

21053 - Fundamentos de Bases de Dados

2014-2015

e-fólio A

Resolução e Critérios de Correção

PARA A RESOLUÇÃO DO E-FÓLIO, ACONSELHA-SE QUE LEIA ATENTAMENTE O SEGUINTE:

- 1) O e-fólio é constituído por 2 perguntas, num total de 10 alíneas, com cotação 0,2 valores cada. A cotação global é de 2 valores.
- 2) 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 2 valores.
- 3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- 4) O nome do ficheiro deve seguir a normal “eFolioA” + <nº estudante> + <nome estudante com o máximo de 3 palavras>
- 5) 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.
- 6) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- 7) A legibilidade, a objectividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

Vetor de cotação:

1	pergunta
1, 2 3, 4 5, 6 7, 8 9, 10	alínea
2, 2 2, 2 2 , 2 2, 2 2, 2	décimas

1) Considere a base de dados de clientes. Os produtos são comprados pelos clientes e é registada a data da compra na transação comercial.

produtos (ID -> nome, preço, cor)

clientes (ID -> nome, morada, cod_postal, país, data_nascimento)

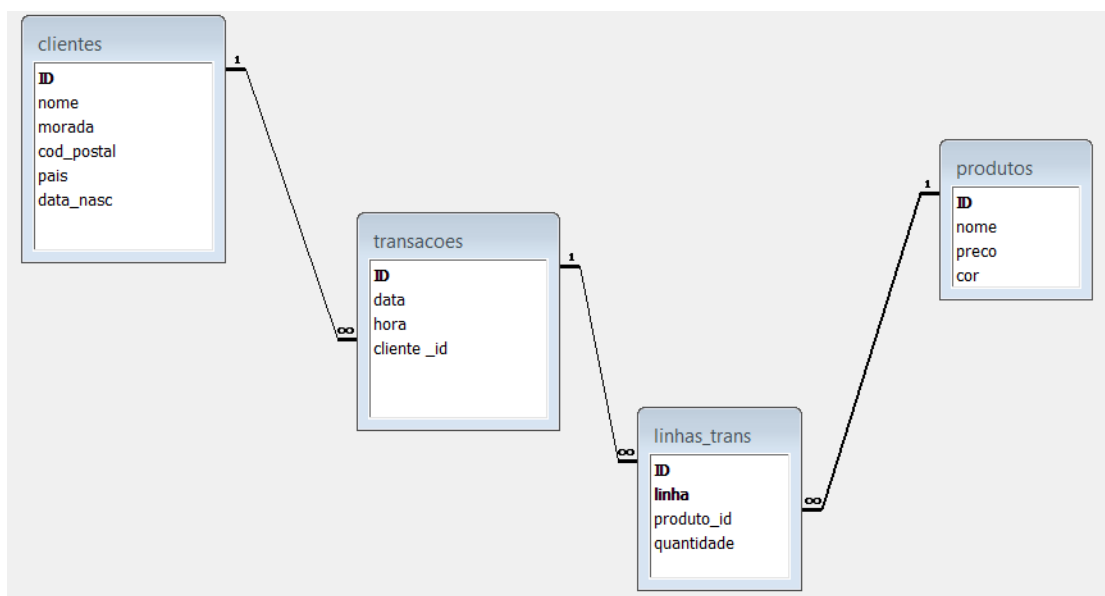
transações (ID -> data, hora, cliente_id)

linhas_transação (ID, linha -> produto_id, 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.

- Chave Principal: atributo ou conjunto de atributos (colunas) que torna única cada linha da tabela;

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



Critérios de correção:

- penalização de (-1 décima) se não cumprir as regras referidas

- comentário: existe uma chave concatenada na tabela linhas_trans constituída por 2 atributos (ID, linha)

Exprima em SQL as seguintes consultas. Evite as cláusulas WITH, TOP e LIMIT:

1.2&3) Quais os nomes dos clientes que compraram produtos vermelhos e verdes?

1.2) utilize uma junção

```
SELECT distinct transacoes.cliente_id
FROM transacoes, produtos, linhas_trans
WHERE produtos.ID = linhas_trans.produto_id
AND transacoes.ID = linhas_trans.ID
AND (((produtos.cor)="verde" OR (produtos.cor)="vermelha"))
```

1.3) utilize UNION

```
SELECT transacoes.cliente_id
FROM transacoes, produtos, linhas_trans
WHERE produtos.ID = linhas_trans.produto_id
AND transacoes.ID = linhas_trans.ID
AND produtos.cor = "verde"
UNION
SELECT transacoes.cliente_id
FROM transacoes, produtos, linhas_trans
WHERE produtos.ID = linhas_trans.produto_id
AND transacoes.ID = linhas_trans.ID
AND produtos.cor = "vermelha"
```

1.4&5) Quais os nomes dos clientes que compraram produtos de cor vermelha?

