Fundação CECIERJ - **Vice Presidência de Educação Superior a Distância**

Curso de Tecnologia em Sistemas de Computação
Disciplina Banco de Dados
AP3 1° semestre de 2017.

Nome –

Assinatura –

Observações:

1. Prova sem consulta e sem uso de máquina de calcular ou qualquer outro dispositivo eletrônico, inclusive celular.
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 (6,0 pontos)

Considere a seguinte base de dados, usada por uma oficina de manutenção de automóveis (a mesma da AP2). As chaves primárias estão sublinhadas.

```
/* tabela de clientes cadastrados na oficina */
CLIENTE (cpf, nome)

/* tabela com dados dos automóveis dos clientes da oficina */
AUTOMOVEL (placa, no_chassis, modelo, cpf);
(cpf) references CLIENTE

/* tabela com as revisões periódicas programadas e feitas – para cada automóvel, a
oficina cadastra todas revisões programadas:
- Km e data_programada são a quilometragem e a data em que deve ser feita a
revisão
- data_ultim_telef serve para informar quando o pessoal da oficina ligou para
o cliente lembrando da provável necessidade de fazer a revisão – caso o cliente não
tenha sido chamado, este campo contém a string vazia (")
- data_executada e Km_executada informa a data e a quilometragem de uma
revisão que já foi executada – caso a revisão não tenha sido executada ainda, estes campos
contêm a string vazia (") */
```

```
REVISAO (placa, Km, data_programada, data_ultim_telef,
data_executada, Km_executada)
(placa) references AUTOMOVEL

/* tabela com as peças usadas em cada revisão */
PEÇA_REVISAO (placa, Km, cod_peça, quantidade)
(placa, Km) references REVISAO
(cod_peça) references PEÇA

/* tabela com as descrições das peças */
PEÇA (cod_peça, descricao_peça)
```

Sobre esta base de dados, resolver as consultas a seguir usando álgebra relacional ou SQL, de acordo com o enunciado de cada questão. Não usar mais tabelas que o estritamente necessário.

(a) Escreva uma consulta em álgebra relacional que retorne o modelo dos automóveis de cada cliente. A consulta deve retornar o CPF e nome do cliente, além do modelo do seu automóvel [1,0 ponto].

$$\pi_{cpf,nomecli,modelo}\,(Cliente\ *\ Automóvel)$$

(b) Escreva uma consulta em álgebra relacional que retorne a descrição das peças que nunca foram usadas em revisões de 10000Km [1,0 ponto].

$$\pi_{descricao\_peça}\left(Peça\ *\ \left(\pi_{cod\_peça}(Peça) - \pi_{cod\_peça}(\sigma_{km=1000}\,(Peça\_Revisão))\right)\right)$$

(c) Escreva uma instrução SQL para inserir uma peça de código R1025 e descrição "escapamento". [1,0 ponto]

```
INSERT INTO PEÇA (cod_peça, descrição_peça)
VALUES ("R1025", "escapamento")
```

(d) Faça uma consulta SQL que retorna o nome e cpf clientes que possuem automóvel modelo Fit. O resultado deve estar ordenado pelo cpf do cliente. [1,0 ponto]

```
SELECT c.nome, c.cpf
FROM Cliente c, Automovel a
WHERE c.cpf = a.cpf
AND a.modelo = "Fit"
ORDER BY c.cpf
```

(e) Faça uma consulta SQL que retorna a quantidade de peças usadas por cada veículo na revisão de 20000Km. A consulta deve retornar a placa do veículo e a quantidade total de peças utilizadas na revisão [1,0 ponto].

```
SELECT pr.placa, SUM(quantidade)
FROM Peca_Revisao pr
WHERE pr.km = 20000
GROUP BY pr.placa
```

(f) Escreva uma instrução SQL para somar 10 na quantidade de parafusos utilizados pelo veículo de placa “KYK0000” em sua revisão de 50000km. [1,0 ponto]

```
UPDATE Peca_Revisao
SET quantidade = quantidade + 10
WHERE cod_peça IN
        (SELECT cod_peça
         FROM Peça
         WHERE descrição_peça = “parafuso”)
AND placa = “KYK0000”
AND km = 50000
```

Questão 2 [2,5 pontos]

Deseja-se construir um sistema WEB que armazene resultados de alunos em uma determinada disciplina. Através de um diagrama entidade-relacionamento, deve ser modelada a base de dados que armazena os dados necessários a este sistema. A base de dados não deve conter redundância de dados. O modelo ER deve ser representado com a notação vista em aula ou com outra notação de poder de expressão equivalente. O modelo deve apresentar, ao menos, entidades, relacionamentos, atributos, especializações, identificadores e restrições de cardinalidade.

