Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**
**Disciplina Banco de Dados**
**AD2 1° semestre de 2011.**

**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.** Considere o seguinte esquema relacional:

Cliente (cid: integer, cnome: string, end: string)
Produto (pid: integer, pnome: string, tid: integer, preço: real)
    tid referencia Tipo
Compra (cid: integer, pid: integer, quantidade: integer)
    cid referencia Cliente
    pid referencia Produto
Tipo (tid: integer, tnome: string)

No esquema acima, as chaves primárias estão sublinhadas. A tabela de Compra lista a quantidade de itens de um mesmo produto comprados por um cliente. Apresente, para cada consulta a seguir, as expressões em álgebra relacional correspondentes às consultas [3 pontos no total, sendo que cada item vale 0,3 ponto].

1. Obtenha o pid do produto de nome “produto A”.
   $\Pi_{pid}$ ( $\sigma_{pnome="Produto\ A"}$ **Produto**)

2. Obtenha o nome dos produtos que foram comprados pelo menos uma vez.

   $\pi_{pnome}$ (Produto ⋈ Compra)

3. Obtenha o nome dos clientes que compraram produtos do tipo "A" e o nome do produto comprado.

   $\pi_{cnome,\ pnome}$ (Cliente ⋈ Compra ⋈ Produto ⋈ $\sigma_{tnome="A"}$ Tipo)

4. Obtenha o nome dos produtos do tipo "C" que tenham preço superior a 150.

   $\pi_{pnome}$ (( $\sigma_{tnome="C"}$ Tipo) ⋈ ($\sigma_{preço>\ 150}$ Produto))

5. Obtenha o pid dos produtos dos tipos "A" ou "B".

   $\pi_{pid}$ (( $\sigma_{tnome="A"\ v\ tnome="B"}$ Tipo) ⋈ Produto)

6. Obtenha o nome dos clientes que compraram algum produto em quantidade superior a 10 unidades do tipo "C".

   $\pi_{cnome}$ (($\sigma_{quantidade>"10"}$ Compra ⋈ (Produto⋈ $\sigma_{tnome="C"}$ Tipo)) ⋈Cliente)

7. Obtenha o cid dos clientes que compraram mais de 20 unidades de um item ou que compraram itens do tipo "B".

   ρ(R1, $\pi_{cid}$($\sigma_{quantidade>"20"}$ Compra))

   ρ(R2, $\pi_{cid}$ ($\pi_{pid}$(($\pi_{tid}\sigma_{tnome="C"}$ Tipo) ⋈ Produto) ⋈Compra))

   R1 ∪ R2

8. Obtenha o nome dos clientes que compraram o produto "Produto Z" e o produto "Produto K" em quantidade maior a 10 unidades cada.

   ρ(R1, $\pi_{cid}$(($\pi_{pid}\sigma_{pnome="Produto\ Z"}$ Produto) $\sigma_{quantidade>'10'}$ Compra))

   ρ(R1, $\pi_{cid}$(($\pi_{pid}\sigma_{pnome="Produto\ K"}$ Produto) $\sigma_{quantidade>'10'}$ Compra))

   ρ(R3, R1 ∩ R2)

   $\pi_{cnome}$(Cliente⋈ R3)

9. Obtenha o nome dos clientes que compraram algum produto de preço abaixo de 100.

   $\pi_{cnome}$ (( $\sigma_{preço<100}$ Produto) ⋈Compra) ⋈Cliente)

10. Obtenha o nome dos tipos dos produtos comprados pelo cliente "Daniel" e que não foram comprados pelo cliente "Pedro" em quantidade superior a 50 itens.

