Elementos do Grupo 33: António Rodrigues n.º40853, José Albuquerque n.º 40251, Ricardo Veloso n.º 44842

Especificação do pacote PL/SQL

```
CREATE OR REPLACE PACKAGE formula_e
IS

PROCEDURE regista_equipa(
        sigla_in IN Equipa.sigla%TYPE,
        nome_in IN Equipa.nome%TYPE ,
        pais_in IN Equipa.pais%TYPE,
        fundacao_in IN Equipa.fundacao%TYPE );

FUNCTION regista_piloto(
        nome_in IN Piloto.nome%TYPE,
        pais_in IN Piloto.pais%TYPE,
        nascimento_in IN Piloto.nascimento%TYPE,
        genero_in IN Piloto.genero%TYPE,
        equipa_in IN Piloto.equipa%TYPE)
RETURN NUMBER;

PROCEDURE regista_corrida(
        ordem_in IN Corrida.ordem%TYPE,
        cidade_in IN Corrida.cidade%TYPE,
        pais_in IN Corrida.pais%TYPE,
        voltas_in IN Corrida.voltas%TYPE);

PROCEDURE regista_participacao(
        piloto_in IN Participa.piloto%TYPE,
        corrida_in IN Participa.corrida%TYPE,
        pontos_in IN Participa.pontos%TYPE);

PROCEDURE remove_equipa(
        nome_in IN Equipa.nome%TYPE);

PROCEDURE remove_piloto(
        id_in IN Piloto.id%TYPE);

PROCEDURE remove_corrida(
        ordem_in IN Corrida.ordem%TYPE);

PROCEDURE remove_participacao(
        piloto_in IN Participa.piloto%TYPE,
        corrida_in IN Participa.corrida%TYPE);

FUNCTION posicao(
        pontos_in IN Participa.pontos%TYPE)
RETURN NUMBER;

END formula_e;
/
```

Elementos do Grupo 33: António Rodrigues n.º40853, José Albuquerque n.º 40251, Ricardo Veloso n.º 44842

Código do pacote PL/SQL

```
CREATE OR REPLACE PACKAGE BODY formula_e IS

-- Exceção - Mensagem
-- -20001 - O número máximo de equipas é 10.
-- -20002 - Já existe uma equipa com essa sigla.
-- -20003 - O número máximo de pilotos por equipa é 2.
-- -20004 - O máximo de corridas é 10.
-- -20005 - Já existe uma corrida nessa cidade desse pai.
-- -20006 - A corrida não existe.
-- -20007 - Estes pontos já foram atribuídos nessa corrida.
-- -20008 - O piloto não existe.
-- -20009 - Já foram atribuídos pontos a esse piloto.
-- -20011 - Não existe esse registo.
-- -20012 - A equipa não existe.
-- -20013 - Não existe essa pontuação.


-- Cria um novo registo de equipa.

PROCEDURE regista_equipa (
        sigla_in    IN equipa.sigla%TYPE,
        nome_in     IN equipa.nome%TYPE,
        pais_in     IN equipa.pais%TYPE,
        fundacao_in IN equipa.fundacao%TYPE)
IS
        numeroEquipas NUMBER :=0;
BEGIN
        SELECT COUNT (*) INTO numeroEquipas
        FROM Equipa;

        IF (numeroEquipas = 10) THEN
                RAISE_APPLICATION_ERROR (-20001, 'O número máximo de equipas é
                10.');
        ELSE
                INSERT INTO equipa(sigla, nome, pais, fundacao)
                        VALUES (sigla_in, nome_in, pais_in, fundacao_in );
        END IF;

EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
                RAISE_APPLICATION_ERROR (-20002, 'Já existe uma equipa com essa
                sigla.');
        WHEN OTHERS THEN RAISE;
END regista_equipa;
```

```
FUNCTION regista_piloto(
        nome_in IN Piloto.nome%TYPE,
        pais_in IN Piloto.pais%TYPE,
        nascimento_in IN Piloto.nascimento%TYPE,
        genero_in IN Piloto.genero%TYPE,
        equipa_in IN Piloto.equipa%TYPE)
        RETURN NUMBER
IS
        numero_Elementos_Equipa NUMBER;
        idPiloto NUMBER;

BEGIN
        -- Verificação do número de elementos da equipa a inserir
        SELECT COUNT(*) INTO numero_Elementos_Equipa
        FROM Piloto P, Equipa E
        WHERE P.equipa = E.sigla AND E.sigla = equipa_in;

        IF (numero_Elementos_Equipa = 2) THEN
                RAISE_APPLICATION_ERROR (-20003, 'O número máximo de pilotos por
                equipa é 2.');
        ELSE
                idPiloto:=seq_id.NEXTVAL;
                INSERT INTO piloto (id, nome, pais, nascimento, genero, equipa)
                        VALUES (idPiloto, nome_in, pais_in, nascimento_in, genero_in,
                        equipa_in);
        END IF;

        RETURN idPiloto;

EXCEPTION
        WHEN OTHERS THEN RAISE;

END regista_piloto;
```

