



Fundação CECIERJ – Vice Presidência de Educação Superior a Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina Banco de Dados
AD2 2º semestre de 2018.

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 enviadas exclusivamente no formato de arquivo PDF.
6. ADs entregues em outros formatos 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 (1 ponto – 0,25 cada item). Considere a consulta em álgebra abaixo e responda as questões.

$R \leftarrow \sigma_{NOME = 'Maria' \wedge SOBRENOME = 'Freitas'} (FUNCIONARIO)$
 $RESULT \leftarrow \pi_{ENDERECO, TELEFONE} (R)$

ATENÇÃO: Não usar mais tabelas que o estritamente necessário. Para responder as questões, utilize a base de dados abaixo.

Funcionario (fcodigo, nome, sobrenome, cpf, endereco, cidade, estado, telefone, dnascimento, dcodigo)

dcodigo referencia Departamento

Departamento (dcodigo, dnome)

- (a) Qual a informação que se pretende obter com a consulta em álgebra?

Endereço e telefone da funcionária cujo nome é “ Maria Freitas”.

- (b) Escreva uma consulta SQL equivalente à essa consulta em álgebra relacional.

```
SELECT Endereco, Telefone
FROM FUNCIONARIO
WHERE NOME='Maria'
AND SOBRENOME='Freitas'
```

- (c) Escreva uma consulta em algebra que também retorne a data de nascimento de “Maria C. Freitas”.

```
R ←  $\sigma_{NOME = 'Maria' \wedge SOBRENOME = 'Freitas'}$  (FUNCIONARIO)
RESULT ←  $\pi_{ENDERECO, TELEFONE, DNASCIMENTO}$  (R)
```

- (d) Ao incluir a data de nascimento, como ficaria a consulta SQL equivalente?

```
SELECT Endereco, Telefone, DNascimento
from FUNCIONARIO
WHERE NOME='Maria'
AND SOBRENOME='Freitas'
```

Questão 2 (2 pontos – 0,4 cada item). Considerando o esquema relacional abaixo, resolva as consultas utilizando SQL.

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

Departamento (did: integer, dnome: string)

- (a) Selecione os nomes dos empregados que ganham um salário superior à 4.000,00?

```
SELECT fnome
FROM Funcionario
WHERE salario > 4000;
```

- (b) Quais os nomes dos departamentos que possuem empregados que ganham salário até 1.500,00?

```
SELECT dnome
FROM Funcionario f, Departamento d
```

**WHERE f.did = d.did
AND f.salario ≤ 1500;**

- (c) Quais os nomes dos empregados que trabalham no departamento de nome “D1”? Ordene o resultado pelos nomes dos empregados.

**SELECT fnome
FROM Funcionario f, Departamento d
WHERE f.did = d.did
AND d.dnome = “D1”
ORDER BY fnome;**

- (d) Qual o nome do departamento em que trabalha o empregado “José da Silva”?

**SELECT dnome
FROM Funcionario f, Departamento d
WHERE f.did = d.did
AND f.fnome = “José da Silva”;**

- (e) Quais os nomes dos departamentos não possuem empregados.

**SELECT dnome
FROM Departamento
WHERE did NOT IN (SELECT did FROM Funcionario f);**

Questão 3 (3 pontos) Sobre a base de dados abaixo, resolva as consultas a seguir usando SQL.

CLIENTE (<u>ccliente</u> , cnome, cpf, endereço, cidade, cep, estado)
PEDIDO (<u>cpedido</u> , data, ccliente, cvendedor)
ccliente referencia CLIENTE
VENDEDOR (<u>cvendedor</u> , vnome, salario)
ITEM_PEDIDO (<u>cpedido</u> , <u>cproduto</u> , quantidade)
cpedido referencia PEDIDO
cproduto referencia PRODUTO
PRODUTO (<u>cproduto</u> , pnome, unidade, descricao, valor)

ATENÇÃO: Não usar mais tabelas que o estritamente necessário. As chaves primárias estão sublinhadas.