    ρ(R1, $\pi_{tnome}$ ((((($\sigma_{cnome\ =\ "Daniel"}$ Cliente) ⋈Compra) ⋈Produto) ⋈ Tipo))

    ρ(R2, $\pi_{tnome}$ ((((($\sigma_{cnome\ =\ "Pedro"}$ Cliente) ⋈$\sigma_{quantidade>50}$ Compra) ⋈Produto) ⋈Tipo))

    R1 - R2

**Questão 2.** Considere o esquema relacional abaixo. As chaves primárias estão sublinhadas.

| |
|---|
| Departamento (CodDepto, Nome)<br>Autor (CodAutor, nome, CodDepto)<br>CodDepto referencia Departamento<br>Conferência(CodConf, Nome)<br>Publicação (CodPub, título, ano, CodConf)<br>CodConf referencia Conferência<br>PublicaçãoAutor (CodAutor, CodPub)<br>CodAutor referencia Autor<br>CodPub REFERENCIA Publicação |

1. Escreva os comandos SQL para criar as tabelas Autor, Publicação e Publicação Autor, incluindo as restrições de integridade que se aplicam. Assuma que quando um departamento é excluído, o código de departamento dos autores relacionados a ele deve ser alterado para NULL. Assuma ainda que ao excluir um autor, todas as suas publicações devem ser excluídas automaticamente. Assuma também que uma conferência não pode ser excluída se houver alguma publicação relacionada. Além disso, ao alterar o código de uma conferência, todas as publicações relacionadas devem ser alteradas automaticamente. Em uma publicação, assuma que o ano não pode ser nulo. Uma tupla na relação PublicaçãoAutor só deve existir se existir uma tupla correspondente na tabela Publicação. Se, por algum motivo, não for possível definir alguma restrição de integridade, justifique [1 ponto].

```
CREATE TABLE AUTOR (
        CODAUTOR INT NOT NULL,
        NOME VARCHAR(30),
        CODDEPTO INT,
        PRIMARY KEY (CODAUTOR),
        FOREIGN KEY CODDEPTO REFERENCES DEPARTAMENTO (CODDEPT) ON DELETE SET
NULL
)

CREATE TABLE PUBLICACAO (
        CODPUB INT NOT NULL,
        TITULO VARCHAR(30),
        ANO INT NOT NULL,
        CODCONF INT,
        PRIMARY KEY (CODPUB),
        FOREIGN KEY (CODCONF) REFERENCES CONFERENCIA (CODCONF) ON DELETE
RESTRICT ON UPDATE CASCADE
)
```

**CREATE TABLE PUBLICACAOAUTOR (**
**CODAUTOR INT NOT NULL,**
**CODPUB INT NOT NULL,**
**PRIMARY KEY (CODAUTOR, CODPUB),**
**FOREIGN KEY (CODAUTOR) REFERENCES AUTOR (CODAUTOR) ON DELETE CASCADE,**
**FOREIGN KEY (CODPUB) REFERENCES PUBLICACAO (CODPUB) ON DELETE CASCADE**
**)**

**É possível definir todas as restrições de integridade.**

2. Escreva comandos SQL para incluir uma nova publicação no banco de dados, com os seguintes dados: Publicação código 111 do autor 05 (assuma que o autor já está cadastrado) na conferência 122 (assuma que a conferência já está cadastrada). Os detalhes da publicação são: título "Many task computing for large clusters", ano 2010 [0,5 ponto].

**INSERT INTO PUBLICACAO (CODPUB, TITULO, ANO, CODCONF)**
**VALUES (111, "Many task computing for large clusters", 2010, 122);**

**INSERT INTO PUBLICACAOAUTOR (CODAUTOR, CODPUB)**
**VALUES (05, 111);**

3. Escreva um comando SQL para excluir todas as publicações do autor "Daniel de Oliveira" [0,5 ponto].

**DELETE FROM PUBLICACAO**
**WHERE CODPUB IN (SELECT CODPUB**
**FROM PUBLICACAOAUTOR p, AUTOR a**
**WHERE p.CODAUTOR=a.CODAUTOR AND**
**a.NOME="Daniel de Oliveira")**