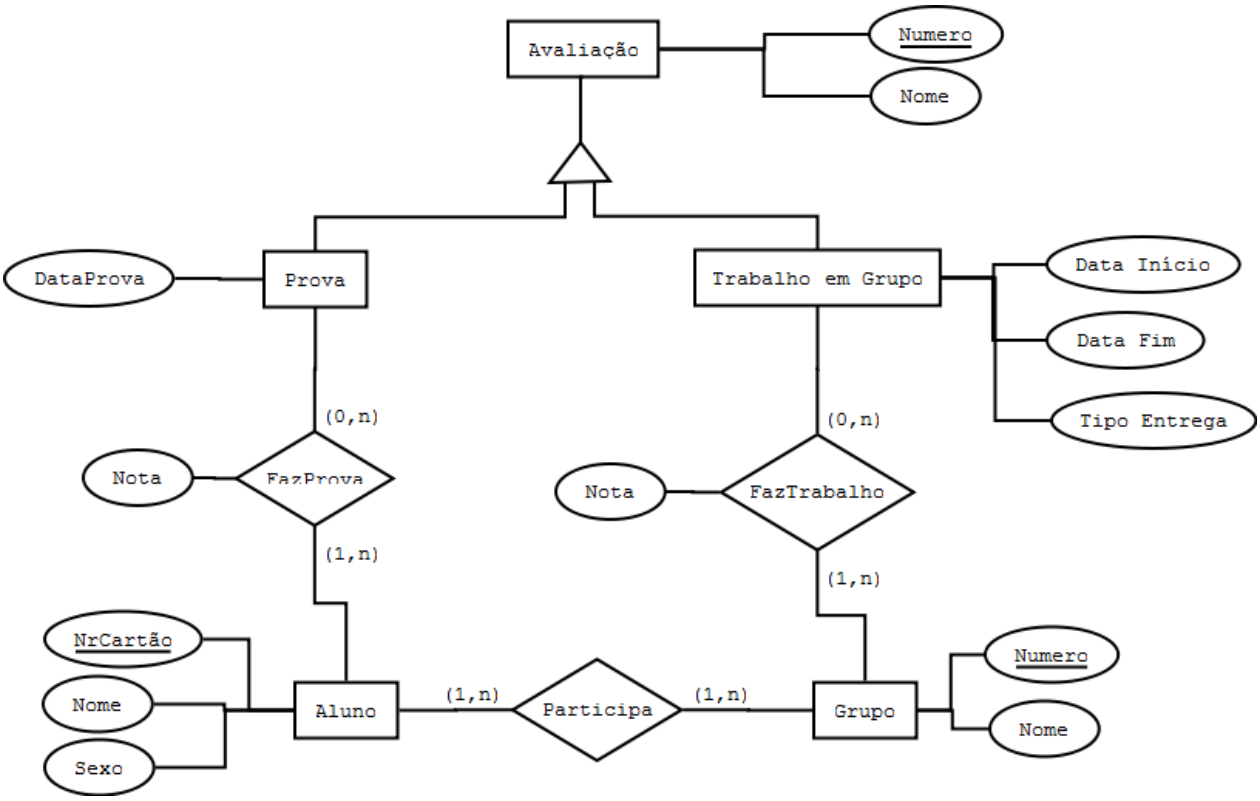
A base de dados mantém informações sobre os alunos de uma disciplina. Cada aluno é identificado por um número de cartão. Além disso, a base de dados deve manter seu nome e sexo.

Para fazer certos trabalhos, os alunos são organizados em grupos, cada um contendo vários alunos. O mesmo grupo de alunos pode fazer vários trabalhos ao longo do semestre. Em diferentes tempos, um determinado aluno pode participar de diferentes grupos. Cada grupo tem um número que serve para identificá-lo e opcionalmente um nome.

Ao longo do semestre ocorrem várias avaliações. As avaliações são identificadas por um número e têm um nome (como "Prova de álgebra relacional" ou "Trabalho de modelagem"). Uma avaliação pode ser uma prova ou um trabalho em grupo.

Uma prova ocorre em uma data determinada e é uma avaliação individual, isto é, cada aluno recebe uma nota na prova.

Em um trabalho em grupo, é atribuída uma nota para um grupo de alunos. O trabalho em grupo deve ser entregue dentro de um período pré-definido de dias (de-até) e pode ser on-line, quando os alunos fazem a entrega através do sistema WEB, ou off-line, quando fazem a entrega diretamente ao professor.



Questão 3 (1,5 ponto)
Considere a tabela abaixo, não necessariamente normalizada.

Tabela (CodModeloPC,DescricaoModeloPC,UCPModeloPC,
(CodCategCompon,NumeroModCompon, QuantidadeEmpregada, DescricaoModCompon,
DescrCategCompon))

Esta tabela foi obtida a partir de uma página WEB que lista dados de PCs. Para cada PC, são informados sua descrição e o modelo de sua UCP, bem como os componentes empregados na montagem do PC. Para cada componente, são listados: o identificador do componente, a quantidade empregada, a descrição do componente e de sua categoria.

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

(CodModeloPC) →DescricaoModeloPC
(CodModeloPC) → UCPModeloPC
(CodModeloPC,CodCategCompon,NumeroModCompon) → QuantidadeEmpregada
(CodCategCompon,NumeroModCompon) → DescricaoModCompon
(CodCategCompon) → DescrCategCompon

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 [1,5 ponto].

1FN: (eliminação das tabelas aninhadas)

Tabela1 (CodModeloPC, DescricaoModeloPC, UCPModeloPC)
Tabela2 (CodModeloPC, CodCategCompon, NumeroModCompon, QuantidadeEmpregada, DescricaoModCompon, DescrCategCompon)

2FN: (eliminação das dependências funcionais parciais)

Tabela1 (CodModeloPC, DescricaoModeloPC, UCPModeloPC)
Tabela21 (CodModeloPC, CodCategCompon, NumeroModCompon, QuantidadeEmpregada)
Tabela22 (CodModeloPC, CodCategCompon, DescricaoModCompon)
Tabela23 (CodCategCompon, DescrCategCompon)

3FN: (eliminação das dependências funcionais transitivas) = 2FN