
Licenciatura em Engenharia Telecomunicações e Informática

Relatório

Projeto de Base de Dados, Parte 3

sexta-feira, 03 de novembro de 2016

Grupo 14
BD8179L04

João Freitas - 81950 (15 horas)
João Carlos Costa – 82528 (15 horas)
Mariana Cruz – 82553 (15 horas)

Criação da Base de Dados

```
drop table fiscaliza cascade;  
drop table posto cascade;  
drop table espaco cascade;  
drop table aluga cascade;  
drop table oferta cascade;  
drop table paga cascade;  
drop table estado cascade;  
drop table arrenda cascade;  
drop table alugavel cascade;  
drop table reserva cascade;  
drop table edificio cascade;  
drop table fiscal cascade;  
drop table user cascade;
```

create table user

```
(nif int not null unique, nome varchar(255) not null, telefone int not null, primary key(nif));
```

create table fiscal

```
(id int not null unique, empresa varchar(255) not null, primary key(id));
```

create table edificio

```
(morada varchar(255) not null unique, primary key(morada));
```

create table alugavel

```
(morada varchar(255) not null, codigo int not null, foto varchar(255), primary key (morada, codigo), constraint foreign key(morada) references edificio(morada) on delete cascade);
```

create table arrenda

```
(morada varchar(255) not null, codigo int not null, nif int not null, primary key(morada, codigo), constraint foreign key(morada, codigo) references alugavel(morada, codigo) on delete cascade, constraint foreign key(nif) references user(nif) on delete cascade);
```

create table fiscaliza

```
(id int not null, morada varchar(255) not null, codigo int not null, primary key(id, morada, codigo), constraint foreign key(id) references fiscal(id) on delete cascade,  
constraint foreign key(morada, codigo) references arrenda(morada, codigo) on delete cascade);
```

create table espaco

```
(morada varchar(255) not null, codigo int not null, primary key(morada, codigo), constraint foreign key(morada, codigo) references alugavel(morada, codigo) on delete cascade);
```

create table posto

```
(morada varchar(255) not null, codigo int not null, codigo_espaco int not null,  
primary key(morada, codigo), constraint foreign key(morada, codigo) references alugavel(morada, codigo) on delete cascade, constraint foreign key(morada, codigo_espaco) references espaco(morada, codigo) on delete cascade);
```

create table oferta

```
(morada varchar(255) not null, codigo int not null, data_inicio date not null, data_fim date not null, tarifa int not null, primary key(morada, codigo, data_inicio), constraint foreign key(morada, codigo) references alugavel(morada, codigo) on delete cascade);
```

create table reserva

```
(numero int not null unique, primary key(numero));
```

create table paga

```
(numero int not null unique, data timestamp not null, metodo varchar(255) not null, primary key(numero), constraint foreign key(numero) references reserva(numero) on delete cascade);
```

create table estado

```
(numero int not null, time_stamp timestamp not null, estado varchar(255) not null, primary key(numero, time_stamp), constraint foreign key(numero) references reserva(numero) on delete cascade);
```

create table aluga

```
(morada varchar(255) not null, codigo int not null, data_inicio date not null, nif int not null, numero int not null, primary key(morada, codigo, data_inicio, nif, numero),
```

```
constraint foreign key(morada, codigo, data_inicio) references oferta(morada, codigo, data_inicio) on delete cascade, constraint foreign key(nif) references user(nif) on delete cascade, constraint foreign key(numero) references reserva(numero) on delete cascade);
```

