



TÉCNICO
LISBOA

Projecto de Bases de Dados Parte 3

LETI 2016-2017 - 1º Semestre

Grupo nº 50 - Turno BD8179L07 (5ª feira às 8h30)

Pedro Silva - 77929 (18 horas – tempo estimado)

Duarte Silva - 79762 (18 horas – tempo estimado)

Ana Rita Rocha - 79779 (18 horas – tempo estimado)

SQL

a)

```
SELECT distinct e.morada, e.codigo
FROM espaco e, aluga a, posto p
WHERE e.codigo = p.codigo_espaco and e.morada = p.morada and (e.morada,
e.codigo) NOT IN (SELECT a.morada, a.codigo FROM aluga a) and
(p.morada,p.codigo) NOT IN (SELECT a.morada, a.codigo FROM aluga a);
```

b)

```
SELECT *
FROM (SELECT morada, COUNT(numero) as reservation FROM aluga
GROUP BY morada) reservas
HAVING (reservation > (SELECT AVG(reservation) FROM reservas));
```

c)

```
SELECT A.nif, U.nome
FROM fiscaliza F LEFT JOIN arrenda A ON F.morada = A.morada LEFT JOIN user U
ON A.nif = U.nif
GROUP BY F.morada, A.nif, U.nome
HAVING count(distinct F.id)=1;
```

d)

```
SELECT morada,codigo_espaco,tarifa,datediff(data_inicio,data_fim)
FROM(
    (SELECT distinct p.morada, p.codigo_espaco
    FROM posto p, aluga a, paga pa
    WHERE p.codigo_espaco = a.codigo and p.morada = a.morada and
    pa.numero=a.numero) UNION
    (SELECT distinct e.morada, e.codigo
    FROM espaco e, aluga a, paga pa
    WHERE e.codigo = a.codigo and e.morada = a.morada and
    pa.numero=a.numero))
WHERE(data_fim like '2016%') and (data_inicio like '2016%');
```

e)

```
SELECT e.morada, e.codigo
FROM posto p, espaco e, aluga a, estado es
WHERE e.morada = p.morada and e.codigo=p.codigo_espaco and es.estado = "aceite"
WHERE NOT EXISTS (SELECT distinct e.morada, e.codigo
                  FROM espaco e, aluga a, posto p
                  WHERE e.codigo = p.codigo_espaco and e.morada =
p.morada and (e.morada,e.codigo) NOT IN (SELECT a.morada, a.codigo
                  FROM aluga a) and (p.morada,p.codigo)
                  NOT IN(SELECT a.morada, a.codigo
                  FROM aluga a));
```

Restrições de Integridade

a)

```
DELIMITER $$
CREATE TRIGGER verify_dates BEFORE INSERT ON oferta
FOR EACH ROW
BEGIN
    DECLARE a INTEGER;
    SET a= (SELECT count(*) FROM oferta WHERE oferta.morada=new.morada
AND oferta.codigo=new.codigo AND (oferta.data_inicio<= new.data_inicio OR
oferta.data_fim>1=new.data_fim));
    IF a > 0 THEN
        CALL raise_application_error(3001, 'oferta existente');
    END IF;
END $$
delimiter ;
```

b)

```
DROP TRIGGER verify_timestamp;
DELIMITER $$
CREATE TRIGGER verify_timestamp BEFORE INSERT ON paga
FOR EACH ROW
BEGIN
    DECLARE last_stamp TIMESTAMP;
    SET last_stamp = (SELECT MAX(E.time_stamp) FROM estado E WHERE
E.numero = new.numero);
    IF NEW.data < last_stamp THEN
        CALL raise_application_error(3001, 'oferta existente');
    END IF;

END $$
delimiter ;
```

Desenvolvimento da aplicação

No desenvolvimento da aplicação foram utilizados os schema.php e populate.php fornecidos pelos docentes da cadeira. A única alteração feita no schema.php foi a colocação de “on delete cascade” para eliminar entradas em tabelas quando eliminadas entradas com os mesmos valores de outras tabelas (exemplo disso é eliminar todos os postos de trabalho quando se elimina um edifício).

Em relação ao PHP e HTML, foi criada uma mainPage.php onde poderão ser feitas todas as operações pedidas. Para cada operação foi criado um formulário e um ‘update’ de forma a executar o pedido. Tanto o formulário como o ‘update’ são semelhantes para todas as operações mudando apenas a query pedida e os atributos consoante as tabelas que se necessita alterar.

```

<html>
  <body>
<?php
  try
  {
    $host = "db.ist.utl.pt";
    $user = "ist179779";
    $password = "yolovidaloka";
    $dbname = $user;

    $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    echo("<h1>InstantOffice</h1>");
    echo("<p><a href='\"ListarEdificios.php\"'> Edificios </a></p>");
    echo("<p> &nbsp; &nbsp; &nbsp; <a href='\"InserirEdificio.php\"'> Inserir Edificio </a></p>");
    echo("<p><a href='\"ListarEspacos.php\"'> Espacos </a></p>");
    echo("<p> &nbsp; &nbsp; &nbsp; <a href='\"InserirEspaco.php\"'> Inserir Espaco </a></p>");
    echo("<p><a href='\"ListarPostos.php\"'> Postos </a></p>");
    echo("<p> &nbsp; &nbsp; &nbsp; <a href='\"InserirPosto.php\"'> Inserir Posto </a></p>");
    echo("<p><a href='\"ListarOfertas.php\"'> Ofertas </a></p>");
    echo("<p> &nbsp; &nbsp; &nbsp; <a href='\"CriarOferta.php\"'> Criar Oferta </a></p>");
    echo("<p><a href='\"ListarReservas.php\"'> Reservas </a></p>");

    $db = null;
  }
  catch (PDOException $e)
  {
    echo("<p>ERROR: {$e->getMessage()}</p>");
  }
?>
  </body>
</html>

```

Figura 1 - mainPage.php

```

<html>
  <body>

    <form action="updateInserirEdificio.php" method="post">
      <p>Morada: <input type="text" name="morada"/></p>
      <p><input type="submit" value="Submit"/></p>
    </form>
  </body>
</html>

```

Figura 2 - InserirEdificio.php

```

<html>
  <body>
<?php
    $morada = $_REQUEST['morada'];
    try
    {
        $host = "db.ist.utl.pt";
        $user = "ist179779";
        $password = "yolovidaloka";
        $dbname = $user;
        $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $db->query("start transaction;");

        $sql = "insert into edificio values ('$morada')";

        echo("<p>Edificio com a morada '$morada' inserido.</p>");

        $db->query($sql);

        $db->query("commit;");

        echo("<p><a href='http://web.ist.utl.pt/ist179779/mainPage.php'> Paginal inicial </a></p>");

        $db = null;
    }
    catch (PDOException $e)
    {
        $db->query("rollback;");
        echo("<p>ERROR: {$e->getMessage()}</p>");
    }
?>
  </body>
</html>

```

Figura 3 - updateInserirEdificio.php