-- Active: 1709377143467@@127.0.0.1@3306

CREATE DATABASE NgolaNurtura2

DEFAULT CHARACTER SET = 'utf8mb4';

USE NgolaNurtura2;

-- Tabelas de Referência

CREATE TABLE Idiomas (

id\_idioma INT AUTO\_INCREMENT PRIMARY KEY,

nome\_idioma VARCHAR(100) NOT NULL,

codigo\_idioma VARCHAR(10) NOT NULL

);

CREATE TABLE Tipos\_Parto (

id\_tipo\_parto INT AUTO\_INCREMENT PRIMARY KEY,

nome\_tipo\_parto VARCHAR(100) NOT NULL

);

CREATE TABLE Tipos\_Leite (

id\_tipo\_leite INT AUTO\_INCREMENT PRIMARY KEY,

nome\_tipo\_leite VARCHAR(100) NOT NULL

);

CREATE TABLE Status (

id\_status INT AUTO\_INCREMENT PRIMARY KEY,

nome\_status VARCHAR(100) NOT NULL

);

CREATE TABLE Tipos\_Grupo\_Apoio (

id\_tipo\_grupo INT AUTO\_INCREMENT PRIMARY KEY,

nome\_tipo\_grupo VARCHAR(100) NOT NULL

);

CREATE TABLE Paises (

id\_pais INT AUTO\_INCREMENT PRIMARY KEY,

nome\_pais VARCHAR(100) NOT NULL,

codigo\_pais VARCHAR(10) NOT NULL

);

CREATE TABLE Estados (

id\_estado INT AUTO\_INCREMENT PRIMARY KEY,

nome\_estado VARCHAR(100) NOT NULL,

id\_pais INT NOT NULL,

FOREIGN KEY (id\_pais) REFERENCES Paises(id\_pais)

);

CREATE TABLE Municipios (

id\_municipio INT AUTO\_INCREMENT PRIMARY KEY,

nome\_municipio VARCHAR(100) NOT NULL,

id\_estado INT NOT NULL,

FOREIGN KEY (id\_estado) REFERENCES Estados(id\_estado)

);

CREATE TABLE Bairros (

id\_bairro INT AUTO\_INCREMENT PRIMARY KEY,

nome\_bairro VARCHAR(100) NOT NULL,

id\_municipio INT NOT NULL,

FOREIGN KEY (id\_municipio) REFERENCES Municipios(id\_municipio)

);

-- Tabela de Usuários (Combinação de Participantes e Usuarios)

CREATE TABLE Usuarios (

id\_usuario SERIAL PRIMARY KEY,

nome\_completo VARCHAR(255) NOT NULL,

data\_nascimento DATE NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

fuso\_horario VARCHAR(100),

id\_idioma INT NOT NULL,

data\_criacao TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

tipo\_usuario ENUM('Mae\_Receptora', 'Doadora', 'Profissional\_Saude', 'Administrador') NOT NULL,

status ENUM('Ativo', 'Inativo') NOT NULL DEFAULT 'Ativo',

data\_parto DATE,

numero\_filhos INT,

amamentando\_atualmente BOOLEAN,

motivo\_desmame TEXT,

observacoes TEXT,

data\_desmame DATE,

grupo\_sanguineo VARCHAR(10),

disponibilidade BOOLEAN,

preferencia\_local\_entrega TEXT,

metodo\_transporte TEXT,

especialidade VARCHAR(255),

crm VARCHAR(50),

historico\_atendimento TEXT,

alergias TEXT,

condicoes\_medicas TEXT,

id\_tipo\_parto INT,

nome\_usuario VARCHAR(255) UNIQUE,

contato\_preferencial VARCHAR(255) UNIQUE,

senha\_hash VARCHAR(255),

salt VARCHAR(255),

google\_id VARCHAR(255) UNIQUE,

facebook\_id VARCHAR(255) UNIQUE,

linkedin\_id VARCHAR(255) UNIQUE,

FOREIGN KEY (id\_idioma) REFERENCES Idiomas(id\_idioma),

FOREIGN KEY (id\_tipo\_parto) REFERENCES Tipos\_Parto(id\_tipo\_parto)

);