O resultado da junção será:

```
SELECT DISTINCT clientes.nome, clientes.ID
FROM clientes, transacoes, linhas_trans, produtos
WHERE clientes.ID = transacoes.cliente_id
AND transacoes.ID = linhas_trans.ID
AND linhas_trans.produto_id = produtos.ID
AND produtos.cor = "vermelha"
```

1.4) utilize sub-consultas com cláusula IN

```
SELECT DISTINCT nome, ID
FROM clientes
WHERE ID IN (SELECT cliente_id
             FROM transacoes
             WHERE ID IN (SELECT ID
                        FROM linhas_trans
                        WHERE produto_id IN (SELECT ID
                                           FROM produtos
                                           WHERE cor="vermelha"))))
```

1.5) utilize sub-consultas com cláusula EXISTS

```
SELECT DISTINCT C.nome, C.id
FROM clientes C
WHERE EXISTS (SELECT *
             FROM transacoes T
             WHERE T.cliente_id=C.ID
             AND EXISTS (SELECT *
                        FROM linhas_trans L
                        WHERE L.ID=T.ID
                        AND EXISTS (SELECT *
                                   FROM produtos P
                                   WHERE P.ID=L.produto_id
                                   AND P.cor="vermelha"))))
```

1.6) Qual o número médio de transações dos clientes que fizeram compras no último mês?

Resposta:

Outra forma de enunciar será: qual a média do número de transações dos clientes que fizeram compras no último mês.

```
SELECT AVG(Conta)
FROM (SELECT T.cliente_id, COUNT(*) AS Conta
      FROM transacoes AS T
      WHERE Month(T.data)=10
      AND   Year(T.data)=2014
      GROUP BY T.cliente_id)
```

1.7) Quais os clientes com número transações maiores que a média dos clientes que fizeram compras no último mês?

```
SELECT T.cliente_id, COUNT(*) AS Conta
FROM transacoes AS T
WHERE Month(T.data)=Month(now)-1
AND   Year(T.data) =Year(now)
GROUP BY T.cliente_id
HAVING COUNT(*)> (SELECT AVG(Conta)
                  FROM (SELECT T.cliente_id, COUNT(*) AS Conta
                        FROM transacoes AS T
                        WHERE Month(T.data)=Month(now)-1
                        AND   Year(T.data)=Year(now)
                        GROUP BY T.cliente_id))
```

1.8) Listar o cliente mais velho, i.e., que tem a transação mais antiga na base de dados.

```
SELECT cliente_id
FROM transacoes
WHERE (data+hora) <= (SELECT MIN(data+hora)
                     FROM transacoes)
```

1.9) Crie uma consulta escrita em Português e traduzida para SQL em que utilize as cláusulas some/any, all.

Resposta:

Listar o cliente mais novo, i.e., que tem a transação mais recente na base de dados.

```
SELECT DISTINCT cliente_id
FROM transacoes
WHERE (data+hora) >= all (SELECT (data+hora)
                        FROM transacoes)
```

1.10) Utilize a mesma base de dados da alínea anterior, consulte <http://www.w3schools.com/SQL/default.asp> e exemplifique o Produto Cartesiano.

Resposta:

```
SELECT DISTINCT produtos.ID, linhas_trans.ID  
FROM produtos, linhas_trans
```

Nota: no produto cartesiano a cláusula WHERE não existe

	produtos.ID	linhas_trans.ID
	a	1
	a	13
	a	2
	a	3
	b	1
	b	13
	b	2
	b	3
	c	1
	c	13
	c	2
	c	3
	d	1
	d	13
	d	2
	d	3
	e	1
	e	13
	e	2
	e	3

...

Comentário final e critérios de correção gerais:

- O SQL não precisa ser comentado.
- As palavras-chave do SQL devem ser escritas em maiúsculas e o restante código em minúsculas.
- O SQL bem indentado deve ter as cláusulas na encostada à esquerda e as sub-consultas destacadas à direita, como foi apresentado nas atividades formativas e como se segue:

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

- A falta de indentação é penalizada por cada questão, com 1 a 2 décimas
- As resposta parcialmente corretas tem a cotação de 50%
- A redundância e a falta de legibilidade têm um panalização de 1 décima cada
- Erro comum: WHERE trabalha sobre as linhas (tuplos)
HAVING trabalha sobre os valores agregados dos grupos
- Como foi referido é de evitar as cláusulas: WITH, TOP e LIMIT