

**Sistemas de Bases de Dados**  
**3º teste 2014/15**  
**Duração: 1 hora + 30 minutos (sem consulta)**

**Grupo 1**

Considere parte de uma base de dados registando informação sobre contribuintes identificados por um NIF, podendo ser individuais ou colectivos (empresas), empresas e seus acionistas e respectivo capital social (montante em dinheiro), e atividade económica realizada pela empresa classificada de acordo com códigos de atividade (CAEs) tendo obrigatoriamente um código primário e no máximo 3 secundários (não primários). Essa base de dados inclui as seguintes tabelas, não sendo permitidos valores nulos em nenhum atributo (onde se sublinham os atributos que constituem a chave primária):

|                                                        |                                                                  |
|--------------------------------------------------------|------------------------------------------------------------------|
| contribuintes( { <u>NIF</u> ,Rua,Local,CodPostal,...}) | peessoas( { <u>NIFPess</u> ,Nome,Sexo,DataNasc,...})             |
| empresas( { <u>NIFEmpr</u> ,Nome,Tipo,DataInício,...}) | acionistas( { <u>NIFEmpr</u> , <u>NIFContribuinte</u> ,Capital}) |
| atividades( { <u>NIFEmpr</u> , <u>CAE</u> ,Primário})  | codigosAtividade( { <u>CAE</u> ,Designação,...})                 |

Todas as chaves são formadas por atributos inteiros. O atributo Tipo da tabela empresas tem 4 valores possíveis e o atributo Primário da tabela atividades é booleano. Para cada uma das 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 *chaves estrangeiras*: de NIFPess em pessoas e NIFEmpr em empresas para contribuintes(NIF), de NIFEmpr em acionistas e atividades para empresas, NIFContribuinte para contribuintes(NIF), e CAE em atividades para codigosAtividade. Os CAE encontram-se numerados sequencialmente a partir de 1.

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

**1 a)** Considere o escalonamento de três transações na tabela abaixo, estando criados previamente os contribuintes com os códigos de 1 a 4. Indique se o escalonamento provoca um deadlock (bloqueio mútuo) entre duas ou mais das transações. Em caso afirmativo, indique em que passo seria detectado pelo SGBD apresentando o estado da estrutura de dados que permitiria detectar o problema.

| Passo | Transação T1:                                    | Transação T2:                                   | Transação T3:                                    |
|-------|--------------------------------------------------|-------------------------------------------------|--------------------------------------------------|
| 1     | insert into pessoas<br>values('1','Carlos',...); |                                                 |                                                  |
| 2     |                                                  | insert into pessoas<br>values('2','Paula',...); |                                                  |
| 3     |                                                  |                                                 | insert into pessoas<br>values('3','Rita',...);   |
| 4     |                                                  | insert into pessoas<br>values('3','Rita',...);  |                                                  |
| 5     |                                                  |                                                 | insert into pessoas<br>values('4','Maria',...);  |
| 6     |                                                  |                                                 | insert into pessoas<br>values('1','Carlos',...); |
| 7     | insert into pessoas<br>values('2','Paula',...);  |                                                 |                                                  |
| 8     | commit;                                          |                                                 |                                                  |
| 9     |                                                  | commit;                                         |                                                  |
| 10    |                                                  |                                                 | commit;                                          |

**1 b)** Apresente um escalonamento ilustrando que com o protocolo de locking em 2 fases é possível ocorrerem rollbacks em cascata. Pode utilizar as operações abstractas **write**, **read**, **lock-x**, **lock-s** e **unlock** na construção do seu escalonamento.

**1 c)** Considere o seguinte registo de *log* de um sistema de gestão de base de dados cujo normal funcionamento foi interrompido por uma falha (crash). Indique o estado final do *log* e da base de dados após recuperação do crash (assuma que os únicos registos na base de dados são o A,B e C).

<T1 start>  
 <T2 start>  
 <T1,A,1,2>  
 <checkpoint {T1,T2}>  
 <T2, B, 1, 3>  
 <T3 start>  
 <T1,A,1>  
 <T3,C,1,4>  
 <T3 commit>  
 <T1 abort>  
 Crash ...

**1 d)** As transações no escalonamento abaixo são executadas em modo READ COMMITTED. Assumindo que as tabelas da base de dados se encontram vazias, indique que valores são escritos pelas instruções DBMS\_OUTPUT sabendo que n1 e n2 são variáveis numéricas. Justifique se as transações são serializáveis.

| Passo | Transação T1:                                  | Transação T2:                                  |
|-------|------------------------------------------------|------------------------------------------------|
| 1     |                                                | SELECT COUNT(*) INTO n2<br>FROM contribuintes; |
| 2     |                                                | DBMS_OUTPUT.PUT_LINE( n2 );                    |
| 3     |                                                | INSERT INTO Contribuintes VALUES('1',...);     |
| 4     | INSERT INTO Contribuintes VALUES('2',...);     |                                                |
| 5     |                                                | SELECT COUNT(*) INTO n2<br>FROM contribuintes; |
| 6     |                                                | DBMS_OUTPUT.PUT_LINE( n2 );                    |
| 7     |                                                | COMMIT;                                        |
| 8     | SELECT COUNT(*) INTO n1<br>FROM contribuintes; |                                                |
| 9     | DBMS_OUTPUT.PUT_LINE( n1 );                    |                                                |
| 10    | COMMIT;                                        |                                                |

**1 e)** A tabela **contribuintes** encontra-se replicada em cinco locais distintos num SGBD distribuído que utiliza o protocolo de maioria como esquema de controlo de concorrência. Descreva uma situação em que transações distribuídas possam ficar mutuamente bloqueadas ao aceder a um único contribuinte (e que não poderia ocorrer num sistema centralizado).

## Grupo 2

**Nota:** Dê respostas breves mas justifique adequadamente.

**2 a)** Explique no que consiste e para que é utilizada a técnica de Write-Ahead Logging (ou regra WAL).

**2 b)** Indique qual o problema principal do protocolo de commit em 2 fases (two phase commit – 2PC) que é endereçado pelo protocolo de commit em 3 fases (three phase commit - 3PC).