

# Bases de Dados

## Relatório do Projecto

### 2ª Parte

---



Duarte Baptista 70542

João Soares 73959

Ricardo Abreu 76370

Docente responsável pela UC:

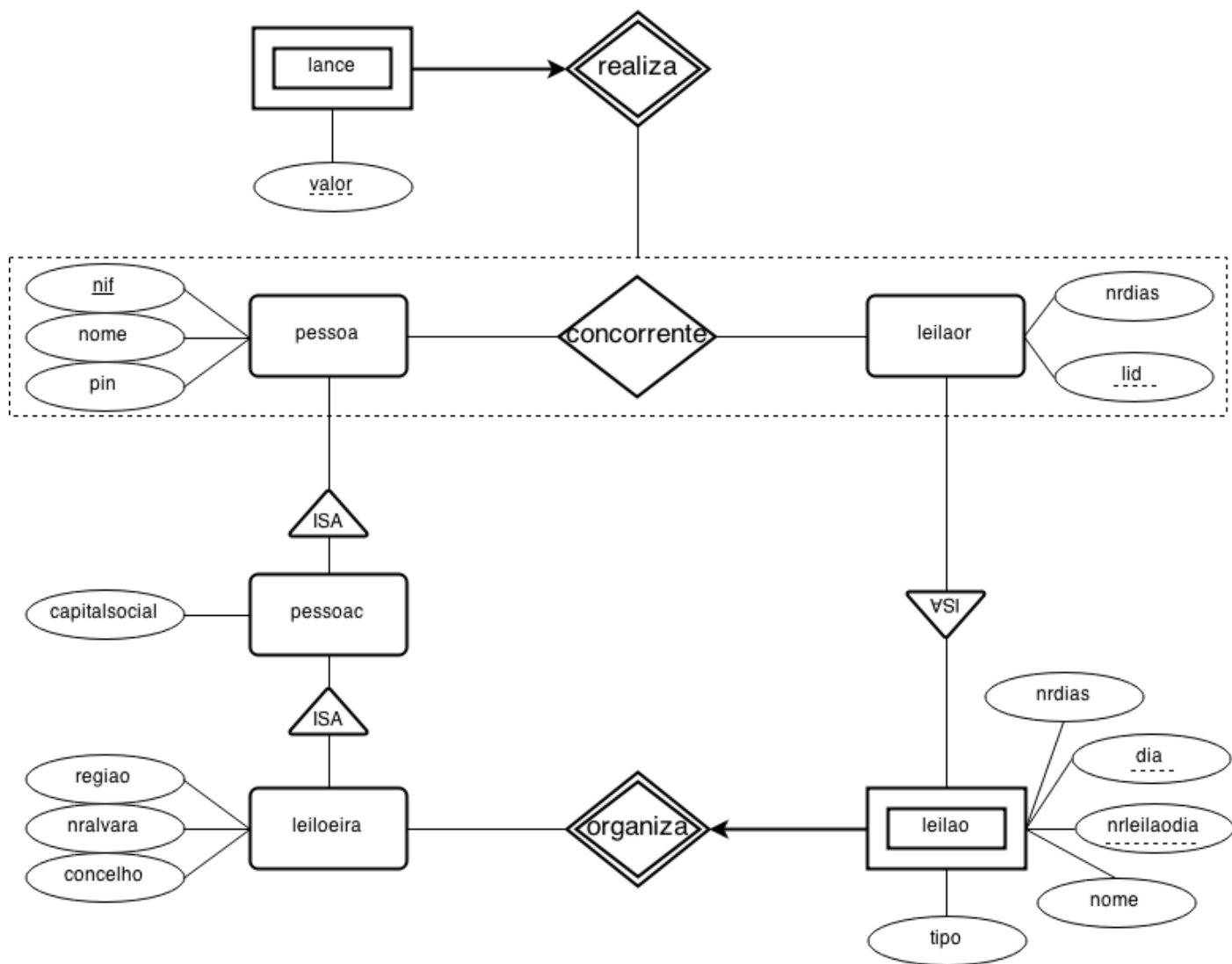
Mário Gaspar da Silva

# Índice

---

1. Modelo Entidade Associação	3
2. Script PHP	4
3. Interrogações SQL	9
4. Trigger	10
5. Transacções	11
6. Índices	13
7. Esquema em Estrela	15
8. Rollup	16
9. View	17

