



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Banco de Dados
AD2 2º semestre de 2017

Nome: _____

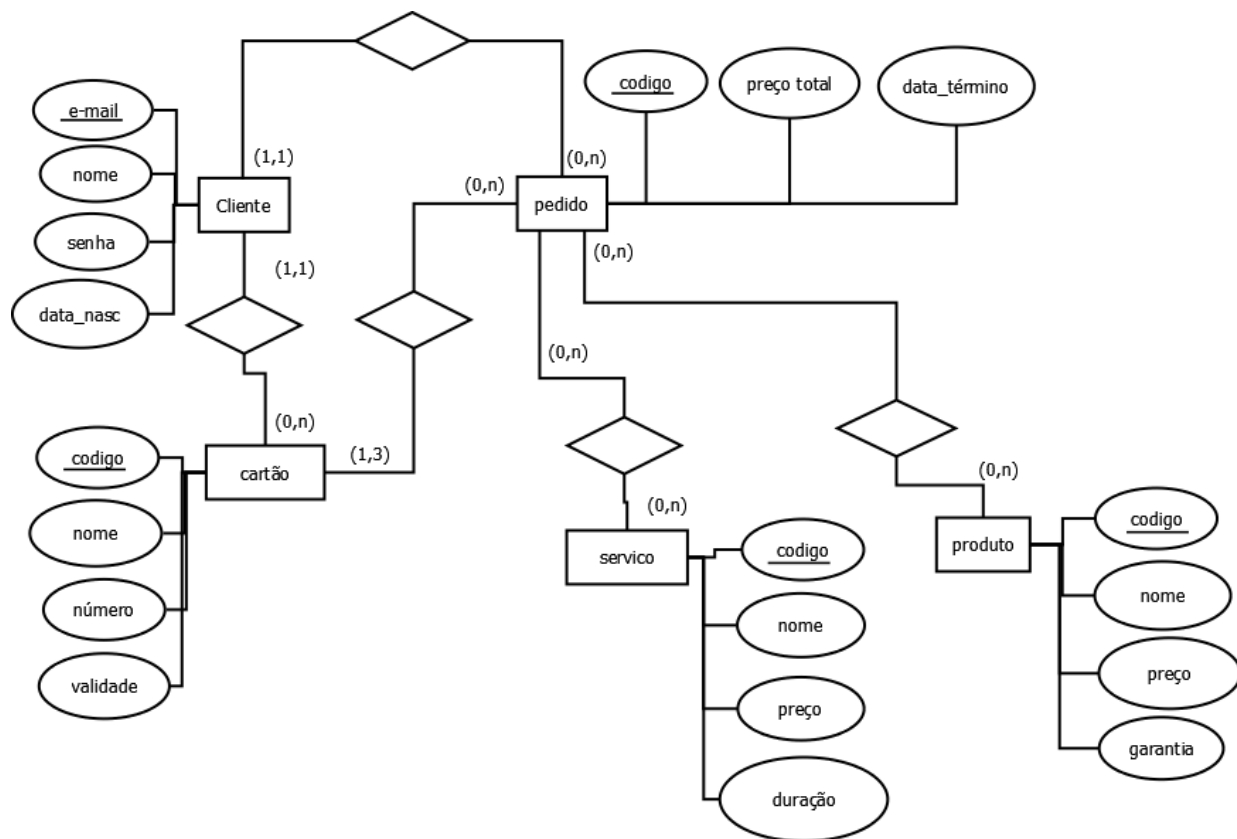
Observações:

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. [4,0 pontos] Considere o diagrama Entidade-Relacionamento (ER) mostrado a seguir. Esse diagrama apresenta as principais entidades e relacionamentos envolvidos na modelagem de uma base de dados para gerenciar os pedidos em uma oficina mecânica, considerando também o pagamento de cada pedido. De acordo com esse diagrama, cada cliente pode possuir vários pedidos de serviços e produtos cadastrados no sistema, assim como cada pedido pode ser pago pelo cliente utilizando, no mínimo, 1 e, no máximo, 3 cartões de crédito. Ao mesmo tempo, o mesmo cartão pode ter sido utilizado diversas vezes em pedidos diferentes.

