

Disciplina de Base de Dados

Enunciado do Projeto – Parte 2

Admita que o seu grupo foi escolhido para desenvolver uma nova aplicação para a gestão de leilões de áreas de concessão de recursos marítimos, sobre uma Base de Dados já existente.

Para além do seu grupo, existem outros a desenvolver outras aplicações sobre as mesmas tabelas. Deverá portanto desenvolver as interrogações SQL para que a aplicação continue a funcionar independentemente de haver alterações às tabelas, nomeadamente após a inserção de novos campos.

1- Trabalho a desenvolver

1. Analise cuidadosamente o script SQL apresentado no Apêndice I e apresente o modelo Entidade-Associação que melhor traduz os conceitos materializados nas respetivas tabelas. Tenha em conta que as tabelas correspondem, com algumas diferenças, a uma concretização simplificada de alguns conceitos da Parte I do projeto. Existem duas novas tabelas, *concorrente* e *lance*, servindo a primeira para registar os participantes nos leilões e a segunda para registar os lances dos mesmos.
2. Escreva os scripts PHP que permitam, a partir de um browser, a participação nos leilões, lendo e escrevendo a informação necessária na Base de Dados. Concretamente, os scripts a desenvolver deverão permitir a um licitante:
 - Autenticar-se através do username (nif) e password (pin) existentes na tabela *pessoa*.
 - Visualizar os leilões em curso ou a iniciar e as respetivas concessões ou áreas de exploração dos recursos marítimos.
 - Inscrever-se, depois de autenticado, num ou mais leilões. É possível a inscrição em leilões até ao momento em que este termina.
 - Fazer lances nos leilões em que está inscrito.
 - Listar o estado dos leilões em curso, apresentando os lances com maiores valores em cada leilão em que está inscrito e o tempo em falta para esse leilão fechar.

No Apêndice II são apresentados scripts PHP que implementam os 2 primeiros itens. Se achar útil, use-os como base para desenvolver os restantes scripts.

Considere que as tabelas já estão preenchidas com todas as informações necessárias, faltando apenas a inscrição dos participantes nos leilões e as respetivas oferta de lances. Consequentemente, assuma que a partir do programa cliente (em PHP) apenas se podem colocar interrogações dos seguintes tipos¹:

- SELECT sobre todas as tabelas indicadas.
 - INSERT nas tabelas *lance* e *concorrente*.
3. Escreva num ficheiro com um script em SQL as interrogações abaixo indicadas:
 1. Quais os participantes inscritos em leilões mas sem lances até à data?
 2. Qual o nome das pessoas coletivas com exatamente duas inscrições em leilões?
 3. Qual o leilão com o maior rácio (valor do melhor lance)/(valor base)?
 4. Quais as pessoas coletivas com o mesmo capital social?
 4. Adicione um trigger que assegure que:

¹Esta limitação é apenas para esta alínea. Na resposta a outras alíneas poderá ter que criar/apagar índices, triggers e tabelas.

- O valor mínimo de um lance num leilão seja o valor base desse leilão.
 - O valor dos lances num determinado leilão seja sempre crescente.
5. Considere agora que as inscrições dos participantes nos leilões iniciados no mesmo dia são feitas de forma atômica, ou seja, o participante seleciona primeiro os leilões em que pretende participar, entre os que começam num determinado dia, e só depois é que o vosso script os deve registar (de forma atômica) na tabela *concorrente*. Escreva uma versão alternativa dos scripts PHP com esta nova funcionalidade, indicando explicitamente as alterações efetuadas.
 6. Suponha que pretende melhorar o desempenho na execução das interrogações 3.3 e 3.4. Escreva em SQL as instruções de criação dos índices que propõe para otimizar o desempenho. Valide a sua solução usando a instrução EXPLAIN do MySQL e comprove indicando os tempos medidos, com e sem índices.
 7. Crie na base de dados o esquema de uma estrela com informação de lances em leilões, em que a tabela de factos regista o lance mais elevado para cada leilão, tendo como dimensões o tempo (por ano, mês e dia) e a localização da empresa leiloeira (por região NUT-III e concelho). Escreva as instruções SQL para carregar a(s) tabela(s) que criou com os dados existentes nas outras tabelas.
 8. Considerando o esquema da estrela, escreva a interrogação em MySQL para obter a receita dos leilões em cada concelho, com rollup por ano e mês do biénio 2012-2013.

2- Relatório

O projeto será avaliado a partir do relatório entregue, que deve conter as respostas a todas as questões da Secção 1. Na tabela seguinte indica-se a valorização de cada parte do trabalho a desenvolver.