-- Demais Tabelas

CREATE TABLE Contatos (

id\_contato INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT NOT NULL, -- Referencia a tabela Usuarios

tipo\_contato ENUM('Telefone', 'Email', 'Outro') NOT NULL,

indicativo VARCHAR(10),

valor\_contato VARCHAR(255) NOT NULL,

observacao TEXT,

whatsapp BOOLEAN DEFAULT FALSE,

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

CREATE TABLE Enderecos (

id\_endereco INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT NOT NULL, -- Referencia a tabela Usuarios

tipo\_endereco ENUM('Residencial', 'Comercial', 'Outro') NOT NULL,

id\_bairro INT NOT NULL,

numero\_residencia VARCHAR(20),

complemento TEXT,

codigo\_postal VARCHAR(20),

latitude DECIMAL(9,6),

longitude DECIMAL(9,6),

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario),

FOREIGN KEY (id\_bairro) REFERENCES Bairros(id\_bairro)

);

CREATE TABLE Doacoes (

id\_doacao INT AUTO\_INCREMENT PRIMARY KEY,

id\_doadora INT NOT NULL, -- Referencia a tabela Usuarios

data\_doacao DATE NOT NULL,

quantidade\_doada DECIMAL(10,2) NOT NULL,

condicoes\_armazenamento TEXT,

id\_tipo\_leite INT NOT NULL,

validade\_leite DATE,

id\_centro\_coleta INT,

FOREIGN KEY (id\_doadora) REFERENCES Usuarios(id\_usuario), -- Referencia a tabela Usuarios

FOREIGN KEY (id\_tipo\_leite) REFERENCES Tipos\_Leite(id\_tipo\_leite),

FOREIGN KEY (id\_centro\_coleta) REFERENCES Centros\_Coleta\_Leite(id\_centro)

);

CREATE TABLE Pedidos (

id\_pedido INT AUTO\_INCREMENT PRIMARY KEY,

data\_pedido DATE NOT NULL,

data\_hora\_pedido DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

id\_mae\_receptora INT NOT NULL, -- Referencia a tabela Usuarios

quantidade\_leite\_solicitada DECIMAL(10,2) NOT NULL,

id\_tipo\_leite\_solicitado INT NOT NULL,

id\_status\_pedido INT NOT NULL,

id\_doadora\_atribuida INT, -- Referencia a tabela Usuarios

data\_hora\_doacao\_confirmada DATETIME,

motivo\_pedido TEXT,

urgencia\_pedido BOOLEAN NOT NULL DEFAULT FALSE,

feedback\_mae\_detalhado TEXT,

feedback\_doadora\_detalhado TEXT,

data\_hora\_entrega DATETIME,

local\_entrega\_desejado TEXT,

data\_hora\_atualizacao DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

tipo\_pedido ENUM('Normal', 'Urgente', 'Emergencial') NOT NULL DEFAULT 'Normal',

FOREIGN KEY (id\_mae\_receptora) REFERENCES Usuarios(id\_usuario), -- Referencia a tabela Usuarios

FOREIGN KEY (id\_tipo\_leite\_solicitado) REFERENCES Tipos\_Leite(id\_tipo\_leite),

FOREIGN KEY (id\_status\_pedido) REFERENCES Status(id\_status),

FOREIGN KEY (id\_doadora\_atribuida) REFERENCES Usuarios(id\_usuario) -- Referencia a tabela Usuarios

);

CREATE TABLE Leite\_Armazenado (

id\_estoque INT AUTO\_INCREMENT PRIMARY KEY,

data\_coleta DATE NOT NULL,

quantidade\_leite DECIMAL(10,2) NOT NULL,

id\_tipo\_leite INT NOT NULL,

local\_armazenamento TEXT,

temperatura\_armazenamento VARCHAR(20),

prazo\_validade DATE,

data\_descarte DATETIME,

id\_status\_leite INT NOT NULL,

id\_centro\_coleta INT NOT NULL,

FOREIGN KEY (id\_tipo\_leite) REFERENCES Tipos\_Leite(id\_tipo\_leite),

FOREIGN KEY (id\_status\_leite) REFERENCES Status(id\_status),

FOREIGN KEY (id\_centro\_coleta) REFERENCES Centros\_Coleta\_Leite(id\_centro)

