12 / 12 / 2014

Projeto de Bases de Dados

PartE 2

**Turno:** 2ª feira 14:00h Lab13

**Grupo:** 76

75657 Paulo Jorge Louseiro Gouveia

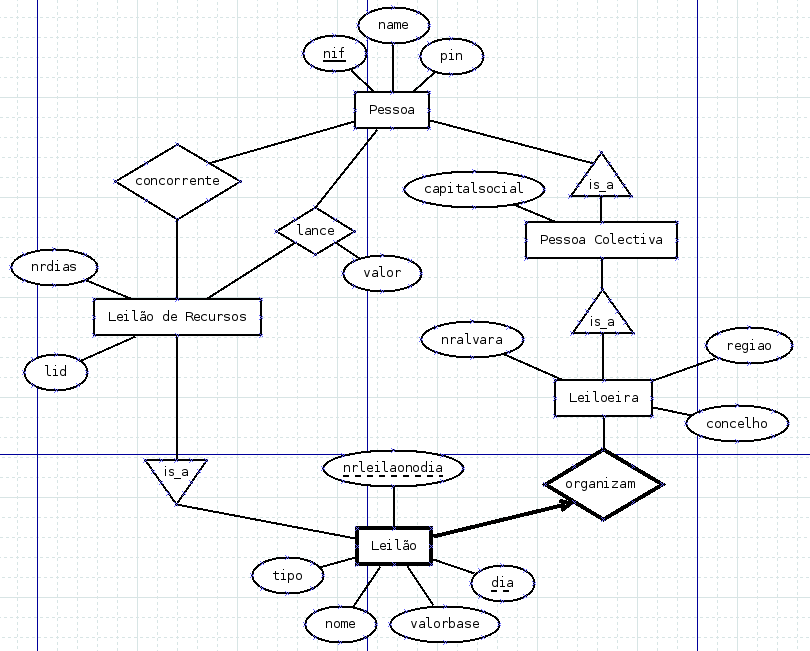
75694 Daniel Machado de Castro Figueira

76119 Ana Marta Correia Nunes

Indice

1. Modelo Entidade Associação 2
2. Scripts PHP e SQL
   1. login.html 3
   2. login\_validate.php 3
   3. home .html 5
   4. signup.php 5
   5. signup\_exec.php 7
   6. logout.php 8
   7. bid.php 8
   8. bid\_amount .php 11
   9. bid\_exec.php 12
3. Interrugações 14
4. Triggers
   1. t\_insert\_lance.sql 15
   2. t\_update\_lance.sql 16
5. Transações
   1. transaction.php 17
   2. transaction\_select.php 18
   3. transaction\_exec.php 20
6. Índices 22
7. Data warehouse e carregamento
   1. wh\_create.sql 23
   2. wh\_load.sql 24
8. Rollup 25

1. Modelo Entidade Associação



**RESTRIÇÕES DE INTEGRIDADE:**

1. Os atributos "nome" e "pin" da entidade "Pessoa" não podem ser nulos.

2. O atributo "capitalsocial" da entidade "Pessoa Colectiva" não pode ser nulo.

3. Os atributos "nralvara", "concelho" e "região" da entidade "Leiloeira" não podem ser nulos.

4. Os atributos "nome" e "tipo" da entidade "Leilão" não podem ser nulos.

5. O atributo "nrdias" da entidade "Leilão de Recursos" não pode ser nulo.

6. O atributo "lid" da entidade "Leilão de Recursos" é único e não nulo.

7. O atributo "valor" da associação "lance" é não nulo.

8. Apenas os pares de atributos "nif"-"lid" que fazem parte da associação "concorrente" podem participar na associação "lance".

2. SCRIPT PHP E SQL

**NOTA**: uma versão funcional da aplicação desenvolvida pode se encontrada em

[*http://web.ist.utl.pt/ist175657*](http://web.ist.utl.pt/ist175657)

**login.html**

<form action="login\_validate.php" method="post">

<h1> Sistema de Leiloes de Recursos Maritimos </h1>

<p> Introduza o seu nif: <input type="text" name="nif" /> </p>

<p> Introduza o seu pin: <input type="text" name="pin" /> </p>

<p> <input type="submit" value="Login"/> </p>

</form>

**login\_validate.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 de variáveis através do metodo POST;

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$nif = test\_input($\_POST["nif"]);

$pin = test\_input($\_POST["pin"]);

}

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre a palavra passe do utilizador

$sql = "select \* from pessoa where nif=".$nif;

$result = $connection->query($sql);

if(!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

// Verifica se o utilizador é válido

if($result->rowCount() == 0){

echo("<p>Nif Invalido!</p>\n");

