



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina Banco de Dados

AP1 1º semestre de 2011

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 (3 pontos)

Deseja-se construir uma base de dados para armazenar informações referentes ao processo de seleção de candidatos a um programa de pós-graduação. O programa está organizado em áreas (por exemplo, "banco de dados" ou "redes de computadores"), cada área com um código e um nome. No programa atuam vários orientadores que têm também um código e um nome. Um orientador pode atuar em diferentes áreas.

Cada candidato deve, ao inscrever-se, indicar obrigatoriamente uma ou mais áreas para as quais está se inscrevendo. Adicionalmente, se assim o desejar, pode indicar com quais orientadores destas áreas está se inscrevendo. Ao indicar um orientador, o candidato deve informar o grau de convicção (alto, médio ou baixo) com o qual fez a indicação.

Além de seu nome e seu CPF (que é usado como identificador) cada candidato deve informar seu e-mail, o nome de seu curso de graduação e o nome da universidade em que fez este curso.

Os candidatos a doutorado devem, adicionalmente, informar o título de seu plano de pesquisa e o resumo de seu plano. Além disso, estes candidatos devem informar a área em que fizeram mestrado e quem foi seu orientador. Aqui há duas situações a considerar. No caso de um aluno que fez mestrado dentro do próprio programa de pós-graduação, ele deve indicar em qual das áreas do Programa fez Mestrado e qual dos orientadores do Programa foi seu orientador. Já um candidato que tenha feito Mestrado em outro

Desenhe um diagrama ER que captura as informações acima. Indique todos os identificadores e cardinalidades. Quando possível deve ser usada generalização/especialização. A base de dados modelada não deve conter redundâncias de dados. Se houver alguma informação que não pode ser concluída a partir da lista acima, indique explicitamente o que você assumiu e como isso foi refletido na modelagem (ex: a cardinalidade mínima da entidade X no relacionamento Y foi definida como 1, pois assumi que X era obrigatório no relacionamento Y).

Diagrama de Modelo Entidade-Relacionamento (MER) para o sistema de Mestrado e Doutorado.

Entidades e Atributos:

- Candidato** (Entidade Base): CPF, Nome, Email, Curso Graduação, Universidade Graduação.
- Candidato a Doutorado** (Entidade Derivada): Titulo do Plano, Resumo do Plano.
- Mestrado Fora da Instituição** (Entidade Derivada): Nome do Orientador, Nome da área do Mestrado, Nome da Universidade do Mestrado.
- Mestrado na Instituição** (Entidade Derivada): (Atributos implícitos).
- Área**: Código, Nome.
- Orientador**: Código, Nome.
- Indicação Orient**: Grau Certeza.

Relacionamentos e Cardinalidades:

- Indicação de Área** (Relacionamento): Conecta **Candidato** (0,n) a **Área** (1,n).
- Indicação Orient** (Relacionamento): Conecta **Candidato a Doutorado** (0,n) a **Orientador** (0,n).
- Área** e **Orientador** (Relacionamento): Conecta **Área** (1,n) a **Orientador** (1,n).
- Área** e **Orientador** (Relacionamento): Conecta **Área** (1,1) a **Orientador** (0,n).
- Área** e **Orientador** (Relacionamento): Conecta **Área** (1,n) a **Orientador** (1,1).

Considere a seguinte base de dados, usada por uma oficina de manutenção de automóveis. As chaves primárias estão sublinhadas.

```
CLIENTE (cpf, nome_cli)
```

/* 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.
Não usar mais tabelas que o estritamente necessário.

(a) Faça uma consulta que retorna os nomes dos clientes que possuem automóvel do modelo X [0,5 ponto].

$\pi_{\text{nomecli, (Cliente} \bowtie (\sigma_{\text{modelo} = \text{"X"}} \text{Automovel))}$

(b) Faça uma consulta que retorna os nomes dos clientes e o modelo do automóvel, cujo automóvel teve uma revisão executada na data de 2011-02-20 [0,5 ponto].

$\pi_{\text{nomecli, modelo, (Cliente} \bowtie \text{Automovel} \bowtie (\sigma_{\text{data_executada} = \text{"2011-02-20"}} \text{Revisao))}$

(c) Faça uma consulta que retorna a descrição das peças que nunca foram usadas em revisões [1 ponto].

$\pi_{\text{descricao_pecã}} (\text{Peça} \bowtie (\pi_{\text{cod_pecã}} (\text{Peça}) - \pi_{\text{cod_pecã}} (\text{Peça_Revisao})))$

(d) Obter as placas e km dos automóveis com a data programada da próxima revisão dos automóveis que, em uma determinada revisão, utilizaram mais do que 3 peças com descrição = “VELA” [1 ponto].

```
 $\pi$  placa, data_executada
(  $\sigma$  descricao_peca = "VELA" AND quantidade > 3
  (REVISAO  $\bowtie$  PECA_REVISAO  $\bowtie$  PECA)
)
```

Questão 3 (1 ponto)

Utilizando o esquema da questão 2, analise as consultas abaixo e diga qual o esquema da tabela retornada por cada uma das consultas. O esquema deve ser informado utilizando a seguinte sintaxe:

```
Tab (TabOrigem1.Atrib1, TabOrigem1.Atrib2, ..., TabOrigemN.Atrib1,
TabOrigemN.Atrib2, ...)
```

Nesta notação, tabOrigem_i é o nome da tabela de onde veio o atributo Atrib_j originalmente.

(a) $\sigma_{\text{quantidade} > 2}$ (Peça \bowtie Peça_Revisão)

```
Tab (Peça.cod_peça, Peça.descricao_peça, Peça_revisão.placa, Peça_revisão.km,
Peça_revisão.data_programada, Peça_revisão.data_ultim_telef,
Peça_revisão.data_executada, Peça_revisão.Km_executada)
```

(b) π_{nomecli} (Cliente)

```
Tab (Cliente.nomeCli)
```

Questão 4 (3 pontos)

Que recursos um SGBD possui para prover *independência de dados e acesso eficiente* aos dados?

Os SGBDS possuem três níveis de representação de dados, e a definição de mapeamentos entre os níveis. Esses esquemas isolam os dados de características internas de armazenamento físico e representação lógica. Os esquemas lógico e externo provêm independência das representações de dados quanto ao armazenamento físico e projeto lógico respectivamente. Assim, estruturas de dados podem evoluir à medida que novos requisitos são definidos.

Para prover acesso eficiente, o SGBD possui mecanismos eficientes de armazenamento e acesso aos dados, contando com a gerência de arquivos muito grandes, estruturas de índices e otimização de consultas.