);

CREATE TABLE Grupos\_Apoio (

id\_grupo INT AUTO\_INCREMENT PRIMARY KEY,

nome\_grupo VARCHAR(255) NOT NULL,

descricao TEXT,

id\_cidade INT NOT NULL,

data\_criacao DATE NOT NULL,

id\_tipo\_grupo INT NOT NULL,

recursos\_adicionais TEXT,

link\_grupo VARCHAR(255),

local\_encontro TEXT,

FOREIGN KEY (id\_cidade) REFERENCES Municipios(id\_municipio),

FOREIGN KEY (id\_tipo\_grupo) REFERENCES Tipos\_Grupo\_Apoio(id\_tipo\_grupo)

);

CREATE TABLE Centros\_Coleta\_Leite (

id\_centro INT AUTO\_INCREMENT PRIMARY KEY,

nome\_centro VARCHAR(255) NOT NULL,

id\_endereco INT NOT NULL,

telefone VARCHAR(20),

email VARCHAR(255),

horario\_funcionamento TEXT,

FOREIGN KEY (id\_endereco) REFERENCES Enderecos(id\_endereco)

);

CREATE TABLE Eventos\_Recursos (

id\_evento\_recurso INT AUTO\_INCREMENT PRIMARY KEY,

titulo VARCHAR(255) NOT NULL,

descricao TEXT,

tipo ENUM('Evento', 'Recurso\_Educativo') NOT NULL,

data\_hora DATETIME,

id\_endereco INT,

capacidade\_maxima INT,

status\_evento VARCHAR(50),

url VARCHAR(255),

id\_idioma INT,

FOREIGN KEY (id\_endereco) REFERENCES Enderecos(id\_endereco),

FOREIGN KEY (id\_idioma) REFERENCES Idiomas(id\_idioma)

);

CREATE TABLE Avaliacoes (

id\_avaliacao INT AUTO\_INCREMENT PRIMARY KEY,

id\_mae\_receptora INT NOT NULL, -- Referencia a tabela Usuarios

id\_doadora INT NOT NULL, -- Referencia a tabela Usuarios

nota INT NOT NULL,

comentario TEXT,

data\_avaliacao DATE NOT NULL,

FOREIGN KEY (id\_mae\_receptora) REFERENCES Usuarios(id\_usuario), -- Referencia a tabela Usuarios

FOREIGN KEY (id\_doadora) REFERENCES Usuarios(id\_usuario), -- Referencia a tabela Usuarios

CONSTRAINT chk\_nota CHECK (nota BETWEEN 1 AND 5)

);

CREATE TABLE Mensagens (

id\_mensagem INT AUTO\_INCREMENT PRIMARY KEY,

id\_remetente INT NOT NULL, -- Referencia a tabela Usuarios

id\_destinatario INT NOT NULL, -- Referencia a tabela Usuarios

data\_hora DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

assunto VARCHAR(255),

conteudo TEXT NOT NULL,

status ENUM('Enviado', 'Lido', 'Excluido') NOT NULL DEFAULT 'Enviado',

FOREIGN KEY (id\_remetente) REFERENCES Usuarios(id\_usuario), -- Referencia a tabela Usuarios

FOREIGN KEY (id\_destinatario) REFERENCES Usuarios(id\_usuario) -- Referencia a tabela Usuarios

);

CREATE TABLE Apoiadores (

id\_apoiador INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(255) NOT NULL,

tipo\_apoiador ENUM('Particular', 'Estatal', 'Privada') NOT NULL,

tipo\_apoio TEXT NOT NULL

);

CREATE TABLE Pedidos\_Doacoes (

id\_pedido\_doacao INT AUTO\_INCREMENT PRIMARY KEY,

id\_pedido INT NOT NULL,

id\_doacao INT NOT NULL,

quantidade\_utilizada DECIMAL(10,2) NOT NULL,

data\_hora\_vinculacao DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_pedido) REFERENCES Pedidos(id\_pedido),

FOREIGN KEY (id\_doacao) REFERENCES Doacoes(id\_doacao),

CONSTRAINT chk\_quantidade\_utilizada CHECK (quantidade\_utilizada > 0)