Trabalho a desenvolver	Cotação (valores)
1- Modelo Entidade Associação	0,5
2- Script PHP e SQL	2,0
3- Interrogações	1,0
4- Trigger	0,5
5- Transações	2,0
6- Índices	1,5
7- Estrela e scripts de carregamento	1,5
8- Rollup	0,5
9- Relatório	0,5

O relatório deverá começar com uma folha de rosto com a indicação “Projeto de Bases de Dados, Parte 2”, o nome e número dos alunos, o número do grupo e o turno a que o grupo pertence. O relatório terá que ser entregue em duas versões:

- a. Versão digital, em formato PDF, a entregar via projeto no Fénix (veja as datas no respetivo projeto). Todo o código PHP e SQL desenvolvido deverá estar presente em anexo no relatório e em ficheiros prontos para serem executados na discussão.
- b. Versão em papel, a entregar na portaria do IST Tagus Park ou na secretaria do DEI na Alameda dentro do mesmo prazo. Não encaderne, apenas agrafe as folhas.

Anexo I - Script de criação da Base de Dados dos Leilões

```
-- define como terminador de comandos o caracter ;
DELIMITER ;

-- desativa a verificaçõ das chaves estrangeiras
SET foreign_key_checks = 0 ;

-- tabela das pessoas
DROP TABLE IF EXISTS pessoa ;
CREATE TABLE pessoa(
    nif                INT ,                                -- ID e username das pessoas pessoa
    nome              VARCHAR(80) NOT NULL,
    pin              INT NOT NULL,                          -- codigo de acesso desta pessoa (pin ou passwd)
    PRIMARY KEY (nif));

-- tabela das pessoas coletivas
DROP TABLE IF EXISTS pessoac;
CREATE TABLE pessoac(
    nif                INT,
    capitalsocial     INT NOT NULL,
    PRIMARY KEY (nif),
    FOREIGN KEY (nif) REFERENCES pessoa(nif));

-- tabela das leiloeiras
DROP TABLE IF EXISTS leiloeira;
CREATE TABLE leiloeira(
    nif                INT,
    nralvara          INT NOT NULL,
    concelho          VARCHAR (80) NULL,
    regiao            VARCHAR (80) NULL,
    PRIMARY KEY (nif),
    FOREIGN KEY (nif) REFERENCES pessoac(nif));

-- tabela dos leilões
DROP TABLE IF EXISTS leilao;
CREATE TABLE leilao(
    dia                DATE,                                -- dia em que o leilão se realiza
    nrleilaonodia     INT ,                                -- Contador diários de leilões: vale 1 para o 1º leilão do dia, 2 para o 2º leilão do dia etc.
    nif                INT ,
    nome              VARCHAR (80) NULL,                   -- Nome da área de concessão do recurso a ser leiloadada neste leilão
    valorbase         INT NOT NULL,                       -- Valor de base do leilão.
    tipo              BOOLEAN NOT NULL,                   -- Tipo de leilão .
                                                         -- TRUE = leilao de área de concessão de recursos,
                                                         -- FALSE = leilao de area de exploracao de infraestrutura

    PRIMARY KEY (nif,dia,nrleilaonodia),
    FOREIGN KEY (nif) REFERENCES leiloeira(nif));

-- tabela dos leilões de áreas de concessões de Recursos
DROP TABLE IF EXISTS leilaor;
CREATE TABLE leilaor(
    dia                DATE NOT NULL,
    nrleilaonodia     INT NOT NULL,
    nif                INT NOT NULL,
    nrdias            INT NOT NULL,                       -- nr de dias em que leilão está aberto
    lid               INT AUTO_INCREMENT,
    FOREIGN KEY (nif,dia,nrleilaonodia) REFERENCES leilao(nif,dia,nrleilaonodia),
    PRIMARY KEY (nif,dia,nrleilaonodia),
    UNIQUE KEY(lid));

-- tabela com os concorrentes registados aos leilões de Recursos
DROP TABLE IF EXISTS concorrente;
CREATE TABLE concorrente(
    pessoa            INT NOT NULL,
    leilao            INT NOT NULL,
    PRIMARY KEY (pessoa,leilao),
    FOREIGN KEY (pessoa) REFERENCES pessoa(nif),
    FOREIGN KEY (leilao) REFERENCES leilaor(lid));

-- tabela com os lances dos concorrentes aos leilões de Recursos
DROP TABLE IF EXISTS lance;
CREATE TABLE lance(
    pessoa            INT NOT NULL,
    leilao            INT NOT NULL,
    valor             INT NOT NULL,                       -- valor de cada lance
    PRIMARY KEY (pessoa,leilao,valor),
    FOREIGN KEY (pessoa,leilao) REFERENCES concorrente(pessoa,leilao));

-- ativa a verificaçõ das chaves estrangeiras
SET foreign_key_checks = 1 ;
```

