

Desenvolvimento de Aplicações com Banco de Dados

Prof. Jorge Viana Doria Junior, M.Sc.

Atividade Prática Supervisionada

1. Visão Geral da Atividade

Desenvolver uma aplicação PDV (Ponto de Venda), utilizando a programação orientação a objetos em **JAVA Swing** e o banco de dados **MySQL**, para efetuar a **INCLUSÃO** e a **EXCLUSÃO** na venda de produtos utilizando as classes/tabelas abaixo: **(2,0 pontos)**

Classe/Tabela: CLIENTE

Atributo	Tipo	Chaves e Descrições
CODCLI	Autonumeração	(PK)
NOME	Texto(35)	
BONUS	Numero	Somente valores superiores a 100 pontos
PERFIL	Texto(1)	P - pequeno; M - médio ou G - Grande
STATUS	Texto(1)	A - ativo ou I - Inativo

Classe/Tabela: LOCALIDADE

Atributo	Tipo	Chaves
CODLOCAL	Autonumeração	(PK)
NOME	Texto(35)	
ENDERECO	Texto(80)	
TELEFONE	Texto(14)	

Classe/Tabela: PRODUTO

Atributo	Tipo	Chaves
CODPROD	Autonumeração	(PK)
CODLOCAL	Numero	(FK)
DESCRICAO	Texto(35)	
QTD_ESTOQUE	Numero	
PRECO_UNITARIO	Moeda	

Classe/Tabela: VENDA

Atributo	Tipo	Chaves
CODCLI	Numero	(PK) e (FK)
CODPROD	Numero	(PK) e (FK)
CODLOCAL	Numero	(PK) e (FK)
QTD_VENDA	Numero	
VALOR_TOTAL	Moeda	
DATA_VENDA	Data	

ALGORITMO DA TRANSAÇÃO DE VENDA: **CADASTRAR VENDA**

Início do Programa INCLUIR_VENDA;

*** OBTER DADOS JUNTO AO USUÁRIO ***

Ler CodCli, Codprod, CodLocal, Quantidade

*** TRATAR ESTOQUE E A EXISTÊNCIA DO PRODUTO ***

EXEC SQL SELECT descricao, local_fab, qtd_estoque, preco_unitario
From Produto Where Codprod = :**Xcodprod**

Se SQL Not Ok! **então**

┌ **Escreva** "Produto não encontrado!";
└ Sair;

Fim se

Se (qtd_estoque < Quantidade) **então**

┌ **Escreva** "Produto sem estoque!";
└ Sair;

Fim se

EXEC SQL UPDATE Produto set qtd_estoque = qtd_estoque - :**QuantidadeVendida**
WHERE Codprod = :**Xcodprod**;

*** CALCULAR O PREÇO TOTAL ***

total = total + (Quantidade * preço);

Escreva "Preço calculado: ", total

*** CALCULAR DESCONTO 1: Com base no bônus e na tabela de Descontos ***

EXEC SQL EXECUTE

BEGIN

total ← CALCULAR_DESCONTO(CodProd, CodCli, Quantidade: Inteiro; Total: Real): Real;

END;

END-EXEC

*** CALCULAR DESCONTO 2: Compras no mesmo local de fabricação tem 10% de desconto ***

Se (CodLocal = LNPROD) **então**

total = total – (total * 10/100);

Escreva “Valor calculado da venda: ”, total;

Fim se

*** GRAVAR VENDA ***

EXEC SQL INSERT INTO Venda (CodCli, CodProd, CodLocal, Qtd_venda, valor_total, data_venda)
VALUES (:XCodcli, :XCodPro, :XCodLocal, :XQuantidade, :Xtotal, :Xdatasistema);

Se SQL = OK **então**

Escreva “Venda incluída com sucesso!”;

Exec SQL Commit;

Senão

Escreva “Erro na gravação da venda!”;

Exec SQL Rollback;

Fim se

*** LISTAR ITENS VENDIDOS ***

EXEC SQL SELECT b.descricao, a.Qtd_venda, a.preco_unitario, a.valor_total
From Venda a, Produto b, Cliente c
Where a.Codprod = b.CodProd
And c.CodCli = :Xcodcli;

Fim do Programa INCLUIR_VENDA

ALGORITMO DA TRANSAÇÃO DE VENDA: **EXCLUIR VENDA**

*** DELETAR VENDA ***

Ler codcli, codprod, dt_venda

Exec SQL Select Total Into VTOT, QTDE Into VQTDE
Codlocal Into Vlocalvenda
From Venda
Where codcli = :codcli And codpro = :codprod and
Datavenda = :dt_venda FOR UPDATE;

