

1. PACOTE PL/SQL

.PKS

create or replace PACKAGE pkg_condominio IS

-- Cria um novo registo de proprietário.

```
PROCEDURE regista_proprietario (  
    nif_in      IN proprietario.nif%TYPE,  
    nome_in     IN proprietario.nome%TYPE,  
    genero_in   IN proprietario.genero%TYPE,  
    piso_in     IN proprietario.piso%TYPE,  
    letra_in    IN proprietario.letra%TYPE);
```

/*-----*/

-- Cria o novo registo de um administrador, dados um proprietário do prédio e o ano em que é administrador

```
PROCEDURE regista_administrador(  
    proprietario_in IN administra.proprietario%TYPE,  
    ano_in          IN administra.ano%TYPE);
```

/*-----*/

-- Cria o novo registo de um contrato e devolve o número sequencial do contrato criado

```
FUNCTION regista_contrato(  
    empresa_in      IN contrato.empresa%TYPE,  
    equipamento_in  IN contrato.equipamento%TYPE,  
    ano_in          IN contrato.ano%TYPE,  
    euros_in        IN contrato.euros%TYPE)
```

```
RETURN contrato.numero%TYPE;
```

/*-----*/

-- Cria o novo registo de uma autorizacao de um contrato por parte de um administrador em dado ano

```
PROCEDURE regista_autorizacao(  
    administrador_in IN AUTORIZA.ADMINISTRADOR%TYPE,  
    ano_in IN AUTORIZA.ANO%TYPE,  
    contrato_in IN AUTORIZA.CONTRATO%TYPE);
```

/*-----*/

-- Remove uma autorização dada por um administrador num determinado ano e contrato

```
PROCEDURE remove_autorizacao (  
    administrador_in IN autoriza.administrador%TYPE,  
    ano_in IN autoriza.ano%TYPE,  
    contrato_in IN autoriza.contrato%TYPE);
```

/*-----*/

-- Remove um contrato e as suas autorizações associadas

```
PROCEDURE remove_contrato (  
    numero_in IN contrato.numero%TYPE);
```

/*-----*/

-- Remove o registo de um administrador para um determinado ano, bem como todas as autorizações que realizou nesse ano

```
PROCEDURE remove_administrador (  
    proprietario_in IN administra.proprietario%TYPE,  
    ano_in IN administra.ano%TYPE);
```

/*-----*/

-- Remove uma proprietário, todos os seus registos como administrador e todas as suas autorizações

```
PROCEDURE remove_proprietario (  

```

SIBD, Etapa 2 do projeto 16/12/2016	António Dias, 47811 Catarina Realinho, 50259 Guilherme Lopes, 49020 TP13
--	---

nif_in IN proprietario.nif%TYPE);

END pkg_condominio;

.PKB

create or replace PACKAGE BODY pkg_condominio IS

-- Cria o novo registo de um proprietário

PROCEDURE regista_proprietario (

nif_in IN proprietario.nif%TYPE,

nome_in IN proprietario.nome%TYPE,

genero_in IN proprietario.genero%TYPE,

piso_in IN proprietario.piso%TYPE,

letra_in IN proprietario.letra%TYPE)

IS

BEGIN

INSERT INTO proprietario (nif, nome, genero, piso, letra)

VALUES (nif_in, nome_in, genero_in, piso_in, letra_in);

EXCEPTION

WHEN DUP_VAL_ON_INDEX THEN

RAISE_APPLICATION_ERROR(-20001, 'Só pode ser registado um proprietario por apartamento');

WHEN OTHERS THEN RAISE;

