DI- FCT/UNL

Sistemas de Bases de Dados

3º teste (Exemplo)

Duração: 1 hora + 30 minutos (sem consulta)

Grupo 1

(Para não estranhar!) considere uma base de dados para registar consultas feitas num hospital e receitas de fármacos feitas em cada consulta. A base de dados inclui as seguintes tabelas (onde em cada uma delas apenas se apresentam parte dos atributos, e os atributos das chaves primárias estão sublinhados):

fármacos({NumF,NomeF,...}) consultas({NumCons, CodMed,NomePaciente, Data,...})
receitas(NumCons,NumF,quantidade,...) médicos({CodMed,Nome,Sx,Especialidade,...})

Considere ainda que são definidas chaves estrangeiras para expressar que: não pode haver consultas de médicos que não estão registados no hospital, não são considerados médicos que não tenham dado pelo menos uma consulta, não pode haver receitas sem estarem associadas a alguma consulta, e não podem ser prescritos fármacos que não estejam registados. Tais chaves podem ser expressas em SQL por:

alter table consultas **add constraint** cfk **foreign key** (CodMed) **references** médicos(CodMed)
alter table médicos **add constraint** mfk **foreign key** (CodMed) **references** consultas(CodMed)
alter table receitas **add constraint** rfk **foreign key** (NumCons) **references** consultas(NumCons)
alter table receitas **add constraint** ffk **foreign key** (NumF) **references** fármacos(NumF)

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 protoloco de *rigorous 2-phase lock*, provoque um *deadlock*.

1 b) Considere as seguintes transações concorrentes:

begin transaction	**begin transaction**
select * from fármacos;	**select * from fármacos;**
insert into fármacos values (1,'aspirina',...);	**insert into fármacos values (2,benuron,...);**
select * from fármacos;	**select * from fármacos;**
commit;	**commit;**

Apresente um escalonamento das ações nestas duas transações em que o resultado seja diferente consoante se esteja em modo de isolamento Read uncommited, Read commited ou Serializable, mostrando qual o resultado para cada um dos 3 modos de isolamento.

1 c) Apresente um escalonamento de **três** transações concorrentes que não seja serializável por conflito mas que seja serializável por vista.

1 d) 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 e) Na definição das propriedades ACID das transações em bases de dados, a consistência só é exigida no final de cada transação, e não no final de cada ação que constituí cada uma das transações. Mostre através de um exemplo de operação sobre a base de dados acima, que exigir a verificação de consistência após cada ação, para além de desnecessário, seria prejudicial.

1 f) Suponha agora que a base de dados acima é distribuída de tal forma que cada uma das tabelas é armazenada num servidor diferente, sem qualquer replicação. Dê um exemplo de uma pergunta SQL para a qual faria sentido usar uma estratégia de semi-join.
Para responder a esta questão pode, se o entender, assumir atributos adicionais nas várias tabelas

Grupo 2

Nota: Dê respostas **breves**.

2 a) *"Os protocolos para controlo de acesso concorrente a bases de dados baseados em locks são usualmente chamados de pessimistas".*

Em que é que os protocolos baseados em *locks* são pessimistas? Por oposição, o que seria um protocolo otimista?

2 b) Que problema existente em bases de dados distribuídas é endereçado pelo protocolo *2-phase commit*?

Na sua resposta refira qual (ou quais) das propriedades ACID é/são mais relevante(s) nesse problema. Mencione ainda porque é que em bases de dados centralizadas não faz sentido sequer falar neste protocolo.