**NÃO é necessário excluir de PublicaçãoAutor devido à restrição DELETE CASCADE.**

**Questão 3.** Considere o esquema relacional abaixo. As chaves primárias estão sublinhadas.

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
Aeroporto(sq_aerop, cd_municipio, ch_nome, dt_exclui)
    cd_municipio referencia Municipio
Cliente(cd_cliente, cd_bairro, cd_telefone, ch_nome, dt_nasc, ch_endereco)
    cd_telefone referencia Telefone
Pais(cd_pais, ch_nome, dt_exclui)
Voo_Escala(cd_trecho, sq_aerop_origem, sq_aerop_destino, cd_voo, dt_origem, dt_destino)
    sq_aerop_origem referencia Aeroporto
    sq_aerop_destino referencia Aeroporto
    cd_voo referencia Voo
Voo(cd_voo, nu_voo, qtd_passagem, dt_partida, dt_chegada, dt_exclui)
Passagem(cd_passagem, cd_cliente, cd_voo, valor)
    cd_cliente referencia Cliente
    cd_voo referencia Voo
Telefone(cd_telefone, cd_cliente, 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 “Maria” e termine com ”Silva” [0,3 ponto].

   **DELETE FROM CLIENTE WHERE CH_NOME LIKE “MARIA%SILVA”**

2. Escreva um comando SQL que atualize o nome do município “Rio das Frores” para “Rio das Flores” [0,3 ponto].

   **UPDATE MUNICIPIO SET CH_NOME = “RIO DAS FLORES” WHERE CH_NOME = “RIO DAS FRORES”**

3. Escreva um comando SQL que lista todas as cidades que possuem mais do que um aeroporto [0,3 ponto].
   **SELECT M.CH_NOME, COUNT(*)**
   **FROM MUNICIPIO M, AEROPORTO A**

```
WHERE M.CD_MUNICIPIO = A.MUNICIPIO_CD_MUNICIPIO
GROUP BT M.CH_NOME
HAVING COUNT(*) > 10
```

4. Escreva um comando SQL que lista todos os passageiros que possuem voos não realizados. Considere nesta questão que existe uma função date() que retorna a data atual do sistema [0,3 ponto].

```
SELECT DISTINCT C.CH_NOME FROM
CLIENTE C, VOO V, PASSAGEM P
WHERE C.CD_CLIENTE = V.CLIENTE_CD_CLIENTE
AND V.CD_VOO = P.VOO_CD_VOO
AND V.DT_PARTIDA > DATE()
```

5. Escreva um comando SQL que lista todos os passageiros que possuem TODOS os seus voos não realizados. Considere nesta questão que existe uma função date() que retorna a data atual do sistema [0,3 ponto].

```
SELECT C.CH_NOME FROM
CLIENTE C
WHERE NOT EXISTS (
                    SELECT * FROM PASSAGEM P, VOO V
                    WHERE P.VOO_CD_VOO = V.CD_CD_VOO
                    AND P.CLIENTE_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, VOO V
                    WHERE P.VOO_CD_VOO = V.CD_CD_VOO
                    AND P.CLIENTE_CD_CLIENTE = C2.CD_CLIENTE
                    AND (V.DT_PARTIDA < DATE() OR DT_PARTIDA = DATE())
)
```

6. Escreva um comando SQL que exclua todos os tipos de telefone que não estão sendo utilizados no momento [0,3 ponto].

```
DELETE FROM TIPO_TELEFONE
WHERE CD_TIPO NOT IN (
                        SELECT DISTINCT TIPO_TELEFONE_CD_TIPO
                        FROM TELEFONE
                        )
```

**Questão 4.** Responda cada uma das questões seguintes. As questões são baseadas no esquema relacional abaixo:

| |
|---|
| 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) |

OBS: No esquema acima, as chaves primárias estão sublinhadas.

