



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina Banco de Dados**

**AP2 2º semestre de 2017**

**Nome –**

**Assinatura –**

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular, celular ou qualquer dispositivo eletrônico.
  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)**

Considere a visão Pedidos\_Usuario abaixo, conforme o esquema da Questão 2.

```
CREATE VIEW PEDIDOS_USUARIO AS
SELECT u.login, u.email, p.pid, p.data
FROM usuario u, pedido p
WHERE u.uid = p.uid
```

Considere a consulta do usuário que seleciona o login dos usuários que fizeram pedidos no dia “2017-10-20”.

```
SELECT login
FROM Pedidos_Usuarios
WHERE p.data = “2017-10-20”
```

(a) Assumindo que o SGBD utiliza a técnica de reescrita de consulta, qual será a consulta SQL que o SGBD vai executar para responder à consulta do usuário acima? [1,0 ponto]

```
SELECT u.login
FROM usuario u, pedido p
WHERE u.uid = p.uid
AND p.data = “2017-10-20”
```

(b) Explique sua resposta [1,0 ponto].

*Na técnica de reescrita de consulta, o SGBD utiliza a consulta de definição da visão. Em cima dessa consulta, adiciona os predicados de seleção da consulta do usuário, e os campos especificados na cláusula SELECT para compor uma nova consulta, que será executada pelo SGBD. Se a consulta do usuário especificar junções, elas também serão incorporadas à consulta reescrita. O resultado da consulta composta é o mesmo resultado que seria obtido se a consulta do usuário fosse executada sobre a visão materializada.*

### **Questão 2 (6,0 pontos)**

Considere o esquema relacional, usando pelo sistema de uma loja de biscoitos online (o mesmo usado na AD2). 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

# Tabela de Pedidos. O atributo data contém a data em que o pedido foi
realizado, e o atributo entregue_prazo registra se o pedido foi entregue no
prazo estipulado no pedido. Para pedidos ainda não entregues, o valor de
entregue_prazo é armazenado como NULL.

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 a base de dados correspondente a esse esquema, resolver as questões a seguir usando SQL. Não usar mais tabelas que o estritamente necessário.

(a) Escreva uma instrução SQL para excluir os usuários que nunca fizeram nenhum pedido. [1 ponto]

```
DELETE FROM usuario  
WHERE uid NOT IN (SELECT uid FROM pedido)
```

(b) Escreva uma instrução SQL para criar a tabela Pedido\_Biscoito, de forma que quando um pedido seja removido do sistema, as tuplas correspondentes de Pedido\_Biscoito sejam automaticamente removidas. [1 ponto]

```
CREATE TABLE Pedido_Biscoito (  
  pid INTEGER NOT NULL,  
  bid INTEGER NOT NULL,  
  quantidade INTEGER,  
  PRIMARY KEY(pid, bid),  
  FOREIGN KEY (pid) REFERENCES Pedido ON DELETE CASCADE,  
  FOREIGN KEY (bid) REFERENCES Biscoito  
)
```

(c) Faça uma consulta SQL que retorna os emails dos usuários que possuem mais de 100 pontos no programa de fidelidade da loja, que fizeram pedido de algum biscoito na data “2017-10-10” e que receberam o pedido fora do prazo estipulado. [1,0 ponto]

```
SELECT u.email  
FROM usuario u, cartao_fidelidade c, pedido p  
WHERE p.data = “2017-10-10”  
AND p.entregue_prazo = False  
AND c.pontos_acumulados > 100  
AND u.uid = c.uid  
AND u.uid = p.uid
```

(d) Faça uma consulta SQL que retorna os números dos pedidos (*pid*) realizados em “2017-10-10,” juntamente com o valor total desses pedidos, ordenados pelo valor total dos pedidos. [1,0 ponto]

```
SELECT p.pid, SUM(pb.quantidade * b.preco) AS total  
FROM pedido p, pedido_biscoito pb, biscoito b  
WHERE p.data = “2017-10-10”  
AND p.pid = pb.pid  
AND pb.bid = b.bid  
GROUP BY pid  
ORDER BY total
```

