

U.C. 21053

Fundamentos de Bases de Dados

2018-2019

## Resolução e Critérios de Correção

### INSTRUÇÕES

- O e-fólio é constituído por 6 alíneas com cotação de 0,5 valores cada. A cotação global é de 3 valores.
- O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 1 a 3 valores.
- Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- O nome do ficheiro deve seguir a normal “eFolioA” + <nº estudante> + <nome estudante com o máximo de 3 palavras>
- Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.
- A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- A legibilidade, a objetividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

A informação da avaliação do estudante está contida no **vetor das cotações**:

Questão: 1.1 1.2.a 1.2.b 1.2.c 2.a 2.b

Cotação: 5, 5 5 5, 5 5 décimas

1) Considere a seguinte tabela de doadores e recetores de sangue com 8 tipos de sangue: A, B, AB e O com fator Rh positivo e negativo. Note que nem todos os tipos de sangue são compatíveis.

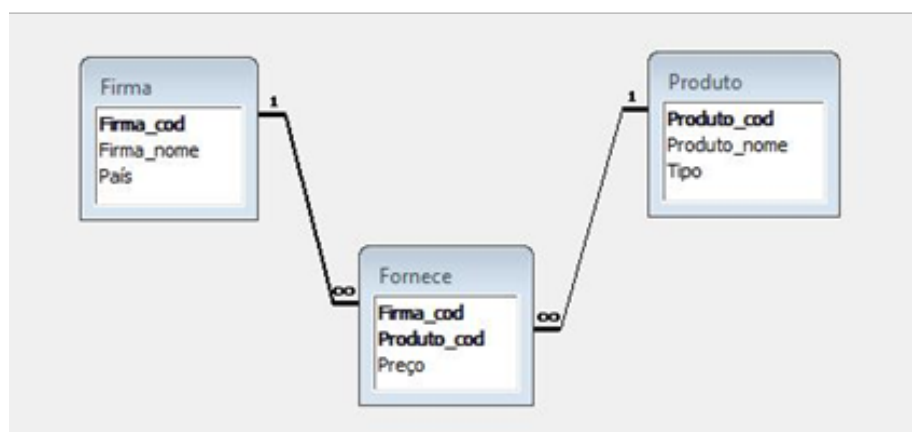
Doutíssima		Doadores							
		O -	O +	B -	B +	A -	A +	AB -	AB +
Receptores	AB +	🩸	🩸	🩸	🩸	🩸	🩸	🩸	🩸
	AB -	🩸		🩸		🩸		🩸	
	A +	🩸	🩸			🩸	🩸		
	A -	🩸				🩸			
	B +	🩸	🩸	🩸	🩸				
	B -	🩸		🩸					
	O +	🩸	🩸						
	O -	🩸							

Arte: Shutterstock

Considere ainda uma base de dados de um conjunto de bancos de sangue com dados e pacientes em que a quantidade de sangue é medida em litros:

- bancoSangue (idBanco-> nome, morada, contactos)
- doador (idDoador -> nome, morada, contactos, tipoSangue, dataNascimento)
- paciente (idPaciente -> nome, morada, contactos, tipoSangue, dataNascimento)
- doadorBanco (idDoador, idBanco -> data, quantidade)
- pacienteBanco (idPaciente, idBanco -> data, quantidade, tipoSangue, motivo)

1.1) Defina chave principal e chave estrangeira. Crie chaves substitutas (“surrogate key”) nas tabelas doadorBanco e pacienteBanco. Represente graficamente a base de dados relacional, com as respectivas tabelas e ligações de chaves estrangeiras. Siga a seguinte regra para a representar: nas ligações de 1:N a tabela com uma única linha é desenhada em cima e da tabela com várias linhas é desenhada por baixo, conforme na figura seguinte:



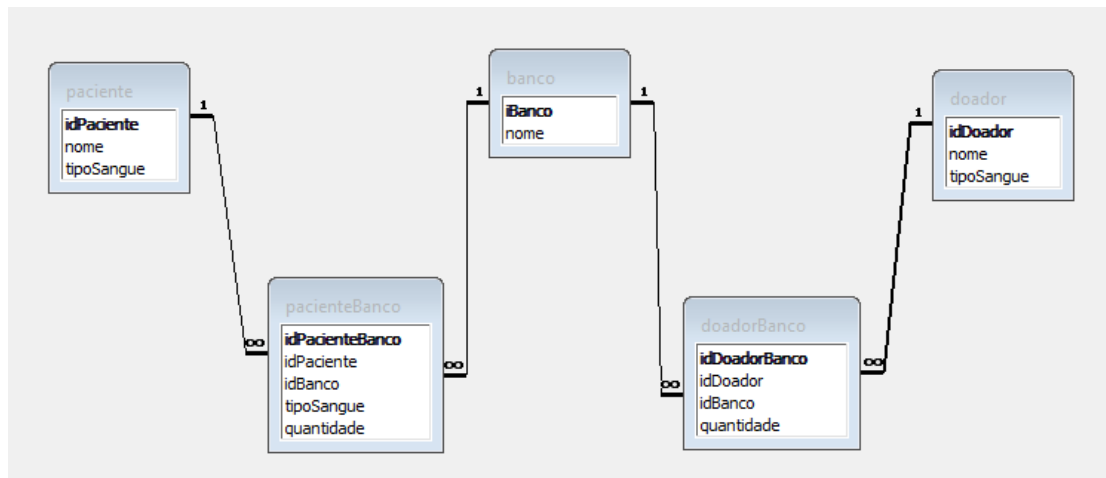
Resposta:

Chave Principal: atributo ou conjunto de atributos que torna única cada linha da tabela.

