

Sistemas de Informação e Bases de Dados

Faculdade de Engenharia da Universidade do Porto
6 de Janeiro 2017

Duração: 2h

Com consulta

1. Uma empresa de mudanças necessita de uma base de dados onde possa guardar informação sobre os seus clientes e serviços a efectuar. Após um estudo detalhado, chegou-se à seguinte lista de requisitos:

- Uma mudança acontece sempre entre duas casas e num determinado dia. Cada mudança deve ter sempre um número único.
- Cada casa tem uma morada composta por uma rua, número de porta, andar (opcional) e código postal. As casas devem ter sempre um id único.
- Uma mudança é sempre feita a pedido de um cliente. Um cliente é identificado pelo seu nif.
- Para cada cliente deve ainda ser guardado o seu nome e número de telefone.
- Uma mudança é sempre efectuada por vários empregados. Cada empregado deve ter um número único e um nome.
- Numa mudança podem ser utilizados vários camiões. Cada camião deve ter uma matrícula (única) que serve como identificador.
- Deve ainda ser guardada a data da última revisão de cada um dos camiões. Um camião pode nunca ter feito uma revisão.

Assuma o que achar mais correcto no que respeita a dados que ache que faltam na lista de requisitos (chaves primárias em falta, campos obrigatórios, ...).

3 $\frac{1}{2}$ val.

(a) Crie um modelo **entidade-associação** que satisfaça os requisitos descritos. Escolha o nome mais apropriado para cada entidade, atributo e associação que ache necessários.

2 $\frac{1}{2}$ val.

(b) Transforme o modelo que criou num **modelo relacional**. Não se esqueça de indicar quais as chaves primárias/únicas, chaves estrangeiras, tipos de dados e colunas não nulas.

2. Considere uma base de dados que contém informação sobre corridas de atletismo, participantes e os seus resultados:

```
1
2 CREATE TABLE atleta (
3     id SERIAL PRIMARY KEY,
4     nome VARCHAR,
5     idade INTEGER
6 );
7
8 CREATE TABLE corrida (
9     num SERIAL PRIMARY KEY,
10    data DATE NOT NULL,
11    cidade VARCHAR NOT NULL,
12    distancia INTEGER NOT NULL      -- A distancia (ex: 10,000 para um corrida de 10km)
13 );
14
15 CREATE TABLE resultado (
16    num SERIAL PRIMARY KEY,
17    id INTEGER REFERENCES atleta,    -- O atleta que obteve este resultado
18    num INTEGER REFERENCES corrida, -- Em que corrida
19    tempo INTEGER NOT NULL,        -- O tempo do atleta na corrida em segundos
20    posicao INTEGER NOT NULL,      -- A posicao do atleta na corrida
21    UNIQUE (id, num),
22    UNIQUE (num, posicao),
23 );
```

Listing 1: Base de Dados

Responda às seguintes perguntas usando SQL retirando repetidos sempre que achar necessário:

- 1 val. (a) Que atletas já ganharam pelo menos uma corrida? (id, nome)
- 1 val. (b) Que atletas participaram em corridas na cidade do Porto? (id, nome)
- 1 val. (c) Quantas corridas houve em cada cidade? (cidade, quantidade)
- 1 val. (d) Que atletas nunca ganharam uma corrida? (id, nome)
- 1 val. (e) Qual o melhor tempo (em segundos) de sempre de um atleta numa corrida de 10km? (tempo)
- 1 val. (f) Que atleta, ou atletas, obtiveram esse tempo? (id, nome)

3. Responda às seguintes questões sobre HTML, CSS e PHP.

Nas perguntas de resposta múltipla, cada resposta errada desconta $\frac{1}{3}$ da cotação mas pode escolher mais do que uma opção.

1 val.

- (a) Em que situações se deve usar o elemento HTML *aside*?
- A. Quando contém elementos de navegação.
 - B. Quando o elemento vai ficar num dos lados da página.
 - C. Quando não contém um *header*.
 - D. Quando o elemento não faz parte do conteúdo principal da página.

1 val.

- (b) Que elementos selecciona o seguinte selector CSS?

```
| header h1, h2 { ... }
```

- A. Elementos *h1* dentro de um *header* e qualquer elemento *h2*.
- B. Elementos *h1* e *h2* dentro de um *header*.
- C. Elementos *h1*, *h2* e *header*.
- D. Nunca selecciona nenhum elemento.

1 val.

- (c) Considere o seguinte código PHP que implementa uma função que retorna todos os produtos de uma certa marca:

```
1 function getProductsFromMake($make) {
2     global $conn;
3     $stmt = $conn->prepare("SELECT * FROM product ...");
4     $stmt->execute(array($make));
5     return $stmt->fetchAll();
6 }
```

Complete a linha 3 do código:

- A. `prepare("SELECT * FROM product WHERE make = '?'");`
- B. `prepare("SELECT * FROM product WHERE make = ?");`
- C. `prepare("SELECT * FROM product WHERE make = '". $make . "'");`
- D. `prepare("SELECT * FROM product WHERE make = '$make'");`

3 val.

- (d) Escreva o código de uma função PHP chamada *mostExpensive* que retorne o nome do produto mais caro de uma determinada marca passada como pãmetro. Parta do princípio que a tabela *product* contém as colunas *id*, *name*, *make* e *price* e que só existe um produto mais caro. Reutilize a função da alínea anterior.

2 val.

- (e) Reutilizado a função da alínea *c*), escreva o código de uma página PHP que receba uma marca enviada a partir do seguinte formulário:

```
1 <form action="expensive.php" method="post">
2   <input type="text" name="make">
3   <input type="submit">
4 </form>
```

E que imprima os produtos dessa marca usando HTML, com o seguinte formato:

```
1 <h2>X</h2>
2 <ul>
3   <li>Y: Z</li>
4   <li>Y: Z</li>
5   <li>...</li>
6 </ul>
```

Em que X seja substituído pela marca passada através do formulário, Y pelo nome do produto e Z pelo preço desse produto.