

U.C. 21053

Fundamentos de Bases de Dados

2019-2020

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

Cotação: 5, 5 5 5, 5 5 décimas

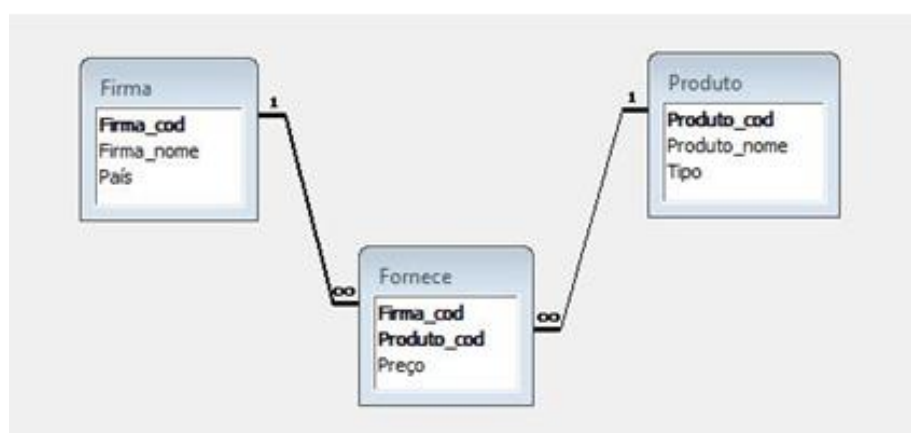
1) Considere uma base de dados relativa ao sistema de ensino básico e secundário (B+S) com os seguintes atributos:

- Ano_curricular = {1, 2, 3, ... 12}
- Nível_ensino = {básico, secundário}
- Natureza = {ensino público, ensino privado}
- Modalidade = {ensino regular básico, ensino profissional básico, ensino regular secundário, ensino profissional secundário, outros básico}
- NUTS = {Área Metropolitana Lisboa, Área Metropolitana Porto, Centro, Região Autónoma Açores, Região Autónoma Madeira}
- Transições = {transita, retido, abandona, reingressa}

E com as seguintes tabelas:

- estudantes (idEstudante -> nome, morada, contacto, encarregado_educação)
- histórico (idFacto -> idEstudante, ano_letivo, ano_curricular, nuts, idModalidade, natureza, transição)
- modalidades (idModalidade -> idNivel)
- níveis (idNivel)

1.1) Defina chave principal e chave estrangeira. A tabela Histórico tem uma chave substituta (“surrogate key”) de uma chave composta. Qual a chave composta equivalente? 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:



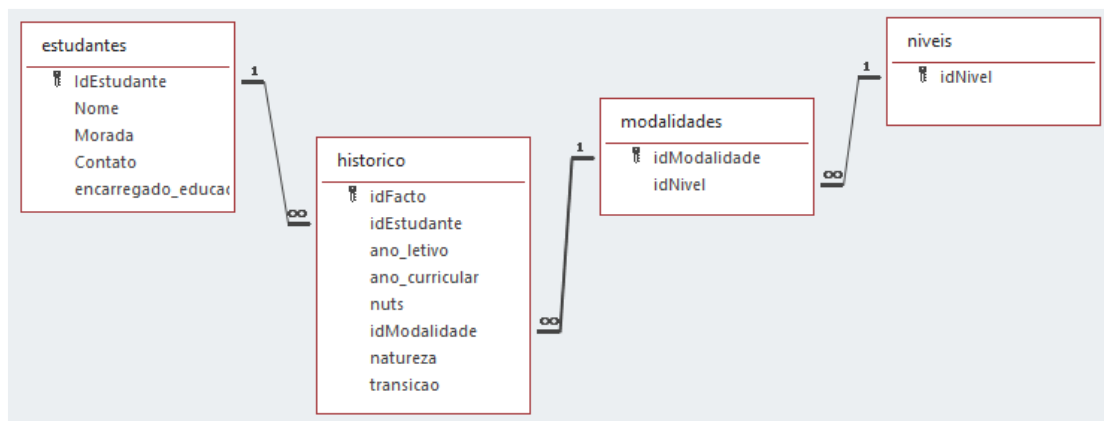
Resposta:

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.

Chave Composta Histórico: idEstudante, ano_letivo

Chaves substitutas (“surrogate key”) Histórico: idFacto



Critérios de correção:

- 1 definição de chave principal e estrangeira
- 2 décimas chave composta
- 2 décimas 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%

1.2) Exprima em SQL as consultas usando ALIAS com a seguinte identificação. Evite, ainda, as cláusulas WITH, TOP e LIMIT.