Elementos do Grupo 33: António Rodrigues n.º40853, José Albuquerque n.º 40251, Ricardo Veloso n.º 44842

```
PROCEDURE regista_corrida(
        ordem_in IN Corrida.ordem%TYPE,
        cidade_in IN Corrida.cidade%TYPE,
        pais_in IN Corrida.pais%TYPE,
        voltas_in IN Corrida.voltas%TYPE)
IS
        total_corridas NUMBER;

BEGIN
        SELECT COUNT(*) INTO total_corridas
        FROM Corrida;

        IF (total_corridas = 10) THEN
                RAISE_APPLICATION_ERROR (-20004, 'O máximo de corridas é 10.');

        ELSE
                INSERT INTO Corrida (ordem, cidade, pais, voltas)
                        VALUES (ordem_in,cidade_in, pais_in,voltas_in);
        END IF;

EXCEPTION
        WHEN DUP_VAL_ON_INDEX THEN
                RAISE_APPLICATION_ERROR (-20005, 'Já existe uma corrida nessa
                        cidade desse país');

        WHEN OTHERS THEN RAISE;

END regista_corrida;
```

```
PROCEDURE regista_participacao(
        piloto_in IN Participa.piloto%TYPE,
        corrida_in IN Participa.corrida%TYPE,
        pontos_in IN Participa.pontos%TYPE)
IS
        verifica_piloto  Piloto.id%TYPE;
        verifica_corrida NUMBER;
        verifica_pontos NUMBER;
 BEGIN
        -- Verifica se existe esse piloto (Exceção NO_DATA_FOUND)
        SELECT Piloto.id INTO verifica_piloto
        FROM Piloto
        WHERE Piloto.id = piloto_in;

        -- Verifica se existe corrida
        SELECT COUNT(*) INTO verifica_corrida
        FROM Corrida
        WHERE Corrida.ordem = corrida_in;

        -- Verifica se já foram atribuídos estes pontos (>0)
        SELECT COUNT(*) INTO verifica_pontos
        FROM Participa
        WHERE Participa.corrida = corrida_in AND Participa.pontos = pontos_in;

        IF (verifica_corrida = 0) THEN
                RAISE_APPLICATION_ERROR (-20006, 'A corrida não existe.');
        END IF;

        IF (verifica_pontos = 0 OR pontos_in = 0 ) THEN
                INSERT INTO Participa (piloto, corrida, pontos)
                        VALUES (piloto_in,corrida_in, pontos_in);
        ELSE
                RAISE_APPLICATION_ERROR (-20007, 'Estes pontos já foram atribuídos
                nessa corrida.');
        END IF;

EXCEPTION
        WHEN NO_DATA_FOUND THEN
                -- Caso o Select não tenha devolvido nada
                RAISE_APPLICATION_ERROR (-20008, 'O piloto não existe.');

        WHEN DUP_VAL_ON_INDEX THEN
                RAISE_APPLICATION_ERROR (-20009, 'Já foram atribuídos pontos a esse
                piloto');

        WHEN OTHERS THEN RAISE;
END regista_participacao;
```

```
PROCEDURE remove_equipa(
        nome_in IN Equipa.nome%TYPE)
IS
        verifica_equipa Equipa.sigla%TYPE;

        CURSOR cursor_pilotos IS SELECT P.id FROM Piloto P, Equipa E WHERE P.equipa =
        E.sigla AND E.nome = nome_in;
        TYPE tab_local_pilotos_da_Equipa_in IS TABLE OF cursor_pilotos%ROWTYPE;

        tabPilotos tab_local_pilotos_da_Equipa_in;

BEGIN
        SELECT E.sigla INTO verifica_equipa
        FROM Equipa E
        WHERE E.nome = nome_in;

        OPEN cursor_pilotos;
        FETCH cursor_pilotos BULK COLLECT INTO tabPilotos;
        CLOSE cursor_pilotos;

        IF (tabPilotos IS NOT EMPTY)THEN
                FOR posicao_atual IN tabPilotos.FIRST .. tabPilotos.LAST LOOP
                        formula_e.remove_piloto(tabPilotos(posicao_atual).id);
                END LOOP;
        END IF;

        DELETE FROM Equipa WHERE sigla = verifica_equipa;

EXCEPTION
        WHEN NO_DATA_FOUND THEN
                -- Caso o Select não tenha devolvido nada
                 RAISE_APPLICATION_ERROR (-20012, 'A equipa não existe.');

        WHEN OTHERS THEN
                BEGIN
                    -- Libertação de recursos (se aplicável).
                        IF (cursor_pilotos%ISOPEN) THEN
                                CLOSE cursor_pilotos;
                        END IF;

                        RAISE;
                END;
END remove_equipa;
```

```
PROCEDURE remove_piloto(
        id_in Piloto.id%Type)
IS
        verificaPiloto Piloto.id%TYPE;

        CURSOR cursor_participacoes IS
                SELECT piloto, corrida FROM PARTICIPA WHERE (piloto=id_in);
        TYPE tabela_local_part_piloto IS TABLE OF cursor_participacoes%ROWTYPE;

        tabParPiloto tabela_local_part_piloto;

BEGIN
        SELECT piloto.id INTO verificaPiloto
        FROM Piloto
        WHERE piloto.id=id_in;

        OPEN cursor_participacoes;
        FETCH cursor_participacoes BULK COLLECT INTO tabParPiloto;
        CLOSE cursor_participacoes;

