

DI- FCT-NOVA

Sistemas de Bases de Dados

Exemplo de 3º teste

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):

pessoas(<u>BI</u> , Nome, Idade, Sx, CatRendimentos, HabLiterarias...)	socios(<u>BI</u> , DataInicio, Quota)
aulas(<u>CodAula</u> , Modalidade, DiaSemana, Hora, NumProf, Local)	inscricoes(<u>BI</u> , <u>CodAula</u>)
movimentos(<u>BI</u> , <u>Ano</u> , <u>Dia</u> , <u>Hora</u> , Local, Tipo)	professores(<u>BI</u> , Categoria)

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 pessoas; de NumProf em aulas, para BI em professores; de BI em inscricoes para BI em socios; de codAula em inscricoes para codAula em aulas. Sabemos ainda que num dado momento a tabela pessoas tem 10.000 tuplos, a de professores tem 40 tuplos, a de socios 1.000 tuplos, a de aulas 500 tuplos, a de inscrições 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. Com exceção da alínea 1e), **todos os exemplos são em SQL**. Além disso, **todas** as respostas deverão conter uma **breve justificação**.

1 a) Apresente um escalonamento de duas transações concorrentes que, caso o sistema de gestão de base de dados recorra ao protocolo de *rigorous 2-phase lock*, provoque um *deadlock*.

1 b) Considere que são executadas as seguintes transações de forma concorrente:

<pre>begin transaction update socios set Quota = Quota*2 where BI=12345678; update socios set Quota = Quota *3 where BI=12345678; commit;</pre>	<pre>begin transaction update socios set Quota = Quota -20 where BI=12345678; commit;</pre>
---	---

Sabendo que antes de ambas as transações a **Quota** do **socio** com **BI** 12345678 é de 100, quais as possíveis **Quotas** desse sócio após as transações, se:

1. ambas as transações correrem em modo *serializable*;
2. ambas as transações correrem em modo *read committed*;

Para cada uma das **Quotas** finais possíveis, apresente um escalonamento das operações que lhe daria origem.

1 c) Apresente um escalonamento de duas transações concorrentes que, se o sistema de gestão de base de dados usasse o protocolo *2-phase lock* então uma das ações ficaria suspensa, mas que se usasse um protocolo baseado em *timestamps* não só nenhuma ação ficaria suspensa, como ambas as transações executariam com sucesso até ao fim (i.e. nenhuma delas teria que fazer *rollback*).

- 1 d) Suponha agora que a base de dados acima é fragmentada por um servidor central para toda a cadeia de ginásios mais um servidor por cada ginásio, da seguinte forma: o servidor central tem as tabelas pessoas, socios, e professores; o servidor de um dado ginásio, tem exatamente os tuplos das tabelas de movimentos e de aulas onde o Local é esse ginásio, e ainda os tuplos de inscricoes correspondentes às aulas onde o Local é o ginásio.

Qual lhe parece que seria o melhor plano de execução para a seguinte pergunta SQL, feita no servidor central, assumindo que o número de sócios com aulas no ginásio da “Almada” é pequeno face ao número total de sócios, mas que cada um desses sócios tem muitos movimentos nesse ginásio?

```
select pessoas.*  
from pessoas natural inner join socios natural inner join movimentos  
      natural inner join inscrições natural inner join aulas  
where aulas.Local = “Almada”
```

- 1 e) Suponha agora que, para além da fragmentação descrita em 1d), as tabelas pessoas, socios, e professores são replicadas nos servidores de todos os ginásios da cadeia. Descreva uma situação (com um escalonamento de uma transação distribuída e ocorrência de falhas) em que o protocolo *3 Phase Commit* seja vantajoso face ao *2 Phase Commit*.

Grupo 2

Nota: Dê respostas breves.

- 2 a) A garantia das propriedades ACID na execução de transações concorrentes é crucial em muitas aplicações de bases de dados.

Dê um exemplo de um cenário concreto sobre uma base de dados (e.g. de um banco) em que seria muito grave o sistema de gestão de bases de dados não garantir a propriedade de **Isolamento**.

- 2 b) Nas transações ACID em bases de dados, a propriedade de **Consistência** só exige que não haja violação de restrições de integridade no final de cada transação, sendo permitida a violação de restrições de integridade a meio de uma transação.

Porque é que assim é? Em particular, porque é que não é importante garantir que não há restrições de integridade violadas a meio das transações, e que problemas existiriam caso se exigisse a não violação de restrições a todo o momento?

- 2 c) Para controlo de concorrência em bases de dados distribuídas com replicação, há, entre outros, protocolos baseados na existência de cópias *primárias* de cada *item*, e protocolos baseados em maiorias. Diga em que situações concretas estes últimos (de maiorias) têm vantagem sobre os primeiros.