

1º parte – pacote PL/SQL

```
--codigo pks
create or replace PACKAGE MUSEU AS

--criar novo registo do Artista
FUNCTION regista_artista (
    nome_in      IN artista.nome%TYPE,
    ano1_in      IN artista.ano1%TYPE,
    ano2_in      IN artista.ano2%TYPE,
    pais_in      IN artista.pais%TYPE)
RETURN NUMBER;

-- Regista um tipo de arte e descricao (única)
PROCEDURE regista_tipo_arte(
    tipo_in      IN tipo_arte.tipo%TYPE,
    descricao_in IN tipo_arte.descricao%TYPE);

--Regista uma habilidade de um artista num tipo de arte
PROCEDURE regista_habilidade(
    artista_in   IN habilidade.artista%TYPE,
    arte_in      IN habilidade.arte%TYPE,
    preferencia_in IN habilidade.preferencia%TYPE);

--Regista uma obra de arte com um título e o ano da sua autoria
FUNCTION regista_obra_arte(
    titulo_in    IN obra_arte.titulo%TYPE,
    ano_in       IN obra_arte.ano%TYPE)
RETURN NUMBER;

--Regista uma obra de arte com um título e o ano da sua autoria
PROCEDURE regista_autoria(
    obra_in      IN autoria.obra%TYPE,
    artista_in   IN autoria.artista%TYPE,
    arte_in      IN autoria.arte%TYPE);

--Remove os dados de um artista
PROCEDURE remove_artista(
    id_in        IN artista.id%TYPE);

--Remove os dados sobre um tipo de arte
PROCEDURE remove_tipo_arte(
    tipo_in      IN tipo_arte.tipo%TYPE);

-- Remove os dados sobre uma habilidade de um artista
PROCEDURE remove_habilidade(
    artista_in   IN habilidade.artista%TYPE,
    arte_in      IN habilidade.arte%TYPE);

-- Remove os dados sobre uma habilidade de um artista
PROCEDURE remove_obra_arte(
    id_in        IN obra_arte.id%TYPE);
```

```
-- Remove os dados da autoria
PROCEDURE remove_autoria(
  obra_in      IN autoria.obra%TYPE,
  artista_in   IN autoria.artista%TYPE,
  arte_in      IN autoria.arte%TYPE);

END MUSEU;

-----codigo pkb-----
create or replace PACKAGE BODY MUSEU AS

--Regista um artista e devolve o seu id
FUNCTION regista_artista (
  nome_in      IN artista.nome%TYPE,
  ano1_in      IN artista.ano1%TYPE,
  ano2_in      IN artista.ano2%TYPE,
  pais_in      IN artista.pais%TYPE)
RETURN NUMBER IS
  novo_id NUMBER := novo_id_artista.NEXTVAL;
BEGIN
  INSERT INTO artista ( id, nome, ano1, ano2, pais )
    VALUES (novo_id, nome_in, ano1_in, ano2_in, pais_in);

  RETURN novo_id;

--Excepção para o mesmo nome
EXCEPTION
  WHEN DUP_VAL_ON_INDEX THEN
    RAISE_APPLICATION_ERROR(-20000, 'Nome de artista já
existe.');
```

```

--Excepcao ano2 > ano1
IF (ano2_in < ano1_in) THEN
  RAISE_APPLICATION_ERROR(-20001, 'O ano de finalização tem
que ser ' ||
                                'posterior ao ano de
inicialização.');
```

```

  END IF;
END regista_artista;

-----
```

Etapa nº2

```
--Regista um tipo de arte
PROCEDURE regista_tipo_arte(
    tipo_in    IN tipo_arte.tipo%TYPE,
    descricao_in IN tipo_arte.descricao%TYPE) AS
BEGIN
    INSERT INTO tipo_arte (tipo, descricao)
        VALUES (tipo_in, descricao_in);

    --Excepção para mesmo tipo
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            RAISE_APPLICATION_ERROR(-20002, 'Tipo de Arte já existe.');
```

```
END regista_tipo_arte;

-- Regista uma habilidade de um artista num tipo de arte
PROCEDURE regista_habilidade(
    artista_in  IN habilidade.artista%TYPE,
    arte_in     IN habilidade.arte%TYPE,
    preferencia_in IN habilidade.preferencia%TYPE) AS
BEGIN
    INSERT INTO habilidade (artista, arte, preferencia)
        VALUES (artista_in, arte_in, preferencia_in);

    -- Exepçao artista == , habilidade /= , preferencia ==
    EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
            RAISE_APPLICATION_ERROR(-20003, 'O artista já é dotado dessa
habilidade' ||
                                'e/ou com esse nível de
preferencia.');
```