Se SQLCA.SQLCODE != 0 **then**
[**Escreva** "Venda Inexistente!";
[**Exec SQL** Rollback;
[Continue;
Fim se;

*** TRATAR ESTOQUE ***

Exec SQL Update Produto SET QTDE = QTDE + VQTDE
Where PN = :codProd;

Se SQLCA.SQLCOD != 0 **then**
[**Exec SQL** Rollback;
[Continue;
Fim se;

*** DEVOLVER BONUS DO CLIENTE ***

Exec SQL Select precounitario Into VPRECO, codlocal Into Vlocalprod
From produto Where PN = :codpro;

VLRTOTALREAL = VPRECO * VQTDE;

Se Vlocalvenda == Vlocalprod **then**
[VLRDESCLOCAL = VLRTOTALREAL * 0.10;
[VTOT = :VTOT + VLRDESCLOCAL;
Fim se;

Exec SQL Update Cliente SET BONUS = BONUS + 100
Where CN = :codcli And :VTOT < :VLRTOTALREAL;

*** DELETAR A VENDA ***

Exec SQL Delete From Venda Where cód_cli = :codcli and datavenda = :dt_venda;

Se SQLCA.SQLCODE != 0 **then**
[**Escreva** "Venda não excluída!"
[**Exec SQL** Rollback;
[Continue;
Else
[**Escreva** "Venda excluída com sucesso!"
[**Exec SQL** Commit;
Fim se;

2) Regra para a construção da **Function** CALCULAR_DESCONTO

Clientes que possuem mais de 100 pontos de bônus tem direito a um desconto, e caso ganhe o desconto pagará 100 pontos de seus bônus. Produtos com descontos são aqueles que estão na tabela desconto com uma determinada quantidade.

Classe/Tabela: DESCONTO

Atributo	Tipo	Chaves e descrições
ID_DESCONTO	Autonumeração	(PK)
CODPROD	Numero	(FK)
PERCENTUAL	Numero	Percentual de desconto Ex: 2, 8, 10, 12...
QTD_MIN	Numero	Quantidade mínima para desconto < QTD_MAX
QTD_MAX	Numero	Quantidade máxima para desconto > QTD_MIN

Observação:

Os produtos podem se repetir, pois terão descontos diferentes para uma determinada quantidade.

ALGORITMO DA FUNÇÃO: **CALCULAR_DESCONTO**

FUNCTION CALCULAR_DESCONTO (CodProd, CodCli, Quantidade: Inteiro; Total: Real): Real;

INÍCIO DA FUNCTION

EXEC SQL SELECT Bonus

From Cliente

Where Codcli = :Xcodcli;

Se Bonus >= 100 **então**

EXEC SQL SELECT Percentual

From Desconto

Where CodProd = :Xcodpro AND

Qtd_min >= :XQuantidade AND Qtd_max <= :XQuantidade;

Se SQL OK! **então**

Retorne \leftarrow valor_total - (valor_total * Percentual/100);

EXEC SQL UPDATE Cliente SET Bonus = Bonus - 100

Where Codcli = :Xcodcli;

