

# Projecto de Bases de Dados, Parte 2

Ricardo Leitão, nº 69632

Fábio Almeida, nº 70227

December 14, 2012

## 1 Criação da base de dados

```
create table Utente
(nick    varchar(255)    not null unique,
nome     varchar(255)    not null,
numero  numeric(5)      not null,
saldo   numeric(9,2)    not null,
primary key(nick));

create table Viatura
(matricula    varchar(8)      not null unique,
marca         varchar(255)    not null,
modelo        varchar(255)    not null,
maxocupantes  numeric(1)      not null,
nick          varchar(255)    not null,
primary key(matricula),
foreign key(nick) references Utente(nick),
check(maxocupantes >= 2 and maxocupantes <= 9));

create table Aluno
(nick    varchar(255)    not null,
curso   varchar(255)    not null,
primary key(nick),
foreign key(nick) references Utente(nick));

create table Docente
(nick    varchar(255)    not null,
primary key(nick),
foreign key(nick) references Utente(nick));
```

```

create table Funcionario
    (nick    varchar(255)    not null,
     primary key(nick),
     foreign key(nick) references Utente(nick));

create table Condutor
    (nick    varchar(255)    not null,
     primary key(nick),
     foreign key(nick) references Utente(nick));

create table Passageiro
    (nick    varchar(255)    not null,
     primary key(nick),
     foreign key(nick) references Utente(nick));

create table Local
    (nome          varchar(255)    not null unique,
     latitude      numeric(3)      not null,
     longitude     numeric(3)      not null,
     primary key(nome));

create table Trajecto
    (nome_origem   varchar(255)    not null,
     nome_destino   varchar(255)    not null,
     primary key(nome_origem, nome_destino),
     foreign key(nome_origem) references Local(nome),
     foreign key(nome_destino) references Local(nome));

create table Boleia
    (nick    varchar(255)    not null,
     data_hora    timestamp(0)    not null,
     custo_passageiro numeric(3,2) not null,
     nome_origem   varchar(255)    not null,
     nome_destino   varchar(255)    not null,
     nick_condutor  varchar(255),
     matricula     varchar(8),
     primary key(nick, data_hora),
     foreign key(nick) references Utente(nick),
     foreign key(nome_origem, nome_destino) references Trajecto(
         nome_origem, nome_destino),
     foreign key(nick_condutor) references Condutor(nick),
     foreign key(matricula) references Viatura(matricula),
     check(custo_passageiro >= 0));

create type frequencia as enum('diaria', 'semanal', 'mensal');

create table BoleiaFrequente
    (nick          varchar(255)    not null,
     data_hora     timestamp(0)    not null,

```

```

        data_termino    timestamp(0)    not null,
        tipo            frequencia      not null,
        primary key(nick, data_hora),
        foreign key(nick, data_hora) references Boleia(nick,
            data_hora));

create table BoleiaUnica
    (nick            varchar(255)    not null,
    data_hora        timestamp(0)    not null,
    primary key(nick, data_hora),
    foreign key(nick, data_hora) references Boleia(nick,
        data_hora));

create table InscricaoP
    (nick_passageiro  varchar(255)    not null,
    nick_organizador  varchar(255)    not null,
    data_hora         timestamp(0)    not null,
    primary key(nick_passageiro, nick_organizador, data_hora),
    foreign key(nick_passageiro) references Passageiro(nick),
    foreign key(nick_organizador, data_hora) references Boleia(
        nick, data_hora));

```

## 2 Normalização

A relação Boleia encontra-se na 1ª Forma, porque apesar de todos os seus atributos serem escalares alguns dos atributos não dependem somente da chave.

A relação Viatura encontra-se na 2ª Forma, porque apesar dos seus atributos não chave serem completamente dependentes do atributo chave e serem valores escalares, estes não são independentes entre si (modelo-¿marca).

