DI-FCT-NOVA **31 de maio de 2021**

Bases de Dados

2º teste, 2020/21 – Versão C

Duração: 2 horas (consulta limitada)

A base de dados das reservas e de serviços da cadeia de restaurantes tem o seguinte esquema, onde as chaves primárias se encontram sublinhadas e chaves estrangeiras a **negrito**.

Restaurantes(nomeR,localidade,NIF)	Serviços(numS,**nomeR**,dataS,horaS,concluído,**codRs**)
Mesas(numM,**nomeR**,lugaresM,xM,yM)	AtribuiçõesMesas(**numS,numM**,num_clientes)
Reservas(codRs,**nomeR**, nomeC,telC,dataRs,horaRs)	Pedidos(numP,**numS**,**numM**,**codE**,**codA**,qtdP, entrP)
MesasReservadas(**codRs,numM**,num_clientes)	Artigos(codA,nome,tipo,preço)
HoráriosAbertura(**nomeR**,dataH,horaH)	Empregados(*codE*,nome,data)

No esquema Restaurantes serão mantidos os restaurantes da cadeia identificados apenas pelo seu nome, indicando-se também a sua localidade e número de identificação fiscal. No esquema Mesas regista-se o número, lugares disponíveis e coordenadas (xM,yM) de cada mesa na planta da sala do restaurante nomeR.

As Reservas para uma determinada data e hora são identificadas por um código único, realizadas para um restaurante por um cliente com nome (nomeC) e telefone (telC). No ato da reserva são imediatamente atribuídas as mesas com o número de lugares reservados em cada uma delas. No esquema HoráriosAbertura indica-se para cada restaurante, data e hora, os períodos de abertura: um tuplo para cada período de quarto de hora em que o restaurante está aberto, fechando todos antes da meia noite. Por exemplo, se o restaurante 'Garfo' estiver aberto entre as 19h00m e as 22h45m do dia 31 de maio de 2021, o conteúdo da tabela HoráriosAbertura conterá os seguintes tuplos para esse dia:

('Garfo', '2021-05-31','19:00')	('Garfo', '2021-05-31','20:00')	('Garfo', '2021-05-31','21:00')	('Garfo', '2021-05-31','22:00')
('Garfo', '2021-05-31','19:15')	('Garfo', '2021-05-31','20:15')	('Garfo', '2021-05-31','21:15')	('Garfo', '2021-05-31','22:15')
('Garfo', '2021-05-31','19:30')	('Garfo', '2021-05-31','20:30')	('Garfo', '2021-05-31','21:30')	('Garfo', '2021-05-31','22:30')
('Garfo', '2021-05-31','19:45')	('Garfo', '2021-05-31','20:45')	('Garfo', '2021-05-31','21:45')	('Garfo', '2021-05-31','22:45')

Os serviços têm um número identificador, a data e hora do seu início e uma indicação se já foi concluído (valor 1) ou não (valor 0). Se o serviço corresponde a uma reserva, coloca-se o código da reserva em codRs, caso contrário o valor do atributo codRs é nulo. As mesas atribuídas a cada serviço encontram-se representadas em instâncias de AtribuiçõesMesas, juntamente com o número de clientes ocupando realmente cada mesa.

Cada pedido regista a quantidade de um artigo solicitado por um empregado para uma mesa de um determinado serviço, assim como se já foi entregue (valor 1) ou não (valor 0); os pedidos são numerados sequencialmente para cada mesa de um serviço. Os artigos têm um código, um nome, um tipo e um preço de venda. Os empregados são identificados por um código e necessitamos de saber o seu nome e data de contratação.

A faturação de um serviço obtém-se somando o valor de todos os pedidos desse serviço, em que o valor de um pedido é a quantidade de artigos multiplicada pelo seu preço.

Com a exceção do atributo codRs em Serviços, nenhum atributo pode conter valores nulos.

Grupo I

O menu de cada restaurante é comunicado ao sistema de informação por intermédio de documentos XML que obedecem à seguinte DTD. Pode encontrar na página seguinte uma instância de um documento XML que que obedece à DTD disponibilizada.

```
<!DOCTYPE Menu[
    <!ELEMENT Menu (Entradas,Pratos,Bebidas,Sobremesas,Combinados*) >
        <!ATTLIST Menu restaurante CDATA #REQUIRED >
    <!ELEMENT Entradas (Item*) >
    <!ELEMENT Pratos (Item*) >
    <!ELEMENT Bebidas (Item*) >
    <!ELEMENT Sobremesas (Item*) >
    <!ELEMENT Item (#PCDATA) >
        <!ATTLIST Item codItem ID #IMPLIED
                       nome CDATA #REQUIRED
                       alérgenos CDATA #IMPLIED
                       tipo CDATA #REQUIRED
                       preço CDATA #REQUIRED >
    <!ELEMENT Combinado EMPTY>
        <!ATTLIST Combinado nome CDATA #REQUIRED
                            itens  IDREFS #REQUIRED
                            desconto CDATA #REQUIRED >
]>
```

1. **[1 valor]** Explique o que se obtém com expressão XPath seguinte, indicando resultado para o documento de exemplo fornecido (ver página seguinte)

   //*[count(*)<=2]/*

2. Apresente expressões XPATH sobre um ficheiro XML de acordo com a DTD acima que devolvam os resultados das seguintes perguntas:

   a) **[1]** quais os itens de tipo vegan que não têm alérgenos?
   b) **[1]** quais os nomes dos pratos combinados com preço total superior a 9€? O preço total de um combinado obtém-se somando o preço dos itens que o constituem e aplicando o desconto do combinado (indicado em percentagem no documento XML).