        IF (tabParPiloto IS NOT EMPTY) THEN
                FOR i IN tabParPiloto.first .. tabParPiloto.last LOOP
                        formula_e.remove_participacao(tabParPiloto(i).piloto,
                                tabParPiloto(i).corrida);
                END LOOP;
        END IF;

        DELETE FROM Piloto WHERE (id_in=Piloto.id);

EXCEPTION
        WHEN NO_DATA_FOUND THEN
                --caso o Select não tenha devolvido nada
                 RAISE_APPLICATION_ERROR (-20008, 'O Piloto não existe.');
        WHEN OTHERS THEN
                BEGIN
                -- Libertação de recursos (se aplicável).
                        IF (cursor_participacoes%ISOPEN) THEN
                                CLOSE cursor_participacoes;
                        END IF;

                        RAISE;
                END;
END remove_piloto;
```

```
PROCEDURE remove_participacao(
        piloto_in IN Participa.piloto%TYPE,
        corrida_in IN Participa.corrida%TYPE)

IS
        verifica_piloto_corrida NUMBER;

BEGIN
        SELECT COUNT (*) INTO verifica_piloto_corrida
        FROM PARTICIPA
        WHERE participa.piloto = piloto_in
        AND participa.corrida = corrida_in;

        IF (verifica_piloto_corrida = 0) THEN
                RAISE_APPLICATION_ERROR (-20011, 'Não existe esse registo');
        ELSE
                DELETE FROM PARTICIPA WHERE participa.piloto=piloto_in
                        AND participa.corrida = corrida_in;
        END IF;

EXCEPTION
        WHEN OTHERS THEN RAISE;
END remove_participacao;
```

```
PROCEDURE remove_corrida(
        ordem_in IN Corrida.ordem%TYPE)

IS
        verificaCorrida corrida.ordem%TYPE;

        -- Selecionar da tabela PARTICIPA todos os pilotos que tem participação
        -- na corrida com ordem_in
        CURSOR cursor_part_na_Corrida IS
                SELECT piloto FROM PARTICIPA WHERE (ordem_in = PARTICIPA.corrida);
        TYPE tabela_part_piloto_corrida IS TABLE OF cursor_part_na_Corrida%ROWTYPE;

        tabParPilotoCorrida tabela_part_piloto_corrida;

 BEGIN
        SELECT ordem INTO verificaCorrida
        FROM Corrida
        WHERE corrida.ordem = ordem_in;

        OPEN cursor_part_na_Corrida;
        FETCH cursor_part_na_Corrida BULK COLLECT INTO tabParPilotoCorrida;
        CLOSE cursor_part_na_Corrida;

        -- Se corrida tiver participações registadas, apaga-as
        IF (tabParPilotoCorrida IS NOT EMPTY) THEN
                FOR i IN tabParPilotoCorrida.FIRST .. tabParPilotoCorrida.last LOOP
                        formula_e.remove_participacao(
                                tabParPilotoCorrida(i).piloto, ordem_in);
                END LOOP;
        END IF;

        DELETE FROM Corrida WHERE (ordem_in = Corrida.ordem);

EXCEPTION
        WHEN NO_DATA_FOUND THEN
                -- Caso o Select nao tenha devolvido nada
                RAISE_APPLICATION_ERROR (-20006, 'A corrida não existe');
        WHEN OTHERS THEN
                BEGIN
                   -- Libertação de recursos (se aplicável).
                        IF (cursor_part_na_Corrida%ISOPEN) THEN
                                CLOSE cursor_part_na_Corrida;
                        END IF;
                        RAISE;
                END;
END remove_corrida;
```

```
FUNCTION posicao(
        pontos_in IN Participa.pontos%TYPE)
        RETURN NUMBER
IS
        classificacao NUMBER;

BEGIN
        CASE pontos_in
                WHEN 25 THEN
                        classificacao := 1;
                WHEN 18 THEN
                        classificacao := 2;
                WHEN 15 THEN
                         classificacao := 3;
                WHEN 12 THEN
                        classificacao := 4;
                 WHEN 10 THEN
                         classificacao := 5;
                WHEN 8 THEN
                        classificacao := 6;
                WHEN 6 THEN
                         classificacao := 7;
                WHEN 4 THEN
                        classificacao := 8;
                WHEN 2 THEN
                        classificacao := 9;
                WHEN 1 THEN
                        classificacao := 10;
                WHEN 0 THEN
                        classificacao := NULL;
                        DBMS_OUTPUT.PUT_LINE ('Classificação na corrida superior a 10');
                ELSE
                        RAISE_APPLICATION_ERROR (-20013, 'Não existe essa pontuação');
        END CASE;

        RETURN classificacao;

EXCEPTION
        WHEN OTHERS THEN RAISE;
END posicao;

END formula_e;
/
```