## 1. Modelo Entidade Associação



## 2. Script PHP e SQL

```
1. # Scripts PHP e MySQL utilizados para gerir a Base de Dados
2. #
3. # Projecto parte 2 Bases de Dados
4. #
5. #
6. # 12-12-2014
7.
8. #####
9. # Script 'registo.php'#
10. #####
11.
12. <html>
13.     <body>
14.         <?php
15.             // inicia sessão para passar variaveis entre ficheiros php
16.             session_start();
17.             // Funcao para limpar os dados de entrada
18.             function test_input($data) {
19.                 $data = trim($data);
20.                 $data = stripslashes($data);
21.                 $data = htmlspecialchars($data);
22.                 return $data;
23.             }
24.
25.             // Carregamento das variaveis username e pin do form HTML atraves do metodo POST;
26.             if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["username"]) && isset($_POST["pin
27.             "])) {
28.                 $_SESSION['username'] = test_input($_POST["username"]);
29.                 $_SESSION['pin'] = test_input($_POST["pin"]);
30.             }
31.
32.             // passa variaveis para a sessao;
33.             $username = $_SESSION['username'];
34.             $pin = $_SESSION['pin'];
35.
36.             //Variaveis de conexao a BD
37.             $host = "db.ist.utl.pt";
38.             // o MySQL esta disponivel nesta maquina
39.             $user = "ist176370";
40.             //-> substituir pelo nome de utilizador
41.             $password = "lldx0788";
42.             //-> substituir pela password dada pelo mysql_reset
43.             $dbname = $user;
44.             // a BD tem nome identico ao utilizador
45.
46.             echo("<p>Projeto Base de Dados Parte II</p><br>");
47.
48.             $connection = new PDO("mysql:host=" . $host . ";dbname=" . $dbname, $user, $password, arra
49.             y(PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING));
50.
51.             //echo("<p>Connected to MySQL database $dbname on $host as user $user</p>\n");
52.
53.             // obtem o pin da tabela pessoa
54.             $sql = "SELECT * FROM pessoa WHERE nif=" . $username;
55.             $result = $connection -> query($sql);
56.
57.             if (!$result) {
58.                 echo("<p> Erro na Query:($sql)</p>");
59.                 exit();
60.             }
61.
62.             foreach ($result as $row) {
```

```

61.             $safepin = $row["pin"];
62.     $nif = $row["nif"];
63.     }
64.     if ($safepin != $pin) {
65.         echo "<p>Pin Invalido! Exit!</p>\n";
66.         $connection = null;
67.         exit ;
68.     }
69.     //echo "<p>Pin Valido!</p>\n";
70.
71.     ?>
72.     <html><form action='logout.php' method='POST'>
73.     <input type='submit' value='LOGOUT'>
74.     </form></html>
75.     <?php
76.
77.     // Apresenta os leilões - versao anterior
78.     //$sql = "SELECT lr.lid,l.dia,l.nif,l.nome,l.valorbase, l.nrleilaonodia FROM leilao l JOI
N leilaor lr where l.nif=lr.nif and l.dia=lr.dia and l.nrleilaonodia=lr.nrleilaonodia and lr.dia <
= date(now()) and date(now()) <= lr.dia + interval lr.nrdias day group by lid";
79.
80.     ?>
81.     <html><h3>Leiloes em curso ou a iniciar</h3></html>
82.     <?php
83.
84.     $sql = "SELECT lr.lid,l.dia,l.nif,l.nome,l.valorbase, l.nrleilaonodia FROM leilao l JOIN
leilaor lr where l.nif=lr.nif and l.dia=lr.dia and l.nrleilaonodia=lr.nrleilaonodia and date(now()
) <= lr.dia + interval lr.nrdias day group by lid";
85.     $result = $connection -> query($sql);
86.     echo("<table border='1' \n");
87.     echo("<tr><td>lid</td><td>nif</td><td>dia</td><td>NrDoDia</td><td>nome</td><td>valorbase</
td></tr>\n");
88.     $idleilao = 0;
89.     foreach ($result as $row) {
90.         $idleilao = $idleilao + 1;
91.         echo("<tr><td>");
92.         echo($row["lid"]);
93.         echo("</td><td>");
94.         echo($row["nif"]);
95.         echo("</td><td>");
96.         echo($row["dia"]);
97.         echo("</td><td>");
98.         echo($row["nrleilaonodia"]);
99.         echo("</td><td>");
100.         echo($row["nome"]);
101.         echo("</td><td>");
102.         echo($row["valorbase"]);
103.         echo("</td><td>");
104.     }
105.     echo("</table>\n<br><br>");
106.
107.     ?>
108.     <html><h3>Lance maximo de cada Leilao em que esta inscrito</h3></html>
109.     <?php
110.
111.     $sql = "select v.lid,v.Maximo, v.diasAteAoFinal
112.     from (select leilao as lid from concorrente where pessoa = " . $username. ") as tab
113.     inner join MyView v on v.lid = tab.lid";
114.
115.     $result = $connection -> query($sql);
116.
117.     echo("<table border='1'>\n");
118.     echo("<tr><td>lid</td><td>Maximo</td><td>diasAteAoFinal</td></tr>\n");
119.     foreach ($result as $row) {
120.         echo("<tr><td>");
121.         echo($row["lid"]);
122.         echo("</td><td>");

