

## **Parte Prática**

**DURAÇÃO 1 HORA e 30 MINUTOS  
(TOLERÂNCIA - 15 MINUTOS)**

Leia a prova com atenção. Seja claro(a) e conciso(a). Responda só ao que se pergunta.

### **Grupo I**

(5,5 valores)

#### **Responda a cada questão numa folha separada**

1. Considere o seguinte problema:

Os responsáveis do Rali ISEP-Dakar, decidiram construir uma base de dados para armazenar informação acerca das provas realizadas ao longo dos anos, sendo cada prova identificada pelo ano. Consideremos, então, os seguintes requisitos.

- No ISEP-Dakar participam várias equipas, cada uma delas tendo um número único de identificação, um nome e uma nacionalidade. Cada equipa pode ser de um (e apenas um) de três tipos, tipos esses que na prática correspondem às três competições existentes no Rali, a saber: há equipas de motos, equipas de carros, e equipas de camiões.

Independentemente do tipo, todas as equipas têm um piloto. Com efeito, cada equipa corre com um veículo (moto, carro ou camião, consoante o tipo de equipa), que é de uma marca e de um tipo (e.g. a equipa portuguesa “Renault Truck/Trifene 200” corre com um camião marca “Renault” e tipo “Kerax”).

É bom não esquecer que, por exemplo, uma equipa de motos não pode correr com um veículo que é um camião. Para além disso, as equipas de carros têm um co-piloto e as de camiões têm um ou dois co-pilotos.

- De cada uma das pessoas participantes na prova (pilotos ou co-pilotos) existe necessidade de armazenar alguma informação relevante, como por exemplo, a nacionalidade. A nacionalidade numa equipa não é completamente independente das nacionalidades dos seus membros (pilotos e co-pilotos). Nada obriga a que todos os membros numa equipa sejam da mesma nacionalidade. Mas a nacionalidade da equipa tem que ser uma (qualquer) das nacionalidades dos seus membros.

Relativamente às pessoas participantes, há que não esquecer que cada uma só pode ser de uma equipa (não faz sentido alguém, por exemplo, ser piloto ou co-piloto numa equipa e simultaneamente piloto ou co-piloto de outra!).

- O Rali possui um conjunto de etapas a serem realizadas pelas diferentes equipas. Cada etapa é realizada em determinados dias, tendo início numa determinada localidade e contém vários troços.

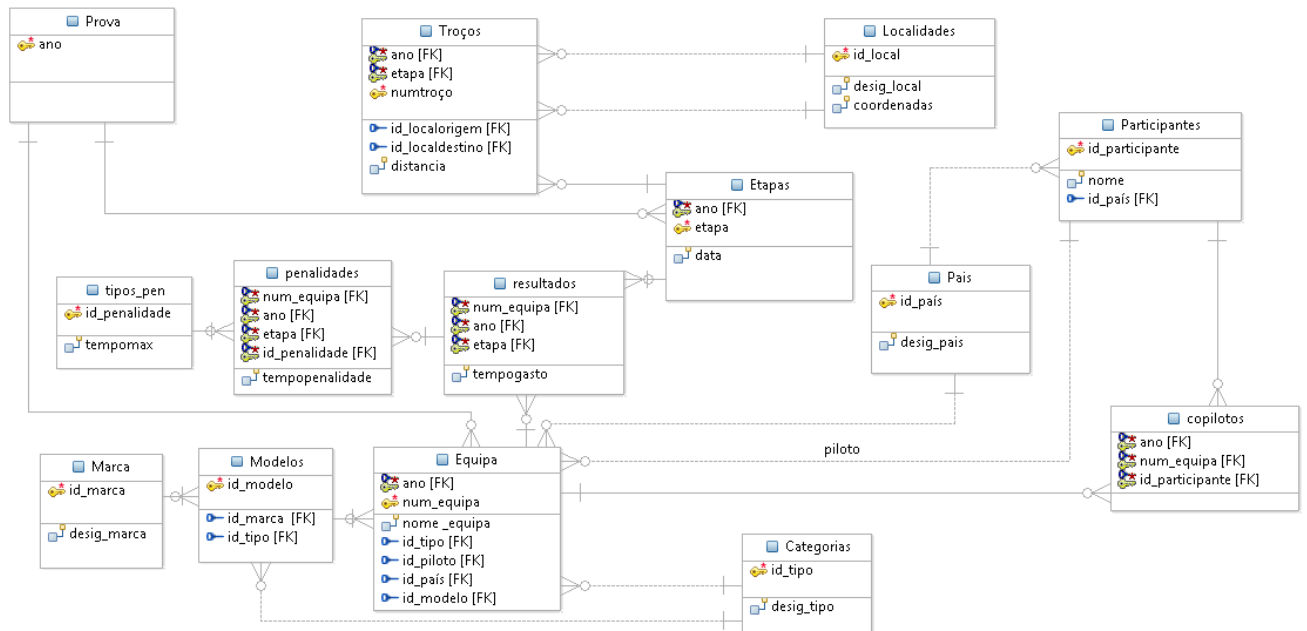
Por exemplo, a etapa 11 que parte de Kayes e chega a Bamako (ambos no Mali), tem 3 troços: um primeiro de 51 km, um segundo de 23 km e um terceiro de 424 km, todos eles com locais de início e de fim identificados por coordenadas.