Elementos do Grupo 33: António Rodrigues n.º40853, José Albuquerque n.º 40251, Ricardo Veloso n.º 44842

Script com comandos de invocação das funções e procedimentos

```
CREATE SEQUENCE seq_id INCREMENT BY 1 MAXVALUE 9999;

-- Inserção da equipa recorrendo a funcao regista_equipa
BEGIN
	formula_e.regista_equipa('AG', 'Amlin Aguri', 'Japao', 2014);
	formula_e.regista_equipa('AF', 'Andretti Formula E', 'USA', 2014);
	formula_e.regista_equipa('AS', 'Audi Sport ABT', 'Alemanha', 2014);
	formula_e.regista_equipa('CR', 'China Racing', 'China', 2014);
	formula_e.regista_equipa('DR', 'Dragon Racing', 'USA', 2014);
END;

-- Tentativa de inserção de uma equipa com sigla já registada
-- Lançamento de uma exceção -20002
BEGIN
	formula_e.regista_equipa('AG', 'Amlin Aguri', 'Japao', 2014);
END;

BEGIN
	formula_e.regista_equipa('ER', 'e.dams-Renault', 'Franca', 2014);
	formula_e.regista_equipa('Tr', 'Trulli', 'Suica', 2014);
	formula_e.regista_equipa('Ve', 'Venturi', 'Monaco', 2014);
	formula_e.regista_equipa('VR', 'Virgin Racing', 'UK', 2014);
	formula_e.regista_equipa('MR', 'Mahindra Racing', 'India', 2014);
END;

-- Tentativa de inserção de uma 11ª equipa na tabela
-- Lançamento de uma exceção -20001
BEGIN
	formula_e.regista_equipa('FR', 'Ferrari', 'Italia', 1900);
END;

-- Visualização do registo de equipas
SELECT * FROM EQUIPA;

-- Inserção de um piloto recorrendo à funcão regista_piloto
VARIABLE idNumber NUMBER;

BEGIN
	:idNumber:=formula_e.regista_piloto('Katherine Legge', 'UK', 1980, 'F', 'AG');
	DBMS_OUTPUT.PUT_LINE(:idNumber);
	:idNumber:=formula_e.regista_piloto('Felix Costa', 'Portugal', 1991, 'M', 'AG');
	DBMS_OUTPUT.PUT_LINE(:idNumber);
END;
```

```
-- Tentativa de inserção de um terceiro piloto numa equipa
-- Lançamento de uma exceção -20003.
BEGIN
	:idNumber:=formula_e.regista_piloto('Costa Felix', 'Portugal', 1991, 'M', 'AG');
END;

BEGIN
	:idNumber:=formula_e.regista_piloto('Franck Montagny', 'Franca', 1978, 'M', 'AF');
	:idNumber:=formula_e.regista_piloto('Charles Pic', 'França', 1990, 'M', 'AF');
	:idNumber:=formula_e.regista_piloto('Lucas di Grassi', 'Brasil', 1984, 'M', 'AS');
	:idNumber:=formula_e.regista_piloto('Daniel Abt', 'Alemanha', 1992, 'M', 'AS');
	:idNumber:=formula_e.regista_piloto('Nelson Piquet', 'Brasil', 1985, 'M', 'CR');
	:idNumber:=formula_e.regista_piloto('Ho-Pin Tung', 'China', 1982, 'M', 'CR');
	:idNumber:=formula_e.regista_piloto('Oriol Servia', 'Espanha', 1974, 'M', 'DR');
	:idNumber:=formula_e.regista_piloto('Jerome DAmbrosio', 'Belgica', 1985, 'M', 'DR');
END;

-- Visualização do registo de pilotos
SELECT * FROM PILOTO;

-- Registo de corridas
BEGIN
	formula_e.regista_corrida(1, 'Pequim', 'China', 20);