## 3 Consultas SQL

a)

```

select B.nick, B.data_hora, I.nick_passageiro, B.custo_passageiro, B.
    nome_origem, B.nome_destino, B.nick_condutor, B.matricula, V.
    marca, V.modelo, V.maxocupantes from Boleia B inner join
    InscricaoP I on( B.nick = I.nick_organizador and B.data_hora = I.
    data_hora) left outer join Viatura V on(B.matricula = V.matricula
    ) where current_timestamp < B.data_hora;

```

b)

```

create view soma as select B1.nick, B1.nick_condutor, sum(B1.
    custo_passageiro) as custo_total from Boleia B1, InscricaoP I
    where B1.nick = I.nick_organizador and B1.nick_condutor is not
    null group by B1.nick, B1.nick_condutor;

```

```
create view media as select S.nick_condutor, avg(S.custo_total) as
    custo_medio from soma S group by S.nick_condutor;
select M.nick_condutor from media M where M.custo_medio = (select
    max(M1.custo_medio) from media M1);
```

c)

```
/* Dada a origem 'Lisboa':*/
create view totais as select count(distinct nome_destino) from
    trajecto where nome_origem='Lisboa';
create view userTraj as select nick_passageiro, count(distinct B.
    nome_destino) from boleia B, inscricao I WHERE B.nick = I.
    nick_organizador and nome_origem='Lisboa' group by I.
    nick_passageiro;
create view condTraj as select nick_condutor, count(distinct B.
    nome_destino) from boleia B WHERE B.nick = B.nick_condutor and
    nome_origem='Lisboa' group by nick_condutor;
select nick_passageiro from (select * from userTraj union select *
    from condTraj) A, totais T where A.count = T.count;
```

## 4 Restrições de Integridade

a)

```
create or replace function passageiro_insert_proc()
returns trigger
as
$$
declare
matricula_nova varchar(8);
max_ocupantes numeric(1);
boleia_ocupantes numeric(1);
saldo_passageiro numeric(9,2);
custo_boleia numeric(9,2);
begin
    if not new.nick_organizador = new.nick_passageiro then
        select matricula into matricula_nova from boleia B where B.nick
            = new.nick_organizador and B.data_hora = new.data_hora;
        select saldo into saldo_passageiro from utente U where U.nick =
            new.nick_passageiro;
        select custo_passageiro into custo_boleia from Boleia B where B.
            nick = new.nick_organizador and B.data_hora=new.data_hora;
        select count(*) into boleia_ocupantes from inscricaoP I where I.
            nick_organizador = new.nick_organizador and I.data_hora = new
                .data_hora;
        if matricula_nova is not null then
            select maxocupantes into max_ocupantes from viatura V where V.
                matricula = matricula_nova;
            if max_ocupantes < boleia_ocupantes then
```

```

        raise exception 'Boleia ja esta cheia';
    end if;
    if saldo_passageiro < custo_boleia then
        raise exception 'Saldo insuficiente';
    else
        update utente set saldo = saldo - custo_boleia where nick =
            new.nick_passageiro;
    end if;
else
    if max_ocupantes < 9 then
        raise exception 'Boleia ja esta cheia';
    end if;
    if saldo_passageiro < custo_boleia then
        raise exception 'Saldo insuficiente';
    else
        update utente set saldo = saldo - custo_boleia where nick =
            new.nick_passageiro;
    end if;
end if;
return new;
end
$$
language plpgsql;

create trigger inscreve_passageiro
before insert on inscricaoP for each row
execute procedure passageiro_insert_proc();

```

b)

```

create or replace function condutor_insert_proc()
returns trigger
as
$$
declare
max_ocupantes numeric(1);
boleia_ocupantes numeric(1);
condutor varchar(255);
begin
    select maxocupantes into max_ocupantes from viatura where
        matricula = new.matricula;
    select count(*) into boleia_ocupantes from inscricaoP where
        nick_organizador = new.nick and data_hora = new.data_hora;
    select nick_condutor into condutor from boleia where nick = new.
        nick and data_hora = new.data_hora;