```

```

123.             echo($row["Maximo"]);
124.             echo("</td><td>");
125.             echo($row["diasAteAoFinal"]);
126.             echo("</td><td>");
127.         }
128.         echo("</table>\n");
129.     ?>
130.
131.     <form action="leilao.php" method="post">
132.         <h3>Escolha o ID do leilao que pretende concorrer</h3>
133.         <p>
134.             ID :
135.             <input type="text" name="lid" />
136.         </p>
137.         <p>
138.             <input type="submit" />
139.         </p>
140.     </form>
141.
142.     <form action="lance.php" method="post">
143.         <h3>Insira o ID do leilão e o valor do respectivo lance</h3>
144.         <p>
145.             ID:
146.             <input type="text" name="lid" />
147.         </p>
148.         <p>
149.             Lance:
150.             <input type="text" name="valor" />
151.         </p>
152.         <p>
153.             <input type="submit" />
154.         </p>
155.     </form>
156.
157.     <form action="leilaoTransact.php" method="post">
158.         <h3>Escolha o ID do leilão que pretende concorrer(inscriÃo atomica) <br> Inserir os
159.         lids dos leiloes pretendidos separados por virgulas EX: 1,2,5,6,7 </h2>
160.         <p>
161.             ID :
162.             <input type="text" name="lids" />
163.         </p>
164.         <p>
165.             <input type="submit" />
166.         </p>
167.     </form>
168. </body>
169. </html>
170.
171.
172. #####
173. # Script 'leilao.php' #
174. #####
175.
176. <html>
177. <body>
178. <?php
179.     // inicia sessÃo para passar variaveis entre ficheiros php
180.     session_start();
181.     $username = $_SESSION['username'];
182.     $nif = $_SESSION['nif'];
183.
184.     // FunÃo para limpar os dados de entrada
185.     function test_input($data) {
186.         $data = trim($data);
187.         $data = stripslashes($data);
188.         $data = htmlspecialchars($data);

