FACULDADE DE TECNOLOGIA DE SÃO PAULO

Functions e Exercícios de Impressão de Pedidos

|  |  |
| --- | --- |
| Nº matrícula | Nome |
| 11206012 | Alexandre Bernardo de Almeida |
| 12100152 | Carlos Kazuo Mochizuki Ishizaka |
| 12100045 | Guilherme da Silva Gonçalves |
| 12110038 | Leonardo Tatsuyoshi Onishi |
| Data: 27/04/2015 |  |

|  |  |
| --- | --- |
| Disciplina: | LBD |
| Prof.: | Bernice |

SÃO PAULO

2015

1. Escrever rotina utilizando cursor para imprimir pedidos conforme o formulario anexo.

CREATE TABLE Clientes (

IdCliente INT NOT NULL IDENTITY(1,1),

Nome VARCHAR(1024) NOT NULL,

CGC\_CPF VARCHAR(50),

ENDERECO VARCHAR(1024),

BAIRRO VARCHAR(512),

CEP VARCHAR(15),

CIDADE VARCHAR(1024),

UF CHAR(02),

CONSTRAINT PK\_Clientes\_IdCliente PRIMARY KEY(IdCliente)

)

GO

CREATE TABLE Vendedores (

IdVendedor INT NOT NULL IDENTITY(1,1),

Nome VARCHAR(1024) NOT NULL,

CONSTRAINT PK\_Vendedores\_IdVendedor PRIMARY KEY(IdVendedor)

)

GO

CREATE TABLE Itens (

IdItem INT IDENTITY(1,1) NOT NULL,

Descricao VARCHAR(50) NOT NULL,

Valor\_Unitario SMALLMONEY,

CONSTRAINT PK\_Itens\_IdItem PRIMARY KEY(IdItem)

)

GO

CREATE TABLE Pedidos (

IdPedido INT IDENTITY(1,1) NOT NULL,

Data DATETIME NOT NULL,

IdCliente INT NOT NULL,

IdVendedor INT NOT NULL,

CONSTRAINT PK\_Pedidos\_IdPedido PRIMARY KEY(IdPedido),

CONSTRAINT FK\_Pedidos\_Clientes\_IdCliente FOREIGN KEY(IdCliente) REFERENCES Clientes (IdCliente),

CONSTRAINT FK\_Pedidos\_Vendedores\_IdVendedor FOREIGN KEY(IdVendedor) REFERENCES Vendedores(IdVendedor)

)

GO

CREATE TABLE Pedido\_Itens (

IdPedido INT NOT NULL,

IdItem INT NOT NULL,

Quantidade INT NOT NULL DEFAULT 0,

Valor\_Unitario SMALLMONEY NULL,

CONSTRAINT PK\_Pedido\_Itens PRIMARY KEY(IdPedido, IdItem),

CONSTRAINT FK\_Pedidos\_IdPedido FOREIGN KEY(IdPedido) REFERENCES Pedidos(IdPedido),

CONSTRAINT FK\_Itens\_IdItem FOREIGN KEY(IdItem) REFERENCES Itens (IdItem)

)

GO

insert into Clientes

(Nome, CGC\_CPF, ENDERECO, BAIRRO, CEP, CIDADE, UF) VALUES

('José', '35715985277', 'Avenida Paulista, 4900', ‘Jardim Paulista’, '04059-001', 'São Paulo', 'SP')

insert into Clientes

(Nome, CGC\_CPF, ENDERECO, BAIRRO, CEP, CIDADE, UF) VALUES

('Maria', '35715985211', 'Rua 1, 25', 'Vila Numero', '09640-001', 'Belo Horizonte', 'MG')

insert into Clientes

(Nome, CGC\_CPF, ENDERECO, BAIRRO, CEP, CIDADE, UF) VALUES

('Ponto Frio LTDA', '98731012000185', 'Rua Barao de Maua, 50', 'Vila Yara', '06490-001', 'Osasco', 'SP')

insert into Clientes

(Nome, CGC\_CPF, ENDERECO, BAIRRO, CEP, CIDADE, UF) VALUES

('Casas Bahia LTDA', '70574626000230', 'Alameda Guarani, 250', 'Centro', '08850-001', 'São Paulo', 'SP')

insert into Itens

(Descricao, Valor\_Unitario) values

('Geladeira', 2500)

insert into Itens

(Descricao, Valor\_Unitario) values

('Fogão', 750)

insert into Itens

(Descricao, Valor\_Unitario) values

('Lavadora', 1550)

insert into Itens

(Descricao, Valor\_Unitario) values

('Televisão', 1900)

insert into Vendedores

(Nome) Values ('Alexandre')

insert into Vendedores

(Nome) Values ('Carlos')

insert into Vendedores

(Nome) Values ('Guilherme')

insert into Vendedores

(Nome) Values ('Leonardo')

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-01-02', 1, 1)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-01-04', 2, 1)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-01-04', 3, 1)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-01-05', 4, 1)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-02-04', 1, 2)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-02-05', 3, 2)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-02-06', 4, 2)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-03-05', 1, 3)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-03-07', 1, 3)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-03-09', 1, 3)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-03-11', 1, 3)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-04-02', 2, 4)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-04-04', 3, 4)

insert into Pedidos (Data, IdCliente, IdVendedor) values ('2015-04-06', 4, 4)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (1, 1, 2, 2500)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (2, 2, 1, 750)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (3, 3, 3, 1550)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (4, 4, 1, 1900)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (5, 1, 2, 2500)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (6, 2, 4, 750)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (7, 3, 3, 1550)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (8, 4, 4, 1900)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (9, 3, 2, 1550)

insert into Pedido\_Itens (IdPedido, IdItem, Quantidade, Valor\_Unitario) Values (10, 4, 1, 1900)

