



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Banco de Dados
AD2 1º semestre de 2013.

Nome: _____

Observações:

1. Prova COM consulta.

Atenção: Como a avaliação à distância é individual, caso seja constatado que provas de alunos distintos são cópias umas 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.

ADs enviadas pelo correio devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.

Questão 1 (1,0 ponto). Considere o esquema relacional a seguir, onde as chaves primárias estão sublinhadas.

Aluno (<u>aid</u> : integer, anome: string, mat: string, vbolsa: real, oid:integer) oid referencia Orientador Orientador (<u>oid</u> : integer, onome: string)
--

Sobre esta base de dados, resolver as consultas a seguir usando álgebra relacional.

ATENÇÃO: Não usar mais tabelas que o estritamente necessário.

(a) Quais os ids dos alunos cujo nome seja "Daniel de Oliveira". [0,2 ponto]

π aid (σ anome="Daniel de Oliveira" Aluno)

(b) Quais os nomes dos orientadores que orientam alunos com bolsa de até 1.500,00. [0,2 ponto]

$\pi_{\text{onome}} (\sigma_{\text{vbolsa} < 1500} \text{ Aluno} * \text{ Orientador})$

- (c) Quais os nomes dos alunos que são orientados pela professora Vanessa Braganholo.
[0,2 ponto]

$\pi_{\text{anome}} (\text{Aluno} * \sigma_{\text{onome} = \text{"Vanessa Braganholo"}} \text{ Orientador})$

- (d) Quais os nomes dos orientadores não possuem alunos. [0,4 ponto]

$\pi_{\text{onome}} (\text{Orientador}) - \pi_{\text{onome}} (\text{Aluno} * \text{ Orientador})$

Questão 2 (1,0 ponto – 0,2 cada). Sobre a base de dados da questão anterior, resolver as consultas utilizando SQL. Não usar mais tabelas que o estritamente necessário.

- (a) Quais os nomes dos alunos que recebem uma bolsa menor que 2.000,00.

**SELECT anome FROM Aluno
WHERE vbolsa < 2000;**

- (b) Quais os nomes dos professores que orientam alunos que ganham bolsas de até 1.500,00.

**SELECT onome FROM Aluno a, Orientador o
WHERE a.oid = o.oid
AND a.vbolsa < 1500;**

- (c) Quais os nomes dos alunos que são orientados pelo professor “Daniel de Oliveira”.

**SELECT anome FROM Aluno a, Orientador o
WHERE a.oid = o.oid
AND o.onome = “Daniel de Oliveira”;**

- (d) Qual o nome do orientador que orienta o aluno “Daniel de Oliveira”.

**SELECT onome FROM Aluno a, Orientador o
WHERE a.oid = o.oid
AND a.anome = “Daniel de Oliveira”;**

- (e) Quais os nomes dos orientadores não possuem orientandos.

**SELECT onome from
Orientador**

**WHERE oid NOT IN (SELECT oid FROM Orientador o, Aluno a
WHERE a.oid = o.oid);**

Questão 3 (6,0 pontos 0,5 ponto cada). Considere a seguinte base de dados relacional, usada por uma empresa que disponibiliza serviço de babás, sob demanda, ou seja, atendimentos esporádicos. As chaves primárias estão sublinhadas.

```
CLIENTE (cpf, nome_cli)
CRIANCA (id, nome, idade, cpf);
    (cpf) references CLIENTE
BABY_SITTER(id, data_agendada, data_ultimo_trabalho, data_atendimento)
    (id) references CRIANCA
MATERIAL_GASTO (id, data_agendada, cod_material, quantidade)
    (id, data_agendada) references BABY_SITTER
    (cod_material) references MATERIAL
MATERIAL (cod_material, descricao)
```

Sobre esta base de dados, resolver as consultas a seguir utilizando SQL.

ATENÇÃO: Não usar mais tabelas que o estritamente necessário.

(a) Faça uma consulta que retorna os nomes dos clientes que sejam pais de uma criança de nome “Pedro Paiva” [0,5 ponto].

**SELECT nomecli from Cliente cli, Crianca cri
WHERE cli.cpf = cri.cpf
AND cri.nome = “Pedro Paiva”;**

(b) Faça uma consulta que retorna os nomes dos clientes e o nome da criança, cuja criança foi cuidada na data de 2013-02-20 [0,5 ponto].

**SELECT cli.nomecli, cri.nome from Cliente cli, Crianca cri, Baby_sitter bs
WHERE cli.cpf = cri.cpf
AND bs.id = cri.id
AND bs.data_atendimento = “2013-02-20”;**

(c) Faça uma consulta que retorna a descrição dos materiais que nunca foram usadas quando uma babá cuidou de uma criança. Esses materiais podem incluir brinquedos, mamadeiras, etc [0,5 ponto].