```

```

    if condutor is not null then
        raise exception 'A boleia ja tem condutor';
    end if;
    if max_ocupantes < boleia_ocupantes then
        raise exception 'A viatura nao tem lugares suficientes';
    end if;
    return new;
end
$$
language plpgsql;

create trigger inscreve_condutor
before update on Boleia for each row
execute procedure condutor_insert_proc();

```

## 5 Desenvolvimento da aplicação

A aplicação encontra-se disponível para ser testada em [web.ist.utl.pt/ist170227](http://web.ist.utl.pt/ist170227). Sugerimos a utilização da base de dados do aluno ist169632 para o teste da aplicação e a base de dados do aluno ist170227 para o teste do índice.

### config.php

```

<?php
$pg_hostname = "db.ist.utl.pt";
$pg_user = "ist169632";//"ist170227"
$pg_password = "ztat5885";//"carpooling"
$pg_port=5432;
$pg_database = $pg_user;
$bd = pg_connect("host=$pg_hostname port=$pg_port user=$pg_user
    password=$pg_password dbname=$pg_database") or die(pg_last_error
    ());
?>

```

### index.php

```

<?php
include("config.php");
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $myusername=addslashes($_POST['username']);
    $mypassword=addslashes($_POST['password']);
    $sql="SELECT nick FROM utente WHERE nick='$myusername'";
    $result=pg_query($sql);
    $count=pg_num_rows($result);
    // If result matched $myusername, table row must be 1 row
    if($count==1){
        session_start();
        $_SESSION['nick']=$myusername;
    }
}

```

```

        $sql="SELECT nome FROM utente WHERE nick='$myusername'";
        $result=pg_query($sql);
        $row=pg_fetch_array($result);
        $nome=$row['nome'];
        $_SESSION['login_nome']=$nome;
        header("location: paginainicial.php");
    }else {
        $error="O utilizador nao existe";
    }
}??>

```

## paginainicial.php

```

<?php
include("config.php");
session_start();
$nick=$_SESSION['nick'];
$sql = "select B.nick, B.data_hora, B.custo_passageiro, B.
        nome_origem, B.nome_destino, B.nick_condutor, B.matricula from
        boleia B, InscricaoP I where current_timestamp < B.data_hora
        and B.nick = I.nick_organizador and B.data_hora = I.data_hora
        and I.nick_passageiro = '$nick'";
$result = pg_query($sql);
$viaturas = pg_query("select * from Viatura where nick = '$nick'");
;
$dados = pg_fetch_assoc(pg_query("select * from utente where nick
        = '$nick'"));
?>

```

## criarboleia.php

```

<?php
include("config.php");
session_start();
$nick=$_SESSION['nick'];
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $horas = $_POST['horas'];
    $minutos = $_POST['minutos'];
    $data_hora = $_POST['orderdate']." ".$horas.":".$minutos;
    $data_termino=$_POST['orderdate1']." 00:00";
    $euros = $_POST['euros'];
    $centimos = $_POST['centimos'];
    $centimos = $centimos / 100;
    $custo = $euros + $centimos;
    $nome_origem = addslashes($_POST['origem']);
    $nome_destino = addslashes($_POST['destino']);
    if(isset($_POST['cond'])){
        $matricula = $_POST['cond_val'];
        pg_query("insert into Boleia values('$nick', '$data_hora',
            $custo, '$nome_origem', '$nome_destino', '$nick', '

```

```

        $matricula'))" or die(pg_last_error());
    }else{
        pg_query("insert into Boleia(nick, data_hora, custo_passageiro
        , nome_origem, nome_destino) values('$nick', '$data_hora',
        $custo, '$nome_origem', '$nome_destino')") or die(
        pg_last_error());
    }
    if(isset($_POST['freq'])){
        $freq_val = $_POST['freq_val'];
        pg_query("INSERT INTO BoleiaFrequente VALUES('$nick', '
        $data_hora', '$data_termino', '$freq_val')") or die(
        pg_last_error());
    }else{
        pg_query("INSERT INTO BoleiaUnica VALUES('$nick', '$data_hora
        ')") or die(pg_last_error());
    }
    echo("<script type=\"text/javascript\"> window.alert(\"A sua
        boleia foi criada.\")</script>");
}??>