```

```

189.         return $data;}

190.
191.     // Carregamento das variáveis username e pin do form HTML através do metodo POST;
192.     if ($_SERVER["REQUEST_METHOD"] == "POST") {
193.         $lid = test_input($_POST["lid"]);}
194.
195.     // Conexão à BD
196.     $host="db.ist.utl.pt";
197.
198.     // o MySQL esta disponivel nesta maquina
199.     $user="ist176370";
200.
201.     //-> substituir pelo nome de utilizador
202.
203.     $password="l1dx0788";
204.
205.     //-> substituir pela password dada pelo mysql_reset
206.     $dbname = $user;
207.
208.     // a BD tem nome identico ao utilizador
209.     $connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,
210.         array(PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING));
211.     //echo("<p>Connected to MySQL database $dbname on $host as user $user</p>\n");
212.
213.     echo("<a href='registo.php'><button>voltar para home</button></a>");
214.
215.     //verifica se esta a decorrer
216.     $sql = "SELECT COUNT(*) AS id FROM leilao WHERE lid = " . $lid. " AND dia + interval nrddias
day > DATE(now())";
217.     $result = $connection -> query($sql);
218.
219.     foreach ($result as $row){$count = $row["id"];}
220.
221.     if($count == 0){
222.         echo("<p> Leilao ja terminou <p>");
223.         exit();
224.     }
225.
226.     $sql = "INSERT INTO concorrente (pessoa,leilao) VALUES ($username,$lid)";
227.     $result = $connection->query($sql);
228.
229.     //verifica se a pessoa ja esta inscrita no leilao
230.     if (!$result) {echo("<p> Pessoa nao registada <p>");
231.         exit();}
232.     echo("<p> Pessoa ($username) Registada no leilao ($lid)</p><br>");
233.
234.     $_SESSION['nif'] = $nif;
235.     $_SESSION['lid'] = $lid;
236.     ?>
237.
238.
239.
240.     #####
241.     # Script 'lance.php'#
242.     #####
243.
244.     <html>
245.     <body>
246.     <?php
247.         // inicia sessão para passar variaveis entre ficheiros php
248.         session_start();
249.         $nif = $_SESSION['username'];
250.
251.         // Função para limpar os dados de entrada
252.         function test_input($data) {
253.             $data = trim($data);
254.             $data = stripslashes($data);

```

```

255.                                     $data = htmlspecialchars($data);
256.         return $data;}
257.
258.         // Carregamento das variáveis username e pin do form HTML através do metodo POST;
259.         if ($_SERVER["REQUEST_METHOD"] == "POST") {
260.             $valor = test_input($_POST["valor"]);
261.             $lid = test_input($_POST["lid"]);}
262.
263.         // Conexão à BD
264.         $host="db.ist.utl.pt";
265.
266.         // o MySQL esta disponivel nesta maquina
267.         $user="ist176370";
268.
269.         //-> substituir pelo nome de utilizador
270.         $password="l1dx0788";
271.
272.         //-> substituir pela password dada pelo mysql_reset
273.         $dbname = $user;
274.
275.         // a BD tem nome identico ao utilizador
276.         $connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,
277.             array(PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING));
278.         //echo("<p>Connected to MySQL database $dbname on $host as user $user</p>\n");
279.
280.         echo("<a href='registo.php'><button>voltar para home</button></a>");
281.         // verifica se a pessoa esta inscrita no respectivo leilao
282.         $sql = "SELECT * FROM concorrente WHERE pessoa=" . $nif . " AND leilao=" . $lid;
283.         $result = $connection -> query($sql);
284.
285.
286.         $sql = "SELECT COUNT(*) AS id FROM concorrente WHERE pessoa=" . $nif . " AND leilao=" . $lid;
287.
288.         $result = $connection -> query($sql);
289.         foreach($result as $row){
290.             $count = $row["id"];
291.         }
292.
293.         // condicao de seguranca
294.         if($count == 0){
295.             echo("<p> Pessoa nao registada no leilao <p>");
296.             exit();
297.         }
298.
299.         // verifica se o leilão ainda está a decorrer
300.         $sql = "SELECT COUNT(*) AS id FROM leilao WHERE lid = " . $lid. " AND dia + interval nrdias
day > DATE(now())";
301.         $result = $connection -> query($sql);
302.
303.         foreach ($result as $row){$count = $row["id"];}
304.
305.         if($count == 0){
306.             echo("<p> Leilao ja terminou <p>");
307.             exit();
308.         }
309.
310.         $sql = "INSERT INTO lance (pessoa,leilao,valor) VALUES ($nif,$lid,$valor)";
311.         $result = $connection->query($sql);
312.
313.         if (!$result) {echo("<p> Lance nao registado <p>");
314.             exit();}
315.         echo("<p> Lance de valor ($valor) registado com sucesso no leilao ($lid)</p>\n");
316.         ?>
317.     </body>
318. </html>