	formula_e.regista_corrida(2, 'Putrajaya', 'Malasia', 12);
	formula_e.regista_corrida(3, 'Punta del Este', 'Uruguai', 20);
	formula_e.regista_corrida(4, 'Buenos Aires', 'Argentina', 12);
	formula_e.regista_corrida(5, 'Miami', 'USA', 30);
END;

-- Visualização da tabela de corrida
SELECT * FROM CORRIDA;

-- Tentativa de inserção de uma corrida num local já existente
-- Lançamento de uma exceção -20005.
BEGIN
	formula_e.regista_corrida(6, 'Miami', 'USA', 26);
END;

-- Registo de corridas
BEGIN
	formula_e.regista_corrida(6, 'Monte Carlo', 'Monaco', 30);
	formula_e.regista_corrida(7, 'Berlim', 'Alemanha', 30);
	formula_e.regista_corrida(8, 'Londres', 'UK', 30);
	formula_e.regista_corrida(9, 'Long Beach', 'USA', 30);
	formula_e.regista_corrida(10, 'Madrid', 'Espanha', 30);
END;
```

```
-- Visualização da tabela de corrida
SELECT * FROM Corrida;

-- Tentativa de inserção de uma 11ª corrida
-- Lançamento de uma exceção -20004.
BEGIN
	formula_e.regista_corrida(11, 'Lisboa', 'Portugal', 30);
END;

-- Inserção de registos de participação do piloto 1
BEGIN
	formula_e.regista_participacao(1,1,0);
	formula_e.regista_participacao(1,2,0);
	formula_e.regista_participacao(1,3,1);
	formula_e.regista_participacao(1,4,1);
	formula_e.regista_participacao(1,5,2);
END;

-- Inserção de registos de participação do piloto 2
BEGIN
	formula_e.regista_participacao(2,1,15);
	formula_e.regista_participacao(2,2,4);
	formula_e.regista_participacao(2,3,6);
	formula_e.regista_participacao(2,4,6);
	formula_e.regista_participacao(2,5,4);
END;

-- Inserção de registos de participação do piloto 3
BEGIN
	formula_e.regista_participacao(3,1,18);
	formula_e.regista_participacao(3,2,10);
	formula_e.regista_participacao(3,3,12);
	formula_e.regista_participacao(3,4,8);
	formula_e.regista_participacao(3,5,6);
END;

-- Inserção de registos de participação do piloto 4
BEGIN
	formula_e.regista_participacao(4,1,0);
	formula_e.regista_participacao(4,2,0);
	formula_e.regista_participacao(4,3,0);
	formula_e.regista_participacao(4,4,0);
	formula_e.regista_participacao(4,5,0);
END;
```

```
-- Inserção de registos de participação do piloto 5
BEGIN
        formula_e.regista_participacao(5,1,0);
        formula_e.regista_participacao(5,2,0);
        formula_e.regista_participacao(5,3,0);
        formula_e.regista_participacao(5,4,0);
        formula_e.regista_participacao(5,5,0);
END;

-- Verificação das alterações da tabela Participa
SELECT P.nome, PART.pontos, COR.cidade, PART.corrida, PART.piloto
FROM Piloto P, Participa PART, Corrida COR
WHERE COR.ordem = PART.corrida
AND P.id = PART.piloto;

-- Tentativa de atribuição de pontos já existentes
-- Lançamento de uma exceção -20007