-- Exepcções 1 < preferencia < 7

```
    IF (preferencia_in NOT BETWEEN 1 AND 7) THEN
        RAISE_APPLICATION_ERROR (-20004, 'O nível de preferêcia ten
que ser ' ||
                                'entre 1 e 7.');
```

END IF;

```
END regista_habilidade;
```

Etapa nº2

```
-- -----  
-- Regista uma obra de arte e devolve o seu Id  
FUNCTION regista_obra_arte(  
    titulo_in    IN obra_arte.titulo%TYPE,  
    ano_in       IN obra_arte.ano%TYPE)  
    RETURN NUMBER IS  
    nova_obra NUMBER := novo_obra_id.NEXTVAL;  
BEGIN  
    INSERT INTO obra_arte (id, titulo, ano)  
        VALUES (nova_obra, titulo_in, ano_in);  
    RETURN nova_obra;  
  
    --Excepção Mesmo título  
EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
        RAISE_APPLICATION_ERROR(-20005, 'Esta obra de arte já  
existe.');
```

```
--Excepção 1500 <ano < 2100  
    IF (ano_in NOT BETWEEN 1500 AND 2100) THEN  
        RAISE_APPLICATION_ERROR (-20006, 'O ano tem de ser entre 1500  
e 2100.');
```

```
    END IF;  
END regista_obra_arte;  
  
-- -----  
-- Regista a autoria de uma obra de arte (atribui um artista)  
PROCEDURE regista_autoria(  
    obra_in      IN autoria.obra%TYPE,  
    artista_in   IN autoria.artista%TYPE,  
    arte_in      IN autoria.arte%TYPE) AS  
BEGIN  
    INSERT INTO autoria (obra, artista, arte)  
        VALUES (obra_in, artista_in, arte_in);  
  
    -- Excepção obra==, artista !=  
EXCEPTION  
    WHEN DUP_VAL_ON_INDEX THEN  
        RAISE_APPLICATION_ERROR(-20007, 'Essa obra de arte já foi  
realizada' ||  
                                           'por outro artista');
```

```
--Excepção artista nao habilitado ao tipo de arte  
    WHEN OTHERS THEN  
        RAISE_APPLICATION_ERROR (-20008, 'Artista não está habilitado  
a este tipo de arte ');  
  
END regista_autoria;  
  
-- -----
```

Etapa nº2

```
-- Remove um artista
PROCEDURE remove_artista(
    id_in          IN artista.id%TYPE) AS
BEGIN
    DELETE FROM autoria WHERE ( artista = id_in);
    DELETE FROM habilidade WHERE ( artista = id_in);
    DELETE FROM artista WHERE (id = id_in);

    -- Nenhuma linha foi afetada pelo comando DELETE.
    IF (SQL%ROWCOUNT = 0) THEN
        RAISE_APPLICATION_ERROR(-20009, 'Artista a remover não
existe.');
```

```
END IF;
END remove_artista;

-- Remove um tipo de arte
PROCEDURE remove_tipo_arte(
    tipo_in        IN tipo_arte.tipo%TYPE) AS
BEGIN
    DELETE FROM autoria WHERE (tipo_in = arte) ;
    DELETE FROM habilidade WHERE (tipo_in = arte) ;
    DELETE FROM tipo_arte WHERE (tipo= tipo_in);

    -- Nenhuma linha foi afetada pelo comando DELETE.
    IF (SQL%ROWCOUNT = 0) THEN
        RAISE_APPLICATION_ERROR(-20010, 'O tipo de arte a remover não
existe.');
```

```
END IF;
END remove_tipo_arte;

-- Remove uma habilidade
PROCEDURE remove_habilidade(
    artista_in     IN  habilidade.artista%TYPE,
    arte_in        IN  habilidade.arte%TYPE) AS
BEGIN
    DELETE FROM autoria WHERE (artista = artista_in AND arte =
arte_in);
    DELETE FROM habilidade WHERE ( arte = arte_in );

    -- Nenhuma linha foi afetada pelo comando DELETE.
    IF (SQL%ROWCOUNT = 0) THEN
        RAISE_APPLICATION_ERROR(-20011, 'A habilidade a remover não
existe.');
```

```
END IF;
END remove_habilidade;
```