Ainda relativamente ao Rali ISEP-Dakar interessa saber o tempo e a classificação que cada equipa obteve em cada etapa realizada, e as eventuais penalizações que tiveram (tipo de penalização e tempo de penalização). As penalizações são tipificadas numa tabela com um descritivo e valor de penalização máxima (tempo em minutos).

Com base nos requisitos descritos,

- a) [4 valores]: Esboce um modelo de dados normalizado explicitando as principais entidades, atributos mais significativos e atributos chave, o tipo de relações e respectivas cardinalidades entre as entidades.

**Nota:** não se esqueça de mencionar as restrições de integridade, se houver alguma, que seja impossível de representar no desenho do modelo.



- Uma equipa só pode participar numa categoria e o tipo de veiculo tem que ser dessa categoria . As equipas de motos não têm co-piloto, as de carros têm apenas um co-piloto e as de camiões têm um ou dois co-pilotos.
  - A nacionalidade da equipa tem que ser uma (qualquer) das nacionalidades dos seus membros.
  - Numa prova cada participante só pode ser de uma equipa
- b) [1,5 valores]: ]: Escreva um comando SQL que permita listar as equipas do rali de 2015 (nome da equipa, categoria, nome do piloto, marca do veiculo, tempo gasto) que não tiveram mais do que uma penalização em toda a prova.

```

Select E.nome_equipa, P.nome, C.desig_tipo, MC.desig_marca, SUM (R.tempogasto)
From Equipa E , Participantes P, Modelos MD, Marca MC, Categorias C, Resultados R
Where E.id_piloto=P.id_participante and E.id_modelo=MD.id_modelo and MD.id_marca=MC.id_marca
and E.id_tipo=C.id_tipo and R.num_equipa=E.num_equipa and R.ano=E. ano and E.ano= 2015
and E.num_equipa not in ( select num_equipa from Penalidades where ano=2015 group by
num_equipa having count(*)>1)
group by E.nome_equipa, P.nome, C.desig_tipo, MC.desig_marca

```

**Grupo II**  
(10 valores)

**Responda a todas as questões numa folha separada**

Considere o seguinte esquema relacional

Schema (**IdEsq**, Descr)

IDESQ	DESCR
1	BDDAD2012
2	BDDAD2013
3	BDDAD2014
4	BDDAD2015

Schema\_Tabela (**IdEsq**, **IdTab**, Nome, Descr, dataCriação)

IDESQ	IDTAB	NOME	DESCR	DATA CRIAÇÃO
1	1	TB-ALUNO	DADOS ALUNOS	16.01.07
1	2	TB-CURSO	DADOS CURSOS	16.01.07
2	3	TB-DISC	DADOS DISC	16.01.10
2	4	TB-PROF	DADOS DOCENTES	16.01.10
4	1	TB-ALUNO	DADOS ALUNOS	16.01.11
4	3	TB-DISC	DADOS DISC	16.01.11