BEGIN
        formula_e.regista_participacao(5,1,15);
END;

-- Tentativa de atribuição de pontos a um piloto não existente
-- Lançamento de uma exceção -20008
BEGIN
        formula_e.regista_participacao(13,5,25);
END;

-- Tentativa de atribuição de pontos a um piloto numa corrida que não existe
-- Lançamento de uma exceção -20006
BEGIN
        formula_e.regista_participacao(3,11,25);
END;

-- Tentativa de atribuição de um novo registo a um piloto que já foi previamente registado
-- Lançamento de uma exceção 20009
BEGIN
        formula_e.regista_participacao(2,1,0);
END;

-- Visualização da tabela das equipas antes da tentativa de remoção de uma delas
SELECT * FROM Equipa;

-- Remoção de uma equipa sem pilotos e sem participações
BEGIN
        formula_e.remove_equipa('Venturi');
END;
```

```
-- Remoção de uma equipa com pilotos e sem participações
BEGIN
	formula_e.remove_equipa('Dragon Racing');
END;

-- Remoção de uma equipa com pilotos que têm participações
BEGIN
	formula_e.remove_equipa('Amlin Aguri');
END;

-- Visualização da tabela das equipas
SELECT * FROM Equipa;

-- Visualização da tabela dos pilotos
SELECT * FROM Piloto;

-- Visualização da tabela das participações
SELECT * FROM Participa;

-- Remoção de um piloto que não tem participações
BEGIN
	formula_e.remove_piloto(8);
END;

-- Tentativa de remoção de um piloto não existente
-- Lançamento de uma exceção -20008
BEGIN
	formula_e.remove_piloto(20);
END;

-- Remoção de um piloto que tem participações
BEGIN
	formula_e.remove_piloto(3);
END;

-- Remoção de uma participação
BEGIN
	formula_e.remove_participacao(4,5);
END;

-- Tentativa de remoção de uma participação não existente
-- Lançamento de uma exceção -20011
BEGIN
	formula_e.remove_participacao(6,1);
END;
```

```
-- Remoção de uma corrida
BEGIN
        formula_e.remove_corrida(1);
END;

-- Remoção de uma corrida que não tem participações
BEGIN
        formula_e.remove_corrida(10);
END;

-- Tentativa de remoção de uma corrida não existente
-- Lançamento de uma exceção -20006
BEGIN
        formula_e.remove_corrida(10);
END;

-- Retorno da posição de um piloto, consoante os pontos
-- Resultado esperado: 1
VARIABLE classf NUMBER;

BEGIN
        :classf:=formula_e.posicao(25);
        DBMS_OUTPUT.PUT_LINE (:classf);
END;

-- Retorno da posição de um piloto, consoante os pontos
-- Resultado esperado: NULL com impressão de uma mensagem
BEGIN
        :classf:=formula_e.posicao(0);
        DBMS_OUTPUT.PUT_LINE (:classf);
END;

-- Obtenção de uma classificação para uma pontuação inexistente
-- Lançamento de uma exceção -20013
BEGIN
        :classf:=formula_e.posicao(50);
        DBMS_OUTPUT.PUT_LINE (:classf);
END;
```