```



NOTA: para facilitar a utilização de certos valores das tabelas nos scripts PHP criamos uma VIEW. Na última página do relatório encontra-se o script SQL para a criação da VIEW

### 3. Interrogações SQL

```
1. # Query 3.1:
2. #
3. # 1. Quais os participantes inscritos em leilões mas sem lances até à data?
4.
5. select distinct c.pessoa
6. from concorrente c
7. where c.pessoa
8. not in (select l.pessoa from lance l)
9. ;
10.
11.
12.
13. # Query 3.2:
14. #
15. # 2. Qual o nome das pessoas coletivas com exatamente duas inscrições em leilões?
16.
17. select pessoa
18. from (select pessoa, count(*) as num from concorrente group by pessoa) tab
19. where tab.num = 2 and tab.pessoa
20. in (select nif from pessoac)
21. ;
22.
23.
24.
25. # Query 3.3:
26. #
27. # 3. Qual o leilão com o maior rácio (valor do melhor lance)/(valor base)?
28.
29. select distinct leilao, (max/valorbase) as racio
30. from (select leilao, max(valor) as max from lance group by leilao) as tab join leilao le join leilao lr
31. on tab.leilao=lr.lid and le.nif=lr.nif and le.nrleilaonodia=lr.nrleilaonodia and le.dia=lr.dia
32. where max/valorbase = (select max(l1.valor/l3.valorbase)
33. from leilao l3, leilao l2, lance l1
34. where l1.leilao = l2.lid
35. and l2.dia = l3.dia
36. and l2.nrleilaonodia = l3.nrleilaonodia
37. and l2.nif = l3.nif)
38. ;
39.
40.
41.
42. # Query 3.4:
43. #
44. # 4. Quais as pessoas coletivas com o mesmo capital social?
45.
46. select p1.nif, p2.nif from pessoac p1, pessoac p2
47. where p1.capitalsocial = p2.capitalsocial AND p1.nif < p2.nif
48. ;
```

## 4. Trigger

```
1. # Trigger 'TriggerLance'
2. #
3. # :: verifica se os lances efectuados nao vao contra as
4. # regras de integridade da Base de Dados
5.
6. delimiter //
7. drop trigger if exists triggerLance //
8. CREATE TRIGGER triggerLance BEFORE INSERT ON lance
9. FOR EACH ROW
10. BEGIN
11.
12.     if new.valor < (select valorbase
13.                     from leilao l cross join leilaor lr on l.dia=lr.dia and l.nrleilaonodia=lr
14.                     .nrleilaonodia and l.nif=lr.nif
15.                     where lid = new.leilao)
16.     then call LanceInferiorAoValorBase();
17.     end if;
18.     if new.valor < (select max(l.valor) from lance l where new.leilao = l.leilao group by leil
19.     ao)
20.     then call LanceInferiorAoLanceActual();
21.     end if;
22. end//
23. delimiter ;
```