END regista_proprietario;

```
/*-----*/
```

```
-- Cria o novo registo de um administrador, dados um proprietário do prédio e o ano em que é administrador
```

```
PROCEDURE regista_administrador(
```

```
    proprietario_in IN administra.proprietario%TYPE,
```

```
    ano_in          IN administra.ano%TYPE)
```

```
IS
```

```
    prop_counter NUMBER(1) := 0;
```

```
BEGIN
```

```
    SELECT COUNT(A.ano) INTO prop_counter FROM administra A WHERE A.ANO = ano_in; --  
Verifica quantos administradores já existem para o ano inserido
```

```
    IF (prop_counter >= 2) THEN      -- Caso já haja dois ou mais, retorna um erro. Caso contrário,  
faz a inserção
```

```
        RAISE_APPLICATION_ERROR(-20003, -- Código definido pelo programador.
```

```
        'Não podem existir mais do que dois administradores por ano.');
```

```
ELSE
```

```
    INSERT INTO administra (proprietario, ano)
```

```
    VALUES (proprietario_in, ano_in);
```

```
END IF;
```

```
END regista_administrador;
```

```
/*-----*/
```

```
-- Cria o novo registo de um contrato e devolve o número sequencial do contrato criado
```

```
FUNCTION regista_contrato(
```

```
    empresa_in    IN contrato.empresa%TYPE,
```

```
    equipamento_in IN contrato.equipamento%TYPE,
```

```
    ano_in        IN contrato.ano%TYPE,
```

SIBD, Etapa 2 do projeto 16/12/2016	António Dias, 47811 Catarina Realinho, 50259 Guilherme Lopes, 49020 TP13
--	---

euros_in IN contrato.euros%TYPE)

RETURN contrato.numero%TYPE

IS

nContratos_ano NUMBER;

nContrato NUMBER;

BEGIN

SELECT MAX(numero) INTO nContrato

FROM contrato;

IF(nContrato IS NULL) THEN

nContrato := 1;

ELSE

nContrato := nContrato + 1;

END IF;

SELECT COUNT (*) INTO nContratos_ano FROM contrato WHERE ano = ano_in AND
equipamento = equipamento_in;

IF nContratos_ano <> 0 THEN

RAISE_APPLICATION_ERROR(-20666,'Já existe um contrato para este equipamento!');

ELSE

INSERT INTO contrato VALUES (nContrato,empresa_in,equipamento_in,ano_in,euros_in);

END IF;

RETURN nContrato;

END regista_contrato;

/*-----*/

-- Cria o novo registo de uma autorizacao de um contrato por parte de um administrador em
dado ano

PROCEDURE regista_autorizacao(

administrador_in IN AUTORIZA.ADMINISTRADOR%TYPE,

ano_in IN AUTORIZA.ANO%TYPE,

contrato_in IN AUTORIZA.CONTRATO%TYPE)

IS

nif_admin INT := 0;

ano_admin INT := 0;

BEGIN

SELECT DISTINCT A.proprietario, A.ano INTO nif_admin, ano_admin FROM administra A
WHERE A.proprietario = administrador_in AND A.ano = ano_in;

-- O primeiro select faz simultaneamente a verificação da existência do administrador e do par
administrador/ano

ano_admin := 0;

SELECT COUNT (ano) INTO ano_admin FROM contrato C WHERE C.ano = ano_in AND C.numero
= contrato_in;

-- O segundo select verifica se o ano introduzido é igual ao que consta no contrato

IF ano_admin = 0 THEN

RAISE_APPLICATION_ERROR(-20668,'Ano que consta no contrato diferente do ano
indicado!');

ELSE

INSERT INTO autoriza (administrador, ano, contrato)

VALUES (administrador_in, ano_in, contrato_in);

END IF;

EXCEPTION WHEN NO_DATA_FOUND THEN

RAISE_APPLICATION_ERROR(-20667,'Administrador e/ou par administrador/ano
inexistente!');

END regista_autorizacao;

/*-----*/

-- Remove uma autorização dada por um administrador num determinado ano e contrato

