

Número:

Nome:

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

Parte Teórica

DURAÇÃO 20 MINUTOS

Grupo I

(5 valores)

1. **[1,5 valores]:** Indique, nas alíneas seguintes, quais as **V(Verdadeiras)** e as **F(Falsas)**. Cada alínea **certa** tem a cotação de **0,25 valores**. Por cada alínea **errada** é atribuído **-1 valores**.
 - a) A integridade para uma transação significa que a execução de uma transação de forma isolada (sem haver concorrência de outra transação) preserva a consistência da base de dados. **V**
 - b) A utilização de *locks* exclusivos e partilhados garante a geração de escalonamentos *conflict-serializable*. **F**
 - c) Um grafo de precedências acíclico corresponde a um escalonamento que não pode resultar em *deadlock*. **F**
 - d) A fase de *REDO* do algoritmo de recuperação *ARIES* inicia-se na localização correspondente ao último *checkpoint*. **F**
 - e) Num índice de árvore B^+ os nós intermédios contêm apontadores para as páginas onde se situam os registos. **F**
 - f) O problema da atualização temporária (ou *dirty read*) acontece quando uma transação lê um item duas vezes e o valor desse item é alterado por outra transação entre essas duas leituras. **F**
2. Considere a relação Auditor, na qual se encontra registada a informação sobre 40,000 auditores.

Nome	Tipo	NIF	Rua	CódigoPostal	Localidade
ISEP	Empresa	503123455	Rua São Tomé	4000-485	Porto
Mário José	Particular	504567987	Av. Aliados	4000-065	Porto
Ana Sofia	Particular	504786887	Av. Aliados	4000-065	Porto
XPTO	Empresa	512444555	Rua São Tomé	1100-561	Lisboa
....

Assuma que o atributo NIF é a chave primária da relação, que podem existir ruas com o mesmo nome em diferentes localidades, que cada código postal corresponde a uma localidade e que todos os atributos são atómicos.

- a) **[1 valores]:** Tendo em conta o significado dos atributos, indique a razão pela qual esta relação não está na 3FN.

Existe uma dependência funcional (transitiva) entre código Postal e localidade

CódigoPostal -> Localidade

b) **[1 valores]**: O SGBD Oracle possui diferentes tipos de índices, nomeadamente os índices do tipo *BITMAP*. Diga, se para otimizar a consulta abaixo indicada, será vantajoso criar um índice *BITMAP* sobre o atributo Tipo. Justifique, convenientemente, a sua resposta.

```
SELECT COUNT(*) FROM AUDITOR WHERE Tipo = 'Particular';
```

"O índice bitmap é um tipo de índice que pode ser utilizado para otimizar consultas que utilizam como filtro de dados, colunas que possuem baixa cardinalidade."

Como o atributo tipo é um atributo de baixa cardinalidade é vantajoso criar o índice.

3. **[1,5 valores]**: Considere a relação Livro (CodLivro, TituloLivro, AutorLivro, PalChave) em que CodLivro é a chave primária.

Relativamente a esta relação suponha que se encontra ordenada pela chave primária e que possui 60,000 registos em que o comprimento de cada atributo é, respetivamente, 10, 40, 30 e 20 bytes. Suponha, ainda, que o ficheiro se encontra guardado num disco com blocos de 1024 bytes e que o apontador para os blocos possui o tamanho de 6 bytes.

Determine quantos acessos diretos a blocos faria se acesse diretamente ao ficheiro e quantos acessos faria se acesse por via da utilização de um índice primário, quando se efetua uma pesquisa sobre o atributo chave.

Slides da aula teórica nr7

Parte Prática**DURAÇÃO 1 HORA e 45 MINUTOS****Grupo II**

(6 valores)

Responda numa folha separada (só uma)**2. Considere o seguinte problema:**

Uma empresa decidiu construir uma base de dados para armazenar informação sobre a programação de várias emissoras de rádio com base nos seguintes requisitos.

