

Nome –

Assinatura –

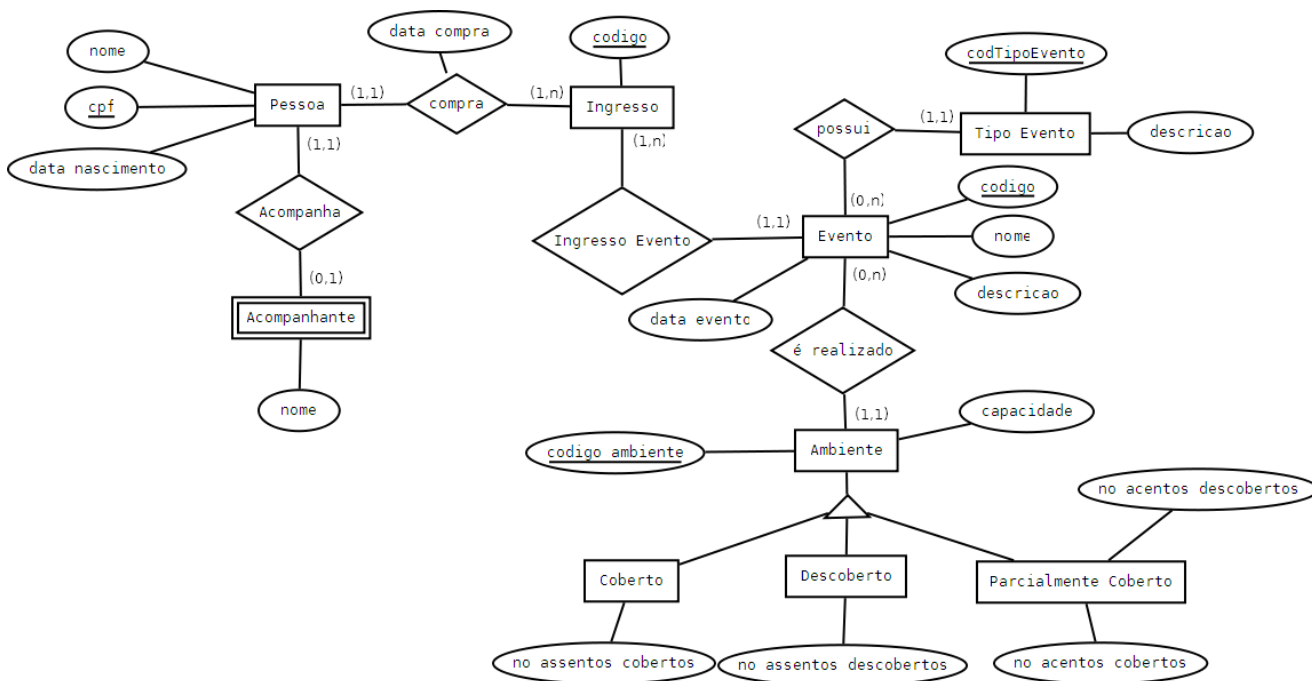
Observações:

1. Prova com consulta.

**Atenção:** Como a avaliação à distância é individual, caso seja constatado que provas de alunos distintos são cópias uma das outras, independentemente de qualquer motivo, a todas será atribuída a nota ZERO. As soluções para as questões podem sim, ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser individual.

**Questão 1 [2,0 pontos]**

Considere o diagrama ER mostrado abaixo.



Construa um esquema relacional equivalente a este diagrama ER.

Considerações:

1. O diagrama encontra-se na notação do DIA (ferramenta que usamos em sala de aula para construir modelos ER).
2. A entidade mostrada com linha dupla (Acompanhante) representa uma entidade fraca.
3. Use uma tabela para cada entidade especializada para mapear a generalização/especialização.