PROCEDURE remove_autorizacao (

```
administrador_in IN autoriza.administrador%TYPE,

ano_in IN autoriza.ano%TYPE,

contrato_in IN autoriza.contrato%TYPE)

IS

BEGIN

    DELETE FROM autoriza WHERE (contrato = contrato_in) AND (administrador =
administrador_in) AND (ano = ano_in);

    EXCEPTION

        WHEN OTHERS THEN RAISE;

END remove_autorizacao;

/*-----*/

-- Remove um contrato e as suas autorizações associadas

PROCEDURE remove_contrato (

    numero_in IN contrato.numero%TYPE)

IS

    TYPE tab_local_aut IS TABLE OF autoriza%ROWTYPE; -- Tabela local (nested table) para a
qual se movem todas linhas da tabela autoriza cujo número

    autorizacoes tab_local_aut; -- de contrato é igual ao introduzido.

BEGIN

    SELECT * BULK COLLECT INTO autorizacoes FROM autoriza A WHERE A.contrato =
numero_in;

    IF (autorizacoes.COUNT > 0) THEN -- Loop onde se percorrem todas as linhas da tabela
'autorizacoes', que são apagadas uma a uma (caso existam)

        FOR posicao_atual IN autorizacoes.FIRST .. autorizacoes.LAST LOOP

            pkg_condominio.remove_autorizacao(authorizacoes(posicao_atual).administrador,
autorizacoes(posicao_atual).ano, autorizacoes(posicao_atual).contrato);

        END LOOP;

    END IF;
```

```
DELETE FROM contrato WHERE (numero = numero_in);

EXCEPTION

    WHEN OTHERS THEN RAISE;

END remove_contrato;

/*-----*/

-- Remove o registo de um administrador para um determinado ano, bem como todas as
autorizações que realizou nesse ano

PROCEDURE remove_administrador (

    proprietario_in IN administra.proprietario%TYPE,

    ano_in IN administra.ano%TYPE)

IS

    TYPE tab_local_aut IS TABLE OF autoriza%ROWTYPE;

    autorizacoes tab_local_aut;

BEGIN

    SELECT * BULK COLLECT INTO autorizacoes FROM autoriza A WHERE A.administrador =
proprietario_in AND A.ano = ano_in;

    IF (autorizacoes.COUNT > 0) THEN

        FOR posicao_atual IN autorizacoes.FIRST .. autorizacoes.LAST LOOP

            pkg_condominio.remove_autorizacao(autorizacoes(posicao_atual).administrador,
autorizacoes(posicao_atual).ano, autorizacoes(posicao_atual).contrato);

        END LOOP;

    END IF;
```


SIBD, Etapa 2 do projeto 16/12/2016	António Dias, 47811 Catarina Realinho, 50259 Guilherme Lopes, 49020 TP13
--	---

DELETE FROM administra WHERE (proprietario = proprietario_in) AND (ano_in = ano);

EXCEPTION

WHEN OTHERS THEN RAISE;

END remove_administrador;

/*-----*/

-- Remove uma proprietário, todos os seus registos como administrador e todas as suas autorizações

PROCEDURE remove_proprietario (

nif_in IN proprietario.nif%TYPE)

IS

TYPE tab_local_aut IS TABLE OF autoriza%ROWTYPE;

autorizacoes tab_local_aut;

TYPE tab_local_admin IS TABLE OF administra%ROWTYPE;

administracao tab_local_admin;

BEGIN

SELECT * BULK COLLECT INTO autorizacoes FROM autoriza A WHERE A.administrador = nif_in;

IF (autorizacoes.COUNT > 0) THEN

FOR posicao_atual IN autorizacoes.FIRST .. autorizacoes.LAST LOOP

pkg_condominio.remove_autorizacao(autorizacoes(posicao_atual).administrador,
autorizacoes(posicao_atual).ano, autorizacoes(posicao_atual).contrato);

END LOOP;

END IF;