```

## inscreverboleia.php

```

<?php
include("config.php");
session_start();
$nick=$_SESSION['nick'];
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $nickp=strtok($_POST['inscricao'],",");
    $data=strtok(",");
    $matricula = $_POST['cond_val'];
    if(isset($_POST['condutor'])){
        pg_query("update boleia set nick_condutor='$nick', matricula='
        $matricula' where nick='$nickp' and data_hora='$data'") or
        die(pg_last_error());
        $data = "Condutor";
    }else{
        pg_query("insert into inscricao values('$nick', '$nickp', '
        $data')") or die(pg_last_error());
        $data = "Passageiro";
    }
    echo("<script type=\"text/javascript\"> window.alert(\"Registou-
        se como " . $data . " na Boleia.\")</script>");
}
$sql = "select * from boleia where current_timestamp < data_hora";
$result = pg_query($sql) or die(pg_last_error());
?>

```

## registrarviatura.php

```

<?php

```



```

include("config.php");
session_start();
$nick=$_SESSION['nick'];
if($_SERVER["REQUEST_METHOD"] == "POST"){
    $marca=$_POST['marca'];
    $modelo=$_POST['modelo'];
    $matricula=$_POST['matricula1'].'-'. $_POST['matricula2'].'-'.
        $_POST['matricula3'];
    $ocupantes=$_POST['ocupantes'];
    pg_query("insert into viatura values('$matricula', '$marca', '
        $modelo', $ocupantes, '$nick')") or die(pg_last_error());
    echo("<script type=\"text/javascript\"> window.alert(\"A sua
        viatura foi registada.\")</script>");
}??>

```

## registar.php

```

<?php
include("config.php");
if($_SERVER["REQUEST_METHOD"] == "POST"){
    // username and password sent from form
    $nick=addslashes($_POST['nick']);
    $nome=addslashes($_POST['nome']);
    $ist_numero=$_POST['ist_numero'];
    $tipo=addslashes($_POST['tipo']);
    $curso=$_POST['curso'];
    if(strlen($ist_numero) > 5){
        $error="N mero ist invalido";
    }else if(pg_num_rows(pg_query("select nick from utente where
        nick='$nick'")) != 0){
        $error="Nick ja existe";
    }else{
        pg_query("INSERT INTO utente VALUES('$nick', '$nome', '
            $ist_numero', 0)");
        if($tipo == "aluno"){
            pg_query("insert into $tipo values ('$nick', '$curso')");
        }else{pg_query("INSERT INTO $tipo VALUES('$nick')");}
        session_start();
        $_SESSION['nick']=$nick;
        $_SESSION['login_nome']=$nome;
        header("location: paginainicial.php");
    }
}??>

```

## logout.php

```

<?php
session_start();
if(session_destroy()){
    header("Location: index.php");}??>

```

## 6 Índices

Um índice adequado será o índice BTREE.

```
create index IndexBoleia on Boleia(nome_origem);
```