Pessoa(cpf, nome, dataNascimento, nomeAcompanhante)

Ambiente (codigoAmbiente, capacidade, tipo)

AmbienteCoberto (codigoAmbiente, noAssentosCobertos)

codigoAmbiente referencia Ambiente

AmbienteDescoberto (codigoAmbiente, noAssentosDescobertos)

codigoAmbiente referencia Ambiente

AmbienteParcialmenteCoberto (codigoAmbiente, noAssentosCobertos,

noAssentosDescobertos)

codigoAmbiente referencia Ambiente

TipoEvento (codTipoEvento, descricao)

Evento (codigo, nome, descrição, dataEvento, codTipoEvento, codigoAmbiente)

codTipoEvento referencia TipoEvento

codigoAmbiente referencia Ambiente

Ingresso (codigo, cpf, codigoEvento, dataCompra)

cpf referencia Pessoa

codigoEvento referencia Evento

## **Questão 2** [3,0 pontos]

Responda:

1. Explique o conceito de VISÃO em bancos de dados relacionais. [0,5 ponto]

Visões são tabelas virtuais, derivadas de outras tabelas ou de outras visões.

2. Explique o que é atualização através de visões em Banco de Dados. [0,5 ponto]

Atualização através de visões é uma técnica que permite atualizar (inserir, excluir ou modificar) tuplas de uma visão, de modo que as tabelas base sobre as quais a visão foi construída sejam automaticamente atualizadas.

3. Qual o principal problema envolvido em atualização através de visões? [0,5 ponto]

O principal problema é o problema de ambigüidade. Este problema acontece porque pode existir mais de um mapeamento de uma atualização sobre a visão para atualizações nas tabelas base.

4. Qual a diferença entre consultas SQL aninhadas correlacionadas e não correlacionadas? **[0,5 ponto]**

Numa consulta aninhada existe a consulta externa e a consulta interna. A diferença é que na consulta aninhada correlacionada:

- a condição da cláusula WHERE da consulta interna referencia um atributo de uma relação declarada na consulta externa; e
- o resultado de uma consulta aninhada correlacionada é diferente para cada tupla (ou combinação de tuplas) da relação(ões) da consulta externa.

Já na consulta aninhada não correlacionada não há essa dependência ou correlação entre a consulta externa e a consulta interna.

5. Exemplifique a diferença entre consultas SQL aninhadas correlacionadas e não correlacionadas apresentando duas consultas sobre o esquema relacional abaixo: uma aninhada correlacionada e outra aninhada não correlacionada. **[1,0 ponto]**

Emp (eid: integer, *ename*: string, *idade*: integer, *salario*: real)

Trabalha (eid: integer, did: integer, *cargahoraria*: integer)

Dept (did: integer, *dnome*: string, *orçamento*: real, *gerenteid*: integer)

Consulta aninhada correlacionada:

```
SELECT e.eid, e.nome
FROM Emp e
WHERE (
  SELECT COUNT(*)
  FROM Trabalha t
    WHERE t.eid = e.eid
) > 2
```

Consulta aninhada não correlacionada:

```
SELECT e.eid, e.nome
FROM Emp e
WHERE e.eid IN (
  SELECT t.eid
  FROM Trabalha t
    WHERE cargahoraria > 20
)
```

**Questão 3 [2 pontos]**

(a) Quais são os objetivos do processo de Normalização? [0,5 ponto]

Reagrupar informações para eliminar redundâncias de dados e para eliminar estruturas inexistentes no modelo relacional (atributos multivalorados).

(b) Analise a tabela abaixo (referente à organização de um departamento de uma universidade). [1,5 pontos]

<i>Departamento</i>							
<i>Cod depto</i>	<i>Nome depto</i>	<i>Sala depto</i>	<i>Tamanho sala</i>	<i>Armários Sala</i>	<i>Disciplina depto</i>		
					<i>Cod</i>	<i>Nome</i>	<i>Carga horária</i>
DCC	Depto de Ciência da Computação	S11-a	250 m2	15	001	Informática e Sociedade	36H
					002	Banco de Dados	40H
					003	Programação I	36H

1. Diga em que forma normal encontra-se a tabela. Justifique sua resposta. [0,5 ponto]

A tabela não se encontra normalizada em nenhuma forma normal, pois existem tabelas aninhadas (não está na 1FN).