(e) Faça uma consulta SQL que retorna o nome do biscoito mais caro que tenha sido vendido em algum pedido de “2017-10-10”. [1,0 ponto]

```
SELECT b.nome, MAX(b.preco)
FROM biscoito b
WHERE b.bid IN (SELECT bp.bid FROM biscoito_pedido bp, pedido p
                WHERE bp.pid = p.pid
                AND p.data = “2017-10-10”)
```

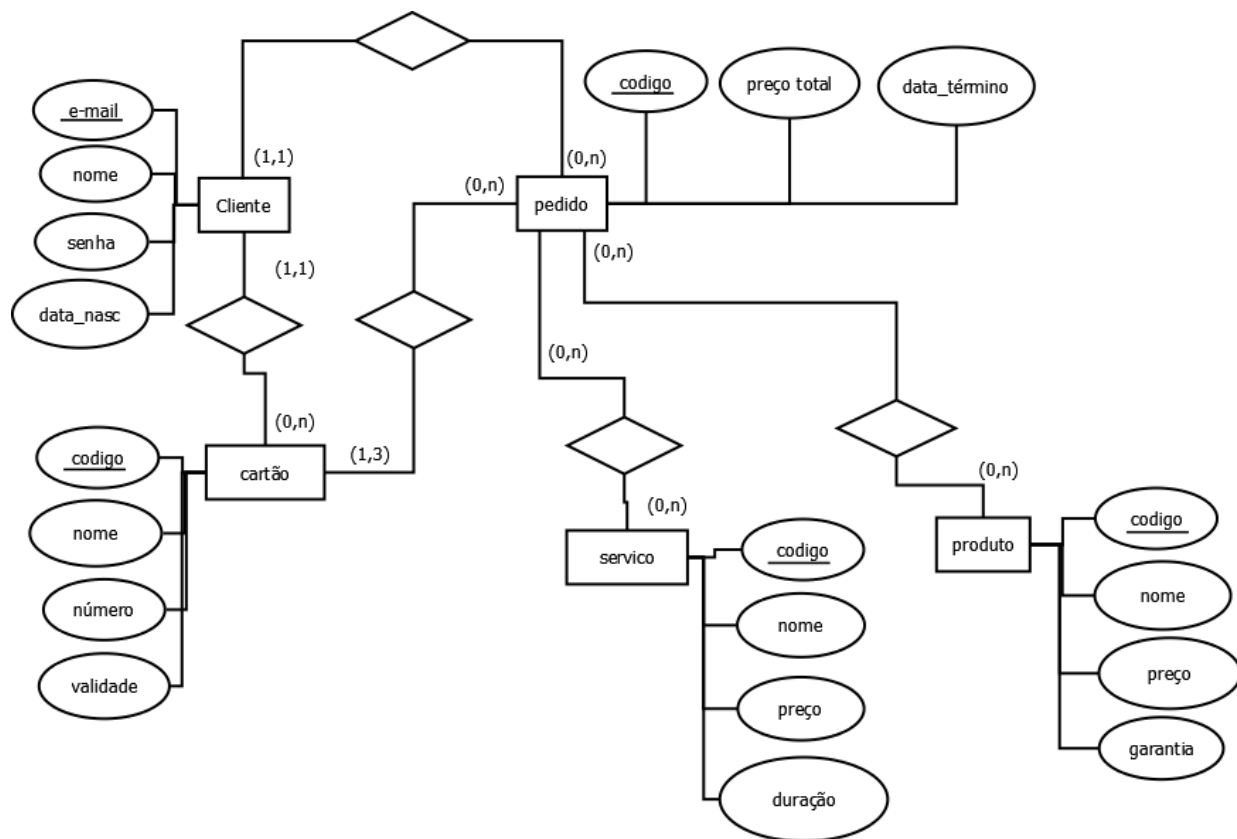
(f) Faça uma consulta SQL que retorna os nomes dos ingredientes necessários para fazer os biscoitos que constem em algum pedido realizado no mês de outubro de 2017. [1,0 ponto]

```
SELECT i.nome
FROM ingrediente i, biscoito_ingrediente bi, pedido p, pedido_biscoito pb
WHERE p.data >= “2017-10-01”
AND p.data <= “2017-10-31”
AND p.pid = pb.pid
AND pb.bid = bi.bi
AND bi.iid = i.iid
```

### **Questão 3 (2,0 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\_termينو)*

*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*