## 5. Transação

```
1. <html>
2. <body>
3. <?php
4.     // inicia sessao para passar variaveis entre ficheiros php
5.     session_start();
6.     $username = $_SESSION['username'];
7.     $nif = $_SESSION['nif'];
8.
9.     // Funcao para limpar os dados de entrada
10.    function test_input($data) {
11.        $data = trim($data);
12.        $data = stripslashes($data);
13.        $data = htmlspecialchars($data);
14.    return $data;}
15.
16. // Carregamento das variaveis username e pin do form HTML através do metodo POST;
17. if ($_SERVER["REQUEST_METHOD"] == "POST") {
18. $lids = test_input($_POST["lids"]);}
19.
20. // Conexao À BD
21. $host="db.ist.utl.pt";
22.
23. // o MySQL esta disponivel nesta maquina
24. $user="ist176370";
25.
26. //-> substituir pelo nome de utilizador
27.
28. $password="lldx0788";
29.
30. //-> substituir pela password dada pelo mysql_reset
31. $dbname = $user;
32.
33. // a BD tem nome identico ao utilizador
34. $connection = new PDO("mysql:host=" . $host. ";dbname=" . $dbname, $user, $password,
35. array(PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING));
36.
37. echo("<a href='registo.php'><button>voltar para home</button></a>");
38.
39. $nif = $username;
40. $vecAux= array();
41.
42. // vector que contem em cada posicao os lids dos leiloes a inscrever
43. $vecTokens = explode(",", $lids);
44.
45.     $arrayDatas = array();
46.
47.     $vecTokensAux = $vecTokens;
48.
49.     // para cada lid escolhido verifica a data do leilao e ve se este ja terminou
50.     foreach($vecTokensAux as $valTok){
51.         $query = "SELECT dia, count(*) as ver
52.                 FROM leilaor
53.                 WHERE lid = $valTok and date(now()) <= dia + interval nrdias day";
54.
55.         $daysWithAuctions= $connection->query($query);
56.
57.         // para cada dia de entre os leiloes escolhidos
58.         foreach ($daysWithAuctions as $rowdwa){
59.             $count = $rowdwa["ver"];
60.
```

```

61.                                     // verifica se o leilao a inscrever ja terminou
62.     if ($count == 0)
63.     {
64.         $kont = 0;
65.         echo("Leilao $valTok ja terminou");
66.
67.         // remove o lid do leilao que ja terminou do vector 'vecTokens'
68.         foreach ($vecTokens as $g){
69.             if($g == $valTok){
70.                 unset($vecTokens[$kont]);
71.             }
72.             $kont = $kont +1;
73.         }
74.         // insere a data correspondente ao lid no vector 'arrayDatas'
75.     }else{
76.         $data = $rowdwa['dia'];
77.         array_push($arrayDatas,$data);
78.     }
79. }
80. // elimina datas duplicadas do vector 'arrayDatas'
81. $arrayDatas = array_unique($arrayDatas);
82. }
83. // para cada data contida em 'arrayDatas', realiza uma Transaction
84. foreach($arrayDatas as $row2){
85.
86.     $query = "SELECT lid FROM leilaor WHERE dia='$row2'";
87.
88.     $auctionForDay= $connection->query($query);
89.
90.     $a=array();
91.
92.     // coloca no vector 'a' todos os lids que pertencam a respectiva data
93.     foreach($auctionForDay as $aux){
94.         foreach($vecTokens as $valLids){
95.             if ($aux['lid']==$valLids){array_push($a,$valLids);}
96.         }
97.     }
98.
99.     $result = array_unique($a);
100.
101.     try {
102.         // realiza uma transacao por cada dia, inserindo todos os lids escolhidos d
103.         esse respectivo dia
104.         $connection->beginTransaction();
105.
106.         foreach($result as $val){
107.             $resultT = $connection->query("INSERT INTO `ist176370`.`concorrente` (`pessoa`, `leilao`) VALUES ($nif, $val)");
108.             if (!$resultT) throw new PDOException("Transacção relativa ao dia $row
109. 2 falhada - ROLLBACK");
110.         }
111.         $connection->commit();
112.         echo("<p>Comit feito da transacao correspondente ao dia $row2</p>\n");
113.
114.         // apenas realiza o commit se nenhuma query desse respectivo dia falhar
115.     } catch(PDOException $ex) {
116.         // faz rollback das querys por dia
117.         $connection->rollBack();
118.         echo $ex->getMessage();
119.     }
120.
121.     unset($a);
122. }
123. $_SESSION['nif'] = $nif;
124. ?>

```

NOTA: no nosso projecto, na parte das transacções o nosso grupo implementou uma solução ligeiramente diferente do que era pedido no enunciado. Em vez de tomarmos por partido que os lids a inscrever pertenciam todos ao mesmo dia, o nosso script após receber os lids, agrupa-os por dia e de seguida realiza uma transacção por dia

## 6. Índices

### Query 4:

Ao efectuar o EXPLAIN da interrogação 3 obtém-se a tabela1.

```
1. create index indCap
2. on pessoac (capitalsocial)
3. using btree;
```

Escolhemos usar este índice pois queremos seleccionar as pessoas baseadas no seu capital social.

