



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina Banco de Dados

AP2 2º semestre de 2009.

Nome –

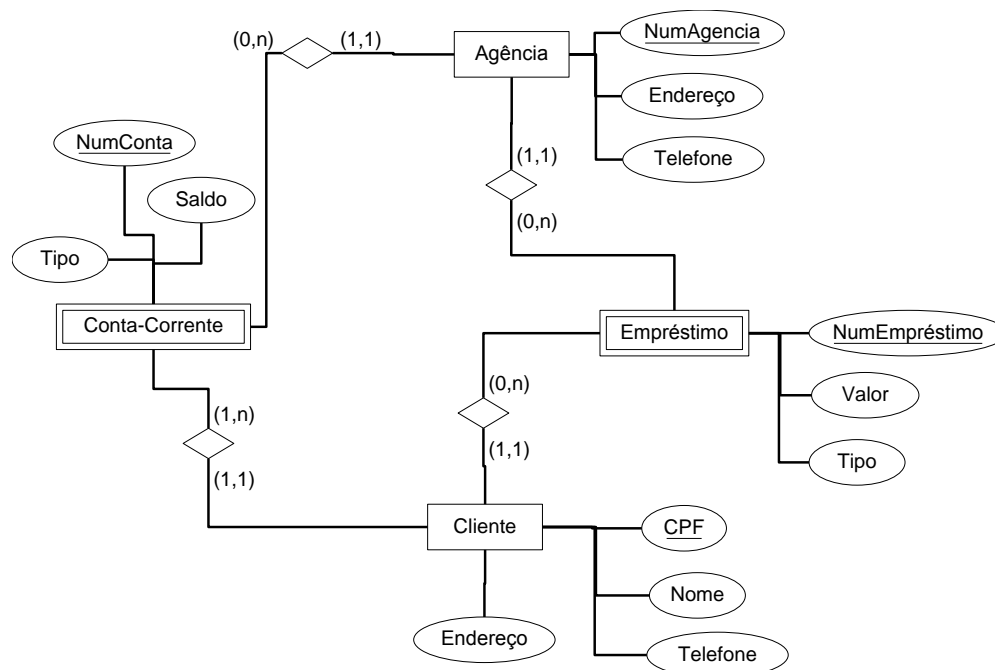
Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

Questão 1 [2,0 pontos]

Analise o modelo ER abaixo. O diagrama encontra-se na notação do DIA (ferramenta que usamos em sala de aula para construir modelos ER). Construa um esquema relacional equivalente a este diagrama ER. A base de dados deve refletir exatamente o especificado no modelo conceitual. O esquema da base de dados relacional deve conter os nomes das tabelas, os nomes dos atributos, atributos que formam a chave primária e as chaves estrangeiras.



Resposta:

Cliente (CPF, Nome, Telefone)

Agência (NumAgencia, Endereço, Telefone)

Conta-Corrente (NumAgencia, NumConta, Saldo, Tipo, CPF)

CPF referencia Cliente

Empréstimo (NumAgencia, NumEmprestimo, Valor, Tipo, CPF)

CPF referencia Cliente

CPF pode ou não ser colocado como chave primária nas tabelas Conta-Corrente e Empréstimo.

Questão 2 [6,0 pontos]

Considere as relações a seguir, retiradas da AP1.

Localidade (codLocal, nomeLocal)

codLocal	nomeLocal
1	Rio de Janeiro
2	Volta Redonda
3	Rio Bonito

Pessoa (codPessoa, nomePessoa, sexo, dataNascimento, codLocalNascimento, codUniaoPais)

codLocalNascimento referencia Localidade

codUniaoPais referencia Uniao

codPessoa	nomePessoa	sexo	dataNascimento	codLocalNascimento	codUniaoPais
1	Ana Rich	F	03/03/1950	3	NULL
2	Pedro Silva	M	02/07/1945	1	NULL
3	João Silva	M	10/02/1980	1	1
4	Maria Silva	F	15/03/1983	1	1

5	Fábio Martins	M	23/10/1978	3	1
6	Fernando Perez	M	12/09/2008	1	2
7	Tatiana Jardins	F	15/12/1977	1	NULL

Uniao (codUniao, codPessoaEsposa, codPessoaMarido, dataUniao, codLocalUniao)

codPessoaEsposa referencia Pessoa

codPessoaMarido referencia Pessoa

codLocalUniao referencia Localidade

codUniao	codPessoaEsposa	codPessoaMarido	dataUniao	codLocalUniao
1	1	2	12/10/1977	3
2	7	5	10/12/2006	1

a) Escreva uma instrução SQL para alterar a tabela Localidade, de modo a adicionar uma coluna para armazenar a população da cidade [1 ponto].

ALTER TABLE Localidade ADD população INTEGER;

b) Escreva uma instrução SQL para inserir uma pessoa com código 8, chamada Marta Moura, do sexo feminino, nascida na localidade de código 3 [1 ponto].

**INSERT INTO Pessoa (codPessoa, nomePessoa, sexo, codLocalNascimento)
VALUES (8, "Marta Moura", "F", 3);**

c) Escreva uma consulta SQL que retorna o nome da pessoa e o nome do Local de nascimento das pessoas que nasceram depois de 01/01/1980 [1 ponto]

**SELECT p.nomePessoa, l.nomeLocal
FROM Localidade l, Pessoa p
WHERE l.codLocal = p.codLocalNascimento AND
p.dataNascimento > 01/01/1980**

d) Escreva o resultado da consulta abaixo. Apresente a tabela resultante com o esquema correspondente [1 ponto].

**SELECT l.codLocal, l.nomeLocal
FROM Localidade l
WHERE l.codLocal NOT IN (SELECT codLocalNascimento FROM Pessoa)**

R(codLocal, nomeLocal)

codLocal	nomeLocal
2	Volta Redonda

e) Escreva uma consulta SQL que retorna o nome e o sexo da pessoa que seja do sexo feminino e cuja união dos pais tenha sido realizada na localidade de Rio Bonito [1 ponto].

SELECT p.nomePessoa, p.sexo

```
FROM Pessoa p, Uniao u, Localidade l
WHERE p.sexo = "F" AND
p.codUniaoPais = u.codUniao AND
u.codLocalUniao = l.codLocal AND
l.nomeLocal = "Rio Bonito"
```

f) Escreva uma consulta SQL que retorna o código da união e o número de filhos resultante dela [1 ponto].

```
SELECT codUniaoPais, COUNT(codPessoa)
FROM Pessoa
GROUP BY codUniaoPais
```

ou

```
SELECT codUniaoPais, COUNT(*)
FROM Pessoa
GROUP BY codUniaoPais
```

Questão 3 [2,0 pontos]

Considere a relação a seguir.

Tab1 (CodDept, NomeDept, SalaDept
 (CodEmp, NomeEmp
 (CodDependente, NomeDependente)))

Transforme-a para a Terceira Forma Normal, considerando que as seguintes dependências funcionais se aplicam:

CodEmp → NomeEmp
CodDept → NomeDept, SalaDept
CodDependente, CodEmp → NomeDependente

Passagem para a Primeira Forma Normal

Departamento (CodDept, NomeDept, SalaDept)
Empregado (CodDept, CodEmp, NomeEmp)
Dependente (CodDept, CodEmp, CodDependente, NomeDependente)

A tabela já se encontra na segunda e terceira formas normais