Elementos do Grupo 33: António Rodrigues n.º40853, José Albuquerque n.º 40251, Ricardo Veloso n.º 44842

Interrogações SQL

-- 1. Anos de fundação e nomes das equipas do Japão com pelo menos um piloto nascido após 1990.
-- Nota 1: pretende-se uma interrogação com apenas um SELECT, isto é, sem sub-interrogações.
-- Nota 2: o resultado deve vir ordenado pelo ano de fundação de forma descendente e pelo nome da equipa de forma ascendente.

```
SELECT DISTINCT E.fundacao, E.nome
FROM equipa E, piloto P
WHERE E.sigla = P.equipa
AND E.pais = 'Japao'
AND P.nascimento > 1990
ORDER BY E.fundacao DESC, E.nome ASC;
```

-- 2. IDs e nomes das mulheres piloto que nasceram após 1990, ou então que participaram em pelo menos uma corrida e nasceram antes de 1980. Nota: pode usar construtores de conjuntos.

```
SELECT DISTINCT P.id, P.nome
FROM piloto P, Participa Part
WHERE P.genero = 'F'
AND ((P.nascimento > 1990) OR (P.nascimento < 1980 AND Part.piloto = P.id));
```

-- 3. Nomes dos pilotos e suas equipas, nascidos depois de 1990, e que pontuaram nas duas primeiras corridas da temporada.
-- Nota 1: pretende-se uma interrogação sem sub-interrogacões.
-- Nota 2: o resultado deve vir ordenado pelo nome do piloto e pelo nome da equipa.

```
SELECT DISTINCT P.nome, E.nome
FROM Piloto P, Participa Part, Participa Part2, Equipa E
WHERE P.nascimento > 1990
AND P.id = Part.piloto
AND P.equipa = E.sigla
AND Part.corrida = 1
AND Part2.corrida = 2
AND Part.pontos > 0
AND Part2.pontos > 0
ORDER BY P.nome ASC, E.nome ASC;
```

-- 4. Nomes dos pilotos e suas equipas que nunca participaram em corridas no país da equipa.

```
SELECT P.nome, E.nome
FROM Piloto P, Equipa E
WHERE P.id NOT IN (SELECT P.id
                   FROM Piloto P, Participa Part, Equipa E, Corrida C
                   WHERE P.equipa = E.sigla
                   AND P.id = Part.piloto
                   AND Part.corrida = C.ordem
                   AND C.pais = E.pais)
AND P.equipa = E.sigla;
```