SELECT m.descricao from Material m

**WHERE m.cod_material NOT IN (SELECT mg.cod_material
FROM Material_gasto mg)**

(d) Obter o nome das crianças com cuidados realizados/agendados e que em um atendimento consumiram mais de um item com descrição = "Mamadeira Lillo" [0,5 ponto].

**SELECT cri.nome from Crianca cri, Baby_sitter bs, Material M, Material_Gasto mg
WHERE bs.id = cri.id
AND bs.id = mg.id
AND bs.data_agendada = mg.data_agendada
AND mg.cod_material = m.cod_material
AND m.descricao = "Mamadeira Lillo"
AND mg.quantidade > 1;**

(e) Escreva uma instrução SQL para inserir uma criança de id igual a 1, nome "João Pedro Monteiro" e associado ao responsável de CPF igual a 097693456-08. [0,5 ponto]

**INSERT INTO CRIANCA (id, nome, cpf)
VALUES (1, "João Pedro Monteiro", "097693456-08");**

(f) Escreva uma instrução SQL para excluir a tabela CRIANCA. [0,5 ponto]

DROP TABLE CRIANCA;

(g) Escreva uma instrução SQL para excluir todas as crianças relacionadas ao responsável "Daniel de Oliveira" [0,5 ponto]

**DELETE FROM Crianca
WHERE cpf IN (SELECT cli.cpf FROM Cliente cli
WHERE cli.nome_cli = "Daniel de Oliveira");**

(h) Escreva uma instrução SQL que retorne os nomes das crianças que foram cuidadas 01/01/2012 e que utilizaram algum material nos cuidados. Retorne o resultado ordenado em ordem decrescente de nome de criança. [0,5 ponto]

**SELECT cri.nome
FROM Crianca cri, Baby_sitter bs, Material M, Material_Gasto mg
WHERE bs.id = cri.id
AND bs.id = mg.id
AND bs.data_agendada = mg.data_agendada
AND bs.data_atendimento = "01/01/2012"
ORDER BY cri.nome DESC;**

(i) Escreva uma instrução SQL que retorne a quantidade total de atendimentos que usaram mais de 3 materiais [0,5 ponto]

**SELECT COUNT(*) AS NumAtendimentos
FROM Material_gasto mg**

```
WHERE mg.quantidade > 3  
GROUP BY mg.id, mg.data_agendada
```

(j) Faça uma consulta SQL que retorna o nome e cpf de clientes que possuem alguma criança cadastrada. O resultado deve estar ordenado pelo nome do cliente. [0,5 ponto]

```
SELECT c.nome, c.cpf  
FROM Cliente c, Crianca co  
WHERE c.cpf = co.cpf  
ORDER BY c.nome_cli;
```

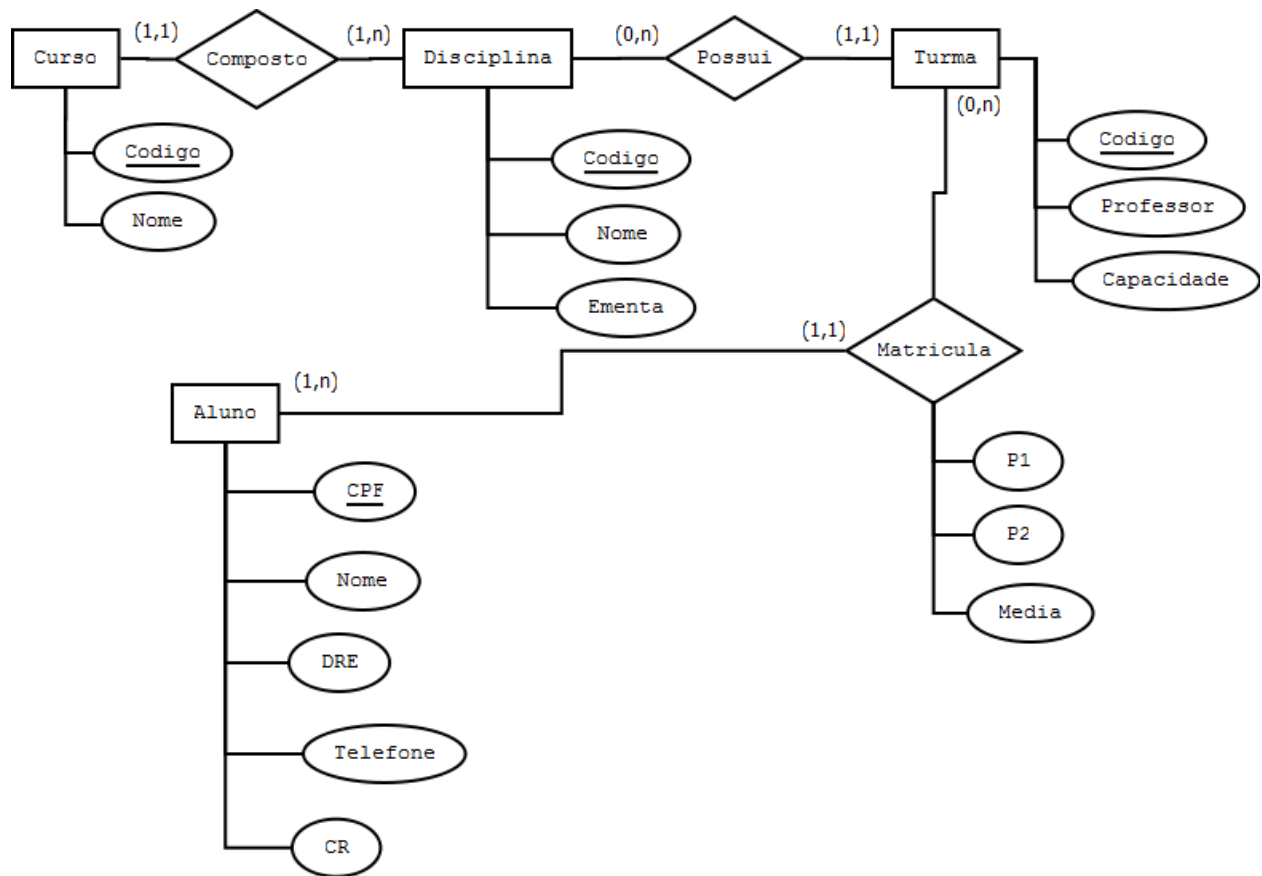
(k) Faça uma consulta SQL que retorna a descrição dos materiais que foram usados no atendimento da criança de id = 3. [0,5 ponto]

```
SELECT m.descricao  
FROM material m, material_gasto mg  
WHERE m.cod_material = mg.cod_material  
AND mg.id = 3;
```

