

DI-FCT-NOVA

3 de julho de 2019

# Bases de Dados

## Exame de Recurso, 2018/19 (Versão B)

### Consulta limitada

#### Parte I

(correspondente à matéria do 1º teste)

#### Grupo I.1

**Nota:** O enunciado deste grupo é *intencionalmente* vago. Ao responder deve, sempre que necessário, dizer o que assumiu e que acha que não está suficientemente detalhado no enunciado.

A PetroPortuguesa dedica-se à comercialização de combustíveis em estações de serviço. Cada estação de serviço tem um código único, um nome, telefone, hora de abertura e hora de fecho, latitude, longitude e morada (campo de texto único). Algumas destas estações podem estar situadas em localidades, pretendendo a PetroPortuguesa registar o distrito e nome da localidade mas obviamente uma localidade pode ter várias estações de serviço. Uma estação de serviço tem várias bombas de combustível numeradas sequencialmente (e.g. bomba 1, bomba 2) podendo estar ou não em pré-pagamento. Cada bomba disponibiliza alguns combustíveis para abastecimento. Um combustível tem um código e uma designação (e.g. "Gasóleo Simples", "GPL", etc...).

Uma tabela de preços indica para um período, especificado com uma data de início e de fim, os preços de todos os combustíveis, podendo a tabela ser aplicada a várias estações de serviço. Contudo, cada estação de serviço tem apenas uma tabela de preços em vigor.

As estações de serviço têm caixas electrónicas com um endereço IP e um campo de texto com a data de instalação do equipamento. Algumas destas caixas são self-service estando localizadas obrigatoriamente numa bomba de combustível (havendo necessidade de saber em qual bomba).

#### 1. [Exame]

- a) Esboce um diagrama de entidades e relações para a base de dados da PetroPortuguesa. **[Cotação em exame: 2 valores; cotação em substituição de teste: 3 valores]**
- b) Proponha uma base de dados relacional para armazenar a informação da PetroPortuguesa. Para cada uma das relações que propuser deve indicar a chave primária, e todas as chaves estrangeiras. Para definição da base de dados pode usar SQL, ou outra qualquer notação, desde que seja claro quais as relações, chaves e atributos. **[2; 3]**

#### 2. [EXTRA]

- a) Sabendo que só existem caixas manuais e self-service, pretende-se registar o histórico de abertura e subsequente fecho das caixas manuais. Repare que podem existir caixas abertas mas que ainda não se encontram fechadas. Deve-se registar o operador que abre e que fecha a caixa (pode não ser o mesmo). O sistema armazena o NIF e nome de cada operador. Altere o diagrama de entidades e relações para modelar estes requisitos, lembrando que não pode utilizar nulos e que, para esta pergunta, não pode adicionar quaisquer outros atributos para além daqueles mencionados no texto. Basta apresentar aquilo que muda e/ou é acrescentado. **[0; 1]**
- b) Proponha um conjunto de relações para armazenar esta informação (incluindo chaves primárias, candidatas e estrangeiras. Pode usar SQL, ou outra qualquer notação, desde que seja claro quais as relações, chaves e atributos. Como poderia simplificar o modelo caso fossem permitidos nulos? **[0; 1]**

**Grupo I.2**

A relação R especifica informação sobre os utilizadores responsáveis em cada momento pela fixação dos preços nas bombas de combustível :

$R = (\text{Estação, Bomba, Combustível, Preço, IP, DataHora, Utilizador})$

e deseja-se impor as seguintes dependências funcionais F:

Estação Bomba Combustível  $\rightarrow$  Preço IP

Estação Combustível  $\rightarrow$  Preço

IP DataHora  $\rightarrow$  Estação Bomba

Estação DataHora  $\rightarrow$  Utilizador

- a) **[Exame]** Apresente as duas chaves candidatas de R, justificando adequadamente. **[1; 1,5]**
- b) **[Exame]** Apresente uma cobertura canónica do conjunto de dependências F. **[1; 1,5]**
- c) **[Exame]** Decomponha R por forma a obter um esquema na 3ª forma normal que preserve as dependências de acordo com o algoritmo estudado. **[1; 1,5]**
- d) **[EXTRA]** Usando o algoritmo lecionado, decomponha sem perdas o esquema R, por forma a obter um conjunto de esquemas na forma normal de Boyce-Codd. **[0; 1,5]**

### Grupo I.3

A PetroPortuguesa pretende registar as vendas de combustíveis nas suas bombas. O esquema da base de dados inclui as seguintes relações, onde os atributos sublinhados constituem a chave primária. Os atributos não contêm nulos excepto nos casos explicitamente indicados:

<i>bombas</i> ( <u><i>idB</i></u> , número, estação)	<i>entregas</i> ( <u><i>idE</i></u> , data, matrícula)
<i>mangueiras</i> ( <u><i>idB</i></u> , <u>número</u> , <i>idDep</i> )	<i>reabastecimentos</i> ( <u><i>idDep</i></u> , <u><i>idE</i></u> , litros)
<i>depósitos</i> ( <u><i>idDep</i></u> , capacidade, <i>idComb</i> )	<i>vendas</i> ( <u><i>idV</i></u> , data, matrícula, preço, litros, <i>b</i> , <i>n</i> )
<i>combustíveis</i> ( <u><i>idComb</i></u> , designação, tipo)	

A relação *bombas* tem um identificador único de cada bomba e o seu número, assim como o identificador da estação de serviço. Na relação *mangueiras* especificam-se as mangueiras de cada bomba, numeradas sequencialmente dentro de cada bomba, e a que o depósito estão ligadas; repare-se que o número da mangueira não tem nada a ver com a coluna número da tabela *bombas*. Cada depósito tem uma capacidade máxima para armazenar um único combustível. O combustível tem um identificador interno, uma designação e um tipo (e.g. “Diesel” ou “Gasolina”).

Os depósitos são reabastecidos em entregas efetuadas numa data por um camião cisterna, cuja matrícula é também registada. Note-se que numa entrega podem ser reabastecidos vários depósitos. A relação *vendas* regista cada venda efetuada nas estações de serviço, com um identificador único, data, preço, litros, a mangueira *n* da bomba *b* utilizada e, finalmente, uma matrícula opcional (pode conter nulos).

Para além das chaves primárias indicadas, estão ainda definidas as seguintes chaves estrangeiras: *idB* em *mangueiras* referencia *idB* em *bombas*; *idDep* em *mangueiras* referencia *idDep* em *depósitos*; *idComb* em *depósitos* referencia *idComb* em *combustíveis*; *idDep* em *reabastecimentos* referencia *idDep* em *depósitos*; *idE* em *reabastecimentos* referencia *idE* em *entregas*. Finalmente, (*b*,*n*) em *vendas* referencia a chave primária de *mangueiras*.

### Grupo I.3

1. Apresente uma expressão em **álgebra relacional** para cada uma das perguntas:
  - a) **[Exame]** Quais os tipos de combustíveis disponíveis na estação de serviço de ‘Aveiras’ ?  
**[1,5; 2]**
  - b) **[Exame]** Quais os identificadores das bombas que não disponibilizam o combustível ‘S95’ ou o combustível ‘D95’ (identificadores dos combustíveis) ?  
**[1,5; 2]**
  - c) **[EXTRA]** Quais os identificadores das bombas que disponibilizam dois ou mais tipos de combustível (sem utilizar agregações)? **[0; 2]**

## Parte II

(correspondente à matéria do 2º teste)

### Grupo II.1

Para a realização deste grupo deve considerar o esquema de base de dados especificado no grupo anterior I.3 na página 3 deste enunciado.

1. Apresente **uma consulta SQL** para cada uma das perguntas do grupo I.3:
  - a) **[Exame]** Quais os tipos de combustíveis disponíveis na estação de serviço de 'Aveiras' ? **[1,5; 1,5]**
  - b) **[Exame]** Quais os identificadores das bombas que não disponibilizam o combustível 'S95' ou o combustível 'D95' (identificadores dos combustíveis)? **[1,5; 1,5]**
  - c) **[EXTRA]** Quais os identificadores das bombas que disponibilizam dois ou mais tipos de combustível (sem utilizar agregações)? **[0; 2]**
2. Apresente **uma consulta datalog** para a pergunta II.1.1b:
  - a) **[Exame]** Quais os identificadores das bombas que não disponibilizam o combustível 'S95' ou o combustível 'D95' (identificadores dos combustíveis)? **[1; 1]**
3. Apresente **uma consulta SQL** para cada uma das perguntas:
  - a) **[Exame]** Quais as bombas que venderam mais de 1000 litros de algum combustível no dia 3 de julho de 2019? **[1; 2]**
  - b) **[EXTRA]** Qual o valor médio diário de reabastecimentos de combustível (em litros)? **[0; 2]**

### Grupo II.2

1. Imponha restrições de integridade sobre a base de dados do grupo I.3, para garantir que:
  - a) **[Exame]** Numa estação não existem bombas com o mesmo número **[0,5; 0,5]**
  - b) **[Exame]** As bombas que disponibilizam GPL só têm uma mangueira **[0,5; 0,5]**
  - c) **[EXTRA]** Os reabastecimentos são feitos sempre em múltiplos de 1000 litros **[0; 0,5]**
2. Adicionou-se um atributo "total\_litros" à relação depósitos para registar o volume total de combustível disponível em cada depósito. Acrescentou-se uma tabela *medições* para indicar para cada depósito a data da última medição e volume medido (em litros). Assuma que esta tabela tem inicialmente uma linha por cada depósito com uma data e litros ambos NULL, significando que é necessário fazer uma medição.