Etapa nº2

```
-- Remove uma obra de arte
PROCEDURE remove_obra_arte(
  id_in      IN obra_arte.id%TYPE) AS
BEGIN
  DELETE FROM autoria WHERE (id_in = obra);
  DELETE FROM obra_arte WHERE (id = id_in ) ;

  -- Nenhuma linha foi afetada pelo comando DELETE.
  IF (SQL%ROWCOUNT = 0) THEN
    RAISE_APPLICATION_ERROR(-20012, 'A habilidade a remover não
existe.');
```

```
END IF;
END remove_obra_arte;

-- Remove uma autoria
PROCEDURE remove_autoria(
  obra_in      IN autoria.obra%TYPE,
  artista_in   IN autoria.artista%TYPE,
  arte_in      IN autoria.arte%TYPE) AS
BEGIN
  DELETE FROM autoria WHERE (obra_in = obra AND artista_in =
artista AND arte_in = arte);

  -- Nenhuma linha foi afetada pelo comando DELETE.
  IF (SQL%ROWCOUNT = 0) THEN
    RAISE_APPLICATION_ERROR(-20013, 'Este artista não realizou
essa obra.');
```

```
END IF;
END remove_autoria;

END MUSEU;

-- -----sequência auxiliares para id-----

DROP SEQUENCE novo_id_artista;
DROP SEQUENCE novo_obra_id;

-- sequencia que define id do artista
CREATE SEQUENCE novo_id_artista
  INCREMENT BY 1 MINVALUE 1 MAXVALUE 999999 NOCYCLE;
-- sequencia que define id da obra de arte
CREATE SEQUENCE novo_obra_id
  INCREMENT BY 1 MINVALUE 1 MAXVALUE 999999 NOCYCLE;
```

2ª parte - o script que demonstra o uso do pacote PL/SQL num cenário de gestão dos dados

```
-- ----- Regista Artista-----  
  
VARIABLE id_1 NUMBER;  
  BEGIN :id_1 := museu.regista_artista('Xiao', 2010, 2013,  
'China'); END;  
/  
PRINT id_1;  
  
VARIABLE id_2 NUMBER;  
  BEGIN :id_2 := museu.regista_artista('Manuel', 1994, 2002,  
'Portugal'); END;  
/  
PRINT id_2;  
  
VARIABLE id_3 NUMBER;  
  BEGIN :id_3 := museu.regista_artista('Anastácio', 1999, 2017,  
'Portugal'); END;  
/  
PRINT id_3;  
  
VARIABLE id_4 NUMBER;  
  BEGIN :id_4 := museu.regista_artista('Raquel', 1997, 2000,  
'Polónia'); END;  
/  
PRINT id_4;  
  
VARIABLE id_5 NUMBER;  
  BEGIN :id_5 := museu.regista_artista('Fernando', 2000 , 2009,  
'Itália'); END;  
/  
PRINT id_5;  
  
VARIABLE id_6 NUMBER;  
  BEGIN :id_6 := museu.regista_artista('Stromea', 2003 , 2009,  
'França'); END;  
/  
PRINT id_6;  
  
  
-- Excepção MESMO NOME  
-- museu.regista_artista('Daniel', 2000 , 2009, 'Portugal');  
  
-- Excepções 1500 < ano1 && ano2 < 2100  
-- museu.regista_artista('Dan ', 1300 , 2009, 'Portugal');  
-- museu.regista_artista('Duarte ', 1560 , 2145, 'Portugal');  
  
-- Excepção ano2 > ano1  
-- museu.regista_artista('Duarte ', 2014 , 1560, 'Portugal');
```

```
-- -----Regista tipo_arte-----
BEGIN
museu.regista_tipo_arte('Pintura', 'Abstrato');
museu.regista_tipo_arte('Escultura', 'Bustos');
museu.regista_tipo_arte('Fotografia', 'Black AND White');
museu.regista_tipo_arte('origami', 'arte de dobrar papel');
museu.regista_tipo_arte('Teatro', 'revista');
museu.regista_tipo_arte('Música', 'Pop');
END ;
/

--Excepção MESMO TIPO
-- museu.regista_tipo_arte('Fotografia', 'BW');

--Excepção mesma descrição
-- museu.regista_tipo_arte('F', 'Black AND White');

-- ----- Regista Habilidade -----
BEGIN
MUSEU.REGISTA_HABILIDADE( 3, 'Pintura', 5);      -- Anastácio
MUSEU.REGISTA_HABILIDADE( 3, 'Fotografia', 4);    -- Anastacio
MUSEU.REGISTA_HABILIDADE( 2, 'Escultura', 6);     -- Manuel
MUSEU.REGISTA_HABILIDADE( 4, 'Escultura', 1);     -- Raquel
MUSEU.REGISTA_HABILIDADE( 1, 'Fotografia', 7);    -- Xiao
MUSEU.REGISTA_HABILIDADE( 1, 'Teatro', 6);       -- Xiao
MUSEU.REGISTA_HABILIDADE( 1, 'Pintura', 5);      -- Xiao
MUSEU.REGISTA_HABILIDADE( 5, 'Fotografia', 7);    -- Fernando
MUSEU.REGISTA_HABILIDADE( 6, 'Música', 5);       -- Stromea
END ;
/

-- Excepção artista == , habilidade ==, preferencia /=
-- MUSEU.REGISTA_HABILIDADE( 3, 'Pintura', 3);

-- Exepção artista == , habilidade /= , preferencia ==
-- MUSEU.REGISTA_HABILIDADE( 3, 'Foto', 4); --Anastacio

-- Exepções 1 < preferencia < 7
-- MUSEU.REGISTA_HABILIDADE( 3, 'maquilhagem', 9);

-- Excepção tipo de arte nao existe
-- MUSEU.REGISTA_HABILIDADE( 3, 'Mercedes', 2); --Anastacio
```