Anexo II - Scripts de acesso e listagem dos leilões

Comece por chamar o login.htm no browser (<http://web.ist.utl.pt/~istXXX/login.htm>). Este form chama o registo.php, que por sua vez chama o leilão.php.

login.htm

```
<form action="registo.php" method="post">
<h2>Sistema de leilões de Recursos Marítimos</h2>
  <p>Introduza o seu nif: <input type="text" name="username" /></p>
  <p>Introduza o seu pin: <input type="text" name="pin" /></p>
  <p><input type="submit" /></p>
</form>
```

Registo.php

```
<html>
<body>
<?php

    // inicia sessão para passar variaveis entre ficheiros php
    session_start();

    // Função para limpar os dados de entrada
    function test_input($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    // Carregamento das variáveis username e pin do form HTML através do metodo POST;
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $username = test_input($_POST["username"]);
        $pin = test_input($_POST["pin"]);
    }
    echo("<p>Valida Pin da Pessoa $username</p>\n");

    // Variáveis de conexão à BD
    $host="db.ist.utl.pt";    // o MySQL esta disponivel nesta maquina
    $user="XXXXX";           // -> substituir pelo nome de utilizador
    $password="XXXXX";       // -> substituir pela password dada pelo mysql_reset
    $dbname = $user;         // a BD tem nome identico ao utilizador
    echo("<p>Projeto Base de Dados Parte II</p>\n");
    $connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING));
    echo("<p>Connected to MySQL database $dbname on $host as user $user</p>\n");

    // obtem o pin da tabela pessoa
    $sql = "SELECT * FROM pessoa WHERE nif=" . $username;
    $result = $connection->query($sql);
    if (!$result) {
        echo("<p> Erro na Query: ($sql)<p>");
        exit();
    }

    foreach($result as $row){
        $safepin = $row["pin"];
        $nif = $row["nif"];
    }
    if ($safepin != $pin ) {
        echo "<p>Pin Invalido! Exit!</p>\n";
        $connection = null;
        exit;
    }
    echo "<p>Pin Valido! </p>\n";

    // passa variaveis para a sessao;
    $_SESSION['username'] = $username;
    $_SESSION['nif'] = $nif;

    // Apresenta os leilões
    $sql = "SELECT * FROM leilao";
    $result = $connection->query($sql);
    echo("<table border=\"1\">\n");
    echo("<tr><td>ID</td><td>nif</td><td>diahora</td><td>NrDoDia</td><td>nome</td><td>tipo</td><td>valo
rbase</td></tr>\n");
    $idleilao = 0;
```

```

        foreach($result as $row){
            $idleilao = $idleilao +1;
            echo("<tr><td>");
            echo($idleilao); echo("</td><td>");
            echo($row["nif"]); echo("</td><td>");
            echo($row["diahora"]); echo("</td><td>");
            echo($row["nrleilaonodia"]); echo("</td><td>");
            echo($row["nome"]); echo("</td><td>");
            echo($row["tipo"]); echo("</td><td>");
            echo($row["valorbase"]); echo("</td><td>");
            $leilao[$idleilao]= array($row["nif"],$row["diahora"],$row["nrleilaonodia"]);
        }
        echo("</table>\n");
    ?>

<form action="leilao.php" method="post">
<h2>Escolha o ID do leilão que pretende concorrer</h2>
    <p>ID : <input type="text" name="lid" /></p>
    <p><input type="submit" /></p>
</form>

</body>
</html>

```

Leilao.php

```

<html>
<body>
<?php
    // inicia sessão para passar variaveis entre ficheiros php
    session_start();
    $username = $_SESSION['username'];
    $nif = $_SESSION['nif'];

    // Função para limpar os dados de entrada
    function test_input($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    // Carregamento das variáveis username e pin do form HTML através do metodo POST;
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $lid = test_input($_POST["lid"]);
    }

    // Conexão à BD
    $host="db.ist.utl.pt"; // o MySQL esta disponivel nesta maquina
    $user="XXXXXX"; // -> substituir pelo nome de utilizador
    $password="XXXXXX"; // -> substituir pela password dada pelo mysql_reset
    $dbname = $user; // a BD tem nome identico ao utilizador
    $connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,
array(PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING));
    echo("<p>Connected to MySQL database $dbname on $host as user $user</p>\n");

    //registra a pessoa no leilão. Exemplificativo apenas.....

    $sql = "INSERT INTO concorrente (pessoa,leilao) VALUES ($nif,$lid)";
    $result = $connection->query($sql);
    if (!$result) {
        echo("<p> Pessoa nao registada: Erro na Query:($sql) <p>");
        exit();
    }

    echo("<p> Pessoa ($username), nif ($nif) Registada no leilao ($lid)</p>\n");

    // to be continued...

    //termina a sessão
    session_destroy();
?>
</body>
</html>

```