- (a) Escreva uma instrução SQL para inserir um cliente chamado ‘Joao Pereira’, CPF 01234432112, que reside em av. Osvaldo Aranha, 64, Centro, São Paulo, CEP 01040-001. O código do cliente é 955 [0.4 ponto]

INSERT INTO CLIENTE (ccliente, cnome, cpf, endereco, cidade, cep, estado)
VALUES (955, "Joao Pereira", "01234432112", "av. Osvaldo Aranha, 64, Centro", "São Paulo", "01040-001", "SP")

(b) Escreva uma instrução SQL para excluir a tabela Vendedor. [0.4 pontos]

DROP TABLE Vendedor

(c) Faça uma consulta SQL que retorne o nome e cpf de clientes que possuem algum pedido cadastrado. O resultado deve estar ordenado pelo nome do cliente. [0.4 ponto]

SELECT c.cnome, c.cpf
FROM Cliente c, Pedido p
WHERE c.ccliente = p.ccliente
ORDER BY c.cnome

(d) Faça uma consulta SQL que retorne os nomes dos clientes que residem no Rio de Janeiro. [0.4 ponto]

SELECT c.cnome
FROM Cliente c
WHERE c.estado = "RJ"

OU

SELECT c.cnome
FROM Cliente c
WHERE c.cidade = "Rio de Janeiro"

(e) Faça uma consulta SQL que retorne o nome, o cpf do cliente e a quantidade de pedidos que ele já realizou. [0.4 ponto]

SELECT c.cpf, c.cnome, COUNT(*)
FROM Cliente c, Pedido p
WHERE c.ccliente = p.ccliente
GROUP BY c.cpf, c.cnome

(f) Faça uma consulta que retorne o código dos pedidos onde foram vendidos mais de 1000 itens (por exemplo, se um pedido tem 400 unidades de um item e 700 de outro, esse pedido deve ser retornado pela consulta). [0.5 ponto]

SELECT p.cpedido

```
FROM Pedido p, Item_Pedido i
WHERE p.cpedido = i.cpedido
GROUP BY c.pedido
HAVING SUM(i.quantidade) >1000;
```

(g) Faça uma consulta SQL que retorne o nome dos clientes que compraram mais de um celular em um pedido. [0.5 ponto]

```
SELECT c.cnome
FROM Cliente c, Pedido p, Item_Pedido i, Produto pr
WHERE c.ccliente = p.ccliente
AND p.cpedido=i.cpedido
AND pr.cproduto=i.cproduto
AND pr.descricao = "celular"
AND i.quantidade >1;
```

Questão 4 (2 pontos – 0.5 cada item). A partir do diagrama ER abaixo, elabore o esquema relacional correspondente e responda as questões abaixo. Obs: No diagrama, as chaves primárias estão com um * do lado.

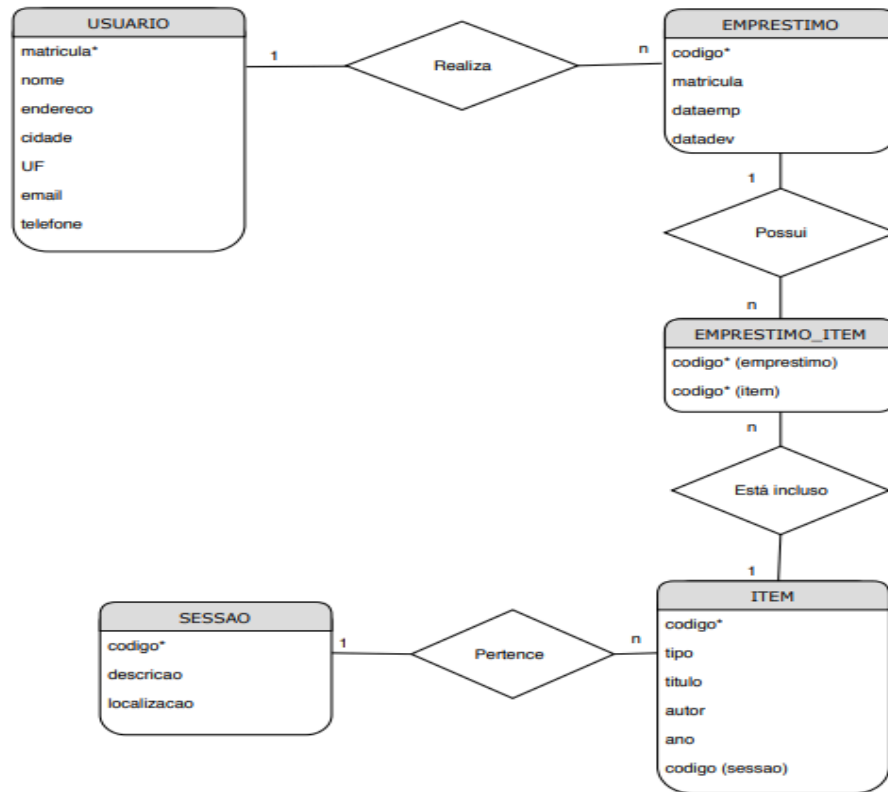


Figura 1. Diagrama gerado pela ferramenta draw.io

(a) Elabore o esquema relacional, sublinhando as chaves primárias.