Etapa nº2

```
-- ----- Regista obra_arte -----  
Variable obra1 NUMBER;  
  BEGIN :obra1 := museu.regista_obra_arte('Grito',1999);  End;  
  /  
  Print obra1;  
  
Variable obra2 NUMBER;  
  BEGIN :obra2 := museu.regista_obra_arte('Favela',1998);  End;  
  /  
  Print obra2;  
  
Variable obra3 NUMBER;  
  BEGIN :obra3 := museu.regista_obra_arte('Somebody',2000);  End;  
  /  
  Print obra3;  
  
Variable obra4 NUMBER;  
  BEGIN :obra4 := museu.regista_obra_arte('cancro',1969);  End;  
  /  
  Print obra4;  
  
Variable obra5 NUMBER;  
  BEGIN :obra5 := museu.regista_obra_arte('olho',1990);  End;  
  /  
  Print obra5;  
  
Variable obra6 NUMBER;  
  BEGIN :obra6 := museu.regista_obra_arte('muita tinta',2016);  
End;  
  /  
  Print obra6;  
  
Variable obra NUMBER;  
  BEGIN :obra := museu.regista_obra_arte('espirro',2016);  End;  
  /  
  Print obra;  
  
--Excepção Mesmo título  
-- museu.regista_obra_arte('Grito', 1893);  
  
--Excepção 1500 <ano < 2100  
-- museu.regista_obra_arte('Menina bonita', 1450);  
-- museu.regista_obra_arte('Espelho', 2300);
```

Etapa nº2

```
-- ----- Regista autoria -----  
Begin  
  museu.regista_autoria(1,3 , 'Pintura'); -- Grito, foi feito pelo  
anastácio  
  museu.regista_autoria(3,5, 'Fotografia'); -- Somebody foi feito  
pelo Fernando  
  museu.regista_autoria(7,3 , 'Pintura'); -- Grito, foi feito pelo  
anastácio  
  museu.regista_autoria(6,3 , 'Pintura'); -- Muita tinta, foi feito  
pelo anastácio  
  museu.regista_autoria(2,4 , 'Escultura');-- Raquel, esculpiu,  
favela  
  museu.regista_autoria(4,6 , 'Música');-- Stromea tocou cancro.  
  
  -- Excepção obra==, artista /=  
  -- museu.regista_autoria(4,2 , 'Pintura'); -- Grito. foi feito  
pelo manuel  
  --Excepção artista nao habilitado ao tipo de arte  
  -- museu.regista_autoria(4,3 , 'Escultura');-- Anastacio nao sabe  
escultura  
END;  
/  
  
-- -----  
-- ----- SELECT -----
```