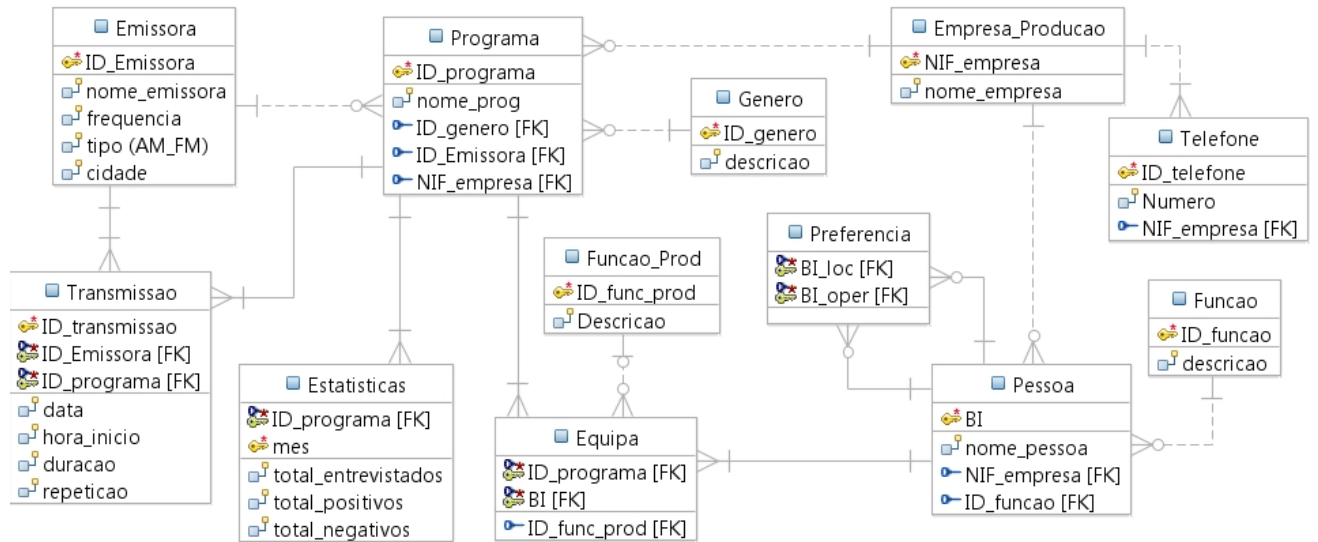
1. Existe um conjunto de emissoras de rádio, caracterizadas por: nome, frequência de transmissão, tipo de transmissão (AM-FM) e cidade de localização.
2. Cada emissora de rádio, emite vários programas que são identificados por um nome e um género (jornalístico, musical, desporto, cultural, etc).
3. Os programas podem ser transmitidos por diferentes emissoras de rádio.
4. Cada emissora de rádio pode emitir cada programa mais do que uma vez. Para cada emissão regista-se: a data, a hora de início, a duração e se é uma repetição ou não.
5. Sobre cada programa sabe-se que é produzido por um único consórcio.
6. Cada consórcio é composto por uma das emissoras de rádio que o transmite e uma empresa de produção.
7. Das empresas de produção conhece-se o seu número de contribuinte, nome, e os seus telefones.
8. Em cada empresa de produção trabalham pessoas relacionadas diretamente com os programas, das quais se conhece o número do bilhete de identidade e o seu nome.
9. Estas pessoas podem ser locutores, operadores ou produtores.
10. As empresas de produção exigem exclusividade aos seus funcionários.
11. Para cada programa interessa registar quem são os seus locutores, os produtores e os operadores.
12. Os produtores podem exercer a função de produção jornalística ou comercial em cada programa que trabalham.
13. Existem locutores que têm preferência em trabalhar com alguns operadores.
14. Cada programa tem pelo menos um operador, um locutor e um produtor, mas pode ter várias pessoas para cada função.
15. Ainda relativamente aos programas emitidos interessa saber, mensalmente, o número total de entrevistados, o número total de pessoas que gostam do programa e o número total de pessoas que não gostam do programa.

Com base nos requisitos descritos,

- 2.1. **[4,5 valores]:** Esboce um modelo de dados normalizado explicitando as principais entidades e os atributos mais significativos. Deve ser claro no modelo quais são os atributos que constituem as chaves primárias e as chaves estrangeiras assim como as relações entre as entidades e respectivas cardinalidades.

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

Uma possível solução



2.2. **[1,5 valores]:** Escreva um comando SQL que permita listar o nome das emissoras de rádio que emitiram todos os programas da empresa de produção “XPTO”.

```

select e.nome_emissora from emissora e
where not exists(select p.id_programa
                from programa p, empresa_producao ep
                where p.nif_empresa = ep.nif_empresa and ep.nome_empresa = 'XPTO'
                minus
                select p.id_programa
                from transmissao t, programa p, empresa_producao ep
                where t.id_programa = p.id_programa and p.nif_empresa = ep.nif_empresa
                and ep.nome_empresa = 'XPTO'
                and t.id_emissora = e.id_emissora);

```

ou

Worksheet

Query Builder

1

2

3

4

5

6






7

8

9

```
select e.nome_emissora from emissora e
where not exists (select * from programa p, empresa_producao ep
where p.nif_empresa = ep.nif_empresa and ep.nome_empresa = 'XPTO'
and p.id_programa not in (select p.id_programa
from transmissao t, programa p, empresa_producao ep
where t.id_programa = p.id_programa and p.nif_empresa = ep.nif_empresa
and ep.nome_empresa = 'XPTO'
and t.id_emissora = e.id_emissora));
```

Script Output x

Task completed in 0,73 seconds

NOME_EMISSORA

emissora1

emissora2

Grupo III
(9 valores)

Responda a cada questão em folhas separadas

Deverá entregar 3 folhas, mesmo que não responda a todas as questões.