$connection = null;

echo("<a href=\"login.html\">return</a>");

exit();

}

$result = $result->fetch();

$safepin = $result["pin"];

$username = $result["nome"];

// Verifica se a palavra passe está correta

if($safepin != $pin ) {

echo("<p>Pin Invalido!</p>\n");

$connection = null;

echo("<a href=\"login.html\">return</a>");

exit();

}

// Passa as variáveis para a sessão;

$\_SESSION["nif"] = $nif;

$\_SESSION["username"] = $username;

?>

<head>

<meta http-equiv="refresh" content="0; URL=home.html">

<meta name="keywords" content="automatic redirection">

</head>

<body>

If the page doesn't automatically load click

<a href="home.html">here</a>

</body>

</body>

</html>

**home.html**

<html>

<h1> Sistema de Leiloes de Recursos Maritimos </h1>

<h2> 2. Inscricao em leiloes, click <a href="signup.php">here</a></h2>

<h2> 2. Ver leiloes e licitar, click <a href="bid.php">here</a></h2>

<h2> 5. Inscrever em leiloes por dia, click <a href="transaction.php">here</a></h2>

<h2> <a href="logout.php">logout</a></h2>

</html>

**signup.php**

<html>

<h1> Sistema de Leiloes de Recursos Maritimos </h1>

<a href="home.html">return</a>

<br><br>

<body>

<?php

// Inicia sessão para passar variaveis entre ficheiros php

session\_start();

$nif = $\_SESSION["nif"];

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre os leilões registados

$sql = "select \* from leilao natural join leilaor order by lid";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

echo("<table border=\"1\">\n");

echo("<tr>

<td>leilao</td>

<td>nif</td>

<td>dia</td>

<td>nrleilaonodia</td>

<td>nome</td>

<td>tipo</td>

<td>valorbase</td>

<td>estado</td>

</tr>\n");

$now = new DateTime(date("Y-m-d H:i:s"));

foreach($result as $row){

// Calcula a data a que o leilão termina

$date = new DateTime($row["dia"]." +".$row["nrdias"]." days");

// Descobre se o utilizador está inscrito no leilão

$sql = "select \* from concorrente where pessoa=".$nif." and leilao=".$row["lid"].";";

$aux = $connection->query($sql);

if (!$aux) {

echo("</table>\n<p> Erro na Query:($sql)<p>");

exit();

}

echo("<tr><td>");

echo($row["lid"]); echo("</td><td>");

echo($row["nif"]); echo("</td><td>");

echo($row["dia"]); 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>");

// Verifica se o leilão já terminou

if($date < $now)

echo("encerrado");

else if($aux->rowCount() != 0)

echo("inscrito");

else

echo("<a href=\"signup\_exec.php?lid={$row["lid"]}\">inscrever</a>\n");

echo("</td></tr>");

}

echo("</table>\n");

?>

</body>

</html>

**signup\_exec.php**

<html>

<body>

<?php

// Inicia sessão para passar variaveis entre ficheiros php

session\_start();

$nif = $\_SESSION["nif"];

$lid = $\_REQUEST["lid"];

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$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");

// Regista o utilizador no leilão

$sql = "insert into concorrente (pessoa,leilao) VALUES ($nif,$lid)";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

?>

<head>

<meta http-equiv="refresh" content="0; URL=signup.php">

<meta name="keywords" content="automatic redirection">

</head>

<body>

If the page doesn't automatically load click

<a href="signup.php">here</a>

</body>

</body>

</html>

**logout.php**

<html>

<body>

<?php

session\_start();

// Termina a sessão do utilizador

session\_destroy();

?>

<head>

<meta http-equiv="refresh" content="0; URL=login.html">

<meta name="keywords" content="automatic redirection">

</head>

<body>

If the page doesn't automatically load click

<a href="login.html">here</a>

</body>

</body>

</html>

**bid.php**

<html>

<h1> Sistema de Leiloes de Recursos Maritimos </h1>

<a href="home.html">return</a>

<br><br>

<body>

<?php

// Inicia sessão para passar variaveis entre ficheiros php

session\_start();

$nif = $\_SESSION['nif'];

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre os leilões registados

$sql = "select \* from

leilao natural join leilaor

order by lid";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

echo("<table border=\"1\">\n");

echo("<tr>

<td>leilao</td>

<td>nif</td>

<td>nome</td>

<td>tempo restante</td>

<td>valor base</td>

<td>valor actual</td>

<td>maior licitador</td>

</tr>\n");