Construa um esquema relacional equivalente a este diagrama ER, indicando chaves primárias e estrangeiras.



As chaves primárias estão sublinhadas.

Cliente (email, nome, senha, data_nasc)

Cartao (codigo, cliente_email, nome, numero, validade)

cliente_email REFERENCIA Cliente

Cartao_Pedido (cartao_codigo, pedido_codigo)

cartao_codigo REFERENCIA Cartao

pedido_codigo REFERENCIA Pedido

Pedido (codigo, cliente_email, preco_total, data_termino)

cliente_email REFERENCIA Cliente

Produto (codigo, nome, preco, garantia)

Pedido_Produto (pedido_codigo, produto_codigo)

pedido_codigo REFERENCIA Pedido

produto_codigo REFERENCIA Produto

Servico (codigo, nome, preco, duracao)

Pedido_Servico (pedido_codigo, servico_codigo)

pedido_codigo REFERENCIA Pedido

servico_codigo REFERENCIA Servico

Questão 2. [4,5 pontos] Considere o esquema relacional a seguir para o sistema de uma loja de biscoitos online, onde as chaves primárias estão sublinhadas.

Usuario (uid:integer, login:string, email:string)

Cartao_Fidelidade (cid:integer, uid:integer, data_criacao:date, pontos_acumulados:integer)

uid REFERENCIA Usuario

Biscoito (bid:integer, nome:string, preco:float)

Ingrediente (iid:integer, nome:string, descricao:string)

Biscoito_Ingrediente (bid:integer, iid:integer)

bid REFERENCIA Biscoito

iid REFERENCIA Ingrediente

Pedido (pid:integer, uid:integer, data:date, hora:hour, prazo_entrega:integer, entregue_prazo:boolean)

uid REFERENCIA Usuario

Pedido_Biscoito (pid:integer, bid:integer, quantidade:integer)

pid REFERENCIA Pedido

bid REFERENCIA Biscoito

Sobre esta base de dados, resolver as consultas utilizando SQL. Não usar mais tabelas que o estritamente necessário.

a) Escreva uma instrução SQL para inserir uma tupla na tabela Biscoito com identificador 1, sendo que tal biscoito possui o nome “chocochips” e custa R\$ 2,50. [0,5 ponto]

INSERT INTO Biscoito(bid, nome, preco)

VALUES (1, “chocochips”, 2,50);

b) Escreva uma instrução SQL que cria a tabela Cartao_Fidelidade (cartão de fidelidade) com as colunas especificadas no enunciado. Adicione uma restrição que garanta que, ao excluir um determinado usuário do banco de dados, todos os cartões de fidelidade criados para ele sejam excluídos automaticamente. [0,5 ponto]

```
CREATE TABLE Cartao_Fidelidade (  
    cid integer,  
    uid integer,  
    data_criacao date,  
    pontos_acumulados integer,  
    PRIMARY KEY (cid),  
    FOREIGN KEY (uid) REFERENCES Usuario(uid) ON DELETE CASCADE  
);
```

c) Faça uma consulta que retorna o identificador, a data e a hora dos pedidos que possuem, pelo menos, um biscoito com o nome chocochips. [0,5 ponto]

```
SELECT p.pid, p.data, p.hora  
  
FROM Pedido p, Pedido_Biscoito pb, Biscoito b  
  
WHERE p.pid = pb.pid  
  
AND pb.bid = b.bid  
  
AND b.nome = "chocochips";
```

d) Faça uma consulta que retorna os identificadores dos pedidos em que os usuários solicitaram mais de 10 biscoitos diferentes. [0,5 ponto]

```
SELECT p.pid  
  
FROM Pedido p, Pedido_Biscoito pb  
  
WHERE p.pid = pb.pid  
  
GROUP BY p.pid  
  
HAVING COUNT(pb.bid) > 10;
```

