

Nutria documentação SQL

Documentação de Banco de Dados: Normalização e Otimização

Este documento descreve a estrutura normalizada, os relacionamentos e os métodos de otimização implementados no banco de dados relacional.

1. Tabelas Originais

Descrição: A versão do 1º ano, é uma versão em que todo o banco foi modelado em Postgres, diferentemente da versão nossa (2º ano) em que apenas usuários e admins são tratados no SQL.

1º Ano

```
----- Criação da tabela usuarios -----
CREATE TABLE usuario (
  id SERIAL PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  email VARCHAR(320) NOT NULL UNIQUE,
  senha VARCHAR(64) NOT NULL,
  telefone VARCHAR(15) NOT NULL UNIQUE CHECK (telefone ~'^([0-9]{2})? ?([0-9]{5})-?([0-9]{4})$'),
  empresa VARCHAR(50) DEFAULT 'Empresa Não Informada',
  foto VARCHAR(255) DEFAULT 'Sem foto'
);

----- Criação da tabela admins -----
CREATE TABLE admin (
  id SERIAL PRIMARY KEY,
  nome VARCHAR(100) NOT NULL,
  email VARCHAR(320) NOT NULL UNIQUE,
  senha VARCHAR(64) NOT NULL,
  telefone VARCHAR(15) NOT NULL UNIQUE CHECK (telefone ~'^([0-9]{2})? ?([0-9]{5})-?([0-9]{4})$'),
  nascimento DATE NOT NULL,
  cargo VARCHAR(64) NOT NULL DEFAULT 'Admin',
  foto VARCHAR(255) DEFAULT 'Sem foto'
);
```

2º Ano

```
----- Criação da tabela usuario -----
CREATE TABLE usuario (
  nCdUsuario SERIAL PRIMARY KEY,
  cNmUsuario VARCHAR(100) NOT NULL,
  cEmail VARCHAR(320) NOT NULL UNIQUE,
  cSenha VARCHAR(255) NOT NULL CHECK (LENGTH(cSenha) >= 8),
  cTelefone VARCHAR(11) NOT NULL UNIQUE CHECK (LENGTH(cTelefone) = 11),
  cEmpresa VARCHAR(50) DEFAULT 'Empresa não informada',
  cFoto VARCHAR(255) DEFAULT 'Sem foto'
);

----- Criação da tabela admin -----
CREATE TABLE admin (
  nCdAdmin SERIAL PRIMARY KEY,
  cNmAdmin VARCHAR(100) NOT NULL,
  cEmail VARCHAR(320) NOT NULL UNIQUE,
  cSenha VARCHAR(255) NOT NULL CHECK (LENGTH(cSenha) >= 8),
  cTelefone VARCHAR(11) NOT NULL UNIQUE CHECK (LENGTH(cTelefone) = 11),
  dNascimento DATE NOT NULL,
  cCargo VARCHAR(64) NOT NULL DEFAULT 'Admin',
  cFoto VARCHAR(255) DEFAULT 'Sem foto'
);
```

Para mais detalhes visite o repositório: <https://github.com/Nutriabr/nutria-bd/tree/main>

2. Tabelas Normalizadas

Descrição: O banco de dados foi estruturado para eliminar redundâncias. Os dados foram divididos nas seguintes tabelas:

- **usuario:** Armazena os dados cadastrais únicos dos usuários da plataforma.
- **admin:** Armazena os dados cadastrais únicos dos administradores.
- **log_admin:** Tabela de log que registra todas as alterações (INSERT, UPDATE, DELETE) feitas na tabela admin.
- **log_usuario:** Tabela de log que registra todas as alterações feitas na tabela usuario.
- **acesso_diario:** Tabela de log para registrar o primeiro acesso de cada usuário por dia (DAU - Daily Active User).
- **log_rpa:** Tabela dedicada a receber logs de status (INFO, ERROR etc.) de processos de Automação (RPA).

3. Relacionamentos

Descrição: Os relacionamentos foram definidos para garantir a integridade dos dados e preservar o histórico de auditoria.

- **Relacionamento 1 (Log Admin):**
 - **Tabelas:** log_admin (filha) → admin (pai).
 - **Regra:** ON DELETE SET NULL. Se um admin for deletado, seu histórico de log é **preservado**, e a referência ao seu ID se torna NULL.
- **Relacionamento 2 (Log Usuário):**
 - **Tabelas:** log_usuario (filha) → usuario (pai).
 - **Regra:** ON DELETE SET NULL. Se um usuario for deletado, seu histórico de log é **preservado**, e a referência ao seu ID se torna NULL.
- **Relacionamento 3 (Acesso Diário):**
 - **Tabelas:** acesso_diario (filha) → usuario (pai).
 - **Regra:** ON DELETE SET NULL. Se um usuario for deletado, seu histórico de acessos (DAU) é **preservado**, e a referência ao seu ID se torna NULL.

4. Tabela de Dados

Descrição: Detalhamento das colunas de cada tabela.

Tabela: usuario

Coluna	Tipo	Restrições	Descrição
nCdUsuario	SERIAL	PRIMARY KEY	Identificador único (auto-incremento).
cNmUsuario	VARCHAR(100)	NOT NULL	Nome do usuário.
cEmail	VARCHAR(100)	NOT NULL, UNIQUE	E-mail de login (único).
cSenha	VARCHAR(255)	NOT NULL, CHECK (>= 8)	Senha (hash armazenado pelo back-end).
cTelefone	VARCHAR(11)	NOT NULL, UNIQUE	Telefone (único).
cEmpresa	VARCHAR(50)	DEFAULT 'Empresa não informada'	Empresa do usuário (opcional).