3º parte – As interrogações

```
-- select nº1 : ID e nome dos artistas que são escultores ou  
pintores,  
--o ano em que iniciaram actividade e o seu país de origem.  
SELECT A.id, A.nome, A.ano1, A.pais  
  FROM artista A, habilidade H  
  WHERE (A.id = H.artista)  
        AND (H.arte = 'Escultura' OR H.arte = 'Pintura')  
 ORDER BY A.AN01, A.pais ASC;  
  
-- select nº2: ID e nome dos artistas que praticam ou dança ou  
arte digital, ou  
-- sejam portuguesa, ou tenham começado depois de 98.  
SELECT DISTINCT A.id,A.nome  
  FROM artista A, habilidade H  
  WHERE ((A.pais = 'Portugal')  
        AND (A.ano1 > 1998))  
        OR (A.id = H.artista  
        AND (H.arte = 'danca'  
        OR H.arte = 'arte_digital'));
```

Etapa nº2

```
-- select nº3: Títulos das obras realizadas no início do milénio
por um
--artista italiano com nome começado por 'F'
SELECT O.titulo
FROM obra_arte O, artista A, autoria S
WHERE (O.ano BETWEEN 2000 AND 2010)
AND (S.obra = O.id)
AND (S.artista = A.id)
AND (A.pais = 'Itália')
AND (A.nome LIKE 'F%');

-- select nº4 : ID e nome dos artistas portugueses, começaram
depois 74,
-- e que nunca realizaram obras depois de 2000
SELECT A.id, A.nome
FROM artista A
WHERE (A.pais = 'Portugal')
AND (A.ano1 > 1974)
AND (NOT EXISTS (SELECT S.obra
FROM obra_arte O, autoria S
WHERE (S.artista = A.id)
AND (S.obra = O.id)
AND (O.ano > 2000)
)
);

-- select nº5 : Títulos e anos das obras que foram feitas pelos
artistas com
--habilidade em música,e nível de preferência superior a 3.
SELECT O.titulo, O.ano
FROM obra_arte O, autoria S
WHERE (S.obra = O.id)
AND (S.artista = (SELECT H.artista
FROM habilidade H
WHERE (H.arte = 'Música')
AND (H.preferencia > 3)
)
)
ORDER BY O.ano DESC, O.titulo ASC;

--Falta testar todos os casos
-- select nº6 : Número de obras de arte realizadas por cada
pintor, em cada ano
SELECT A.id, A.nome, O.ano, COUNT (O.id)
FROM artista A, obra_arte O, autoria S, habilidade H
WHERE (A.id = S.artista)
AND (O.id = S.obra)
AND (A.id = H.artista)
AND (H.arte = 'Pintura')
GROUP BY O.ANO, A.id, A.nome
ORDER BY A.nome ASC, O.ano DESC;
```

```
-- select nº7 : ID e nome dos artistas com mais obras para cada ano
SELECT O.ano, A.id, A.nome
FROM artista A, obra_arte O LEFT JOIN autoria S
                                ON O.id = S.obra
WHERE A.id = S.artista
      AND o.id = s.obra
GROUP BY A.id, A.nome, O.ano
HAVING (COUNT (S.obra) >= ALL (SELECT COUNT(S1.obra)
                                FROM autoria S1, obra_arte O1
                                WHERE (O1.id = S1.obra)
                                AND (O1.ano = O.ano)
                                GROUP BY S1.artista, O1.ano))

ORDER BY O.ano;
```

```
-- select nº8 : ID e nome dos artistas que realizaram menos de 2
SELECT A.id , A.nome
      FROM artista A LEFT JOIN autoria S
                                ON A.id = S.artista
      GROUP BY A.id, A.nome
HAVING ( COUNT( DISTINCT S.obra) < 2);
```

```
-- select nº9 : Para cada país de origem, o ID e nome do artista
que tem mais
--habilidades, dizendo qual a habilidade preferida e a menos
preferida
SELECT DISTINCT A.pais, A.id, A.nome, H.arte, H.preferencia
FROM artista A, habilidade H, (SELECT H1.artista,
                                MAX(H1.preferencia) as ma,
                                MIN(H1.preferencia) as mi,
                                COUNT(H1.arte) as n_hab
                                FROM habilidade H1
                                GROUP BY H1.artista) P

WHERE (A.id = H.artista)
      AND (A.id = P.artista)
      AND (P.n_hab >= ALL (SELECT COUNT(H2.arte)
                          FROM habilidade H2, artista A2
                          WHERE (H2.artista = A2.id)
                          AND (A.pais = A2.pais)
                          GROUP BY H2.artista
                          ))
      AND ((H.preferencia = P.ma)
            OR (H.preferencia = P.mi))
GROUP BY A.pais, A.id, A.nome, H.arte, H.preferencia
ORDER BY A.pais, A.id;
```