

e-Fólio A

U.C. 21053

Fundamentos de Bases de Dados

2020-2021

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 será penalizada.

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

Questão:	1	2.a	2.b	2.c	2.d	3
Cotação:	5,	5	5	5	5,	5

décimas

Considere uma base de dados de um departamento de gestão portuária que inclui as seguintes tabelas:

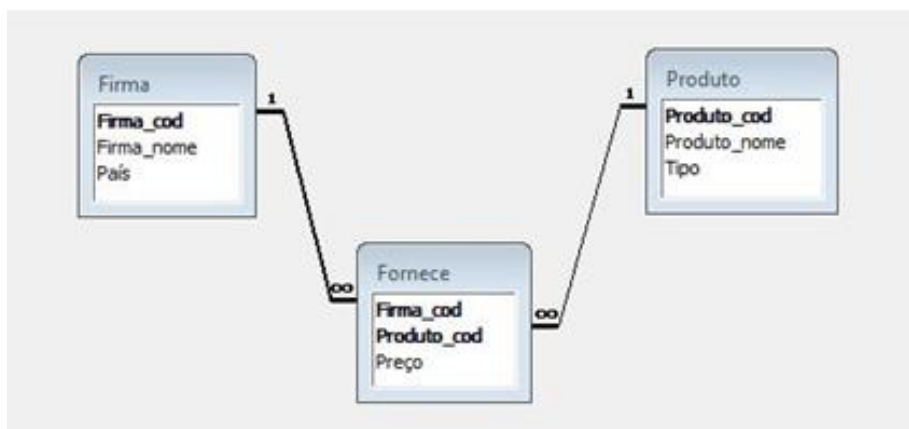
- agências (idAgencia -> nome, contactos, morada)
- contentores (idContentor -> idAgencia, dimensões)
- navios (idNavio -> nome, país, capacidade)
- posicaoNavios (idNavio, data -> longitude, latitude, origem, destino)
- cargaNavios (idNavio, data, IdContentor -> peso)
- portos (idPorto -> nome, cidade, contacto)
- cargaPortos (idPorto, data, idContentor -> peso, origem, destino)

1) Esquema da base de dados (0,5 valores)

i) Defina chave principal e chave estrangeira. O que entende por chave substituta (“surrogate key”) de uma chave composta? Na base de dados substitua todas as chaves com mais de dois atributos por "surrogate keys”.

ii) Represente graficamente a base de dados relacional, com as respetivas tabelas e ligações de chaves estrangeiras.

Nota importante: Para representar o esquema da base de dados siga a seguinte regra: 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:

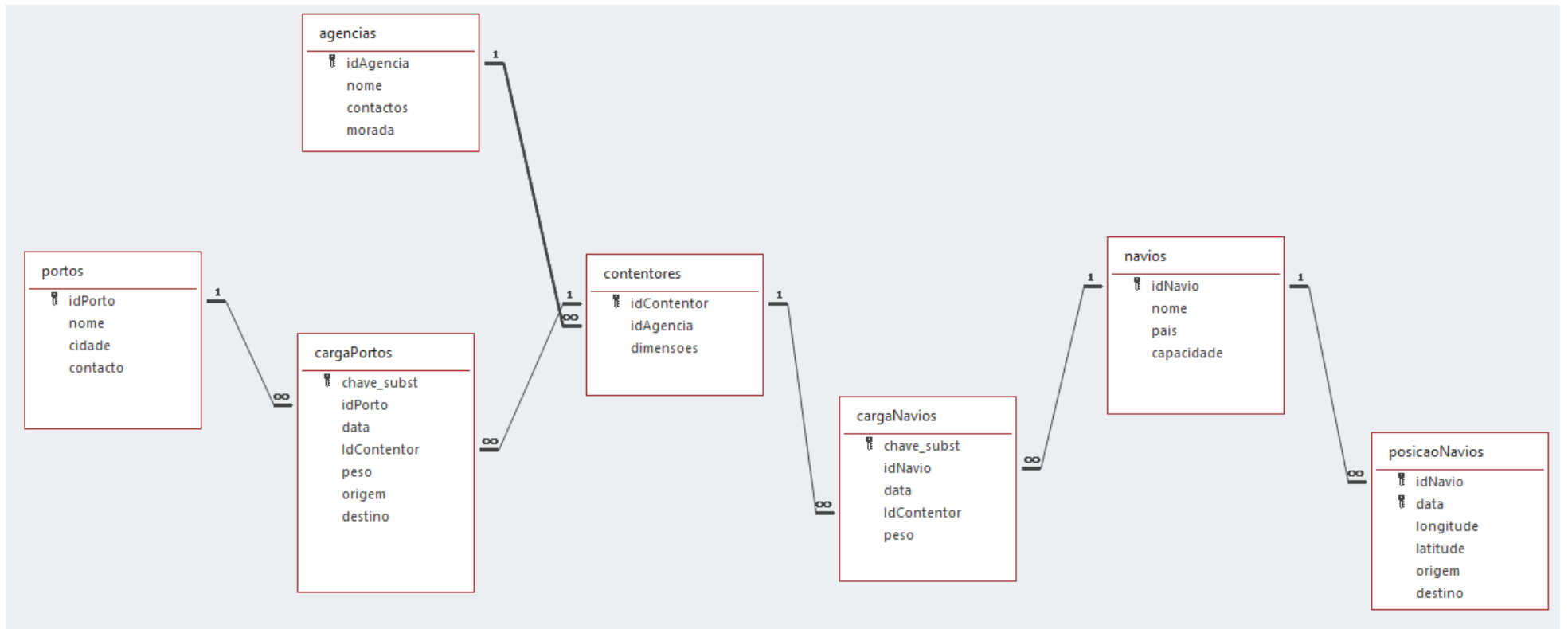
i) Chave Principal: atributo ou conjunto de atributos que torna única cada linha da tabela.

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

Chave Substituta: um único atributo substitui vários atributos de uma chave composta.

- cargaNavios (chaveSubst -> idNavio, data, IdContentor, peso)
- cargaPortos (chaveSubst -> idPorto, data, idContentor, peso, origem, destino)

ii) Representação gráfica da base de dados relacional, com as respetivas tabelas e ligações de chaves estrangeiras.



Critérios de correção:

- 2 definição de chave principal, chave estrangeira e chaves substitutas
- 3 décimas para esquema da BD com as regras de representação
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- visualização do esquema BD desadequada: -50% a -100%

2) Exprima em SQL as consultas usando colunas ALIAS com a indentação indicada em baixo. Evite, ainda, as palavras chave JOIN, WITH, TOP e LIMIT.

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

2.a) (0,5 valores) No dia de hoje, quais os navios que se deslocam de Faro para a cidade do Porto e do Porto para Faro?

- utilize o operador de união
- utilize outra forma

Resposta:

Com união:

```
(SELECT N.nome  
FROM navios N, posicaoNavios PN  
WHERE PN.data = Date()  
AND PN.idNavio = N.idNavio  
AND PN.origem = 'Faro'  
AND PN.destino = 'Porto')  
UNION  
(SELECT N.nome  
FROM navios N, posicaoNavios PN  
WHERE PN.data = Date()  
AND PN.idNavio = N.idNavio  
AND PN.origem = 'Porto'  
AND PN.destino = 'Faro');
```

De outra forma, com uma junção:

```
SELECT N.nome  
FROM navios N, posicaoNavios PN  
WHERE PN.data = Date()  
AND PN.idNavio = N.idNavio  
AND (PN.origem = 'Faro' AND PN.destino = 'Porto')  
OR (PN.origem = 'Porto' AND PN.destino = 'Faro')
```

Critérios de correção:

- 3 décimas para união
- 2 décimas para outra forma
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- desconto 50% se usar as palavras chave JOIN, WITH, TOP e LIMIT.

