



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina Banco de Dados**  
**AD2 1º semestre de 2014.**

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.

**A entrega das ADs será feita exclusivamente via plataforma em formato PDF.**

**ADs entregues em formatos diferentes de PDF não serão corrigidas.**

**Não serão aceitas ADs entregues via correio ou no pólo.**

**Questão 1 (2,0 pontos – 0,5 ponto cada).** Considere a seguinte base de dados, usada para gerenciar a produção de uma fábrica. As chaves primárias estão sublinhadas.

Maquina ( <u>codMaq</u> , nomeMaq) % Tabela com máquinas usadas em uma fábrica para fabricar produtos%
Produto ( <u>codProd</u> , descrProd, pesoProd) % Tabela com os produtos fabricados%
MaqProd ( <u>codMaq</u> , <u>codProd</u> , tempo) % Tabela que informa em que máquinas cada produto pode ser fabricado e qual o respectivo tempo de produção %
codMaq referencia Maquina
codProd referencia Produto
ComposProd ( <u>codProdComposto</u> , <u>codProdComponente</u> , quant) % Tabela que informa a composição de cada produto - Para cada produto composto, informa quais os produtos que o compõem em que quantidade %
codProdComposto referencia Produto
codProdComponente referencia Produto

Sobre esta base de dados, resolver as consultas a seguir usando **SQL**. Não usar mais tabelas que o estritamente necessário em cada consulta.

(a) Faça uma consulta que retorna os nomes das máquinas e as descrições dos produtos que podem ser fabricados naquela máquina. Os resultados devem estar ordenados pelo nome da máquina.

```
SELECT m.nomeMaq, p.descrProd
FROM MAQUINA m, MAQPROD mp, PRODUTO p
WHERE m.codMaq = mp.codMaq
AND p.codProd = mp.codProd
ORDER BY nomeMaq
```

(b) Faça uma consulta que retorna as descrições dos produtos que demoram mais de 20h para serem produzidos.

```
SELECT p.descrProd
FROM MAQPROD mp, PRODUTO p
WHERE p.codProd = mp.codProd
AND mp.tempo > 20
```

(c) Faça uma consulta que retorna a média dos pesos dos produtos fabricados pela máquina de código 15.

```
SELECT mp.codMaq, AVG(p.pesoProd) AS pesoMedio
FROM MAQPROD mp, PRODUTO p
WHERE p.codProd = mp.codProd
AND mp.codMaq = 15
GROUP BY mp.codMaq
```

(d) Faça uma consulta que retorna a descrição dos produtos, seus componentes, e a quantidade que precisam para serem produzidos.

```
SELECT p1.descrProd, p2.descrProd, cp.quant
FROM PRODUTO p1, PRODUTO p2, COMPOSPROD cp
WHERE p1.codProd = cp.codProdComposto
AND p2.codProd = cp.codProdComponente
```

**Questão 2 (3,0 pontos – 0,6 cada)** Considere a seguinte base de dados genérica de fornecimento de peças por um fornecedor qualquer para uma série de projetos que usam essas peças. As chaves primárias estão sublinhadas.

Fornecedor ( <u>fcod</u> : integer, fnome: string, status: string, cidade: string)
Pecas ( <u>pcod</u> : integer, pnome: string, cor:string, peso: float, cidade:string)
Projeto ( <u>pjcod</u> :integer, pjnome:string, pjcidade:string)
Fornecimento ( <u>fid</u> :integer, <u>pid</u> :integer, <u>pjid</u> :integer, quantidade: integer)
fcod referencia Fornecedor
pcod referencia Pecas

Sobre esta base de dados, resolver as consultas a seguir usando **SQL**. Não usar mais tabelas que o estritamente necessário em cada consulta.

1. Obtenha todas as cidades em que ao menos um fornecedor, uma peça ou um projeto esteja localizado.

```
SELECT cidade FROM Fornecedor
UNION
SELECT cidade FROM PECAS
UNION
SELECT cidade FROM PROJETO
```

2. Obtenha os códigos dos projetos fornecidos por pelo menos um fornecedor da mesma cidade que esse projeto.

```
SELECT pj.pcod
FROM Projeto pj, Fornecimento fn, Fornecedor f
WHERE pj.pjcod = fn.pjcod
AND fn.fcod = f.fcod
AND pj.cidade = f.cidade
```

3. Obter nomes dos fornecedores (sem repetição) que fornecem todas as peças.

```
SELECT distinct F.nome
FROM Fornecedor F
WHERE not exists (
    SELECT * FROM Pecas P
    WHERE P.pcod NOT IN (
        SELECT FP.pcod
        FROM Fornecimento FP
        WHERE F.fcod = FP.fcod
        AND FP.pcod = P.pcod ) )
```

4. Obtenha os códigos dos fornecedores que fornecem a mesma peça a todos os projetos

```
SELECT f.fcod FROM FORNECEDOR
WHERE EXISTS
    (SELECT * FROM PECAS P
    WHERE NOT EXISTS
        (SELECT * FROM PROJETO PJ
        WHERE NOT EXISTS
```

```
(SELECT * FROM FORNECIMENTO FN
WHERE fn.fcod = f.fcod
AND fn.pcod = p.pcod
AND fn.pjcod = pj.pjcod)))
```

5. Obter os códigos das peças fornecidas por mais de um fornecedor.

```
SELECT fn.pcod
FROM FORNECIMENTO fn
GROUP BY fn.pcod
HAVING COUNT(distinct fn.fcod) > 1
```

**Questão 3 (1,0 ponto – 0,25 cada item).** Considere o seguinte esquema relacional, onde as chaves primárias estão sublinhadas.