Como se pode constatar pela tabela2, após a inserção deste índice a instrução SQL passa a ser realizada fazendo uso do mesmo.

Os tempos medidos não foram conclusivos pois as tabelas eram de pequena dimensão. A validação foi feita com base no comando EXPLAIN.

**Tabela 1**

#	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	1	PRIMARY	<derived2>	ALL	NULL	NULL	NULL	NULL	274	
2	1	PRIMARY	pessoac	ALL	PRIMARY	NULL	NULL	NULL	274	Range checked for each reco
3	2	DERIVED	p2	ALL	NULL	NULL	NULL	NULL	274	

**Tabela 2**

#	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	1	PRIMARY	<derived2>	ALL	NULL	NULL	NULL	NULL	274	
2	1	PRIMARY	pessoac	ref	PRIMARY,indCap	indCap	4	tab1.cap2	1	Using where; Using index
3	2	DERIVED	p2	index	NULL	indCap	4	NULL	274	Using index

### Query 3:

Ao efectuar o EXPLAIN da interrogação 4 obtém-se a tabela3.

```
1. create index indValor
2. on lance (valor)
3. using btree;
```

Após a análise da tabela 3, concluímos que apesar da query sql já estar a usar índices, não estava a usar os ideais para cada join/where.

Desta forma, escolhemos implementar um índice no valor de cada lance, pois calculamos o lance máximo para cada leilão em parte da query.

Com base na tabela 4, podemos verificar que essa foi uma decisão correcta pois agora, o índice utilizado é o indValor que tem um comprimento (key\_len) mais pequeno que o índice da primary key.

Os tempos medidos não foram conclusivos pois as tabelas eram de pequena dimensão. A validação foi feita com base no comando EXPLAIN.

**Tabela 3**

#	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	1	PRIMARY	<derived2>	ALL	<b>NULL</b>	<b>NULL</b>	<b>NULL</b>	<b>NULL</b>	12	Using temporary
2	1	PRIMARY	lr	eq_ref	PRIMARY,lid	lid	4	tab.leilao	1	Using index
3	1	PRIMARY	le	eq_ref	PRIMARY	PRIMARY	11	ist176370.lr.nif,ist1763	1	Using where
4	3	SUBQUERY	l1	index	<b>NULL</b>	PRIMARY	12	<b>NULL</b>	13	Using index
5	3	SUBQUERY	l2	eq_ref	PRIMARY,lid	lid	4	ist176370.l1.leilao	1	Using index
6	3	SUBQUERY	l3	eq_ref	PRIMARY	PRIMARY	11	ist176370.l2.nif,ist1763	1	
7	2	DERIVED	lance	index	<b>NULL</b>	PRIMARY	12	<b>NULL</b>	13	Using index; Using

**Tabela 4**

#	id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	1	PRIMARY	<derived2>	ALL	<b>NULL</b>	<b>NULL</b>	<b>NULL</b>	<b>NULL</b>	12	Using temporary
2	1	PRIMARY	lr	eq_ref	PRIMARY,lid	lid	4	tab.leilao	1	Using index
3	1	PRIMARY	le	eq_ref	PRIMARY	PRIMARY	11	ist176370.lr.nif,ist1763	1	Using where
4	3	SUBQUERY	l1	index	<b>NULL</b>	indValor	4	<b>NULL</b>	13	Using index
5	3	SUBQUERY	l2	eq_ref	PRIMARY,lid	lid	4	ist176370.l1.leilao	1	Using index
6	3	SUBQUERY	l3	eq_ref	PRIMARY	PRIMARY	11	ist176370.l2.nif,ist1763	1	
7	2	DERIVED	lance	index	<b>NULL</b>	indValor	4	<b>NULL</b>	13	Using index; Using