e) Crie uma visão que contenha o identificador do pedido e o seu preço total, sendo que esse preço total corresponde à soma da multiplicação do preço pela quantidade de cada biscoito solicitado. [0,5 ponto]

```
CREATE VIEW preco_total_pedido AS  
  
SELECT p.pid, SUM(b.preco*pb.quantidade)  
  
FROM Pedido p, Pedido_Biscoito pb, Biscoito b  
  
WHERE p.pid = pb.pid  
  
AND pb.bid = b.bid  
  
GROUP BY p.pid;
```

f) Escreva uma instrução SQL para excluir todos os biscoitos que possuem chocolate branco como ingrediente e que são vendidos por mais de R\$ 3,50. [1 ponto]

```
DELETE FROM Biscoito  
  
WHERE preco > 3.50  
  
AND bid IN (  
  
    SELECT bi.bid  
  
    FROM Biscoito_Ingrediente bi, Ingrediente i  
  
    WHERE bi.iid = i.iid  
  
    AND i.nome = "chocolate branco"  
  
);
```

g) Faça uma consulta que retorna os nomes e os preços dos biscoitos que não apresentam chocolate meio amargo como ingrediente. [1 ponto]

```
SELECT nome, preco  
  
FROM Biscoito  
  
WHERE bid NOT IN (  
  
    SELECT DISTINCT bi.bid  
  
    FROM Biscoito_Ingrediente bi, Ingrediente i  
  
    WHERE bi.iid = i.iid  
  
    AND i.nome = "chocolate meio amargo"  
  
);
```

Questão 3 (1,5 ponto). Devido ao grande volume de livros no seu estoque, o dono de uma livraria desenvolveu um sistema para gerenciar esses livros. Assim, ele propôs definir todas as categorias de livros existentes na loja, assim como alocar cada livro a apenas uma categoria. Um projetista de banco de dados propôs as seguintes tabelas como solução para armazenar os dados referentes aos livros, supondo que cada livro pertença a apenas uma editora. As chaves primárias estão sublinhadas.

Livro (CodLivro, CodCategoria, NomeLivro, Sinopse, Preco, CodEditora, NomeEditora)

Categoria (CodCategoria, NomeCategoria, DescricaoCategoria)

O significado de cada coluna é o seguinte:

- CodLivro: código do livro
- CodCategoria: código da categoria
- NomeLivro: nome do livro
- Sinopse: Sinopse do livro
- Preco: Preço do livro
- CodEditora: Código da editora que publicou o livro
- NomeEditora: Nome da editora
- NomeCategoria: Nome da categoria do livro
- DescricaoCategoria: Descrição da categoria

As dependências funcionais (podendo incluir dependências transitivas) existentes nestas tabelas são as seguintes:

- CodLivro → NomeLivro, Sinopse, Preco, CodCategoria, CodEditora
- CodCategoria → NomeCategoria, DescricaoCategoria
- CodEditora → NomeEditora

a) Assumindo que o profissional não conhece o conceito de normalização, explique para ele em que forma normal encontra-se a tabela. [0,5 ponto]

Está na segunda forma normal (2FN), pois ainda existe um atributo não-chave (NomeEditora) que possui dependência funcional parcial.

b) Caso a tabela não se encontre na terceira forma normal, mostre a(s) transformação(ões) para a terceira forma normal. Mostre cada forma normal intermediária, entre aquela em que a tabela se encontra e a terceira forma normal. [1,0 ponto]

Passagem para a Terceira Forma Normal:

Categoria (CodCategoria, NomeCategoria, DescricaoCategoria)

Editora (CodEditora, NomeEditora)

Livro (CodLivro, NomeLivro, Sinopse, Preco, CodCategoria, CodEditora)

CodCategoria REFERENCIA Categoria

CodEditora REFERENCIA Editora