USUARIO (matricula, unome, endereço, cidade, estado, Email, Telefone).

EMPRESTIMO (ecodigo, matricula, dataemp, datadev)

matricula referencia USUARIO

ITEM (icodigo, tipo, titulo, autor, ano, scodigo)

scodigo referencia SESSAO

EMPRESTIMO_ITEM (ecodigo, icodigo)

ecodigo referencia EMPRESTIMO

icodigo referencia ITEM

SESSAO (scodigo, descricao, localizacao)

(b) Se houvesse um campo para informar mais de um telefone (atributo 'telefones') na tabela USUARIO, ela estaria normalizada? Justifique.

Não. Estaria na 1FN, pois a coluna de telefones é multivalorada.

(c) Realize uma consulta em SQL que permita identificar a matricula dos usuários que nunca realizaram um empréstimo.

```
SELECT matricula FROM USUARIO u
WHERE u.matricula NOT IN (SELECT matricula FROM EMPRESTIMO);
```

- (d) Para cadastrar um dependente para cada usuário, qual seria a melhor forma de alterar a tabela USUARIO? Se houvesse uma coluna dependente em USUARIO, a tabela estaria normalizada?

Criar uma tabela DEPENDENTE, pois o usuário pode ter mais de um dependente. Se houvesse uma coluna dependente em USUARIO a tabela não estaria normalizada.

Questão 5 (2 pontos). Escreva os comandos SQL considerando as tabelas abaixo. As chaves primárias estão sublinhadas.

Funcionario (codfunc, nome, coddepto)
coddepto referencia Departamento
Departamento (coddepto, depnome, localizacao)
Dependente (coddep, dnome, datan)
DependenteFuncionario (codfunc, coddep)
codfunc referencia Funcionario
coddep referencia Dependente

- (a) **(1 ponto)** Criação das tabelas Funcionário, Dependente, DependenteFuncionario e Departamento, incluindo as restrições de integridade que se aplicam. Assuma que quando um departamento é excluído, o código de departamento dos funcionários relacionados a ele deve ser alterado para NULL. Assuma ainda que ao excluir um funcionário, todas os dependentes devem ser excluídos automaticamente. Se, por algum motivo, não for possível definir alguma restrição de integridade, justifique.

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

```
CREATE TABLE DEPENDENTEFUNCIONARIO (
  CODFUNC INT NOT NULL,
  CODDEP INT NOT NULL,
```

```
PRIMARY KEY (CODFUNC, CODDEP),  
FOREIGN KEY (CODFUNC) REFERENCES FUNCIONARIO (CODFUNC) ON DELETE CASCADE,  
FOREIGN KEY (CODDEP) REFERENCES DEPENDENTE (CODDEP) ON DELETE CASCADE  
)
```

```
CREATE TABLE DEPENDENTE (  
CODDEP INT NOT NULL ,  
DNOME VARCHAR (30),  
DATAN DATE ,  
PRIMARY KEY (CODDEP)  
)
```

```
CREATE TABLE DEPARTAMENTO (  
CODDEPTO INT NOT NULL,  
DEPNOME VARCHAR (30),  
LOCALIZACAO VARCHAR (50),  
PRIMARY KEY (CODDEPTO)  
)
```

(b) **(0.5 ponto)** Inclusão de um novo dependente, com os seguintes dados: código 3, nome “Ana Paula Silva”, data de nascimento 27/03/1995 vinculado ao funcionário código 5123.

```
INSERT INTO DEPENDENTE (CODDEP, DNOME, DATAN)  
VALUES (3, “Ana Paula Silva”, 27/03/1955);
```

```
INSERT INTO DEPENDENTEFUNCIONARIO (CODFUNC, CODDEP)  
VALUES (5123, 3);
```

(c) **(0.5 ponto)** Exclusão de todas os dependentes do funcionário “Marcos de Oliveira”.

```
DELETE FROM DEPENDENTE  
WHERE CODDEP IN (SELECT CODDEP  
FROM DEPENDENTEFUNCIONARIO d, FUNCIONARIO f  
WHERE d.CODFUNC=f.CODFUNC AND  
f.NOME=”Marcos de Oliveira”)
```