Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Banco de Dados
AD2 2° semestre de 2014.

Nome:_____

Observações:

1. Prova COM consulta.
2. As ADs deverão ser postadas na plataforma antes do prazo final de entrega estabelecido no calendário de entrega de ADs.
3. Lembre-se de enviar as ADs para avaliação. Cuidado para não deixar a AD como "Rascunho" na plataforma!
4. **ADs em forma de "Rascunho" não serão corrigidas!**
5. As ADs devem ser **obrigatoriamente postadas em formato PDF**. **Outros formatos não serão aceitos.**
6. Não serão aceitas ADs enviadas em arquivos compactados. **Somente serão aceitas ADs postadas em um ÚNICO arquivo PDF**.
7. Se utilizar foto para gerar o PDF, assegure-se de usar um aplicativo que usa a câmera do celular como scanner. Existem vários que produzem um bom resultado. Assegure-se de que o PDF final é legível. **ADs não legíveis não serão corrigidas**.

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.

Questão 1 (5,0 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 por uma babá 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" [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"
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 mg.cod_material = m.cod_material
AND mg.quantidade > 1
ORDER BY cri.nome DESC;
```

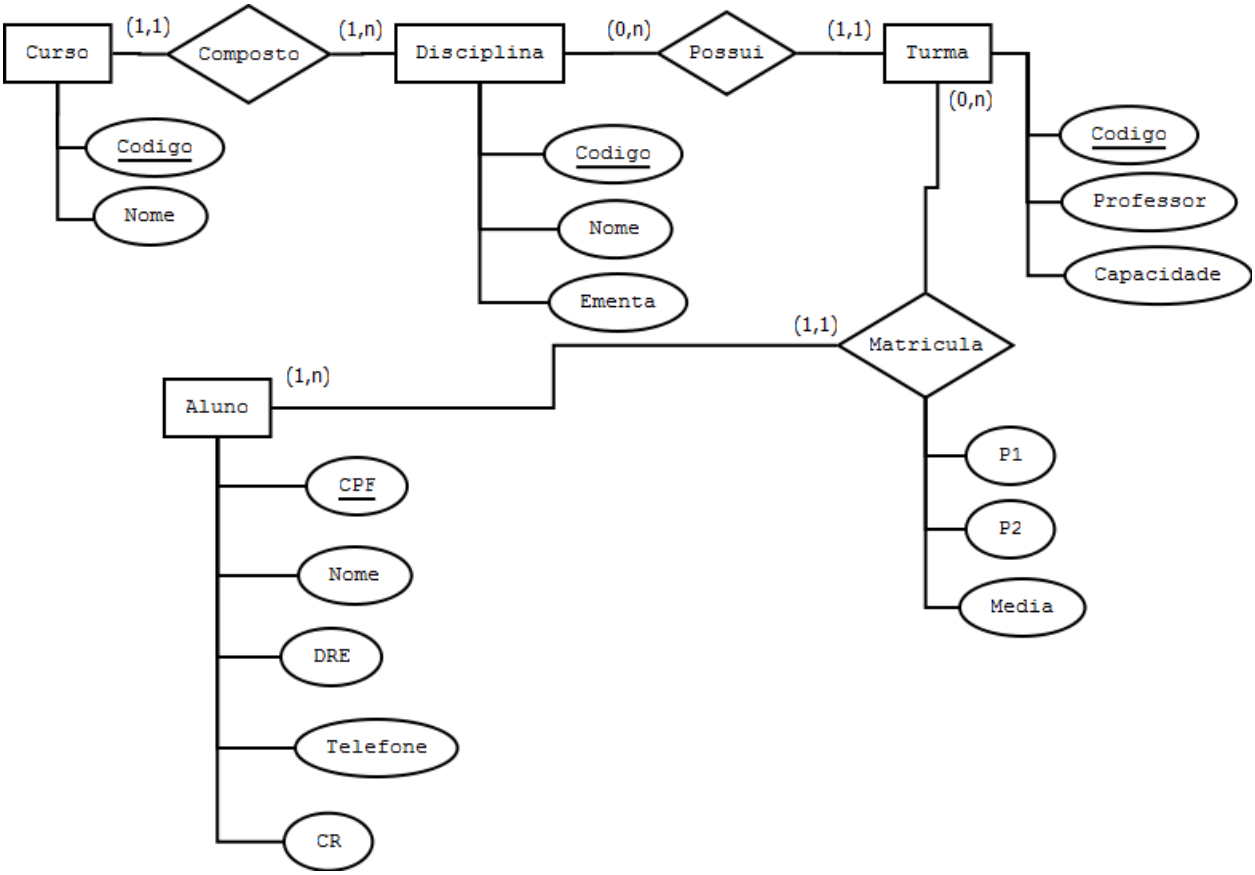
(i) Escreva uma instrução SQL que retorne a quantidade total de atendimentos que usaram mais de 3 unidades de algum material [0,5 ponto]

```
SELECT COUNT(*) AS NumAtendimentos
FROM BABY_SITER b
WHERE b.id, b.data_agendada IN (SELECT id, data_agendada FROM MATERIAL_GASTO mg WHERE mg.quantidade > 3)
```

(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;
```

Questão 2 (1,0 ponto). Considere o diagrama ER mostrado abaixo, 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 3 (2,0 pontos – 0,4 cada). Considere o esquema relacional abaixo de uma companhia de ônibus da cidade do Rio de Janeiro. As chaves primárias estão sublinhadas no esquema.