Tipo\_Índice(**IdTipoi**, Descricao)

IDTIPOI	DESCRICAO
1	hashing
2	primario
3	arvore B+

Índice (**IdEsq**, **IdTab**, **IdInd**, NomeÍndice, **IdTipoi**, AscDes)

IDESQ	IDTAB	IDIND	NOMEÍNDICE	IDTIPOI	ASCDES
1	1	1	índice	1	A
1	1	2	índice2	2	A
2	3	1	índice3	2	A
2	4	2	índice4	1	A

Atributo (**IdEsq**, **IdTab**, **IdAtr**, Nome, IdTipo, Chave)

IDESQ	IDTAB	IDATR	NOME	IDTIPO	CHAVE
1	1	1	codaluno	number	S
1	1	2	nome	varchar	N
1	1	3	morada	varchar	N
1	2	1	codcurso	integer	S
1	2	2	designacao	varchar	N
2	3	1	sigla	char	S
2	3	2	nomedisc	varchar	N
2	3	3	creditos	integer	N

Atributo\_Índice (**IdEsq**, **IdInd**, **IdTab**, **IdAtr**, Posição)

IDESQ	IDIND	IDTAB	IDATR	POSICAO
1	1	1	1	1
1	1	1	2	1
1	2	1	3	3
2	1	3	2	1
2	2	4	1	1
2	2	4	2	3
2	2	4	3	2

com as chaves primárias sublinhadas e as chaves estrangeiras em itálico.

Suponha que se verificam as seguintes restrições de integridade:

- R1-** o atributo AscDes indica se o índice é ascendente ou descendente no conjunto de atributos que o compõem. Ele só pode tomar valores de A ou D;
- R2-** o atributo Descr é um atributo de comprimento variável de 10 caracteres.
- R3-** o nome de qualquer tabela, tem o tamanho obrigatório de 8 caracteres e inicia-se sempre por TB, possuindo na 3ª posição o carácter "-".
- R4-** a dataCriação é por omissão a data do sistema.

1. [1 valores]: Escreva os comandos SQL que permitem criar as tabelas Schema\_Tabela e Índice, respeitando as restrições enunciadas.

```
CREATE TABLE SCHEMA_Tabela(  
    IdEsq number(3) NOT NULL,  
    IdTab number(3) NOT NULL,  
    Nome char(8),  
    Descr varchar2(10),  
    datacriacao date DEFAULT sysdate,  
    CONSTRAINT CK_tabela_nome CHECK (Nome LIKE 'TB-____'),  
    CONSTRAINT PK_schema_tabela PRIMARY KEY (IdEsq, IdTab),  
    CONSTRAINT FK_schema_tabela FOREIGN KEY (IdEsq) REFERENCES schema(IdEsq));  
  
CREATE TABLE Indice(  
    IdEsq number(3) NOT NULL,  
    IdTab number(3) NOT NULL,  
    IdInd number(3) NOT NULL,  
    NomeIndice varchar2(10),  
    IdTipol varchar2(5),  
    AscDesc char(1),  
    CONSTRAINT CK_Indice_ascdesc CHECK (AscDesc LIKE 'A' or AscDesc LIKE 'D'),  
    CONSTRAINT FK_Indice_IdEsq_IdTab FOREIGN KEY (IdEsq,IdTab) REFERENCES  
        SCHEMA_Tabela(IdEsq, IdTab),  
    CONSTRAINT FK_Indice_IdTipol FOREIGN KEY (IdTipol) REFERENCES  
        TipoIndice(IdTipol),  
    CONSTRAINT PK_Indice PRIMARY KEY (IdEsq, IdTab, IdInd));
```

2. Escreva o comando SQL que:

- a) [1,5 valores]: Liste a descrição do schema e o nome da tabela que possui o menor número de atributos.

```
SELECT S.Descr,ST.Nome  
FROM Atributo A, SCHEMA1 S, SCHEMA1_TABELA ST  
WHERE S.Idesq = ST.Idesq AND ST.IdTab = A.IdTab AND ST.IdEsq = A.IdEsq  
GROUP BY S.Descr,ST.Nome  
HAVING COUNT(Idatr) = (SELECT MIN (contador) FROM (SELECT COUNT(IdAtr)  
as contador  
FROM Atributo A  
GROUP BY IdEsq, IdTab));
```