$now = new DateTime(date("Y-m-d H:i:s"));

foreach($result as $row){

// Calcula o tempo restante para o leilão encerrar

$date = new DateTime($row["dia"]." +".$row["nrdias"]." days");

$timeleft = $date->diff($now);

// Verifica se o leilão já terminou

if($date > $now){

echo("<tr><td>");

echo($row["lid"]); echo("</td><td>");

echo($row["nif"]); echo("</td><td>");

echo($row["nome"]); echo("</td><td>");

echo($timeleft->format("%a days, %H:%I:%S")); echo("</td><td>");

echo($row["valorbase"]); echo("</td><td>");

// Descobre o valor do maior lance

$sql = "select max(valor) as max, pessoa

from lance where leilao=".$row["lid"];

$aux1 = $connection->query($sql);

if (!$aux1) {

echo("</table>\n<p> Erro na Query:($sql)<p>");

exit();

}

$aux1 = $aux1->fetch();

if(!is\_null($aux1["max"])){

echo($aux1["max"]);echo("</td><td>");

// Descobre a pessoa que efectuou o lance mais alto

$sql = "select pessoa from lance

where leilao=".$row["lid"].

" and valor =".$aux1["max"];

$aux2 = $connection->query($sql);

if (!$aux2) {

echo("</table>\n<p> Erro na Query:($sql)<p>");

exit();

}

$aux2 = $aux2->fetch();

echo($aux2["pessoa"]);echo("</td><td>");

}

else{

echo("-");echo("</td><td>");

echo("-");echo("</td><td>");

}

// Descobre se o utilizador está inscrito no leilão

$sql = "select \* from concorrente where pessoa=".$nif." and leilao=".$row["lid"].";";

$aux3 = $connection->query($sql);

if (!$aux3) {

echo("</table>\n<p> Erro na Query:($sql)<p>");

exit();

}

if($aux3->rowCount() != 0)

echo("<a href=\"bid\_amount.php?lid={$row["lid"]}\">licitar</a>\n");

echo("</td></tr>");

}

}

echo("</table>\n");

?>

<head>

<meta http-equiv="refresh" content="10; URL=bid.php">

<meta name="keywords" content="automatic redirection">

</head>

</body>

</html>

**bid\_amount.php**

<html>

<h1> Sistema de Leiloes de Recursos Maritimos </h1>

<body>

<?php

$lid = $\_REQUEST['lid'];

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=". $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre o valor do maior lance

$sql = "select max(valor) as min from lance where leilao=".$lid.";";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

$result = $result->fetch();

if(is\_null($result["min"])){

// Descobre o valor base do leilão

$sql = "select valorbase as min from leilao natural join leilaor where lid=".$lid.";";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

$result = $result->fetch();

}

else

$result["min"] = $result["min"] + 1;

echo("<p>Valor minimo: ".$result["min"]."</p>");

?>

<form action="bid\_exec.php" method="post">

<p><input type="hidden" name="lid" value="<?=$lid?>"/></p>

<p>Novo valor: <input type="text" name="valor"/></p>

<p><input type="submit" value="Submeter"/></p>

</form>

</body>

<a href="bid.php">return</a>

</html>

**bid\_exec.php**

<html>

<body>

<?php

// Inicia sessão para passar variaveis entre ficheiros php

session\_start();

$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 de variáveis através do metodo POST;

if ($\_SERVER["REQUEST\_METHOD"] == "POST"){

$lid = test\_input($\_POST["lid"]);

$valor = test\_input($\_POST["valor"]);

if(empty($valor))

$valor = 0;

}

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=". $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre se o utilizador já licitou no leilão

$sql = "select \* from lance where pessoa=".$nif." and leilao=".$lid.";";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

if($result->rowCount() == 0){

// Cria uma nova licitação

$sql = "insert into lance(pessoa, leilao, valor)

values(".$nif.",".$lid.",".$valor.");";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

} else {

// Actualiza o valor da licitação

$sql = "update lance set valor=".$valor."

where pessoa=".$nif." and leilao=".$lid.";";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

}

?>

<head>

<meta http-equiv="refresh" content="0; URL=bid.php">

<meta name="keywords" content="automatic redirection">

</head>

<body>

If the page doesn't automatically load click

<a href="bid.php">here</a>

</body>

</body>

</html>

3. Interrogações

**3.1 Quais os participantes inscritos em leilões mas sem lances até à data?**

select distinct pessoa.nome

from pessoa

inner join concorrente on concorrente.pessoa = pessoa.nif

left join lance on concorrente.pessoa = lance.pessoa

where lance.pessoa is null;