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

1.2.a) No ano letivo 2014-2015 e no ano 2015-2016, quais os estudantes (idEstudante) que reingressaram no 12º ano, na NUTS da RA Açores e na RA Madeira?

- utilize o operador de união
- utilize outra forma

Resposta:

Com união:

```
(SELECT H.idEstudante
FROM historico H
WHERE H.ano_curricular = 12
AND H.transicao = 'reingressa'
AND (H.nuts = 'RAA' OR H.nuts = 'RAM')
AND H.ano_letivo = '2015-2016')
UNION
(SELECT H.idEstudante
FROM historico H
WHERE H.ano_curricular = 12
AND H.transicao = 'reingressa'
AND (H.nuts = 'RAA' OR H.nuts = 'RAM')
AND H.ano_letivo = '2014-2015')
```

De outra forma:

```
SELECT DISTINCT H.idEstudante
FROM historico H
WHERE H.ano_curricular = 12
AND (H.ano_letivo = '2015-2016' OR H.ano_letivo = '2014-2015')
AND (H.nuts = 'RAA' OR H.nuts = 'RAM')
AND H.transicao = 'reingressa'
```

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 estudantes do ano letivo 2015-2016 e no ano curricular N, que no ano letivo anterior (2014-2015) se encontravam num ano inferior a N-1. Exemplo:

(est12345, 2014-2015, 7)

(est12345, 2015-2016, 10)

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

Resposta:

Com cláusula IN:

```
SELECT H.idEstudante
FROM historico H
WHERE H.ano_letivo = '2015-2016'
AND H.idEstudante IN (SELECT H2.idEstudante
                      FROM historico H2
                      WHERE H2.ano_letivo = '2014-2015'
                      AND H2.ano_curricular < H.ano_curricular - 1)
```

Com cláusula EXISTS:

```
SELECT H.idEstudante
FROM historico H
WHERE H.ano_letivo = '2015-2016'
AND EXISTS (SELECT *
            FROM historico H2
            WHERE H2.idEstudante = H.idEstudante
            AND H2.ano_letivo = '2014-2015'
            AND H2.ano_curricular < H.ano_curricular - 1)
```

Critérios de correção:

- 2 décimas para IN

- 3 décimas para EXISTS

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

1.2.c) Quais os estudantes com mais retenções do que a média das retenções de todos os estudantes?

Resposta:

Primeiro podemos criar uma tabela auxiliar com o número de retenções por estudante

```
SELECT H.idEstudante, COUNT(H.idFacto) AS NumRetencoes
FROM historico H
WHERE H.transicao = 'retido'
GROUP BY H.idEstudante
```

A consulta final será:

```
SELECT R.idEstudante, R.NumRetencoes
FROM EstudanteRetencoes R
WHERE R.NumRetencoes > (SELECT AVG(R2.NumRetencoes)
                        FROM EstudanteRetencoes R2)
```

OU

A consulta final será:

```
SELECT H.idEstudante
FROM historico H
WHERE H.transicao = 'retido'
GROUP BY H.idEstudante
HAVING COUNT(*) > (SELECT AVG(R2.NumRetencoes)
                  FROM EstudanteRetencoes R2)
```

Critérios de correção:

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

2) Para a mesma base de dados, exprima em SQL as operações de manipulação da base de dados. Considere as regras definidas nas alíneas anteriores.

2.a) apague todos os estudantes em duplicado, i.e. um estudante que aparece duas ou mais vezes no mesmo ano letivo;

Resposta:

```
DELETE FROM historico
WHERE idFacto in (SELECT A.idFacto
                  FROM historico A, historico B
                  WHERE A.ano_letivo=B.ano_letivo
                  AND A.idEstudante=B.idEstudante
                  AND A.idFacto > B.IdFacto)
```

Critérios de correção:

- 2 décimas para DELETE

- 3 décimas para WHERE

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

2.b) altere o ano curricular de N para N+6 dos estudantes com modalidade 'EFA' (educação e formação de adultos);

Resposta:

```
UPDATE historico  
SET ano_curricular = ano_curricular + 6  
WHERE idModalidade = 'EFA'
```

Crerios de correção:

- 3 drcimas para UPDATE table SET
- 2 drcimas para WHERE
- erros, omissões, redundâncias ou indentação desadequada: -20% a -100%