|  |  |
| --- | --- |
| NOME | TURMA |
| Izael Reis de Oliveira Junior | 1ADSB |

# PRÁTICA 07 - NN

Diagrama

Descrição gerada automaticamente

-- a) Criar um banco de dados chamado Venda.

CREATE DATABASE venda;

-- b) Selecionar esse banco de dados.

USE venda;

-- c) Criar as tabelas correspondentes à sua modelagem

CREATE TABLE cliente (

idCliente int primary key auto\_increment,

nome varchar(45),

email varchar(45),

logradouro varchar(45),

numero varchar (45),

bairro varchar(45),

cidade varchar(45),

complemento varchar(45),

fkIndicacao int

);

CREATE TABLE venda (

idVenda int primary key auto\_increment,

dtVenda date,

total double,

fkCliente int

) auto\_increment = 100;

CREATE TABLE produto (

idProduto int primary key auto\_increment,

nome varchar(45),

descricao varchar(45),

preco double

) auto\_increment = 1000;

CREATE TABLE carrinho (

idCarrinho int auto\_increment,

fkProduto int,

fkVenda int,

quantidade int,

desconto varchar(45),

constraint pkComposta primary key (idCarrinho, fkProduto, fkVenda),

constraint fkProdutoCarrinho foreign key (fkProduto) references produto(idProduto),

constraint fkVendaCarrinho foreign key (fkVenda) references venda(idVenda)

) auto\_increment = 10000;

-- d) Inserir dados nas tabelas, de forma que exista mais de uma venda para cada

-- cliente, e mais de um cliente sendo indicado por outro cliente.

INSERT INTO cliente VALUES

(default, 'Carlos Silva', 'carlos.silva@email.com', 'Rua A', '123', 'Centro', 'São Paulo', 'Ap 45', NULL),

(default, 'Maria Santos', 'maria.santos@email.com', 'Rua B', '456', 'Jardim', 'São Paulo', 'Casa 1', 1),

(default, 'João Pereira', 'joao.pereira@email.com', 'Av. C', '789', 'Vila', 'São Paulo', 'Ap 101', 1),

(default, 'Ana Costa', 'ana.costa@email.com', 'Rua D', '101', 'Centro', 'São Paulo', '', 2);

INSERT INTO produto VALUES

(default, 'Notebook', 'Notebook Dell i5', 3500.00),

(default, 'Smartphone', 'iPhone 12', 4500.00),

(default, 'Tablet', 'Samsung Galaxy Tab', 1200.00);

INSERT INTO venda VALUES

(default, '2023-10-01', 8000.00, 1),

(default, '2023-10-05', 1200.00, 1),

(default, '2023-10-07', 4500.00, 2),

(default, '2023-10-10', 3500.00, 3);

INSERT INTO carrinho VALUES

(default, 1000, 100, 1, '0%'),

(default, 1001, 101, 1, '0%'),

(default, 1002, 102, 1, '10%'),

(default, 1002, 103, 1, '5%'),

(default, 1000, 103, 1, '0%');

-- e) Exibir todos os dados de cada tabela criada, separadamente.

SELECT \* FROM cliente;

SELECT \* FROM produto;

SELECT \* FROM venda;

SELECT \* FROM carrinho;

-- f) Fazer os acertos da chave estrangeira, caso não tenha feito no momento da criação

-- das tabelas.

ALTER TABLE cliente

ADD CONSTRAINT fkIndicacaoCliente foreign key (fkIndicacao) references cliente(idCliente);

ALTER TABLE venda

ADD CONSTRAINT fkClienteVenda foreign key (fkCliente) references cliente(idCliente);

-- g) Exibir os dados dos clientes e os dados de suas respectivas vendas.

SELECT c.\*, v.\* FROM cliente as c JOIN venda as v ON v.fkCliente = c.idCliente;

-- h) Exibir os dados de um determinado cliente (informar o nome do cliente na consulta)

-- e os dados de suas respectivas vendas.

SELECT c.\*, v.\* FROM cliente as c JOIN venda as v ON v.fkCliente = c.idCliente

WHERE c.nome LIKE 'Maria%';

-- i) Exibir os dados dos clientes e de suas respectivas indicações de clientes.

SELECT c.\*, i.\* FROM cliente as c LEFT JOIN cliente as i ON c.fkIndicacao = i.idCliente;

-- j) Exibir os dados dos clientes indicados e os dados dos respectivos clientes

-- indicadores, porém somente de um determinado cliente indicador (informar o nome

-- do cliente que indicou na consulta).

SELECT c.\*, i.\* FROM cliente as c JOIN cliente as i ON c.fkIndicacao = i.idCliente

WHERE i.nome LIKE 'Maria%';

-- l) Exibir os dados dos clientes, os dados dos respectivos clientes que indicaram, os

-- dados das respectivas vendas e dos produtos.

SELECT c.\*, i.\*, v.\*, p.\*, ca.\* FROM carrinho as ca

JOIN produto as p ON ca.fkProduto = p.idProduto

JOIN venda as v ON ca.fkVenda = v.idVenda

JOIN cliente as c ON v.fkCliente = c.idCliente

left JOIN cliente as i ON c.fkIndicacao = i.idCliente;

-- m) Exibir apenas a data da venda, o nome do produto e a quantidade do produto

-- numa determinada venda.

SELECT ca.quantidade, v.dtVenda, p.nome FROM carrinho as ca

JOIN venda as v

JOIN produto as p

WHERE v.idVenda = 103;

-- n) Exibir apenas o nome do produto, o valor do produto e a soma da quantidade de

-- produtos vendidos agrupados pelo nome do produto.

SELECT p.nome, p.preco, sum(ca.quantidade) as QuantidadeVendido FROM produto as p

JOIN carrinho as ca ON ca.fkProduto = p.idProduto

GROUP BY p.nome, p.preco;

-- o) Inserir dados de um novo cliente. Exibir os dados dos clientes, das respectivas

-- vendas, e os clientes que não realizaram nenhuma venda.

INSERT INTO cliente VALUES

(default, 'Matheus Campos', 'matheus.campos@email.com', 'Rua Damasco', '1469', 'Vila Matilda', 'São Paulo', 'Ap 03', 3);

SELECT c.\*, v.\* FROM cliente as c LEFT JOIN venda as v ON v.fkCliente = c.idCliente;

-- p) Exibir o valor mínimo e o valor máximo dos preços dos produtos;

SELECT min(preco) as PrecoMinimo, max(preco) as PrecoMaximo FROM produto;

-- q) Exibir a soma e a média dos preços dos produtos;

SELECT sum(preco) as SomaPreco, avg(preco) as MediaPreco FROM produto;

-- r) Exibir a quantidade de preços acima da média entre todos os produtos;

SELECT COUNT(\*) as QuantidadeAcima FROM produto WHERE preco > (SELECT avg(preco) FROM produto);

-- s) Exibir a soma dos preços distintos dos produtos;

SELECT sum(DISTINCT preco) as SomaPrecoDistintos FROM produto;

-- t) Exibir a soma dos preços dos produtos agrupado por uma determinada venda

SELECT ca.fkVenda, sum(p.preco) SomaPreco

FROM carrinho as ca

JOIN produto as p ON ca.fkProduto = p.idProduto

JOIN venda as v ON ca.fkVenda = v.idVenda

GROUP BY ca.fkVenda;;