DECLARE

@IdPedido INT,

@IdCliente INT,

@DataPedido DATETIME,

@NomeCliente NVARCHAR(300),

@CGC\_CPF NVARCHAR(300),

@Endereco NVARCHAR(300),

@Bairro NVARCHAR(300),

@CEP NVARCHAR(300),

@Cidade NVARCHAR(300),

@UF NVARCHAR(300),

@IdVendedor NVARCHAR(300),

@NomeVendedor NVARCHAR(300),

@IdItem INT,

@Descricao NVARCHAR(300),

@Quant INT,

@VlItem SMALLMONEY,

@VlUnit SMALLMONEY,

@VlTotal SMALLMONEY,

@VlTotalDesc SMALLMONEY

DECLARE cursor\_pedido CURSOR FOR

SELECT IdPedido, Data, IdCliente, IdVendedor FROM Pedidos

OPEN cursor\_pedido

FETCH NEXT FROM cursor\_pedido INTO @IdPedido, @DataPedido, @IdCliente, @IdVendedor

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT ' '

PRINT '--------------------------------------------------------------------------'

PRINT CONCAT('COD.PEDIDO: ' , @IdPedido, ' ', ' ', 'DATA PEDIDO: ', Convert(varchar(10),CONVERT(date,@DataPedido,106),103))

SELECT

@NomeCliente = Nome,

@CGC\_CPF = CGC\_CPF,

@Endereco = Endereco,

@Bairro = Bairro,

@CEP = CEP,

@Cidade = Cidade,

@UF = UF

FROM

Clientes

WHERE

IdCliente = @IdCliente

PRINT CONCAT('COD.CLIENTE: ' , @IdCliente, ' ', ' ', ' ')

PRINT CONCAT('NOME CLIENTE: ' , @NomeCliente, ' ', 'CGC\_CPF: ', @CGC\_CPF)

PRINT CONCAT('ENDERECO: ' , @Endereco, ' ', 'Bairro: ', @Bairro)

PRINT CONCAT('CEP: ' , @CEP, ' ', 'CIDADE: ', @Cidade, ' ', 'UF: ', @UF)

SELECT @NomeVendedor = Nome FROM Vendedores WHERE IdVendedor = @IdVendedor

PRINT CONCAT('COD.VENDEDOR: ' , @IdVendedor, ' ', 'NOME VENDEDOR: ', @NomeVendedor)

PRINT ' '

PRINT 'ITENS'

PRINT CONCAT('COD.ITEM' , ' ', 'QTD', ' ', 'DESCRIÇÃO', ' ', 'VR.UNIT', ' ', 'VR.TOTAL', ' ', '%DESCTO', ' ')

SET @VlTotal = 0

SET @VlTotalDesc = 0

DECLARE cursor\_item\_pedido CURSOR FOR

SELECT IdItem, Quantidade, Valor\_Unitario FROM Pedido\_Itens WHERE IdPedido = @IdPedido

OPEN cursor\_item\_pedido

FETCH NEXT FROM cursor\_item\_pedido INTO @IdItem, @Quant, @VlItem

WHILE @@FETCH\_STATUS = 0

BEGIN

SELECT @Descricao = Descricao, @VlUnit = Valor\_Unitario FROM Itens WHERE IdItem = @IdItem

PRINT CONCAT(@IdItem , ' ', @Quant, ' ', @Descricao, ' ', @VlUnit, ' ', @VlUnit \* @Quant, ' ', @VlUnit - @VlItem)

SET @VlTotal = @VlTotal + (@VlUnit \* @Quant)

SET @VlTotalDesc = @VlTotalDesc + @VlUnit - @VlItem

FETCH NEXT FROM cursor\_item\_pedido INTO @IdItem, @Quant, @VlItem

END

CLOSE cursor\_item\_pedido

DEALLOCATE cursor\_item\_pedido

PRINT ' '

PRINT CONCAT('VALOR TOTAL DO PEDIDO: ', @VlTotal)

PRINT CONCAT('VALOR LIQUIDO DO PEDIDO: ', @VlTotal - @VlTotalDesc)

FETCH NEXT FROM cursor\_pedido INTO @IdPedido, @DataPedido, @IdCliente, @IdVendedor

END

CLOSE cursor\_pedido

DEALLOCATE cursor\_pedido

2. escrever FUNCTION para editar CPF e CGC recebendo um numero de 15 posições, identificar quando é CPF quando[cod.de](http://cod.de/) filial igual a zeros (9999999999000099).

CREATE FUNCTION ALTERA\_CGC\_CPF(

@IdCliente INT,

@NOVO\_CGC\_CPF NVARCHAR(16))

RETURNS VARCHAR(50)

BEGIN

DECLARE

@CGC\_CPF NVARCHAR(50)

SELECT @CGC\_CPF = CGC\_CPF FROM Clientes WHERE IdCliente = @IdCliente

IF LEN(@CGC\_CPF) = 11

BEGIN

IF LEN(@NOVO\_CGC\_CPF) = 11

RETURN 'É CPF, E PODE SER ALTERADO'

ELSE

RETURN 'NÃO FOI POSSÍVEL ALTERAR O CPF. CLIENTE POSSUI CGC.'

END

ELSE

BEGIN

IF LEN(@NOVO\_CGC\_CPF) = 14

IF CHARINDEX('0000', @NOVO\_CGC\_CPF) > 0

RETURN 'CÓDIGO DE FILIAL IGUAL A 0000'

ELSE

RETURN 'É CGC, E PODE SER ALTERADO'

ELSE

RETURN 'NÃO FOI POSSÍVEL ALTERAR O CGC. NOVO É CPF'

END

RETURN ''

END