Fim se

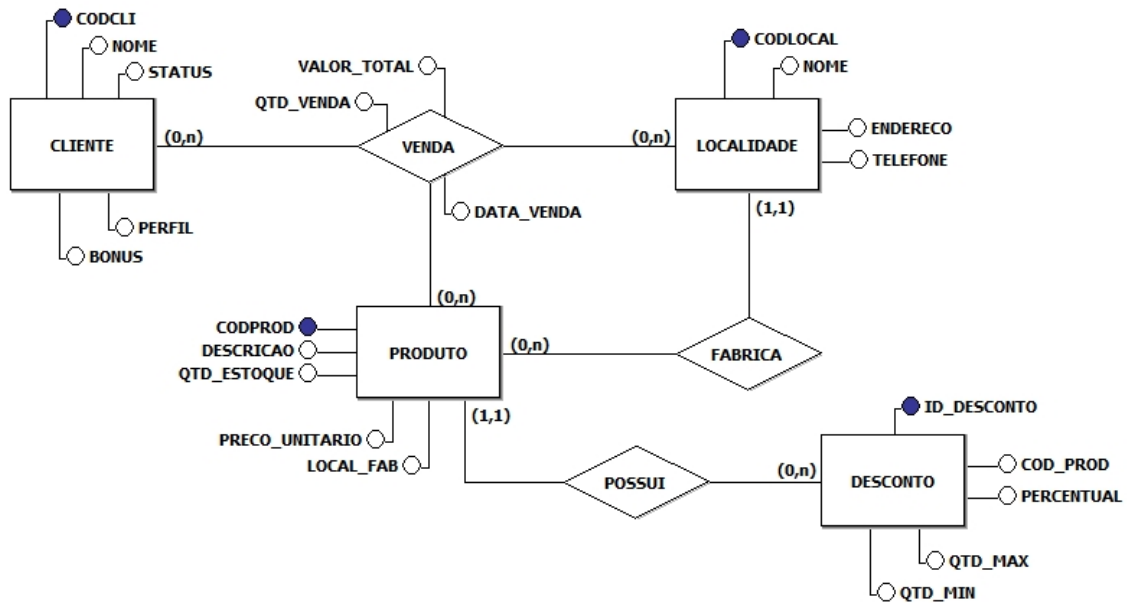
senão

Escreva "Cliente não possuirá desconto!";

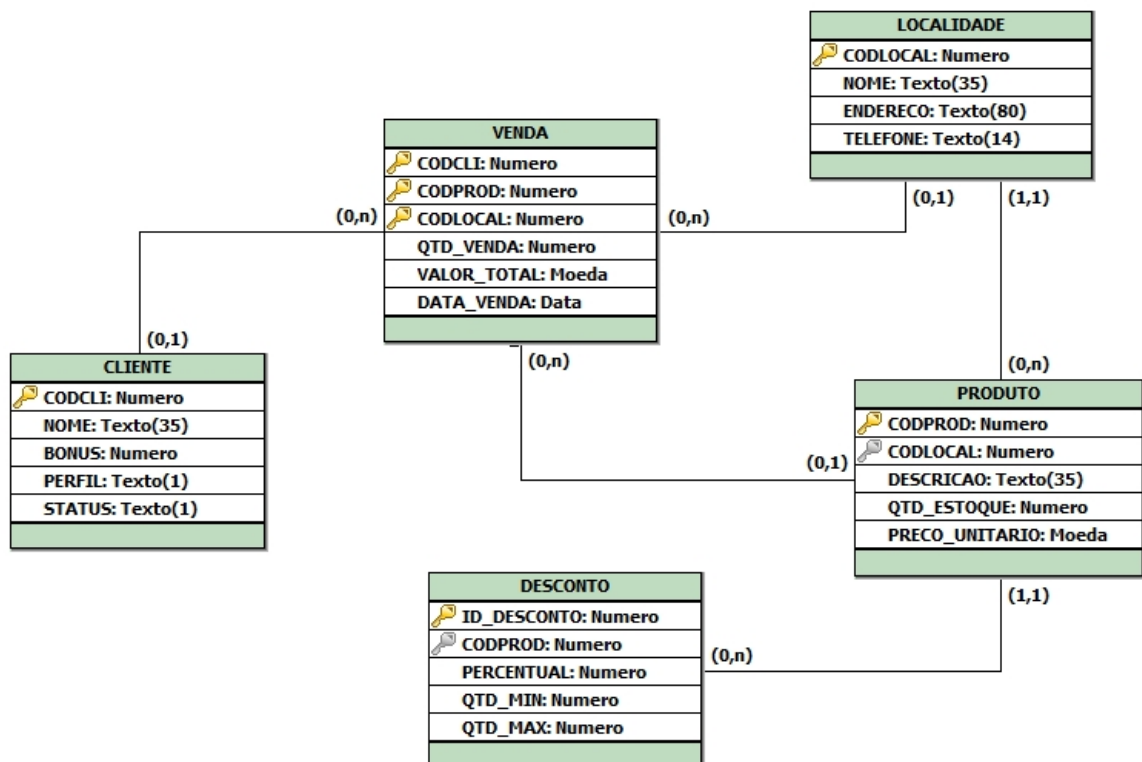
Fim se

FIM DA FUNCTION

MODELO CONCEITUAL DO BANCO DE DADOS



MODELO LÓGICO DO BANCO DE DADOS



PROPOSTA DE PROTÓTIPO DO SISTEMA

PDV - Ponto de Venda

CAIXA LIVRE


Cliente Selecionado: Local de venda:

Código do Produto: Quantidade:

Descrição do produto

Produto	Quantidade	Preço Unitário	Valor Total

Total da Compra:

 Fechar

2. Critérios de Avaliação

As tarefas totalizam 20% da AV2.

3. Quanto à Entrega do Trabalho

O dia da entrega do trabalho será em **22/11/2019**. Deverá ser entregue **SOMENTE** pelo AVA para avaliação do professor. Utilize a tarefa **ATIVIDADE PRÁTICA SUPERVISIONADA** para upload do arquivo **ZIP** (contendo o código fonte e o script do banco de dados). Façam as resoluções nas ferramentas **NetBeans 8.x** ou **Eclipse IDE**.