogo e, para o mesmo jogo, por lugar. Justifique que desta forma não é possível criar deadlocks em transações que atualizem e utilizem somente a informação da tabela de bilhetes relativamente a lugares ocupados e/ou NIF (pode ilustrar a argumentação com exemplos).
- 1 f) Na tabela abaixo pode encontrar o log de um SGBD que utiliza o algoritmo de recuperação baseado em logs com checkpoints estudado. Sabendo que após o último registo ocorreu crash, indique cada uma das operações realizadas pelo sistema de recuperação a partir da informação existente, explicitando **todas** as alterações efetuadas na base de dados.

#		#	
1	<T ₁ , start >	7	<T ₃ , start >
2	<T ₁ , A, 0, 1 >	8	<T ₃ , C, 3, 4 >
3	<T ₁ , A, 1, 2 >	9	<T ₂ , B, 2 >
4	<T ₂ , start >	10	<T ₃ , commit >
5	<T ₂ , B, 2, 3 >	11	CRASH
6	<checkpoint {T ₁ , T ₂ } >		

- 1 g) Considere a seguinte transação distribuída que decorre em dois locais distintos de um adepto a comprar dois bilhetes. Indique os registos de log (por ordem) de controlo de concorrência em cada um dos locais sabendo que a transação é o coordenada pela máquina 1, que teve sucesso e que se recorre a um protocolo de controlo de compromisso em duas fases (2 Phase Commit).

#	Local 1	Local 2
1	begin transaction	
2	UPDATE bilhetes SET ocupado=TRUE, NIF='PT123456789' WHERE codJogo = 1 AND numLugar = 2000;	
3		UPDATE bilhetes SET ocupado=TRUE, NIF='PT123456789' WHERE codJogo = 1 AND numLugar = 2001;
4	End transaction	

- 1 h) Mostre através de um exemplo que podem existir transações em modo de isolamento READ COMMITTED que não são serializáveis.

Grupo 2

Nota: A resposta a cada uma das alíneas deste grupo **não pode em caso algum exceder uma página.**

- 2 a) Como sabe, as árvores B+ são a estrutura de dados de excelência para indexar informação em bases de dados. Contudo, para carregar grandes volumes de informação poderão ser ineficientes. Assumindo, que uma determinada tabela se encontra vazia explique quais as alternativas para implementar um comando de carregamento em lote (*bulk upload*).
- 2 b) Alguns sistemas de gestão de bases de dados constroem histogramas dos dados presentes nas tabelas (por amostragem parcial ou completa). Explique para que são utilizados, concretizando os seus benefícios com exemplo(s).
- 2 c) Apresente as vantagens e desvantagens de um sistema de controlo de locks distribuído baseado na técnica da cópia primária.