SIBD, Etapa 2 do projeto 16/12/2016	António Dias, 47811 Catarina Realinho, 50259 Guilherme Lopes, 49020 TP13
--	---

```
SELECT * BULK COLLECT INTO administracao FROM administra A WHERE A.proprietario =  
nif_in;
```

```
IF (administracao.COUNT > 0) THEN
```

```
FOR posicao_atual IN administracao.FIRST .. administracao.LAST LOOP
```

```
    pkg_condominio.remove_administrador(administracao(posicao_atual).proprietario,  
administracao(posicao_atual).ano);
```

```
END LOOP;
```

```
END IF;
```

```
DELETE FROM proprietario WHERE (nif = nif_in);
```

```
EXCEPTION
```

```
WHEN OTHERS THEN RAISE;
```

```
END remove_proprietario;
```

```
END pkg_condominio;
```

2. SCRIPT

```
VARIABLE n_contrato NUMBER;
```

```
BEGIN
```

```
pkg_condominio.regista_proprietario(123456789, 'Gervásio', 'M', 1, 'A');
```

```
pkg_condominio.regista_proprietario(123456788, 'Catarina', 'F', 1, 'B');  
pkg_condominio.regista_proprietario(123456787, 'Guilherme', 'M', 1, 'C');  
pkg_condominio.regista_proprietario(123456786, 'António', 'M', 1, 'D');  
pkg_condominio.regista_proprietario(123456785, 'Pedro', 'M', 1, 'E');  
pkg_condominio.regista_proprietario(123456784, 'Deolinda', 'F', 2, 'A');  
pkg_condominio.regista_proprietario(123456783, 'Matilde', 'F', 3, 'F');  
pkg_condominio.regista_proprietario(123456782, 'Emma', 'F', 2, 'B');  
pkg_condominio.regista_proprietario(123456781, 'Ferreira', 'M', 4, 'A');  
pkg_condominio.regista_proprietario(123456780, 'Abhishek', 'M', 1, 'F');
```

```
pkg_condominio.regista_administrador(123456789, 2016);  
pkg_condominio.regista_administrador(123456789, 2015);  
pkg_condominio.regista_administrador(123456789, 2014);  
pkg_condominio.regista_administrador(123456788, 2016);  
pkg_condominio.regista_administrador(123456788, 2011);  
pkg_condominio.regista_administrador(123456780, 2015);  
pkg_condominio.regista_administrador(123456781, 2012);  
pkg_condominio.regista_administrador(123456780, 2010);  
pkg_condominio.regista_administrador(123456782, 2009);  
pkg_condominio.regista_administrador(123456785, 2010);  
pkg_condominio.regista_administrador(123456784, 2011);
```

```
:n_contrato := pkg_condominio.regista_contrato('Empresa A', 'Extintores', 2016, 5001.04);  
pkg_condominio.regista_autorizacao(123456789, 2016, :n_contrato);
```

```
:n_contrato := pkg_condominio.regista_contrato('Empresa B', 'Extintores', 2015, 3000.00);
```

SIBD, Etapa 2 do projeto 16/12/2016	António Dias, 47811 Catarina Realinho, 50259 Guilherme Lopes, 49020 TP13
--	---

pkg_condominio.regista_autorizacao(123456789, 2015, :n_contrato);

pkg_condominio.regista_autorizacao(123456780, 2015, :n_contrato);

:n_contrato := pkg_condominio.regista_contrato('Empresa B', 'Elevadores', 2015, 4000.55);

pkg_condominio.regista_autorizacao(123456780, 2015, :n_contrato);

:n_contrato := pkg_condominio.regista_contrato('Empresa C', 'Extintores', 2010, 3000.66);

pkg_condominio.regista_autorizacao(123456785, 2010, :n_contrato);

:n_contrato := pkg_condominio.regista_contrato('Empresa D', 'Luzes', 2011, 2000.00);

pkg_condominio.regista_autorizacao(123456784, 2011, :n_contrato);

pkg_condominio.regista_autorizacao(123456788, 2011, :n_contrato);

pkg_condominio.remove_contrato(:n_contrato);

