

U.C. 21053

Fundamentos de Bases de Dados

2017-2018

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**:

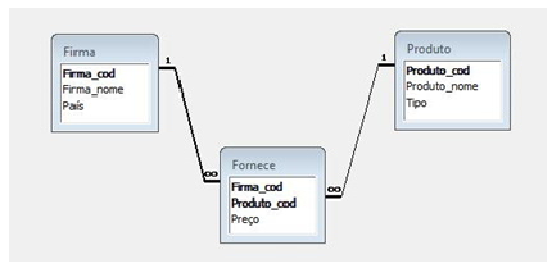
Questão: 1.1 1.2.a 1.2.b 1.2.c 2.a 2.b

Cotação: 5, 5 5 5, 5 5 décimas

1) Considere a base de dados de clientes de uma cadeia de restaurantes:

- clientes (idCliente -> nomeCliente, morada, codPostal, contacto)
- produto (idProduto -> nomeProduto, IdClasse, preco, quantidadeStock)
- classe (idClasse-> nomeClasse)
- empregado (idEmpregado -> nomeEmpregado, contacto, comissao)
- encomenda (idEncomenda -> idCliente, idEmpregado, data)
- linha_encomenda (idEncomenda, idLinha -> idProduto, quantidade)

1.1) Defina chave principal e chave estrangeira. Represente graficamente a base de dados relacional, com as respetivas 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:



Crie ainda, uma chave substituta (“surrogate key”) na tabela linha-encomenda. Qual a nova representação gráfica da base de dados?

Resposta parcial:

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.

Tabela Linha_encomenda:

idLinhaEncomenda (“surrogate key”) ->

idEncomenda, idLinha, idProduto, quantidade

Critérios de correção:

- 1 definição de chave principal e estrangeira
- 2 décimas esquema da BD com as regras de representação
- 2 décimas chave substituta
- 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 clientes que compraram produtos com classe “X” e “Y”.

- utilize uma junção
- utilize uma união

Com a junção:

```
SELECT DISTINCT clientes.nomeCliente
FROM clientes, classe, produto, encomenda, linha_encomenda
WHERE clientes.idCliente=encomenda.idCliente
AND encomenda.idEncomenda=linha_encomenda.idEncomenda
AND produto.idProduto=linha_encomenda.idProduto
AND classe.idClasse=produto.IdClasse
AND (classe.nomeClasse='X' OR classe.nomeClasse='Y')
```

Com a união:

```
SELECT clientes.nomeCliente
FROM clientes, classe, produto, encomenda, linha_encomenda
WHERE clientes.idCliente=encomenda.idCliente
AND encomenda.idEncomenda=linha_encomenda.idEncomenda
AND produto.idProduto=linha_encomenda.idProduto
AND classe.idClasse=produto.IdClasse
AND classe.nomeClasse='X'
UNION
SELECT clientes.nomeCliente
FROM clientes, classe, produto, encomenda, linha_encomenda
WHERE clientes.idCliente=encomenda.idCliente
AND encomenda.idEncomenda=linha_encomenda.idEncomenda
AND produto.idProduto=linha_encomenda.idProduto
AND classe.idClasse=produto.IdClasse
AND classe.nomeClasse='Y'
```

Critérios de correção:

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

1.2.b) Quais os nomes dos clientes que compraram produtos da classe “X”?

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

b) Utilizando IN

```
SELECT nomeCliente
FROM clientes
WHERE idCliente IN
    (SELECT IDcliente
     FROM encomendas
     WHERE IDEncomenda IN
         (SELECT IDEncomenda
          FROM linhas_encomendas
          WHERE idproduto IN
              (SELECT idproduto
               FROM produtos
               WHERE idClasse IN
                   (SELECT idClasse
                    FROM classes
                    WHERE nomeClasse = 'X')))))
```

Utilizando EXISTS

```
SELECT nomeCliente
FROM clientes C
WHERE EXISTS
    (SELECT *
     FROM encomendas E
     WHERE C.idCliente = E.idCliente
     AND EXISTS
         (SELECT *
          FROM linhas_encomendas LE
          WHERE E.idEncomenda = LE.idEncomenda
          AND EXISTS
              (SELECT *
               FROM produtos P
               WHERE LE.idproduto = P.idProduto
               AND EXISTS
                   (SELECT *
                    FROM classes C
                    WHERE P.idClasse = C.idClasse
                    AND nomeClasse = 'X')))))
```

Critérios de correção:

- 2 décimas para IN
- 3 décimas para EXISTS
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%
- para aparecer um nome único usar SELECT DISTINCT em vez de GROUP BY

1.2.c) Cada cliente comprou N produtos, sendo N a soma das quantidades. Quais os clientes que compraram mais produtos que a média dos produtos comprados por cliente?

Resposta:

i) a média dos produtos comprados por cliente

```
(SELECT AVG(M.Med)
FROM (SELECT SUM(LE.quantidade) AS Med
FROM clientes C, encomendas E, linhas_encomendas LE
WHERE C.idCliente = E.idCliente
AND E.idEncomenda = LE.idEncomenda
GROUP BY C.idCliente) AS M)
```

ii) consulta final

```
SELECT c.nomeCliente
FROM clientes C, encomendas E, linhas_encomendas LE
WHERE C.idCliente = E.idCliente
AND E.idEncomenda = LE.idEncomenda
GROUP BY C.idCliente, C.nomeCliente
HAVING SUM(LE.quantidade) > (SELECT AVG(M.Med)
FROM (SELECT SUM(LE.quantidade) AS Med
FROM clientes C, encomendas E, linhas_encomendas
WHERE C.idCliente = E.idCliente
AND E.idEncomenda = LE.idEncomenda
GROUP BY C.idCliente) AS M)
```

Crítérios de correção:

- 2 décimas para i)
- 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 clientes, da tabela cliente, que nunca fizeram compras;

```
DELETE
FROM clientes
WHERE idCliente NOT IN (SELECT idcliente
                        FROM encomendas)
```

Critérios de correção:

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

2.b) na tabela produtos, aumente os preços em 5% aos 10 produtos mais vendidos e diminua os preços em 5% aos 10 produtos menos vendidos.

Aumentar 5% os produtos mais vendidos

```
UPDATE produtos
SET preco = preco * 1.05
WHERE idProduto IN (SELECT TOP 10 p.idproduto
                    FROM produtos p, linhas_encomendas le
                    WHERE p.idproduto = le.idProduto
                    GROUP BY p.idproduto
                    ORDER BY SUM(le.quantidade) DESC)
```

Baixar 5% os produtos menos vendidos

```
UPDATE produtos
SET preco = preco * 0.95
WHERE idProduto IN (SELECT TOP 10 p.idproduto
                    FROM produtos p, linhas_encomendas le
                    WHERE p.idproduto = le.idProduto
                    GROUP BY p.idproduto
                    ORDER BY SUM(le.quantidade) ASC)
```

Nota: $\text{preço} * (1 - \text{taxa}) \neq \text{preço} / (1 + \text{taxa})$, também conhecida como o falso do desconto do vendedor.

Critérios de correção:

- 2 décimas para aumento

- 2 décimas para abaixamento

- 1 décima percentagens

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