Populate

```
insert into user values (456693446, 'joao', 913674287);
insert into user values (764379359, 'mariana', 916758325);
insert into user values (523768478, 'facas', 925476895);
```

```
insert into fiscal values (64676498, 'ist');
insert into fiscal values (65743833, 'maca lda');
insert into fiscal values (93578205, 'xp company');
```

```
insert into edificio values ('coimbra');
insert into edificio values ('lisboa');
insert into edificio values ('cascais');
```

```
insert into alugavel values ('coimbra', 100000, null);
insert into alugavel values ('lisboa', 100000, 'empregado do mes');
insert into alugavel values ('porto salvo', 100000, 'empregado do mes');
insert into alugavel values ('laranjeiras', 100000, 'proprio');
```

```
insert into alugavel values ('coimbra', 100001, 'chefe');
insert into alugavel values ('lisboa', 300003, 'empresa');
insert into alugavel values ('lisboa', 300004, 'casual');
insert into alugavel values ('porto', 300002, 'atitude positiva');
insert into alugavel values ('mafra', 100003, null);
insert into alugavel values ('laranjeiras', 100002, 'proprio');
(...)
```

```
insert into arrenda values ('coimbra', 100002, 523768478);
insert into arrenda values ('oeiras', 300000, 564325746);
insert into arrenda values ('laranjeiras', 100004, 534669654);
insert into arrenda values ('mafra', 100005, 523768478);
insert into arrenda values ('coimbra', 100004, 678658567);
insert into arrenda values ('lisboa', 200002, 764379359);
```

```
insert into fiscaliza values (35564633, 'coimbra', 100002);
insert into fiscaliza values (64676498, 'oeiras', 300000);
insert into fiscaliza values (43456783, 'porto', 200002);
insert into fiscaliza values (19496540, 'laranjeiras', 100004);
insert into fiscaliza values (35564633, 'mafra', 100005);
insert into fiscaliza values (87365399, 'porto salvo', 100000);
```

```
insert into espaco values ('coimbra', 100000);
insert into espaco values ('porto', 300000);
insert into espaco values ('oeiras', 300000);
insert into espaco values ('porto salvo', 100000);
insert into espaco values ('laranjeiras', 100000);
(...)
```

```
insert into posto values ('coimbra', 100001, 100000);
insert into posto values ('lisboa', 300005, 300000);
insert into posto values ('cascais', 200001, 200000);
insert into posto values ('porto', 100002, 100000);
insert into posto values ('laranjeiras', 100002, 100000);
(...)
```

```
insert into oferta values ('coimbra', 100000, date('2016-01-01'), date('2016-01-31'), 470);
insert into oferta values ('lisboa', 100002, date('2016-01-01'), date('2016-01-31'), 700);
(...)
```

```
insert into reserva values (1);
insert into reserva values (4);
insert into reserva values (6);
insert into reserva values (7);
(...)
```

```
insert into aluga values ('coimbra', 100000, date('2016-01-01'), 456693446, 1);
insert into aluga values ('cascais', 200001, date('2016-05-11'), 564325746, 11);
insert into aluga values ('porto', 200002, date('2016-01-01'), 564325746, 10);
(...)
```

```
insert into estado values (1, '2016-01-02 10:43:41', 'Pendente');
insert into estado values (1, '2016-01-02 16:43:41', 'Aceite');
insert into estado values (3, '2016-03-03 12:54:12', 'Declinada');
insert into estado values (4, '2016-01-10 13:45:00', 'Cancelada');
(...)
```

```
insert into paga values (1, '2016-01-03 16:43:41', 'visa');
insert into paga values (5, '2016-01-02 10:17:48', 'dinheiro');
insert into paga values (10, '2016-01-02 14:35:59', 'paypal');
(...)
```

SQL

1.

```
SELECT morada, t_codigo codigo FROM (SELECT morada, t_codigo, CASE WHEN codigo_espaco IS NULL THEN codigo ELSE codigo_espaco
END codigo FROM (SELECT t.morada, t.codigo t_codigo, codigo_espaco, u.codigo FROM (SELECT espaco.morada, codigo, codigo_espaco
FROM espaco LEFT JOIN (SELECT morada, codigo_espaco, z.c FROM (SELECT morada, codigo_espaco, COUNT(morada) c
FROM (SELECT * FROM (SELECT * FROM posto NATURAL JOIN aluga NATURAL JOIN estado WHERE estado = 'Aceite') x GROUP BY
morada, codigo) y GROUP BY morada, codigo_espaco) z NATURAL JOIN (
SELECT *, COUNT(morada) c FROM posto GROUP BY morada, codigo_espaco) w) v ON espaco.morada = v.morada AND espaco.codigo =
codigo_espaco) t LEFT JOIN (SELECT espaco.morada, espaco.codigo, estado FROM espaco LEFT JOIN aluga NATURAL JOIN estado ON
espaco.morada = aluga.morada AND espaco.codigo = aluga.codigo WHERE estado = 'Aceite') u ON t.morada = u.morada AND t.codigo=u.codigo)
r) s WHERE codigo IS NULL;
```