-- 5. Nomes das equipas cujos pilotos (todos) participaram em pelo menos uma corrida com mais de 20 voltas ao circuito.

```
SELECT E.nome
FROM Participa PART, Piloto P, Equipa E, Corrida COR
WHERE P.equipa = E.sigla
AND COR.ordem = PART.corrida
AND PART.piloto = P.id
AND COR.voltas > 20
GROUP BY E.nome
HAVING (COUNT (PART.piloto)>=2);
```

-- 6. Soma dos pontos obtidos por cada equipa em cada corrida.
-- Nota: o resultado deve vir ordenado pelo nome da equipa e pelo número de ordem da corrida, ambos de forma ascendente.

```
SELECT E.nome, PART.corrida, SUM(Part.pontos) AS Pontos
FROM Equipa E, Piloto P, Participa PART, Corrida COR
WHERE E.sigla = P.equipa
AND PART.piloto = P.id
AND COR.ordem = PART.corrida
GROUP BY E.nome, PART.corrida
ORDER BY E.nome ASC, COR.ordem ASC;
```

-- 7. Nomes das equipas com mais pontos por corrida.

```
SELECT B.equipa, A.corrida, A.Max_Pontos
FROM (SELECT CORRIDA, MAX (pontos) As Max_Pontos
        FROM (SELECT E.nome AS EQUIPA, PART.corrida, SUM(PART.pontos) AS PONTOS
                FROM Equipa E, Piloto P, Participa PART, Corrida COR
                WHERE E.sigla = P.equipa
                AND PART.piloto = P.id AND COR.ordem = PART.corrida
                GROUP BY E.nome, PART.corrida
                ORDER BY E.nome ASC, COR.ordem ASC)
                GROUP BY CORRIDA
                ORDER BY CORRIDA) A, (SELECT E.nome AS EQUIPA, PART.corrida, SUM
                (PART.pontos) AS PONTOS
                                        FROM Equipa E, Piloto P, Participa PART, Corrida COR
                                        WHERE E.sigla = P.equipa
                                        AND PART.piloto = P.id AND COR.ordem = PART.corrida
                                        GROUP BY E.nome, PART.corrida
                                        ORDER BY E.nome ASC, COR.ordem ASC) B
                WHERE B.corrida = A.corrida AND B.pontos = A.Max_Pontos;
```

-- 8. Média dos pontos obtidos por pilotos que nasceram depois de 1990 e que participaram em todas as corridas do seu país.

```
SELECT P.nome, AVG (Part.pontos) AS MEDIA_PONTOS
FROM Piloto P, Participa Part
WHERE P.nascimento > 1990
AND Part.piloto = P.id
AND NOT EXISTS (SELECT C.ordem
                        FROM Corrida C, Participa Part2
                        WHERE Part2.corrida = C.ordem
                        AND C.pais = P.pais
                        AND Part2.corrida NOT IN (SELECT Part3.corrida
                                                        FROM Participa Part3
                                                        WHERE Part3.piloto = P.id))
GROUP BY P.nome;
```

Elementos do Grupo 33: António Rodrigues n.º40853, José Albuquerque n.º 40251, Ricardo Veloso n.º 44842

-- Extra:

-- 9. Crie duas vistas, chamadas tabela_classificativa_pilotos e tabela_classificativa_equipas
-- com os pontos acumulados pelos pilotos e equipas nas várias corridas da temporada.

-- TABELA_CLASSIFICATIVA_PILOTOS

```
CREATE OR REPLACE VIEW tabela_classificativa_pilotos AS
SELECT P.nome, NVL (SUM (PART.PONTOS), 0) AS PONTOS
FROM Piloto P LEFT OUTER JOIN Participa PART ON P.id = PART.piloto
GROUP BY P.nome, P.id
ORDER BY PONTOS DESC;
```

-- TABELA_CLASSIFICATIVA_EQUIPAS

```
CREATE OR REPLACE VIEW tabela_classificativa_equipas AS
SELECT E.nome, NVL (SUM (PART.PONTOS), 0) AS PONTOS
FROM Piloto P LEFT OUTER JOIN Participa PART ON PART.piloto = P.id, Equipa E
WHERE E.sigla = P.equipa
GROUP BY E.nome
ORDER BY PONTOS DESC;
```