*medições*(idDep, data, litros\_medidos)

Implemente um mais triggers para garantir que:

- a) **[Exame]** ao efetuar-se uma venda ou um reabastecimento seja atualizada a coluna total\_litros da tabela depósitos de forma a manter correcto o volume de combustível disponível em cada depósito. Se esse total for negativo ou exceder a capacidade do depósito, a correspondente entrada na tabela *medições* deve ficar com os campos data e litros\_medidos a NULL. Trate só do caso de reabastecimentos dos depósitos. **[1; 1]**
- b) **[EXTRA]** ao se atualizar as colunas *data* e *litros\_medidos* com valores não NULL, o valor da coluna total\_litros fique igual a litros\_medidos. **[0; 1]**

**Grupo II.3**

Considere o seguinte escalonamento de três transações em que o total\_litros da tabela depósitos tem inicialmente o valor 0 para os depósitos 1, 2 e 3.

Transação 1	Transação 2	Transação 3
UPDATE depósitos SET total_litros += 100 WHERE idDep = 1;		
	UPDATE depósitos SET total_litros += 200 WHERE idDep = 1;	
UPDATE depósitos SET total_litros += 500 WHERE idDep = 3;		
	UPDATE depósitos SET total_litros -= 200 WHERE idDep = 1;	
	UPDATE depósitos SET total_litros += 100 WHERE idDep = 2;	
	<b>commit;</b>	
		UPDATE depósitos SET total_litros += 300 WHERE idDep = 2;
		<b>commit;</b>
UPDATE depósitos SET total_litros += 400 WHERE idDep = 2;		
<b>commit;</b>		

1. **[Exame]** Sabendo que as transações são executadas em modo **read committed**, e que as transações não bloqueiam, indique justificando se o escalonamento apresentado é serializável. **[1; 1]**
2. **[EXTRA]** Indique se o mesmo escalonamento é serializável de conflito, justificando adequadamente a sua resposta não esquecendo de indicar as operações conflitantes. **[0;1]**

### Grupo II.3

Para transferir dados sobre os concessionárias de marcas, optou-se por usar ficheiros XML de acordo com a seguinte DTD:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE estações[
<!ELEMENT estações (estação+)>
<!ELEMENT estação ((estrada|localidade),combustível+,rápido?)>
  <!ATTLIST estação nome CDATA #REQUIRED cod ID #IMPLIED>
<!ELEMENT localidade (#PCDATA)>
<!ELEMENT rápido EMPTY >
<!ELEMENT estrada EMPTY >
  <!ATTLIST estrada km CDATA #REQUIRED cod CDATA #REQUIRED>
<!ELEMENT combustível (#PCDATA)>
]>
```

1. **[Exame]** Apresente o texto dum documento XML que use esta DTD, e que contenha a seguinte informação: **[1; 1]**
  - A estação 'Aveiras' é localizada ao Km 44 da estrada AE1 e vende os combustíveis 'Gasóleo', 'Gasolina Simples' e 'Gasolina aditivada', tendo também carregadores rápidos para carros eléctricos;
  - A estação com o nome 'Aeroporto' localiza-se em Lisboa e tem o código LIS-1 vendendo apenas 'Gasóleo'.
2. Apresente expressões XPATH sobre um ficheiro de acordo com o DTD acima que devolvam os resultados das seguintes perguntas:
  - a) **[Exame]** Quais os nomes das estações com carregadores rápidos? **[0,5; 0,75]**
  - b) **[Exame]** Quais os combustíveis vendidos na estrada AE-1? **[0,5; 0,75]**
  - c) **[EXTRA]** Quais as estações que vendem pelo menos 2 tipos de combustível mas que não têm carregador rápido? **[0; 0,75]**
3. **[EXTRA]** O que faz a seguinte pergunta XQuery, quando aplicada a um ficheiro XML de acordo com a DTD apresentada acima? Exemplifique, mostrando o seu resultado no ficheiro XML da pergunta 1 deste grupo. **[0;0,75]**

```
<lista>
{ for $x in distinct-values(//combustível/text())
  return <combustível c="{ $x }">
    { for $y in //estação[combustível = $x]/*[1]
      return <estação>{ $y } </estação>
    }
  </combustível >
}
</lista>
```