3. Considere o seguinte esquema de uma base de dados relacional, que tem como objectivo a gestão do registo de cruzeiros efectuados por empresas.

Porto = { CodPorto, NomePorto, Pais }

Cruzeiro = { NrCruzeiro, NomeCruzeiro, Empresa, Preco }

Itinerario = { NrCruzeiro, CodPorto, NrParagem }

GuiaTuristico = { CodGuia, Nome, dataNascimento }

ServicoGuia = { Data, CodGuia, NrCruzeiro, CodPorto }

3.1. [3 valores]: Escreva um procedimento em PL/SQL que receba como parâmetros de entrada um itinerário (assumido como existente) e uma data. O procedimento deve inserir os dados na tabela ServicoGuia atribuindo o primeiro guia turístico livre por ordem alfabética (assume-se que não há guias turísticos com o mesmo nome). O procedimento deve lançar uma exceção caso não exista nenhum guia turístico disponível para essa data.

```
create procedure proc_3_1(p_nrcruzeiro itinerario.nrcruzeiro%type, p_codigoporto
itinerario.codigoporto%type, p_data date) is
cursor c is
  select codguia from guia where codguia not in (select codguia from service_guia where data =
p_data) order by nome;
begin
  open c;
  fetch c into r;
  if c%found then
    insert into servico_guia(data, codguia, nrcruzeiro, codigoporto) values (p_data, r.codguia,
p_nrcruzeiro, p_codigoporto);
  else
    raise_application_error(-20000, '...');
  end if;
  close c;
end;
```

3.2. **[3 valores]**: Escreva uma função em PL/SQL que receba como parâmetro de entrada o código de um determinado guia e retorne a maior diferença de dias entre duas datas consecutivas do respectivo Serviço de Guia (tabela ServicoGuia). A função deverá retornar NULL no caso do parâmetro fornecido ser NULL ou no caso de um código de guia não existente. **Deverá usar cursores explícitos.**

```
create or replace FUNCTION func_maior_diferenca(p_cod_guia servico_guia.cod_guia%TYPE)
RETURN NUMBER
AS
CURSOR c IS SELECT DISTINCT data FROM servico_guia WHERE cod_guia=p_cod_guia
ORDER BY 1 DESC;
v_data_1  DATE;
v_data_2  DATE;
v_dif     NUMBER :=0;
v_dif_max NUMBER :=0;
v_cont    NUMBER;
BEGIN
OPEN c
FETCH c INTO v_data_1;
LOOP
    FETCH c INTO v_data_2;
    EXIT WHEN c%NOTFOUND;
    v_dif := v_data_1 - v_data_2;
    IF v_dif > v_dif_max THEN
        v_dif_max := v_dif;
    END IF;
    v_data_2 := v_data_1;
END LOOP;
v_cont := c%ROWCOUNT;
CLOSE c;
    IF v_cont>1 THEN
        RETURN v_dif_max;
    END IF;
RETURN NULL;
END;
```

3.3. **[3 valores]:** Escreva um trigger em PLSQL que não permita a atualização do preço de cruzeiros que não façam escala em portos da Grécia e cujo preço seja inferior a 1000.

```
CREATE OR REPLACE TRIGGER trg_imped_atual_preco_cruzeiro
BEFORE UPDATE OF preco ON cruzeiro
FOR EACH ROW WHEN (OLD.preco<1000)
DECLARE
cont  NUMBER;
BEGIN
    SELECT COUNT(*) INTO cont
    FROM itinerario I, porto P
    WHERE I.cod_porto=P.cod_porto AND I.nr_cruzeiro=:OLD.nr_cruzeiro AND
UPPER(P.pais)='GRECIA';
IF (cont=0) THEN
RAISE_APPLICATION_ERROR(-20100, 'AtualizaÁ,,o do preÁo do cruzeiro ' || :OLD.nome_cruzeiro || '
È inv·lida.' );
END IF;
END;

/
```