:n_contrato := pkg_condominio.regista_contrato('Empresa E', 'Luzes', 2011, 1999.99);

pkg_condominio.regista_autorizacao(123456784, 2011, :n_contrato);

pkg_condominio.regista_autorizacao(123456788, 2011, :n_contrato);

pkg_condominio.remove_autorizacao(123456788, 2011, :n_contrato);

pkg_condominio.remove_administrador(123456788, 2011);

pkg_condominio.remove_proprietario(123456788);

pkg_condominio.remove_proprietario(123456789);

END;

/

3. INTERROGAÇÕES

1) SELECT P.nif, P.nome, P.piso, P.letra

FROM proprietario P, administra A

WHERE A.ano > 2000

ORDER BY P.piso DESC, P.letra DESC;

2) SELECT P1.nif, P1.nome

FROM proprietario P1, administra A

WHERE (P1.nif = A.proprietario)

AND A.ano BETWEEN 2000 AND 2010

UNION

SELECT P2.nif, P2.nome

FROM proprietario P2

WHERE P2.piso > 5 AND P2.genero = 'F';

3) SELECT C.empresa, A.proprietario

FROM contrato C, administra A

WHERE C.equipamento = 'elevadores'

AND C.ano BETWEEN 2010 AND 2015

AND A.proprietario LIKE 'P%';

4) SELECT A1.proprietario, P.nome

FROM administra A1, proprietario P

WHERE A1.proprietario = P.nif

AND P.genero= 'M'

MINUS

SELECT A2.administrador, P2.nome

FROM autoriza A2, contrato C, proprietario P2

WHERE C.equipamento = 'extintores'

AND C.euros > 5000;

5) SELECT C.empresa, C.equipamento, C.ano, C.euros

FROM contrato C

WHERE NOT EXISTS (SELECT *

FROM administra A

WHERE A.ano = C.ano

AND NOT EXISTS (SELECT *

FROM autoriza A1

WHERE A1.contrato = C.numero

AND A1.administrador = A.proprietario))

ORDER BY C.ano DESC, C.euros DESC, C.empresa ASC, C.equipamento ASC;

6) SELECT A.administrador, P.nome, A.ano, SUM(C.euros) AS total, SUM(C.euros * 0.06) as IVA

FROM autoriza A, proprietario P, contrato C

WHERE A.administrador = P.nif

AND A.contrato = C.numero

GROUP BY A.administrador, P.nome, A.ano

ORDER BY A.administrador ASC, P.nome ASC, A.ano DESC;

```
7) SELECT A.ano, P.nif, P.nome, COUNT (*) AS n_autorizacoes
FROM autoriza A, proprietario P, contrato C
WHERE A.administrador = P.nif
AND A.contrato = C.numero
GROUP BY A.ano, P.nif, P.nome
HAVING (COUNT(A.administrador) >= ALL (SELECT COUNT(*)
FROM contrato C1, autoriza A1
WHERE C1.numero = A1.contrato
AND A1.ano = A.ano
GROUP BY A1.administrador, A1.ano))
ORDER BY A.ano DESC, P.nome ASC, P.nif ASC;
```

```
8) SELECT DISTINCT P.genero,
A.proprietario,
P.nome,
COUNT (A.proprietario) AS n_anos,
MIN (A.ano) AS a_primeiro,
MAX (A.ano) AS a_ultimo
FROM proprietario P, administra A
WHERE A.proprietario = P.nif
GROUP BY P.genero, A.proprietario, P.nome
HAVING (COUNT (A.proprietario) >= ALL (SELECT COUNT(*)
FROM proprietario P1, administra A1
WHERE P1.nif = A1.proprietario
AND P1.genero = P.genero
GROUP BY (P1.genero, P1.nif)));
```