



Instituto Superior de Engenharia de Lisboa
Área Departamental
de Engenharia de Electrónica e Telecomunicações e de Computadores
Licenciatura Engenharia Informática e de Computadores (2º Sem 2020/2021)
Sistemas de Informação - exame de EN (2022/06/23) Duração: 2:30 horas
Docentes: Afonso Remédios, Nuno Datia e Walter Vieira

1) Numa plataforma que permite a atribuição de tarefas a funcionários foram criadas as seguintes tabelas:

```
create table Funcionario(  
    num int primary key,  
    nome varchar(255) not null,  
    idade int not null  
);  
  
create table tarefa(  
    id int primary key,  
    nome varchar(255) not null  
);
```

```
create table funcionario_Tarefa(  
    num_func int references Funcionario,  
    id_tarefa int references Tarefa,  
    primary key(num_func, id_tarefa)  
);
```

- a) Crie em PL/pgSQL o procedimento *inicializaFuncionariosSemTarefas* que:
- Cria a tarefa de nome 'Sem tarefa' caso ainda não exista
 - caso existam funcionários sem tarefas, que os associe à tarefa de nome 'Sem tarefa'
- Garanta a conveniente gestão de exceções.
Indique como poderia chamar o procedimento, garantindo a conveniente gestão transaccional.
- b) Crie uma função que devolva todos os funcionários (**num**, **nome**) bem como a soma de tarefas associadas a cada um, eliminando os funcionários com idade superior ao parâmetro *limiteInferior* passado à função e excluindo da soma de tarefas a tarefa de nome 'Sem tarefa'.
- c) Crie um gatilho associado a inserções na tabela **Funcionario_Tarefa**, que remova a ligação do funcionário à tarefa de nome 'Sem tarefa', caso ela exista.
- d) Apresente o código das classes de modelo (entidades) para modelar as tabelas apresentadas anteriormente de modo a serem utilizadas em JPA. Inclua todas as associações. Use anotações JPA.
- e) Usando JPA, Implemente o método **criar** do repositório **FuncionarioRep** que, garantindo gestão transaccional, insira o funcionário passado como parâmetro e que o associe à tarefa 'Sem tarefa' (se a tarefa não existir deverá ser gerada uma exceção).
- f) Usando JPA Implemente o método **obterFuncionariosSemTarefas** da classe **FuncionarioRep** que retorna os funcionários associados à tarefa 'Sem tarefa'.
- g) Usando JPA, Implemente o método **inserirFuncionarios(List<Funcionario> lf)** da classe **BusinessLogic** que, na mesma transação insere os funcionários constantes da lista **lf** passada como parâmetro, associando-os à tarefa 'Sem tarefa' (se a tarefa não existir deverá ser gerada uma exceção e abortada a transação). Deve usar a classe **FuncionarioRep** das alíneas e) e f), indicando as alterações que nela tiverem de ser feitas, se for caso disso.

- 2) Considere a tabela seguinte criada em postgresql, com os registos indicados bem como as duas transações indicadas, executadas após a criação e preenchimento da tabela:

<pre> create table t(a int primary key, b int); insert into t values(1,1),(2,2),(3,3); (1.1) start transaction; (1.2) set transaction isolation level read committed; (1.3) update t set b = 11 where a = 1; (1.4) update t set b = 22 where a = 2; (1.5) commit; </pre>	<pre> (2.1) start transaction; (2.2) set transaction isolation level ? (2.3) select * from t where a = 3; (2.4) select * from t where a = 2 for share; (2.5) select * from t where a = 1; (2.6) commit; </pre>
--	--

- Indique quais são os escalonamentos possíveis iniciados por: <(2.1),(2.2),(2.3),(1.1),(1.2),(1.3),...>
- Para cada um dos escalonamentos identificados em a), indique, para os níveis de isolamento *read committed* e *repeatable read*, o que observa como resultado da execução das várias instruções *select*. Deve indicar possíveis erros que ocorram nessas execuções.
- Considere que na instrução (2.4) se retira a cláusula *for share* e que este código se executava num sistema com o protocolo *two phase lock* com o nível de isolamento *repeatable read*. Indique quais seriam os escalonamentos possíveis iniciados por: <(2.1),(2.2),(2.3),(1.1),(1.2),(1.3),...>.
- Para cada um dos escalonamentos identificados em c), indique o que observaria como resultado da execução das várias instruções *select*.

- 3) Sobre outros modelos de representação de dados, responda às seguintes questões:

- O que entende por agregado e qual a sua importância em alguns dos modelos de bases de dados NoSql. Exemplifique com **Faturas, Itens e Produtos**, dizendo como seria a representação normal num SGBD relacional e como poderia ser numa base de dados *document oriented*, se se usasse este princípio.
- Qual é a distinção fundamental entre uma base de dados *Graph oriented* e uma base de dados relacional?

Cotação:

alínea	1.a	1.b	1.c	1.d	1.e	1.f	1.g	2.a	2.b	2.c	2.d	3.a	3.b	Total
cotação	2	2	2	1,5	1,5	1,5	1,5	2	2	1	1	1	1	20