2.

```
SELECT * FROM (SELECT morada, COUNT(morada) cn, y.a FROM aluga, (SELECT AVG(c) a FROM (SELECT *, COUNT(morada) c FROM
aluga GROUP BY morada) x) y GROUP BY morada) z WHERE z.cn > z.a;
```

3.

```
SELECT nif FROM (SELECT *, COUNT(nif) c FROM (SELECT * FROM arrenda NATURAL JOIN fiscaliza GROUP BY nif, id) A GROUP BY nif) B
WHERE c = 1;
```

4.

```
SELECT espaco.morada, espaco.codigo, CASE WHEN sum IS NULL THEN 0 ELSE sum END sum FROM espaco LEFT JOIN (SELECT morada,
codigo, SUM(tarifa_total) sum FROM ( SELECT a_morada morada, CASE WHEN codigo_espaco IS NULL THEN a_codigo ELSE codigo_espaco
END as codigo, tarifa_total FROM (SELECT * FROM (SELECT morada a_morada, codigo a_codigo, (data_diff+1)*tarifa tarifa_total FROM
(SELECT *, DATEDIFF(data_fim, data_inicio) data_diff FROM oferta NATURAL JOIN aluga NATURAL JOIN paga) x
WHERE data like '2016_%') y LEFT JOIN posto p
ON a_morada = p.morada AND a_codigo = p.codigo) w) v GROUP BY morada, codigo) u
ON espaco.morada = u.morada AND espaco.codigo = u.codigo;
```

5.

```
SELECT morada, codigo FROM (SELECT e_morada, codigo, CASE WHEN t.morada IS NULL THEN u.morada ELSE t.morada END morada
FROM (SELECT e.morada e_morada, e.codigo, v.morada FROM espaco e LEFT JOIN (SELECT morada, codigo, estado FROM espaco NATURAL
JOIN aluga NATURAL JOIN estado
WHERE estado = 'Aceite' GROUP BY morada, codigo) v ON e.morada = v.morada AND e.codigo = v.codigo) t LEFT JOIN (SELECT morada,
codigo_espaco, z.c FROM (SELECT morada, codigo_espaco, COUNT(morada) c FROM (SELECT * FROM (SELECT * FROM posto NATURAL
JOIN aluga NATURAL JOIN estado WHERE estado = 'Aceite') x GROUP BY morada, codigo) y GROUP BY morada, codigo_espaco) z NATURAL
JOIN (SELECT *, COUNT(morada) c FROM posto GROUP BY morada, codigo_espaco) w) u ON e_morada = u.morada AND codigo =
codigo_espaco) r WHERE morada IS NOT NULL;
```