);

CREATE TABLE Relatorios\_Metricas (

id\_metrica INT AUTO\_INCREMENT PRIMARY KEY,

tipo\_relatorio ENUM(

'Taxa\_Sucesso', 'Solicitacoes', 'Avaliacoes', 'Doacoes', 'Participantes',

'Leite\_Armazenado', 'Grupos\_Apoio', 'Centros\_Coleta'

) NOT NULL,

data\_referencia DATE NOT NULL,

id\_pais INT,

id\_estado INT,

id\_municipio INT,

valor\_metrica DECIMAL(10,2),

descricao\_metrica TEXT,

unidade\_metrica VARCHAR(50),

FOREIGN KEY (id\_pais) REFERENCES Paises(id\_pais),

FOREIGN KEY (id\_estado) REFERENCES Estados(id\_estado),

FOREIGN KEY (id\_municipio) REFERENCES Municipios(id\_municipio)

);

-- Índices

CREATE INDEX idx\_participante ON Usuarios(id\_usuario); -- Índice na tabela Usuarios

CREATE INDEX idx\_tipo\_participante ON Usuarios(tipo\_usuario); -- Índice na tabela Usuarios

CREATE INDEX idx\_mae\_receptora ON Pedidos(id\_mae\_receptora);

CREATE INDEX idx\_doadora ON Doacoes(id\_doadora);

CREATE INDEX idx\_tipo\_pedido ON Pedidos(tipo\_pedido);

DELIMITER //

CREATE PROCEDURE CalcularMetricas(data\_referencia DATE)

BEGIN

-- Declarar variáveis

DECLARE total\_pedidos INT;

DECLARE pedidos\_atendidos INT;

DECLARE taxa\_sucesso DECIMAL(5,2);

DECLARE total\_doacoes INT;

DECLARE quantidade\_leite\_doada DECIMAL(10,2);

DECLARE media\_avaliacoes DECIMAL(3,2);

-- Métricas de Pedidos

SELECT COUNT(\*) INTO total\_pedidos FROM Pedidos WHERE DATE(data\_pedido) = data\_referencia;

SELECT COUNT(\*) INTO pedidos\_atendidos

FROM Pedidos

WHERE DATE(data\_pedido) = data\_referencia

AND id\_status\_pedido IN (SELECT id\_status FROM Status WHERE nome\_status = 'Atendido');

SET taxa\_sucesso = IFNULL(pedidos\_atendidos / total\_pedidos, 0);

INSERT INTO Relatorios\_Metricas (tipo\_relatorio, data\_referencia, valor\_metrica, descricao\_metrica, unidade\_metrica)

VALUES

('Solicitacoes', data\_referencia, total\_pedidos, 'Total de pedidos', 'pedidos'),

('Taxa\_Sucesso', data\_referencia, taxa\_sucesso, 'Taxa de sucesso de atendimento de pedidos', '%');

-- Métricas de Doações

SELECT COUNT(\*) INTO total\_doacoes FROM Doacoes WHERE DATE(data\_doacao) = data\_referencia;

SELECT SUM(quantidade\_doada) INTO quantidade\_leite\_doada FROM Doacoes WHERE DATE(data\_doacao) = data\_referencia;

INSERT INTO Relatorios\_Metricas (tipo\_relatorio, data\_referencia, valor\_metrica, descricao\_metrica, unidade\_metrica)

VALUES

('Doacoes', data\_referencia, total\_doacoes, 'Total de doações', 'doacoes'),

('Doacoes', data\_referencia, quantidade\_leite\_doada, 'Quantidade total de leite doado', 'litros');

-- Métricas de Avaliações

SELECT AVG(nota) INTO media\_avaliacoes FROM Avaliacoes WHERE DATE(data\_avaliacao) = data\_referencia;

IF media\_avaliacoes IS NOT NULL THEN

INSERT INTO Relatorios\_Metricas (tipo\_relatorio, data\_referencia, valor\_metrica, descricao\_metrica, unidade\_metrica)

VALUES ('Avaliacoes', data\_referencia, media\_avaliacoes, 'Média de avaliações das doadoras', '');

