CREATE DATABASE miniWEG;

USE miniWEG;

CREATE TABLE produtos (

id\_produto INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL UNIQUE, -- não há 2 produtos com o mesmo nome

descricao TEXT,

preco\_unitario DECIMAL(10,2) NOT NULL

);

CREATE TABLE materia\_prima (

id\_materia\_prima INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL,

unidade\_de\_medida ENUM ('litros', 'gramas', 'metros', 'unidade') NOT NULL,

custo\_unitario DECIMAL(10,2) NOT NULL

);

-- tabela associativa:

CREATE TABLE produto\_materia (

id\_produto INT,

id\_mp INT,

quantidade DECIMAL(10,2) NOT NULL,

/\*a atividade pede para usar decimal(10,2),

mas eu usaria int, pois a quantidade sempre será um número natural (inteiro e >0)\*/

PRIMARY KEY (id\_produto, id\_mp),

FOREIGN KEY (id\_produto) REFERENCES produtos(id\_produto),

FOREIGN KEY (id\_mp) REFERENCES materia\_prima(id\_materia\_prima)

);

CREATE TABLE fornecedores (

id\_fornecedor INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL,

cnpj VARCHAR(18) UNIQUE NOT NULL,

telefone VARCHAR(20) UNIQUE NOT NULL,

email VARCHAR(100) UNIQUE

);

-- tabela associativa:

CREATE TABLE fornecimento (

id\_fornecedor INT,

id\_mp\_fornecida INT,

PRIMARY KEY (id\_fornecedor, id\_mp\_fornecida),

FOREIGN KEY (id\_fornecedor) REFERENCES fornecedores(id\_fornecedor),

FOREIGN KEY (id\_mp\_fornecida) REFERENCES materia\_prima(id\_materia\_prima)

);

CREATE TABLE departamentos (

id\_departamento INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL UNIQUE

-- não existem dois departamentos com o mesmo nome

);

CREATE TABLE funcionarios (

id\_funcionario INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL,

cargo VARCHAR(50),

dpto\_pertencente INT,

FOREIGN KEY (dpto\_pertencente) REFERENCES departamentos(id\_departamento)

);

CREATE TABLE clientes (

id\_cliente INT PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL,

cnpj VARCHAR(18) UNIQUE NOT NULL,

telefone VARCHAR(20) UNIQUE NOT NULL,

/\*não há dois clientes com o mesmo telefone, e é obrigatório,

para podermos entrar em contato com eles, já o email é opicional\*/

email VARCHAR(100) UNIQUE

);

-- tabela superclasse:

CREATE TABLE ordem\_producao (

id\_ordem INT PRIMARY KEY AUTO\_INCREMENT,

data\_op DATE NOT NULL,

func\_responsavel INT,

observacoes TEXT,

FOREIGN KEY (func\_responsavel) REFERENCES funcionarios(id\_funcionario)

);

-- tabela subclasse:

CREATE TABLE ordem\_item (

id\_ordem INT,

id\_produto INT,

quantidade INT NOT NULL,

PRIMARY KEY (id\_ordem, id\_produto),

FOREIGN KEY (id\_ordem) REFERENCES ordem\_producao(id\_ordem),

FOREIGN KEY (id\_produto) REFERENCES produtos(id\_produto)

);

-- tabela superclasse:

CREATE TABLE vendas (

id\_venda INT PRIMARY KEY AUTO\_INCREMENT,

id\_cliente INT NOT NULL,

data\_venda DATE NOT NULL,

valor\_total DECIMAL(10,2) NOT NULL,

FOREIGN KEY (id\_cliente) REFERENCES clientes(id\_cliente)

);

-- tabela subclasse:

CREATE TABLE venda\_item (

id\_venda INT,

id\_produto INT,

quantidade INT NOT NULL,

preco\_unitario DECIMAL(10,2) NOT NULL,

/\*é o preço do momento exato da venda,

por isso não devemos usar uma chave estrangeira para preencher esse campo\*/

PRIMARY KEY (id\_venda, id\_produto),

FOREIGN KEY (id\_venda) REFERENCES vendas(id\_venda),

FOREIGN KEY (id\_produto) REFERENCES produtos(id\_produto)

);

/\*troquei a ordem de algumas entidades para ficar uma legibilidade melhor,

tendo uma subclasse (classe filho) logo depois da sua superclasse (classe pai),

e mantendo também as tabelas associativas o mais próximo possível das tabelas associadas\*/