**3.2 Qual o nome das pessoas coletivas com exatamente duas inscrições em leilões?**

select pessoa.nome

from pessoa

natural join pessoac

inner join concorrente on pessoa.nif = concorrente.pessoa

group by pessoa.nome

having count(concorrente.leilao) = 2;

**3.3 Qual o leilão com o maior rácio (valor do melhor lance)/(valor base)?**

select leilao.nome

from leilao natural join leilaor, lance

where leilaor.lid = lance.leilao

group by lance.leilao

having max(lance.valor/leilao.valorbase) >= all(

select max(lance.valor/leilao.valorbase)

from leilao natural join leilaor, lance

where leilaor.lid = lance.leilao

group by lance.leilao);

**3.4 Quais as pessoas coletivas com o mesmo capital social?**

select p.nome, pc.capitalsocial

from pessoa p, pessoac pc

where p.nif = pc.nif

group by pc.capitalsocial;

4. Triggers

**t\_insert\_lance.sql**

-- trigger para que o valor mínimo de um lance num leilão seja o valor base desse leilão.

drop trigger if exists t\_insert\_lance;

delimiter //

create trigger t\_insert\_lance before insert on lance

for each row

begin

select max(valor)

from lance

where leilao = new.leilao

into @min;

if(@min is NULL) then

select valorbase

from leilao natural join leilaor

where lid = new.leilao

into @min;

else

set @min = @min + 1;

end if;

if new.valor <= @min then

set new.valor = @min;

end if;

end//

delimiter ;

**t\_update\_lance.sql**

-- trigger para que o valor dos lances num determinado leilão seja sempre crescente.

drop trigger if exists t\_update\_lance;

delimiter //

create trigger t\_update\_lance before update on lance

for each row

begin

select max(valor)

from lance

where leilao = new.leilao

into @min;

if new.valor <= @min then

set new.valor = @min + 1;

end if;

end//

delimiter ;

5. Transacções

**transaction.php**

<html>

<h1> Sistema de Leiloes de Recursos Maritimos </h1>

<a href="home.html">return</a>

<br><br>

<body>

<?php

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre os dias em que foram iniciados leilões

$sql = "select distinct dia from leilao order by dia";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

foreach($result as $row){

echo("<a href=\"transaction\_select.php?day={$row["dia"]}\">".$row["dia"]."</a>\n");

echo("<br>");

}

echo("</table>\n");

?>

</body>

</html>

**transaction\_select.php**

<html>

<h1> Sistema de Leiloes de Recursos Maritimos </h1>

<a href="transaction.php">return</a>

<br><br>

<body>

<?php

// Inicia sessão para passar variaveis entre ficheiros php

session\_start();

$nif = $\_SESSION["nif"];

$day = $\_REQUEST["day"];

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre os leilões que começaram num determinado dia

$sql = "select \* from leilao natural join leilaor

where dia=\"".$day."\"";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

echo("<form action=\"transaction\_exec.php?day={$day}\" method=\"post\">");

echo("<table border=\"1\">\n");

echo("<tr>

<td>leilao</td>

<td>nif</td>

<td>dia</td>

<td>nrleilaonodia</td>

<td>nome</td>

<td>tipo</td>

<td>valorbase</td>

<td>inscrever</td>

</tr>\n");

$i = 0;

$now = new DateTime(date("Y-m-d H:i:s"));

foreach($result as $row){

// Calcula a data a que o leilão termina

$date = new DateTime($row["dia"]." +".$row["nrdias"]." days");

// Descobre se o utilizador está inscrito no leilão

$sql = "select \* from concorrente where pessoa=".$nif." and leilao=".$row["lid"].";";

$aux = $connection->query($sql);

if (!$aux) {

echo("</table>\n<p> Erro na Query:($sql)<p>");

exit();

}

// Verifica se o utilizador se encontra inscrito no leilão e se este está activo

if($date > $now and $aux->rowCount() == 0){

echo("<tr><td>");

echo($row["lid"]); echo("</td><td>");

echo($row["nif"]); echo("</td><td>");

echo($row["dia"]); 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>");

echo("<input type=\"checkbox\" name=\"checkbox[".$i."]\" value=\"selected\

" /></>\n");

echo("</td></tr>");

}

$i = $i+1;

}

echo("</table>\n");

echo("<input type=\"submit\" value=\"Inscrever\"/>");

echo("</form>");

?>

</body>

</html>

**transaction\_exec.php**

<html>

<body>

<?php

// Inicia sessão para passar variaveis entre ficheiros php

session\_start();

$nif = $\_SESSION['nif'];

$day = $\_REQUEST["day"];