Restrições de Integridade

1.

```
delimiter //
CREATE TRIGGER data_oferta_check BEFORE INSERT ON oferta FOR EACH ROW BEGIN IF EXISTS (SELECT * FROM oferta WHERE
NEW.morada = oferta.morada AND NEW.codigo = oferta.codigo AND NEW.data_inicio > oferta.data_inicio AND NEW.data_inicio < oferta.data_fim)
THEN INSERT INTO oferta VALUES (NEW.morada, NEW.codigo, NEW.data_inicio, NEW.data_fim, NEW.tarifa);ELSEIF EXISTS (SELECT *
FROM oferta WHERE NEW.morada = oferta.morada AND NEW.codigo = oferta.codigo AND NEW.data_fim > oferta.data_inicio AND NEW.data_fim
< oferta.data_fim) THEN INSERT INTO oferta VALUES (NEW.morada, NEW.codigo, NEW.data_inicio, NEW.data_fim, NEW.tarifa);ELSEIF EXISTS
(SELECT * FROM oferta WHERE NEW.morada = oferta.morada AND NEW.codigo = oferta.codigo AND NEW.data_inicio < oferta.data_inicio AND
NEW.data_fim > oferta.data_fim) THEN INSERT INTO oferta VALUES (NEW.morada, NEW.codigo, NEW.data_inicio, NEW.data_fim,
NEW.tarifa);END IF;END //
delimiter ;
```

2.

```
delimiter //
CREATE TRIGGER data_paga_check BEFORE INSERT ON paga FOR EACH ROW BEGIN IF EXISTS(SELECT * FROM (SELECT * FROM
estado e NATURAL JOIN (SELECT numero, MAX(time_stamp) max_time_stamp FROM estado GROUP BY numero) A WHERE e.numero =
A.numero AND e.time_stamp = A.max_time_stamp) B WHERE NEW.numero = B.numero AND NEW.data < B.time_stamp) THEN INSERT INTO
paga VALUES (NEW.numero, NEW.data, NEW.metodo); END IF; END //
delimiter ;
```

PHP

edificio_remove.php

```
<html>
<body>
<?php
$morada = $_REQUEST['morada'];
try{
    $host = "db.ist.utl.pt";
    $user = "ist181950";
    $password = "evlw3510";
    $dbname = $user;
    $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $db->query("start transaction;");
    $sql = "DELETE FROM edificio WHERE morada = '$morada'";
    echo("<p>$sql</p>");
```

```

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

```

edificio_update.php

```

<html>
<body>
<?php
$morada = $_REQUEST['morada'];
try{
    $host = "db.ist.utl.pt";
    $user = "ist181950";
    $password = "evlw3510";
    $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>$sql</p>");
    $db->query($sql);
    $db->query("commit;");
    $db = null; }
catch (PDOException $e){
    $db->query("rollback;");
    echo("<p>ERROR: {$e->getMessage()}</p>"); }
?>
</body>
</html>

```

espaço_remove.php

```

<html>
<body>
<?php
$morada = $_REQUEST['morada'];
$codigo = $_REQUEST['codigo'];
try {
    $host = "db.ist.utl.pt";
    $user = "ist181950";
    $password = "evlw3510";
    $dbname = $user;
    $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->query("start transaction;");

    $sql = "DELETE FROM alugavel WHERE codigo IN (SELECT codigo FROM posto WHERE morada = '$morada' AND codigo_espaço = '$codigo');";
    $db->query($sql);
    $db->query("commit;");
    $db = null; }
catch (PDOException $e) {
    $db->query("rollback;");
    echo("<p>ERROR: {$e->getMessage()}</p>"); }
?>
</body>
</html>

```

espaço_update

```

<html>
<body>
<?php
$morada = $_REQUEST['morada'];
$codigo = $_REQUEST['codigo'];
try{
    $host = "db.ist.utl.pt";
    $user = "ist181950";
    $password = "evlw3510";
    $dbname = $user;
    $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->query("start transaction;");

```

```

        $sql = "INSERT INTO alugavel VALUES ('$morada', '$codigo', '');";
        $sql1 = "INSERT INTO espaco VALUES ('$morada', '$codigo');";

        echo("<p>$sql</p>");
        $db->query($sql);
        $db->query($sql1);
        $db->query("commit;");
        $db = null;}
    catch (PDOException $e){
        $db->query("rollback;");
        echo("<p>ERROR: {$e->getMessage()}</p>");}
?>
</body>
</html>