- b) [1,5 valores]: Liste os idEsq, os id\_Tab e a data de criação das tabelas que possuem todos os tipos de índices.

```
SELECT ST.Idesq, ST.IdTab, ST.dataCriação  
FROM SCHEMA1_Tabela ST  
WHERE NOT EXISTS (  
    SELECT ti.idtipoi  
    FROM TIPOINDICE ti
```

```

WHERE NOT EXISTS (
    SELECT l.idesq, l.idtab
    FROM indice i
    WHERE i.idtipoi = ti.idtipoi AND i.idtab = st.idtab AND
           i.idesq = st.idesq));

```

c) [1,5 valores]: Liste o nome dos schemas onde não são criadas tabelas há mais de um mês.

```

SELECT s.Descr
FROM Schema s, Schema_Tabela st
WHERE s.IdEsq = st.IdEsq
AND s.IdEsq NOT IN (SELECT IdEsq
                    FROM Schema_Tabela
                    WHERE TRUNC(dataCriação) > TRUNC(SYSDATE - 30);

```

Ou

```

SELECT s.Descr
FROM Schema s
WHERE NOT EXISTS (SELECT *
                  FROM Schema_Tabela
                  WHERE IdEsq = s.IdEsq
                  AND TRUNC(dataCriação) > TRUNC(SYSDATE - 30);

```

d) [1,5 valores]: Liste para cada schema, o nome das tabelas e o nome dos respectivos atributos, de todas as tabelas que tenham mais do que 5 atributos do tipo *integer*. A lista deverá estar ordenada por ordem crescente pelo nome da tabela e por ordem decrescente do nome do atributo.

```

SELECT s.Descr, st.Nome, a.Nome
FROM Schema s, Schema_Tabela st, Atributo a
WHERE s.IdEsq = st.IdEsq
AND st.IdEsq = a.IdEsq
AND st.IdTab = a.IdTab
AND a.IdTab IN (SELECT IdTab
                FROM Atributo
                WHERE IdEsq = st.IdEsq
                AND IdTipo = 'integer'
                GROUP BY IdTab
                HAVING COUNT(*) > 5);

```

3. [1,5 valores]: Escreva um trigger que impeça de inserir ou atualizar os atributos numa tabela que possua simultaneamente atributos do tipo *varchar* e chave primária definida.

```

CREATE OR REPLACE TRIGGER trg_atributos
BEFORE insert or UPDATE on Atributo
FOR EACH ROW
DECLARE NrChave number(1);
        NrAtrib number(10);
BEGIN
        SELECT count(1) into NrChave
        FROM Indice, TipoIndice
        WHERE Indice.idTab = :new.IdTab AND Indice.IDTipol = TipoIndice.IdTipol
                AND TipoIndice.Descricao = 'primario';
        IF NrChave > 0 then
                Select count(1) into NrAtrib
                from Atributo
                Where Atributo.IdTab= :new.IdTab and Atributo.IdTipo = 'varchar';
        IF NrAtrib > 0 THEN
                RAISE_APPLICATION_ERROR (-20001,'n,o pode...');
        END IF;
        END IF;
END trg_atributos;

```

4. [1,5 valores]: Escreva um procedimento que receba como parâmetro um determinado schema e permita eliminar todos os índices do tipo “árvore B+”, de todas as tabelas que pertencem somente a esse schema. Caso não exista nenhuma tabela que verifique estas condições, deverá ser levantada uma exceção. Deverá também apresentar o programa de PL/SQL que chame o procedimento criado.

```

create or replace procedure elimina_Indices(esq indice.IDEsq%TYPE) as
Begin
delete from indice
where IDTipol in (Select IdTipol
                from TipoIndice
                where Descricao = 'arvore B+')
and IDEsq = esq;
        IF SQL%ROWCOUNT = 0 THEN
                RAISE_APPLICATION_ERROR (-20001,'n,o existe...');
        END IF;
end;
--- para executar
begin
        elimina_Indices(1);
end;

```