Coluna	Tipo	Restrições	Descrição
cFoto	VARCHAR(255)	DEFAULT 'Sem foto'	URL ou caminho da foto (opcional).

Tabela: admin

Coluna	Tipo	Restrições	Descrição
nCdAdmin	SERIAL	PRIMARY KEY	Identificador único (auto-incremento).
cNmAdmin	VARCHAR(100)	NOT NULL	Nome do administrador.
cEmail	VARCHAR(320)	NOT NULL, UNIQUE	E-mail de login (único).
cSenha	VARCHAR(255)	NOT NULL, CHECK (>= 8)	Senha (hash armazenado pelo back-end).
cTelefone	VARCHAR(11)	NOT NULL, UNIQUE	Telefone (único).
dNascimento	DATE	NOT NULL	Data de nascimento.
cCargo	VARCHAR(64)	NOT NULL, DEFAULT 'Admin'	Cargo do administrador.
cFoto	VARCHAR(255)	DEFAULT 'Sem foto'	URL ou caminho da foto (opcional).

Tabela: log_admin

Coluna	Tipo	Restrições	Descrição
nCdLog	SERIAL	PRIMARY KEY	ID único do registro de log.
cTabelaAfetada	VARCHAR(100)	NOT NULL	Nome da tabela que sofreu a ação.
cOperacao	VARCHAR(10)	NOT NULL	Tipo de operação (INSERT, UPDATE, DELETE).
nCdAdminAfetado	INT	FK (admin)	ID do admin que sofreu a ação.
cDadosAntigos	JSONB	DEFAULT NULL	Estado completo da linha (antes da ação).

Coluna	Tipo	Restrições	Descrição
cDadosNovos	JSONB	DEFAULT NULL	Estado completo da linha (depois da ação).
dHorario	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Data e hora exata da operação.

Tabela: log_usuario

Coluna	Tipo	Restrições	Descrição
nCdLog	SERIAL	PRIMARY KEY	ID único do registro de log.
cTabelaAfetada	VARCHAR(100)	NOT NULL	Nome da tabela que sofreu a ação.
cOperacao	VARCHAR(10)	NOT NULL	Tipo de operação (INSERT, UPDATE, DELETE).
nCdUsuarioAfetado	INT	FK (usuario)	ID do usuário que sofreu a ação.
cDadosAntigos	JSONB	DEFAULT NULL	Estado completo da linha (antes da ação).
cDadosNovos	JSONB	DEFAULT NULL	Estado completo da linha (depois da ação).
tHorario	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Data e hora exata da operação.

Tabela: acesso_diario

Coluna	Tipo	Restrições	Descrição
nCdAcesso	SERIAL	PRIMARY KEY	ID único do registro de acesso.
nCdUsuario	INT	FK (usuario), UNIQUE (com data)	ID do usuário que acessou.
dDataAcesso	DATE	DEFAULT CURRENT_DATE	Data do acesso (sem hora).

Tabela: log_rpa

Coluna	Tipo	Restrições	Descrição
nCdLog	SERIAL	PRIMARY KEY	ID único do registro de log.
cStatus	VARCHAR(7)	NOT NULL	Status do log (INFO, ERROR, SUCCESS).
dDate	DATE	NOT NULL	Data da ocorrência.
tTime	TIME	NOT NULL	Hora da ocorrência.
cText	VARCHAR(255)	NOT NULL	Mensagem de log.

5. Métodos de Otimização (Índices)

Para garantir a performance das consultas, os seguintes índices foram implementados:

1. Índices Únicos (Automáticos):

- O Postgres cria índices automaticamente para todas as colunas definidas como PRIMARY KEY (nCdUsuario, nCdAdmin, etc.) e UNIQUE.
- Portanto, as colunas cEmail e cTelefone (nas tabelas usuario e admin) já estão otimizadas para buscas rápidas.

2. Índices Manuais (Otimização de Chaves Estrangeiras): Para otimizar consultas que usam cláusulas WHERE ou JOINS nas colunas de chave estrangeira (FK), índices manuais foram criados.

- **Índice em log_usuario(nCdUsuarioAfetado):**
 - Acelera a busca por todo o histórico de um usuário específico.
 - *(CREATE INDEX idx_log_usuario_fk ON log_usuario(nCdUsuarioAfetado);)*
 - **Índice em log_admin(nCdAdminAfetado):**
 - Acelera a busca por todo o histórico de auditoria de um admin específico.
 - *(CREATE INDEX idx_log_admin_fk ON log_admin(nCdAdminAfetado);)*
 - **Índice em acesso_diario(nCdUsuario):**
 - Acelera as consultas de DAU (Daily Active User) para um usuário específico.
 - *(CREATE INDEX idx_acesso_diario_fk ON acesso_diario(nCdUsuario);)*
-

Outros meios de otimização podem ser encontrados na documentação do banco não relacional Mongo: <https://github.com/Nutria-Oficial/Fast-API-Features/tree/main/docs%2FModels%2FMongo>