(l) Faça uma consulta SQL que retorna o nome, o cpf do cliente e a quantidade de crianças que são atendidas na empresa. [0,5 ponto]

```
SELECT c.cpf, c.nome_cli, COUNT(*)  
FROM Cliente c, Crianca co  
WHERE c.cpf = co.cpf  
GROUP BY c.cpf, c.nome_cli;
```

Questão 4 (1,0 ponto). Considere o diagrama ER mostrado a seguir, relativo a um sistema de controle de matrículas de alunos em um curso universitário. Construa um esquema relacional equivalente a este diagrama ER. O diagrama encontra-se na notação do DIA (ferramenta que usamos nas aulas para construir modelos ER).



CURSO (ccod, nome);

DISCIPLINA (dcod,nome,ementa,ccod)
(ccod) references CURSO

TURMA (tcod, professor, capacidade, dcod)
(dcod) references DISCIPLINA

ALUNO (cpf,nome,dre,telefone,cr)

MATRICULA (cpf,tcod,p1,p2,media)
(cpf) references ALUNO
(tcod) references TURMA

Questão 5 (1,0 ponto). A “vídeo Center of Europe Ltda”. é uma cadeia de locadoras de DVDs. Ela precisa manter dados sobre os DVDs que têm para locação, os filmes dos DVDs, e locadoras da rede e suas respectivas localidades. Cada DVD para locação tem um número de série único. Um projetista de banco

de dados inexperiente propôs a seguinte tabela como solução para armazenar os dados da rede de locadoras. Suponha que cada filme tenha apenas um diretor (as chaves primárias estão sublinhadas):

Locadora (CodLocadora, Nome, CodLocal, NomeLocal (no_serie, Filme, Diretor, Ano, Tipo))

O significado de cada coluna é o seguinte:

- CodLocadora: código da locadora
- Nome: nome da locadora
- CodLocal: código da localidade onde a locadora se localiza
- NomeLocal: nome da localidade onde a locadora se localiza
- no_serie: Número de série do DVD
- Filme: nome do filme
- Diretor: nome do diretor
- Ano: ano de filmagem do filme
- Tipo: classificação do filme (ação, policial, terror, etc.)

As dependências funcionais (podendo incluir dependências transitivas) que existem nesta tabela são as seguintes:

CodLocadora → Nome

CodLocal → NomeLocal

CodLocadora → CodLocal

(CodLocadora, no_serie) → Filme, Diretor, Ano, Tipo

Filme → Diretor

Assumindo que o profissional não conhece o conceito de normalização, explique para ele em que forma normal encontra-se a tabela.

Não se encontra normalizada.

Caso a tabela não se encontre na terceira forma normal, ensine ao profissional como a transformar para a terceira forma normal. Mostre cada forma normal intermediária, entre aquela em que a tabela se encontra e a terceira forma normal.

Passagem para a Primeira Forma Normal

Locadora (CodLocadora, Nome, CodLocal, NomeLocal)

DVD (CodLocadora, no_serie, Filme, Diretor, Ano, Tipo)

Passagem para a Segunda Forma Normal

Já está.

Passagem para a Terceira Forma Normal

Locadora (CodLocadora, Nome, CodLocal)

Localidade (CodLocal, NomeLocal)

DVD (no_serie, Filme)

Filme (Filme, Diretor, Ano, Tipo)