## 7. Estrela e Scripts de carregamento

```
1. DROP TABLE IF EXISTS Factos ;
2. CREATE TABLE Factos(
3.     localID          INT NOT NULL,
4.     tempoID          Date NOT NULL,
5.     valor            INT,
6.     leilao           INT,
7.     PRIMARY KEY (localID,tempoID,leilao),
8.     FOREIGN KEY (localID) references LocalDim(localID),
9.     FOREIGN KEY (tempoID) references Tempo(tempoID)
10. );
11.
12. DROP TABLE IF EXISTS LocalDim ;
13. CREATE TABLE LocalDim(
14.     localID          INT AUTO_INCREMENT,
15.     Regiao           VARCHAR(80),
16.     Concelho         VARCHAR(80),
17.     PRIMARY KEY (localID)
18. );
19.
20. DROP TABLE IF EXISTS Tempo ;
21. CREATE TABLE Tempo(
22.     tempoID          Date NOT NULL,
23.     Dia              INT,
24.     Mes              INT,
25.     Ano              INT,
26.     PRIMARY KEY (tempoID)
27. );
28.
29. select * from Factos;
30. select * from Tempo;
31. select * from LocalDim;
32.
33. #popula Tempo
34. INSERT INTO Tempo (tempoID,Ano,Mes,Dia) select distinct dia, YEAR(dia),MONTH(dia),DAY(dia) from leilao;
35.
36. #popula Local
37. INSERT INTO LocalDim (Regiao,Concelho) select distinct regiao, concelho from leiloeira;
38.
39. #popula tabela Factos
40. insert into Factos(tempoID, localID, valor,leilao)
41. select dia, LocalDim.localID, max(valor),leilao
42. from lance natural join leilaor natural join leiloeira, LocalDim
43. where lance.leilao=leilaor.lid and LocalDim.Concelho=leiloeira.concelho and LocalDim.Regiao=leiloeira.regiao
44. group by lid;
```

NOTA: na tabela Factos usamos como primary key a combinação de atributos (LocalID,TempoID,lid) pois só assim podemos identificar univocamente lance mais elevado para cada leilão

## 8. Rollup

```
1. #Considerando o esquema da estrela, escreva a interrogacao em MySQL para obter a receita dos
2. # leiloes em cada concelho, com rollup por ano e mes do bienio 2012-2013.
3.
4. select Concelho,sum(Valor)as quantity,Ano,Mes
5. from Factos f natural join LocalDim natural join Tempo
6. where Ano = 2012 or Ano=2013
7. group by Concelho,Ano,Mes with rollup;
8. where Concelho is not null
```



## 9. View

```
1. CREATE
2. VIEW `MyView` AS
3.     select
4.         `l`.`nif` AS `nif`,
5.         `l`.`dia` AS `dia`,
6.         `l`.`nrleilaonodia` AS `nrleilaonodia`,
7.         `lr`.`lid` AS `lid`,
8.         `lr`.`nrdias` AS `nrdias`,
9.         `l`.`valorbase` AS `valorbase`,
10.        (select
11.            max(`laa`.`valor`)
12.        from
13.            `lance` `laa`
14.        where
15.            (`laa`.`leilao` = `lr`.`lid`)) AS `Maximo`,
16.        (select
17.            (max(`laa`.`valor`) / `l`.`valorbase`)
18.        from
19.            `lance` `laa`
20.        where
21.            (`laa`.`leilao` = `lr`.`lid`)) AS `Racio`,
22.        (select (to_days((`l`.`dia` + interval `lr`.`nrdias` day)) -
23.            to_days(cast(now() as date)))) AS `diasAteAoFinal`
24.    from
25.        (`leilao` `l`
26.        join `leilaor` `lr` ON (((`l`.`dia` = `lr`.`dia`)
27.            and (`l`.`nif` = `lr`.`nif`)
28.            and (`l`.`nrleilaonodia` = `lr`.`nrleilaonodia`))));
```