END IF;

-- Métricas de Participantes

INSERT INTO Relatorios\_Metricas (tipo\_relatorio, data\_referencia, id\_pais, id\_estado, id\_municipio, valor\_metrica, descricao\_metrica, unidade\_metrica)

SELECT

'Participantes', data\_referencia, p.id\_pais, e.id\_estado, m.id\_municipio, COUNT(\*),

CONCAT('Número de participantes - ', tipo\_usuario), 'participantes' -- Usa tipo\_usuario da tabela Usuarios

FROM Usuarios u

JOIN Enderecos ed ON u.id\_usuario = ed.id\_usuario

JOIN Bairros b ON ed.id\_bairro = b.id\_bairro

JOIN Municipios m ON b.id\_municipio = m.id\_municipio

JOIN Estados e ON m.id\_estado = e.id\_estado

JOIN Paises p ON e.id\_pais = p.id\_pais

WHERE u.status = 'Ativo'

GROUP BY p.id\_pais, e.id\_estado, m.id\_municipio, tipo\_usuario;

-- Métricas de Leite Armazenado (com melhorias)

INSERT INTO Relatorios\_Metricas (tipo\_relatorio, data\_referencia, id\_pais, id\_estado, id\_municipio, valor\_metrica, descricao\_metrica, unidade\_metrica)

SELECT

'Leite\_Armazenado', data\_referencia, p.id\_pais, e.id\_estado, m.id\_municipio, SUM(la.quantidade\_leite),

CONCAT('Quantidade de leite armazenado - ', tl.nome\_tipo\_leite), 'litros'

FROM Leite\_Armazenado la

JOIN Tipos\_Leite tl ON la.id\_tipo\_leite = tl.id\_tipo\_leite

JOIN Centros\_Coleta cc ON la.id\_centro\_coleta = cc.id\_centro

JOIN Enderecos ed ON cc.id\_endereco = ed.id\_endereco

JOIN Bairros b ON ed.id\_bairro = b.id\_bairro

JOIN Municipios m ON b.id\_municipio = m.id\_municipio

JOIN Estados e ON m.id\_estado = e.id\_estado

JOIN Paises p ON e.id\_pais = p.id\_pais

WHERE DATE(la.data\_coleta) <= data\_referencia AND (la.data\_descarte IS NULL OR DATE(la.data\_descarte) > data\_referencia)

GROUP BY p.id\_pais, e.id\_estado, m.id\_municipio, tl.nome\_tipo\_leite;

-- Métricas de Grupos de Apoio

INSERT INTO Relatorios\_Metricas (tipo\_relatorio, data\_referencia, id\_pais, id\_estado, id\_municipio, valor\_metrica, descricao\_metrica, unidade\_metrica)

SELECT

'Grupos\_Apoio', data\_referencia, p.id\_pais, e.id\_estado, m.id\_municipio, COUNT(\*),

'Número de grupos de apoio', 'grupos'

FROM Grupos\_Apoio ga

JOIN Municipios m ON ga.id\_cidade = m.id\_municipio

JOIN Estados e ON m.id\_estado = e.id\_estado

JOIN Paises p ON e.id\_pais = p.id\_pais

GROUP BY p.id\_pais, e.id\_estado, m.id\_municipio;

-- Métricas de Centros de Coleta

INSERT INTO Relatorios\_Metricas (tipo\_relatorio, data\_referencia, id\_pais, id\_estado, id\_municipio, valor\_metrica, descricao\_metrica, unidade\_metrica)

SELECT

'Centros\_Coleta', data\_referencia, p.id\_pais, e.id\_estado, m.id\_municipio, COUNT(\*),

'Número de centros de coleta', 'centros'

FROM Centros\_Coleta cc

JOIN Enderecos ed ON cc.id\_endereco = ed.id\_endereco

JOIN Bairros b ON ed.id\_bairro = b.id\_bairro

JOIN Municipios m ON b.id\_municipio = m.id\_municipio

JOIN Estados e ON m.id\_estado = e.id\_estado

JOIN Paises p ON e.id\_pais = p.id\_pais

GROUP BY p.id\_pais, e.id\_estado, m.id\_municipio;

END //

DELIMITER ;