De modo a testar a eficácia do índice, criámos (em conjunto com outro grupo) um script que gera Utentes, Locais, Trajectos e Boleias:

```
#!/bin/sh
i=1;
while [ $i -le 30 ]
do
    number='expr 100000 + $i';
    echo "INSERT INTO utente values('user"$i"', 'Mr.User"$i"', '$number"', 5);"
    i='expr $i + 1'
done

j=1;
while [ $j -le 100 ]
do
    echo "INSERT INTO local values('local"$j"', '$j"', '$j')";"
    j='expr $j + 1'
done

l=2;
while [ $l -le 100 ]
do
    echo "INSERT INTO trajecto values('local"$l"', 'local"$l"', '$l')";"
    l='expr $l + 1'
done

m=1;
while [ $m -le 12 ]
do
    d=1;
    while [ $d -le 30 ]
    do
        h=0;
        while [ $h -le 23 ]
        do
            min=0;
            while [ $min -le 15 ]
            do
                echo "INSERT INTO boleia(nick, data_hora, custo_passageiro,
                    nome_origem, nome_destino) values('user"$d"', '2012-$m-$d-$h-$min-";"
            done
        done
    done
done
```

```

        "$d $h":"$min":00', 5, 'local"$d"', 'local'"'expr $d + 1'"
    ');"
    min='expr $min + 1'
done
h='expr $h + 1'
done
d='expr $d + 1'
done
m='expr $m + 1'
done

```

Depois de carregar os dados gerados pelo script, aplicámos a interrogação apresentada no enunciado (“traduzida” para SQL), obtendo um tempo de execução de 160ms. Depois de criado o índice voltámos a aplicar a mesma interrogação e obtivemos um resultado significativamente menor, 42ms.

```

#ist170227=> \timing
#Timing is on.
#ist170227=> select * from boleia where nome_origem = 'local5' and
    data_hora > date_trunc('minute', timestamp '2001-02-16 20:38:40')
;
#Time: 160.387 ms
#ist170227=> create index IndexBoleia on Boleia(nome_origem);
#CREATE INDEX
#Time: 3389.180 ms
#ist170227=> select * from boleia where nome_origem = 'local5' and
    data_hora > date_trunc('minute', timestamp '2001-02-16 20:38:40')
;
#Time: 42.645 ms

```

Utilizamos como local de origem o ‘local5’, pois o script não gera nenhum local chamado ‘Tagus’ (como referido no enunciado), no entanto, o que é importante notar é a melhoria da performance na interrogação e não qual o local de origem utilizado na interrogação. Podemos concluir que a interrogação está de facto a beneficiar da existência do índice.

## 7 Transações

Incluimos no nosso código alguns triggers para garantir a atomicidade de uma inscrição numa boleia que é chamados depois da inserção na tabela. Assim garantimos que um utilizador que crie uma boleia como passageiro seja inserido na tabela inscricaoP. Incluimos também triggers para quando são adicionados novos utentes e viaturas, garantindo que são também inseridos nas respectivas tabelas (passageiro e condutor).

```

/*Trigger passageiro_insert chama passageiro_proc()
*
*Introduz automaticamente um utente novo na tabela de passageiros
*/
create or replace function passageiro_proc()
returns trigger

```

```

as
$$
begin
    insert into passageiro values(new.nick);
    return new;
end
$$
language plpgsql;

create trigger passageiro_insert
after insert on utente for each row
execute procedure passageiro_proc();

/*Trigger condutor_insert chama condutor_proc()
*
*Introduz automaticamente na tabela condutor um utente que registre
uma viatura
*/
create or replace function condutor_proc()
returns trigger
as
$$
declare
existe numeric(1);
begin
    select count(nick) into existe from condutor where nick=new.nick;
    if existe = 0 then
        insert into condutor values(new.nick);
    end if;
    return new;
end
$$
language plpgsql;

create trigger condutor_insert
after insert on viatura for each row
execute procedure condutor_proc();

/*Trigger inscricao_insert chama inscricao_proc()
*
*Introduz nas tabelas inscricaoP o passageiro, se for o caso,
aquando da cria ao de uma boleia
*/
create or replace function inscricao_proc()
returns trigger
as
$$
begin
    if new.nick_condutor is null then

```

```
        insert into inscricaoP values(new.nick, new.nick, new.data_hora)
        ;
    end if;

    return new;
end
$$
language plpgsql;

create trigger inscricao_insert
after insert on boleia for each row
execute procedure inscricao_proc();
```