Chave estrangeira: atributo ou conjunto de atributos que é chave principal numa outra tabela.

Chaves substitutas (“surrogate key”):

- Tabela doadorBanco:
- idDoadorBanco (“surrogate key”) -> idDoador, idBanco, data, quantidade
- Tabela pacienteBanco: de forma idêntica à anterior



Critérios de correção:

- 1 definição de chave principal e estrangeira
- 2 décimas esquema da BD com as regras de representação
- 2 décimas chaves substitutas (‘surrogate key’)
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- visualização do esquema BD desadequada: -50% a -100%

1.2) Exprima em SQL as consultas com a seguinte indentação. Evite, ainda, as cláusulas WITH, TOP e LIMIT:

```
SELECT.....  
FROM .....  
WHERE.....  
AND.....(SELECT.....  
          FROM.....  
          WHERE.....  
GROUP BY .....  
HAVING .....
```

1.2.a) Quais os nomes dos doadores com sangue do tipo ‘doador universal’, O-, e com sangue do tipo ‘recetor universal’, AB+ ?

- utilize uma união
- utilize outra forma

Resposta:

Com união:

```
(SELECT D.idDoador, D.tipoSangue  
FROM doador D  
WHERE D.tipoSangue="O-")  
UNION (SELECT D.idDoador, D.tipoSangue  
FROM doador D  
WHERE D.tipoSangue="AB+");
```

De outra forma:

```
SELECT D.idDoador, D.tipoSangue  
FROM doador D  
WHERE D.tipoSangue="O-"  
OR D.tipoSangue="AB+";
```

Critérios de correção:

- 3 décimas para união
- 2 décimas para outra forma
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

1.2.b) Quais os pacientes que receberam sangue incompatível? Por exemplo, um paciente A+ receber sangue B-. Pode criar uma nova tabela de incompatibilidades.

- utilize sub-consultas com cláusula IN
- utilize sub-consultas com cláusula EXISTS

Resposta:

Fui criada a seguinte tabela de incompatibilidades:

id	sangue1	sangue2
1	AB-	O+
2	AB-	B+
3	AB-	A+
4	A+	B-
5	A+	B+
6	A-	O+
7	A-	B-
8	A-	B+
9	B-	O+
10	O+	AB-
11	B+	AB-
12	A+	AB-
13	B-	A+
14	B+	A+
15	O+	A-
16	B-	A-
17	B+	A-
18	O+	B-
0		

Com cláusula IN:

```
SELECT P.idPaciente
FROM paciente P
WHERE P.idPaciente IN (SELECT PB.idPaciente
                       FROM pacienteBanco PB
                       WHERE PB.tipoSangue IN ( SELECT I.sangue1
                                               FROM incompativel I
                                               WHERE I.sangue2= P.tipoSangue))
```

Com cláusula EXISTS:

```
SELECT P.idPaciente
FROM paciente P
WHERE EXISTS (SELECT *
              FROM pacienteBanco PB
              WHERE P.idPaciente = PB.idPaciente
              AND EXISTS (SELECT *
                          FROM incompativel I
                          WHERE PB.tipoSangue= I.sangue1
                          AND P.tipoSangue= I.sangue2))
```

Critérios de correção:

- 1 décima tabela incompatibilidades
- 2 décimas para IN
- 2 décimas para EXISTS
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

1.2.c) Quais os pacientes que receberam mais litros de sangue que a média dos pacientes?

Resposta:

Primeiro podemos criar uma tabela auxiliar com a soma da quantidade de litros de sangue recebidos por cada paciente:

```
SELECT PB.idPaciente, SUM(PB.quantidade) AS SomaQuant INTO PacienteQuant
FROM pacienteBanco PB
GROUP BY PB.idPaciente
```

A consulta final será:

```
SELECT A.idPaciente, A.SomaQuant
FROM PacienteQuant A
WHERE A.SomaDequant > (SELECT AVG(A2.SomaQuant)
                        FROM PacienteQuant A2 )
```

Crítérios de correção:

- 2 décimas para a tabela auxiliar
- 3 décimas para consulta final
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

2) Exprima em SQL as operações de manipulação da base de dados. Considere as regras definidas nas alíneas anteriores.

2.a) apague todos os doadores, da tabela doadores, que nunca doaram sangue;

```
DELETE
FROM doador
WHERE idDoador NOT IN (SELECT idDoador
                       FROM doadorBanco);
```

Critérios de correção:

- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%

2.b) na tabela doadores, crie um novo atributo com a quantidade de litros doados e atualize o referido atributo.

```
ALTER TABLE doador
ADD qtDoado int;

UPDATE doador D
SET qtDoado = (SELECT SUM(quantidade)
              FROM doadorBanco B
              WHERE B.idDoador = D.idDoador);
```

Critérios de correção:

- 2 décimas para ALTER

- 3 décimas para UPDATE

- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%