

```

-- Ativar foreign keys
PRAGMA foreign_keys = ON;

-- Limpar tabelas existentes (se houver)
DROP TABLE IF EXISTS Noticias;
DROP TABLE IF EXISTS Relatorio;
DROP TABLE IF EXISTS Doacao;
DROP TABLE IF EXISTS Campanha;
DROP TABLE IF EXISTS Usuario;

-- =====
-- CRIANDO TABELAS
-- =====

-- Tabela Usuario (base - sem dependências)
CREATE TABLE Usuario (
    cd_cliente INTEGER PRIMARY KEY AUTOINCREMENT,
    nome_completo TEXT NOT NULL,
    telefone TEXT,
    cpf TEXT UNIQUE,
    cep TEXT,
    nome_usuario TEXT UNIQUE NOT NULL,
    senha TEXT NOT NULL,
    email TEXT UNIQUE NOT NULL
);

-- Tabela Campanha (base - sem dependências)
CREATE TABLE Campanha (
    cd_campanha INTEGER PRIMARY KEY AUTOINCREMENT,
    nome_campanha TEXT NOT NULL,
    meta_arrecadacao REAL,
    inicio DATE,
    fim DATE
);

-- Tabela Doacao (depende de Usuario e Campanha)
CREATE TABLE Doacao (
    cd_doacao INTEGER PRIMARY KEY AUTOINCREMENT,
    cd_cliente INTEGER NOT NULL,
    cd_campanha INTEGER NOT NULL,
    nome_doacao TEXT NOT NULL,
    tipo_doacao TEXT NOT NULL,
    forma_arrecadacao TEXT,
    status_arrecadacao TEXT DEFAULT 'Pendente',
    CONSTRAINT fk_doacao_usuario FOREIGN KEY (cd_cliente)
        REFERENCES Usuario(cd_cliente),
    CONSTRAINT fk_doacao_campanha FOREIGN KEY (cd_campanha)
        REFERENCES Campanha(cd_campanha)
);

```

```
-- Tabela Relatorio (depende de Campanha)
CREATE TABLE Relatorio (
    cd_relatorio INTEGER PRIMARY KEY AUTOINCREMENT,
    cd_campanha INTEGER NOT NULL,
    tipo_relatorio TEXT NOT NULL,
    valor_gasto REAL,
    data_relatorio DATE NOT NULL,
    CONSTRAINT fk_relatorio_campanha FOREIGN KEY (cd_campanha)
        REFERENCES Campanha(cd_campanha)
);
```

```
-- Tabela Noticias (depende de Campanha)
CREATE TABLE Noticias (
    cd_noticias INTEGER PRIMARY KEY AUTOINCREMENT,
    cd_campanha INTEGER NOT NULL,
    titulo_noticia TEXT NOT NULL,
    data_noticia DATE NOT NULL,
    autor TEXT,
    conteudo TEXT,
    CONSTRAINT fk_noticias_campanha FOREIGN KEY (cd_campanha)
        REFERENCES Campanha(cd_campanha)
);
```

```
-- =====
-- CRIANDO ÍNDICES PARA PERFORMANCE
-- =====
```

```
-- Índices para Usuario
CREATE INDEX idx_usuario_email ON Usuario(email);
CREATE INDEX idx_usuario_telefone ON Usuario(telefone);
CREATE INDEX idx_usuario_cpf ON Usuario(cpf);
```

```
-- Índices para Doacao
CREATE INDEX idx_doacao_cliente ON Doacao(cd_cliente);
CREATE INDEX idx_doacao_campanha ON Doacao(cd_campanha);
CREATE INDEX idx_doacao_status ON Doacao(status_arrecadacao);
```

```
-- Índices para Relatorio
CREATE INDEX idx_relatorio_campanha ON Relatorio(cd_campanha);
```

```
-- Índices para Noticias
CREATE INDEX idx_noticias_campanha ON Noticias(cd_campanha);
CREATE INDEX idx_noticias_data ON Noticias(data_noticia);
```

```
-- Índices para Campanha
CREATE INDEX idx_campanha_nome ON Campanha(nome_campanha);
```

```

-- =====
-- CRIANDO INSERT PARA TESTES
-- =====

-- Inserir dados de teste
INSERT INTO Usuario (nome_completo, telefone, cpf, nome_usuario, senha, email)
VALUES
('João Silva Santos', '11999999999', '12345678901', 'joao123', 'senha123',
'joao@email.com'),
('Maria Oliveira', '11888888888', '98765432100', 'maria456', 'senha456',
'maria@email.com');

INSERT INTO Campanha (nome_campanha, meta_arrecadacao, inicio, fim) VALUES
('Campanha do Agasalho 2024', 50000.00, '2024-06-01', '2024-08-31'),
('Natal Solidário', 30000.00, '2024-11-01', '2024-12-25');

INSERT INTO Doacao (cd_cliente, cd_campanha, nome_doacao, tipo_doacao,
forma_arrecadacao) VALUES
(1, 1, 'Doação de Roupas', 'Roupas', 'Entrega'),
(2, 1, 'Doação em Dinheiro', 'Dinheiro', 'PIX');

-- =====
-- VERIFICAÇÃO DAS TABELAS
-- =====

-- Verificar criação
SELECT 'Tabelas e índices criados com sucesso!' as resultado;
SELECT name FROM sqlite_master WHERE type='table' ORDER BY name;
SELECT name, tbl_name FROM sqlite_master WHERE type='index' AND name LIKE
'idx_%' ORDER BY tbl_name;

```