3. **[1]** Apresente código XQuery para traduzir um documento XML com a estrutura anterior num documento em que o conteúdo do elemento Combinado em vez de ser vazio passa conter cada um dos itens que forma o prato combinado. Ou seja, uma DTD exatamente igual à anterior mas agora com a seguinte especificação para o elemento Combinado:

```
<!ELEMENT Combinado (Item+)>
<!ATTLIST Combinado nome CDATA #REQUIRED
                    desconto CDATA #REQUIRED >
```

Os itens que ocorrem dentro dos elementos Combinado não deverão ter o atributo codItem. Lembre-se que a notação para a avaliação de uma expressão {...} também pode ser utilizada dentro de atributos.

Grupo I (cont.)

Exemplo de documento

```
<?xml version="1.0" encoding="UTF-8" ?>

<Menu restaurante="Garfo">

  <Entradas>
        <Item codItem="E1" nome="Croquetes" tipo="carne" preço="1.2">Croquetes especiais</Item>
        <Item codItem="E2" nome="Azeitonas" tipo="vegan" preço="0.95">Sortido de azeitonas</Item>
  </Entradas>

  <Pratos>
        <Item codItem="P1" nome="Bitoque" tipo="carne" preço="5.0">Bitoque com ovo a cavalo</Item>
        <Item codItem="P2" nome="Sardinhas" tipo="peixe" preço="6.0">Sardinhas na grelha</Item>
        <Item codItem="P3" nome="Salada" tipo="vegan" alérgenos="amêndoa" preço="4.0">Salada de verdes</Item>
  </Pratos>

  <Bebidas>
        <Item codItem="B1" nome="Água" tipo="bebida" preço="0.75">Água 1,5L</Item>
         <Item codItem="B2" nome="Sangria" tipo="bebida" preço="3.0">Sangria tinta</Item>
        <Item codItem="B3" nome="Café" tipo="bebida" preço="0.8">Café da Arábia</Item>
  </Bebidas>

  <Sobremesas>
  </Sobremesas>

  <Combinado nome="Combinado de carnes" itens="E1 P1 B2 B3" desconto="5"/>
  <Combinado nome="Combinado Vegetal" itens="E2 P3 B1 B3" desconto="5"/>

</Menu>
```

Grupo II

1. Apresente uma **consulta em SQL** para cada uma das perguntas:

a) **[2,0]** listar os nomes dos artigos, sem repetições e por ordem alfabética, pedidos ao empregado com o nome Francisco em serviços entre os dias 23 e 30 de maio de 2021 (inclusive).

b) **[2,0]** criar uma vista para listar para cada serviço as correspondentes atribuições de mesas e o número de lugares previamente reservados nessas mesas (deverá ser nulo caso o serviço não corresponda a uma reserva). Deverá constar toda a informação das tabelas envolvidas na consulta.

c) **[2,0]** criar uma vista MesasDisponíveis com número da mesa (numM), a data (dataD), hora (horaD), e lugares (lugaresD). Uma mesa está disponível à hora horaD do dia dataD se não foi reservada no intervalo]horaD-2h;horaD+2h[. Para simplificar, pode tratar tempos como se fossem números reais.

d) **[2,0]** listar os restaurantes e respetivos dias em que a faturação média por serviço concluído tenha sido superior a 100€.

e) **[2,0]** Recorrendo à linguagem **datalog**, obtenha para cada empregado (código e nome) os artigos (código e nome) que ele nunca vendeu. Assuma que as tabelas não têm valores nulos.

Grupo III

1. Como já reparou, a base de dados acima tem uma série de problemas de desenho e omissão de algumas restrições. Apresente o código SQL (de alteração de tabelas, introdução de novas restrições e/ou introdução de asserções) para impor cada uma das seguintes restrições de integridade:

a) **[1,5]** Todas as mesas reservadas para uma reserva têm de pertencer ao restaurante no qual se efetuou a reserva.

b) **[1,5]** As reservas só podem ser efetuadas em datas e horas presentes em HoráriosAbertura.

2. **[2,0]** Considere a vista ReservaClientes(codRs, nomeR, nomeC, telC, dataRs, horaRs, numPessoas) que indica o número de pessoas de cada reserva. Implemente um trigger que ao se inserir nesta vista um tuplo com codRS a nulo será reservada uma mesa disponíveis à horaRs da dataRs (pode utilizar a vista da questão II1.c); se as pessoas não couberem numa mesa, não será efetuada a reserva. Assuma que está criada a sequência **cod_reservas**.

3. **[1,0]** Considere o seguinte escalonamento de reserva de mesas efetuadas em modo **snapshot isolation**.

Transação 1	Transação 2	Transação 3
INSERT INTO MesasReservadas VALUES (1,1,6);		
	INSERT INTO MesasReservadas VALUES (2,2,4);	
	commit;	
		INSERT INTO MesasReservadas VALUES (3,3,5);
SELECT SUM(num_clientes) INTO :x FROM MesasReservadas; DBMS_OUTPUT.PUT_LINE(x);		
commit;		
		INSERT INTO MesasReservadas VALUES (3,4,2);
		SELECT SUM(num_clientes) INTO :z FROM MesasReservadas; DBMS_OUTPUT.PUT_LINE(z);
		commit;

Justifique apropriadamente se o escalonamento é serializável (recorde que DBMS_OUTPUT.PUT_LINE escreve no ecrã). Assuma que as transações não ficam bloqueadas, que se iniciam com a primeira instrução e que no início a tabela MesasReservadas se encontra vazia.