1. Dê um exemplo de uma chave estrangeira que envolva (referencie) a tabela Dept [0,2 ponto].
   **Coluna did da tabela Trabalha referencia a tabela Dept.**

2. Escreva os comandos SQL necessários para criar as relações acima, incluindo as versões apropriadas de todas as restrições de chave primária e chave estrangeira. Considere que a tabela Trabalha relaciona empregados com departamentos e a tabela Dept se relaciona com Emp a partir do atributo *gerenteid* [0,5 ponto].

```
CREATE TABLE EMP(
  EID INTEGER NOT NULL,
  ENAME VARCHAR(100),
  IDADE INTEGER,
  SALARIO DOUBLE PRECISION
);

ALTER TABLE EMP ADD CONSTRAINT EMP_PK PRIMARY KEY(EID);

CREATE TABLE TRABALHA(
  EID INTEGER NOT NULL,
  DID INTEGER NOT NULL,
  CARGAHORARIA INTEGER
);

ALTER TABLE TRABALHA ADD CONSTRAINT TRABALHA_PK PRIMARY KEY(EID, DID);

CREATE TABLE DEPT(
```

```
 DID INTEGER NOT NULL,
 DNOME VARCHAR(100),
 ORCAMENTO DOUBLE PRECISION,
 GERENTEID INTEGER
);

ALTER TABLE DEPT ADD CONSTRAINT DEPT_PK PRIMARY KEY(DID);

ALTER TABLE TRABALHA ADD CONSTRAINT TRABALHA_EMP_FK FOREIGN KEY(EID) REFERENCES EMP(EID);

ALTER TABLE TRABALHA ADD CONSTRAINT TRABALHA_EMP_FK FOREIGN KEY(DID) REFERENCES DEPT(DID);

ALTER TABLE DEPT ADD CONSTRAINT DEPT_EMP_FK FOREIGN KEY(GERENTEID) REFERENCES EMP(EID);
```

3. Defina a tabela Dept em SQL de modo que cada departamento tenha necessariamente um gerente [0,2 ponto].
   **ALTER TABLE DEPT ADD ALTER COLUMN GERENTEID INTEGER NOT NULL;**

4. Escreva uma instrução SQL para adicionar “Marta Mattoso” como um empregado com eid = 125, idade = 35 e salário = 15000 [0,2 ponto].
   **INSERT INTO Emp(eid, ename, idade, salario) VALUES (122, “Marta Mattoso”, 32, 15000);**

5. Escreva uma instrução SQL para dar a todos os empregados um aumento de 12% [0,2 ponto].
   **UPDATE Emp SET salario = salario * 1.12;**

6. Escreva uma instrução SQL que altere a carga horária de todos os empregados que trabalham no departamento “Vendas” para 20 [0,2 ponto].
   **UPDATE Trabalha SET cargahoraria = 20 WHERE did IN (SELECT did FROM dept WHERE dnome = ‘Vendas’)**

7. Escreva uma consulta SQL que retorne a média salarial dos empregados, por departamento [0,2 ponto].
   **SELECT t.did, AVG(e.salario) FROM Emp e, Trabalha t WHERE t.eid = e.eid GROUP BY t.did**

8. Escreva uma consulta que retorna os nomes dos empregados que trabalham no departamento que possui o maior orçamento [0,5 ponto].

```
SELECT e.ename
FROM Emp e
WHERE exists (
```

```
        select * from Trabalha t, Dept d
        where e.eid = t.eid
        AND t.did = d.didid
        And d.orcamento =
                (SELECT max(x.orcamento) FROM Dept x)
)
```

**Questão 5.** Considere a seguinte tabela, não necessariamente normalizada, de uma base de dados referente a uma rede de locadoras de vídeo (as chaves primárias estão sublinhadas):

Locadora (CodLocadora, Nome, CodLocal, NomeLocal (CodFita, 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
- CodFita: código da Fita/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, CodFita) → Filme, Diretor, Ano, Tipo
Filme → Diretor

1. Diga em que forma normal encontra-se a tabela [0,2 ponto]

   **Não se encontra normalizada**

2. Caso a tabela não se encontre na terceira forma normal, 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 [0,8 ponto].

**Passagem para a Primeira Forma Normal**

**Locadora (CodLocadora, Nome, CodLocal, NomeLocal)**
**Fita (CodLocadora, CodFita, 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)**
**Fita (CodFita, Filme)**
**Filme (Filme, Diretor, Ano, Tipo)**