```

oferta_remove

```

<html>
    <body>
<?php
    $morada = $_REQUEST['morada'];
    $codigo = $_REQUEST['codigo'];
    $data_inicio = $_REQUEST['data_inicio'];
    try{
        $host = "db.ist.utl.pt";
        $user = "ist181950";
        $password = "evlw3510";
        $dbname = $user;
        $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $db->query("start transaction;");

        $sql = "DELETE FROM oferta WHERE morada = '$morada' AND codigo = '$codigo' AND data_inicio = '$data_inicio'";
        echo("<p>$sql</p>");
        $db->query($sql);
        $db->query("commit;");
        $db = null;}
    catch (PDOException $e){
        $db->query("rollback;");
        echo("<p>ERROR: {$e->getMessage()}</p>");}
?>
</body>
</html>

```

oferta_update

```

<html>
    <body>
<?php
    $morada = $_REQUEST['morada'];
    $codigo = $_REQUEST['codigo'];
    $data_inicio = $_REQUEST['data_inicio'];
    $data_fim = $_REQUEST['data_fim'];
    $tarifa = $_REQUEST['tarifa'];
    try{
        $host = "db.ist.utl.pt";
        $user = "ist181950";
        $password = "evlw3510";
        $dbname = $user;
        $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $db->query("start transaction;");
        echo ((bool)strtotime($data_inicio));
        if ((filter_var($_POST['tarifa'], FILTER_VALIDATE_FLOAT)) && (bool)strtotime($data_inicio) && (bool)strtotime($data_fim)) {
            $sql = "INSERT INTO oferta VALUES ('$morada', '$codigo', '$data_inicio', '$data_fim', '$tarifa')";
            echo("<p>$sql</p>");
            $db->query($sql);
            $db->query("commit;");
            $db = null; }
        catch (PDOException $e) {
            $db->query("rollback;");
            echo("<p>ERROR: {$e->getMessage()}</p>");}
?>
</body>
</html>

```

posto_remove

```

<html>
    <body>
<?php
    $morada = $_REQUEST['morada'];
    $codigo = $_REQUEST['codigo'];
    try{
        $host = "db.ist.utl.pt";
        $user = "ist181950";
        $password = "evlw3510";

```

```

$dbname = $user;
$db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$db->query("start transaction;");
$sql = "DELETE FROM alugavel WHERE morada = '$morada' AND codigo = '$codigo'";
echo("<p>$sql</p>");
$db->query($sql);
$db->query("commit;");
$db = null;
}
catch (PDOException $e){
    $db->query("rollback;");
    echo("<p>ERROR: {$e->getMessage()}</p>");
}
?>
</body>
</html>

```

```

posto_update
<html>
<body>
<?php
    $morada = $_REQUEST['morada'];
    $codigo = $_REQUEST['codigo'];
    $codigo_espaco = $_REQUEST['codigo_espaco'];
    try{
        $host = "db.ist.utl.pt";
        $user = "ist181950";
        $password = "evlw3510";
        $dbname = $user;
        $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $db->query("start transaction;");
        $sql = "INSERT INTO alugavel VALUES ('$morada', '$codigo', '')";
        $sql1 = "INSERT INTO posto VALUES ('$morada', '$codigo', '$codigo_espaco')";
        echo("<p>$sql</p>");
        $db->query($sql);
        $db->query($sql1);
        $db->query("commit;");
        $db = null;
    }
    catch (PDOException $e){
        $db->query("rollback;");
        echo("<p>ERROR: {$e->getMessage()}</p>");
    }
?>
</body>
</html>

```

```

reserva_paga_update
<html>
<body>
<?php
    $numero = $_REQUEST['numero'];
    $data = $_REQUEST['data'];
    $metodo = $_REQUEST['metodo'];
    try {
        $host = "db.ist.utl.pt";
        $user = "ist181950";
        $password = "evlw3510";
        $dbname = $user;
        $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $db->query("start transaction;");
        $sql1 = "SELECT MAX(numero)+1 FROM reserva;";
        $result = $db->query($sql1);
        foreach($result as $row){
            $nmax = ($row[0]); }
        $sql = "INSERT INTO paga VALUES ('$numero', '$data', '$metodo')";
        $sql2 = "INSERT INTO estado VALUES ('$numero', '$data', 'Aceite')";
        echo("<p>$sql</p>");
        $db->query($sql);
        $db->query($sql2);
        $db->query("commit;");
        $db = null;
    }
    catch (PDOException $e){
        $db->query("rollback;");
        echo("<p>ERROR: {$e->getMessage()}</p>");
    }
?>
</body>
</html>