$checkbox = $\_POST["checkbox"];

// Variáveis de conexão à BD

$host="db.ist.utl.pt";

$user="ist175657";

$password="dlno1642";

$dbname = $user;

$connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,

array(PDO::ATTR\_ERRMODE => PDO::ERRMODE\_WARNING));

// Descobre os leilões que começaram num determinado dia

$sql = "select \* from leilao natural join leilaor

where dia=\"".$day."\"";

$result = $connection->query($sql);

if (!$result) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

// Inicia uma transacção

$sql = "start transaction;";

$action = $connection->query($sql);

if (!$action) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

$i = 0;

$now = new DateTime(date("Y-m-d H:i:s"));

foreach($result as $row){

// Calcula a data a que o leilão termina

$date = new DateTime($row["dia"]." +".$row["nrdias"]." days");

// Descobre se o utilizador está inscrito no leilão

$sql = "select \* from concorrente where pessoa=".$nif." and leilao=".$row["lid"].";";

$aux = $connection->query($sql);

if (!$aux) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

if($date > $now and $aux->rowCount() == 0 and $checkbox[$i]){

// Regista o utilizador no leilão

$sql = "insert into concorrente (pessoa,leilao) VALUES (".$nif.", ".$row["lid"].")";

$action = $connection->query($sql);

if (!$action) {

echo("<p> Erro na Query:($sql) <p>");

// Cancela e transacção

$sql = "rollback;";

$action = $connection->query($sql);

if (!$action)

echo("<p> Erro na Query:($sql)<p>");

exit();

}

}

$i = $i+1;

}

// Termina a transacção

$sql = "commit;";

$action = $connection->query($sql);

if (!$action) {

echo("<p> Erro na Query:($sql)<p>");

exit();

}

?>

<head>

<meta http-equiv="refresh" content="0; URL=transaction.php">

<meta name="keywords" content="automatic redirection">

</head>

<body>

If the page doesn't automatically load click

<a href="transaction">here</a>

</body>

</body>

</html>

6. Índices

**index\_create.sql**

-- for 3.3.

create index valorbase\_idx on leilao(valorbase);

create index leilao\_idx on lance(leilao);

create index valor\_idx on lance(valor);

/\* embora ‘leilao’ e ‘valor’ façam parte da primary key (pessoa, leilão, valor) que é por default indexada,

na query 3.3. nao é usado esse índice pois nenhuma das condições envolve o elemento mais a esquerda

ou uma combinação dos três \*/

-- for 3.4.

create index capitalsocial\_idx on pessoac(capitalsocial);

7. data warehouse e CArregamento

**wh\_create.sql**

set foreign\_key\_checks = 0 ;

-- time dimension

drop table if exists time;

create table time(

time\_id int not null,

day date,

primary key (time\_id)

);

-- adress dimension

drop table if exists address;

create table address(

address\_id int not null,

town VARCHAR (80) default null,

region VARCHAR (80) default null,

primary key (address\_id)

);

-- facts table

drop table if exists bids ;

create table bids(

time\_id int not null,

address\_id int not null,

lid int not null,

bid int not null default '0',

primary key (time\_id, address\_id, lid),

foreign key (time\_id) references time(time\_id),

foreign key (address\_id) references address(address\_id),

foreign key (lid) references leilaor(lid)

);

set foreign\_key\_checks = 1 ;

**wh\_load.sql**

set foreign\_key\_checks = 0 ;

truncate table time;

truncate table address;

truncate table bids;

set foreign\_key\_checks = 1 ;

-- time inserts

lock tables time write, leilaor read;

insert into time(time\_id, day)

select lid, dia

from leilaor

;

unlock tables;

-- address inserts

lock tables address write, leilaor as lr read, leiloeira as la read;

insert into address(address\_id, town, region)

select lr.lid, la.concelho, la.regiao

from leilaor lr, leiloeira la

where lr.nif = la.nif

;

unlock tables;

-- bids inserts

lock tables bids write, leilaor as lr read, lance as lc read, time read, address read;

insert into bids(time\_id, address\_id, lid, bid)

select lr.lid, lr.lid, lr.lid, max(lc.valor)

from leilaor lr left join lance lc on lr.lid = lc.leilao

group by lr.lid

;

unlock tables;

8. ROllup

**rollup.sql**

select a.town, extract(year from t.day) as year, extract(month from t.day) as month, sum(b.bid) as sum

from bids b, time t, address a

where b.time\_id = t.time\_id and b.address\_id = a.address\_id

group by a.town, extract(year from t.day), extract(month from t.day) with rollup

having year = 2012 | year = 2013

;