2. Caso a tabela não esteja normalizada, normalize-a mostrando as transformações da tabela para a terceira forma normal. Mostre cada forma normal intermediária entre aquela em que a tabela se encontra e a terceira forma normal. [1,0 ponto]

0FN:

Tab1(CodDepto, nomeDepto, SalaDepto, tamanhoSala, armariosSala, (Cod, nome, cargaHoraria))

1FN:

Tab1(CodDepto, nomeDepto, SalaDepto, tamanhoSala, armariosSala)  
DisciplinaDepto(Cod, nome, cargaHoraria, CodDepto)

2FN:

Tab1(CodDepto, nomeDepto, SalaDepto)  
Sala(SalaDepto, tamanhoSala, armariosSala)  
DisciplinaDepto(Cod, nome, cargaHoraria, CodDepto)

3FN:

Já está.

**Atenção TUTOR: a pontuação desta questão será atribuída da seguinte forma:**

0,5 ponto para a 1FN

0,5 ponto para a 2FN

0,5 ponto para a 3FN

**Questão 4 [3,0 pontos]**

Considere o esquema relacional abaixo. As chaves primárias estão sublinhadas.

Proprietário (Nome, <u>CPF</u> )
Apartamento ( <u>Número</u> , Andar, CPF)
CPF referencia Proprietário (CPF)
Cobranca ( <u>NumApartamento</u> , <u>Data</u> , Valor)
NumApartamento referencia Apartamento (Numero)

(a) Escreva os comandos SQL para criar as tabelas, incluindo as restrições de integridade que se aplicam. **[1,0 ponto]**

Assuma que:

1. Um proprietário não pode ser excluído se houver alguma informação associada a ele.
2. Ao alterar o número de um apartamento, todas as referências na tabela Cobranca devem ser atualizadas.
3. Um Apartamento não pode ser excluído caso hajam Cobranças associadas.

```
CREATE TABLE PROPRIETARIO (  
    CPF VARCHAR(9) NOT NULL,  
    NOME VARCHAR(30)  
    PRIMARY KEY (CPF)  
)
```

```
CREATE TABLE APARTAMENTO (  
    Numero INTEGER NOT NULL,  
    Andar INTEGER,  
    CPF VARCHAR(9),  
    PRIMARY KEY (Numero),
```

```
FOREIGN KEY CPF references PROPRIETARIO (CPF) ON DELETE  
RESTRICT  
)
```

```
CREATE TABLE COBRANCA (  
    NumApartamento INTEGER NOT NULL,  
    Data DATE NOT NULL,  
    Valor DOUBLE,  
    PRIMARY KEY (NumApartamento, Data)  
    FOREIGN KEY NumApartamento references APARTAMENTO (Numero) ON  
UPDATE CASCADE ON DELETE RESTRICT  
)
```

(b) Escreva o comando SQL necessário para incluir um novo proprietário no banco de dados, com os seguintes dados: CPF 0000000000-01, nome “Maria Lima”. **[0,5 ponto]**

```
INSERT INTO PROPRIETARIO  
VALUES (“00000000001”, “Maria Lima”);
```

(c) Escreva um comando SQL para excluir todas as cobranças dos apartamentos do proprietário “João Paulo”. **[0,5 ponto]**

```
DELETE FROM COBRANCA  
WHERE numApartamento IN  
    (SELECT numero  
    FROM apartamento a, proprietario p  
    WHERE a.cpf=p.cpf AND p.nome = “João Paulo”)
```

(d) Crie uma visão que contenha o nome, o CPF e o numero de apartamentos que cada proprietário possui. **[0,5 ponto]**

```
CREATE VIEW V (Nome, CPF, NumApartamentos) AS  
SELECT p.Nome, p.CPF, COUNT(*)  
FROM Proprietario p, Apartamento a  
WHERE p.CPF = a.CPF  
GROUP BY p.Nome, p.CPF
```

(e) Escreva uma instrução SQL que retorna o numero do apartamento, o andar, a data e o valor das cobranças dos apartamentos cujo proprietário tem CPF 000000000001. **[0,5 ponto]**

```
SELECT a.Numero, a.Andar, c.Data, c.Valor  
FROM apartamento a, cobrança c  
WHERE a.numero = c.numApartamento  
AND a.cpf= “00000000001”
```