2.b) (0,5 valores) No dia de hoje, quais os portos com contentores com origem em Faro e destino no Porto?

- utilize subconsultas com cláusula IN
- utilize subconsultas com cláusula EXISTS

Resposta:

Com cláusula IN:

```
SELECT P.nome
FROM portos P
WHERE P.idPorto IN (SELECT CP.idPorto
                    FROM cargaPortos CP
                    WHERE CP.data = Date()
                    AND CP.origem = 'Faro'
                    AND CP.destino = 'Porto');
```

Com cláusula EXISTS:

```
SELECT P.nome
FROM portos P
WHERE EXISTS (SELECT *
             FROM cargaPortos CP
             WHERE P.idPorto = CP.idPorto
             AND CP.data = Date()
             AND CP.origem = 'Faro'
             AND CP.destino = 'Porto');
```

Critérios de correção:

- 2 décimas para IN
- 3 décimas para EXISTS
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- desconto 50% se usar as palavras chave JOIN, WITH, TOP e LIMIT.

2.c) (0,5 valores) No dia de hoje, quais os navios mais carregados (i.e., com mais carga, dado o peso dos contentores) do que a média da carga dos navios?

Resposta:

Primeiro podemos criar uma tabela auxiliar com a soma do peso de todos os contentores de cada navio:

```
SELECT CN.idNavio, SUM(CN.peso) AS pesoCargas INTO navioPesoContentores
FROM cargaNavios CN
WHERE CN.data = Date()
GROUP BY CN.idNavio;
```

Com esta tabela auxiliar já posso correr a consulta final:

```
SELECT A.idNavio, A.pesoCargas
FROM navioPesoContentores A
WHERE A.pesoCargas > (SELECT AVG(B.pesoCargas)
                      FROM navioPesoContentores B)
```

Critérios de correção:

- 2 décimas para a tabela auxiliar
- 3 décimas para consulta final
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- desconto 50% se usar as palavras chave JOIN, WITH, TOP e LIMIT.

2.d) (0,5 valores) Qual a agência com mais contentores?

Resposta:

Utilizando o mesmo método da alínea anterior, criei uma tabela auxiliar da contagem de contentores por agências, de forma a evitar subconsultas no FROM:

```
SELECT C.idAgencia, count(C.idContendor) AS numCont INTO AgenciaContentores
FROM contentores C
GROUP BY C.idAgencia
```

Assim, podemos utilizar a tabela auxiliar para realizar a consulta desejada:

```
SELECT A.idAgencia, A.numCont
FROM AgenciaContentores A
WHERE A.SomaCont = (SELECT MAX(B.numCont)
                    FROM AgenciaContentores B)
```

Critérios de correção:

- 2 décimas para a tabela auxiliar
- 3 décimas para consulta final
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- desconto 50% se usar as palavras chave JOIN, WITH, TOP e LIMIT.

3) (0,5 valores) Operações de manipulação da base de dados: para o dia de hoje, apague todos os contentores em duplicado, i.e. um contentor que aparece duas ou mais vezes num porto ou num navio.

Resposta:

Vamos usar a informação das seguintes tabelas:

- cargaNavios (idNavio, data, IdContentor -> peso)
- cargaPortos (idPorto, data, idContentor -> peso, origem, destino)

```
DELETE
FROM cargaNavios
WHERE idContentor IN (SELECT A.idContentor
                     FROM cargaNavios A, cargaNavios B
                     WHERE A.data = Date()
                     AND B.data=Date()
                     AND A.idContentor = B.idContentor)
```

```
DELETE
FROM cargaPortos
WHERE idContentor IN (SELECT A.idContentor
                     FROM cargaPorto A, cargaPorto B
                     WHERE A.data = Date()
                     AND B.data=Date()
                     AND A.idContentor = B.idContentor)
```

Critérios de correção:

- 2 décimas para DELETE
- 3 décimas para WHERE
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- desconto 50% se usar as palavras chave JOIN, WITH, TOP e LIMIT.