```

```

reserve_update
<html>
<body>
<?php
    $morada = $_REQUEST['morada'];
    $codigo = $_REQUEST['codigo'];
    $data_inicio = $_REQUEST['data_inicio'];

```

```

$nf = $_REQUEST['nif'];
try{
    $host = "db.ist.utl.pt";
    $user = "ist181950";
    $password = "evlw3510";
    $dbname = $user;
    $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->query("start transaction;");
    $sql1 = "SELECT MAX(numero)+1 FROM reserva;";
    $result = $db->query($sql1);
    foreach($result as $row){
        $nmax = ($row[0]);
        $sql = "INSERT INTO reserva VALUES ('$nmax');";
        $sql2 = "INSERT INTO aluga VALUES ('$morada', '$codigo', '$data_inicio', '$nf', '$nmax');";
        $sql3 = "INSERT INTO estado VALUES ('$nmax', '$data_inicio', 'Pendente'); ";
        echo("<p>$sql</p>");
        echo("<p>$sql2</p>");
        echo("<p>$sql3</p>");
        $db->query($sql);
        $db->query($sql2);
        $db->query($sql3);
        $db->query("commit;");
        $db = null;
    }
    catch (PDOException $e){
        $db->query("rollback;");
        echo("<p>ERROR: {$e->getMessage()}</p>"); }
?>
</body>
</html>

total_realizado
<html>
<body>
<h3>Alugaveis</h3>
<?php
    $morada = $_REQUEST['morada'];
    try{
        $host = "db.ist.utl.pt";
        $user = "ist181950";
        $password = "evlw3510";
        $dbname = $user;
        $db = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
        $sql = "SELECT espaco.morada, espaco.codigo, CASE WHEN sum IS NULL THEN 0 ELSE sum END sum
        FROM espaco
        LEFT JOIN (
        SELECT morada, codigo, SUM(tarifa_total) sum
        FROM (
        SELECT a_morada morada, CASE WHEN codigo_espaco IS NULL THEN a_codigo ELSE codigo_espaco END as codigo, tarifa_total
        FROM (
        SELECT *
        FROM (
        SELECT morada a_morada, codigo a_codigo, (data_diff+1)*tarifa tarifa_total
        FROM (
        SELECT *, DATEDIFF(data_fim, data_inicio) data_diff
        FROM oferta
        NATURAL JOIN aluga
        NATURAL JOIN paga) x) y
        LEFT JOIN posto p
        ON a_morada = p.morada AND a_codigo = p.codigo) w) v
        GROUP BY morada, codigo) u
        ON espaco.morada = u.morada AND espaco.codigo = u.codigo
        WHERE espaco.morada = '$morada';
        ";
        $result = $db->query($sql);
        echo("<table border='0' cellspacing='5'>\n");
        echo("<td>morada</td>\n");
        echo("<td>codigo do espaco</td>\n");
        echo("<td>montate realizado</td>\n");
        foreach($result as $row){
            echo("<tr>\n");
            echo("<td>{$row['morada']}</td>\n");
            echo("<td>{$row['codigo']}</td>\n");
            echo("<td>{$row['sum']}</td>\n");
            echo("<tr>\n");
        }
        echo("</table>\n");
        echo("<td><a href='espaco_inserir.php?morada=$morada'>Inserir Espaco</a></td>\n");
        $db = null;
    }
    catch (PDOException $e) {
        echo("<p>ERROR: {$e->getMessage()}</p>"); }
?>
</body>
</html>

```