Funcionario (fid: integer, fnome: string, salario:real, did:integer)  
                   did referencia Departamento  
 Departamento (did: integer, dnome: string)

João era um aluno que sabia bastante álgebra relacional, porém não conhecia muito SQL. Ele elaborou várias consultas em álgebra relacional e precisa de ajuda para “traduzi-las” para SQL. Você pode ajudar o João? Para cada uma das expressões abaixo apresentadas, informe a consulta SQL equivalente.

(a)  $\pi_{fnome} (\sigma_{salario < 2000} Funcionario)$

```
SELECT fnome FROM FUNCIONARIO
WHERE salario < 2000
```

(b)  $\pi_{dnome} (\sigma_{salario < 1500} Funcionario * Departamento)$

```
SELECT dnome FROM FUNCIONARIO f, DEPARTAMENTO d
WHERE f.did = d.did
AND f.salario < 1500
```

(c)  $\pi_{fnome} (Funcionario * \sigma_{dnome = "D3"} Departamento)$

```
SELECT fnome FROM FUNCIONARIO f, DEPARTAMENTO d
WHERE f.did = d.did
AND d.dnome = "D3"
```

(d) Quais os nomes dos departamentos não possuem empregados.  
 $\pi_{dnome} (DEPARTAMENTO) - \pi_{dnome} (DEPARTAMENTO * FUNCIONARIO)$

**SELECT d.nome FROM DEPARTAMENTO d  
WHERE d.did NOT IN (SELECT did FROM FUNCIONARIO)**

**Questão 4 (3,0 pontos).** Considere a seguinte base de dados, usada por uma empresa de computadores (Dell, HP, por exemplo) e que disponibiliza manutenção de computadores e *upgrades*. As chaves primárias estão sublinhadas.

```
CLIENTE (cpf, nome_cli)

COMPUTADOR (no_serie, modelo, cpf);

(cpf) references CLIENTE

/* tabela com os upgrades periódicos programados e realizados – para cada computador, a
empresa cadastra todos os upgrades programados. */

UPGRADE_REVISAO (no_serie, data_programada, data_ultimo_upgrade,
data_executada)

(no_serie) references COMPUTADOR

PEÇA_UPGRADE_REVISAO (no_serie, data_programada, cod_peça, quantidade)

(no_serie, data_programada) references UPGRADE_REVISAO

(cod_peça) references PEÇA

PEÇA (cod_peça, descricao_peça)
```

Sobre a base de dados acima apresentada, resolver as consultas a seguir usando SQL.

**ATENÇÃO:** Não usar mais tabelas que o estritamente necessário.

(a) Escreva uma instrução SQL para inserir um computador de número de série M4N68T-M, modelo igual a Acer Aspire , e CPF igual a 1010. [0,6 ponto]

**INSERT INTO COMPUTADOR (no\_serie, modelo, cpf)  
VALUES ("M4N68T-M", "Acer Aspire", "1010")**

(b) Escreva uma instrução SQL para excluir a tabela Cliente. [0.6 ponto]

**DROP TABLE Cliente**

(c) Faça uma consulta SQL que retorna o nome e cpf de clientes que possuem algum computador cadastrado. O resultado deve estar ordenado pelo nome do cliente. [0.6 ponto]

**SELECT c.nome, c.cpf  
FROM Cliente c, Computador co**

```
WHERE c.cpf = co.cpf
ORDER BY c.nome_cli
```

(d) Faça uma consulta SQL que retorna a descrição das peças que foram usadas no upgrade do computador de número de série M4N68T-M. [0.6 ponto]

```
SELECT p.descricao_pecas
FROM Peca p, Peca_Upgrade_Revisao pur
WHERE p.cod_pecas = pur.cod_pecas
AND pr.no_serie = "M4N68T-M"
```

(e) Faça uma consulta SQL que retorna o nome, o cpf do cliente e a quantidade de computadores que fazem revisão ou upgrade na empresa. [0.6 ponto]

```
SELECT c.cpf, c.nome, COUNT(*)
FROM Cliente c, Computador co
WHERE c.cpf = co.cpf
GROUP BY c.cpf, c.nome
```

**Questão 5 (1 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

1. 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.**

2. 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)**