BAIRRO (cd_bairro, cd_municipio, ch_nome, dt_exclui)
    cd_municipio referencia MUNICIPIO
MUNICIPIO (cd_municipio, cd_pais, ch_nome, dt_exclui)
    cd_pais referencia PAIS
PAIS (cd_pais, ch_nome, dt_exclui)
RODOVIARIA (sq_rod, cd_municipio, ch_nomme, dt_exclui)
    cd_municipio referencia MUNICIPIO
CLIENTE (cd_cliente, cd_bairro, cd_telefone, ch_nome, dt_nasc, ch_endereco)
    cd_telefone referencia TELEFONE
VIAGEM (cd_viagem, nu_viagem, qtd_passagem, dt_partida, dt_chegada, dt_exclui)
PASSAGEM (cd_passagem, cd_cliente, cd_viagem, valor)
    cd_cliente referencia CLIENTE
    cd_viagem referencia VIAGEM
TELEFONE (cd_telefone, cd_tipo_telefone, nu_ddd, nu_telefone)
    cd_tipo_telefone referencia TIPO_TELEFONE
TIPO_TELEFONE (cd_tipo_telefone, ch_codigo, ch_nome)

1. Escreva um comando SQL para excluir todos os cientes cujo nome comece por “Marta” e termine com “Mattoso”.

   DELETE FROM CLIENTE WHERE CH_NOME LIKE “MARTA%MATTOSO”

2. Escreva um comando SQL que atualize o nome do município “Frorianopolis” para “Florianópolis”

UPDATE MUNICIPIO SET CH_NOME = “FRORIANOPOLIS” WHERE CH_NOME = “FLORIANÓPOLIS”

3. Escreva um comando SQL que lista todos os municípios que possuem mais do que uma rodoviária.

   SELECT M.CH_NOME, COUNT(*)
   FROM MUNICIPIO M, RODOVIARIA R
   WHERE M.CD_MUNICIPIO = R. CD_MUNICIPIO
   GROUP BY M.CH_NOME
   HAVING COUNT(*) > 1

4. Escreva um comando SQL que lista todos os passageiros que possuem viagens não realizadas. Considere nesta questão que existe uma função date() que retorna a data atual do sistema.

```
SELECT DISTINCT C.CH_NOME FROM
CLIENTE C, VIAGEM V, PASSAGEM P
WHERE C.CD_CLIENTE = P. CD_CLIENTE
AND V.CD_VIAGEM = P. CD_VIAGEM
AND V.DT_PARTIDA > DATE()
```

5. Escreva um comando SQL que lista todos os passageiros que possuem TODAS as suas viagens não realizadas. Considere nesta questão que existe uma função date() que retorna a data atual do sistema.

```
SELECT C.CH_NOME FROM
CLIENTE C
WHERE NOT EXISTS (
                    SELECT * FROM PASSAGEM P, VIAGEM V
                    WHERE P.CD_VIAGEM = V. CD_ VIAGEM
                    AND P. CD_CLIENTE = C.CD_CLIENTE
                    AND (V.DT_PARTIDA < DATE() OR DT_PARTIDA = DATE())
)
```

OU

```
SELECT C.CH_NOME FROM
CLIENTE C
WHERE C.CD_CLIENTE NOT IN(
                    SELECT C2.CD_CLIENTE FROM
                    CLIENTE C2, PASSAGEM P, VIAGEM V
                    WHERE P. CD_VIAGEM = V. CD_VIAGEM
                    AND P. CD_CLIENTE = C2.CD_CLIENTE
                    AND (V.DT_PARTIDA < DATE() OR DT_PARTIDA = DATE())
)
```

Questão 4 (2,0 pontos). Considere a seguinte tabela feita por um profissional que não foi aluno do CEDERJ, que não se encontra normalizada, de uma base de dados referente a uma rede de lanchonetes *fast food* (as chaves primárias estão sublinhadas):

Lanchonete (CodLanchonete, Nome, CodLocal, NomeLocal (CodSanduiche, NomeSanduiche, Tipo))

O significado de cada coluna é o seguinte:

- CodLanchonete: código da lanchonete
- Nome: nome da lanchonete
- CodLocal: código da localidade onde a lanchonete se localiza
- NomeLocal: nome da localidade onde a lanchonete se localiza
- CodSanduiche: código do sanduiche
- Nome: nome do sanduiche
- Hora: hora em que o sanduiche foi feito
- Validade: hora em que o sanduiche perde a validade
- Tipo: classificação do sanduiche

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

CodLanchonete→ Nome
CodLocal → NomeLocal
CodLanchonete→ CodLocal
(CodLanchonete, CodSanduiche) → Tipo
CodSanduiche→ NomeSanduiche

1. Mostre a transformação 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.

   Passagem para a Primeira Forma Normal

   Lanchonete (CodLanchonete, Nome, CodLocal, NomeLocal)
   Sanduiche (CodSanduiche, NomeSanduiche, Tipo)

   Passagem para a Segunda Forma Normal

   Já está.

   Passagem para a Terceira Forma Normal

   Lanchonete (CodLanchonete, Nome, CodLocal)
   Localidade (CodLocal, NomeLocal)
   Sanduiche (CodSanduiche, NomeSanduiche, Tipo)