

BANCO DE DADOS

ENTIDADES

Fornecedores:

- ID (Chave Primária) - Inteiro
- Nome da Empresa - Texto
- CNPJ - Texto
- Endereço Completo - Texto
- Contato (Telefone, Email) - Texto
- Informações de Pagamento (Conta Bancária, Forma de Pagamento) - Texto

Produtos:

- ID (Chave Primária) - Inteiro
- Nome - Texto
- Descrição - Texto
- Categoria (ex: Hortaliças, Frutas, Grãos) - Texto
- Preço Unitário - Decimal ou Float
- Quantidade em Estoque - Inteiro
- Data de Validade - Data
- Fornecedor Associado (Chave Estrangeira referenciando a tabela de Fornecedores) - Inteiro

Clientes:

- ID (Chave Primária) - Inteiro
- Nome Completo - Texto
- CPF - Texto
- Endereço Completo - Texto
- Contato (Telefone, Email) - Texto

Pedidos:

- ID (Chave Primária) - Inteiro
- Data do Pedido - Data e Hora
- Cliente (Chave Estrangeira referenciando a tabela de Clientes) - Inteiro
- Status do Pedido (ex: Pendente, Em Processamento, Concluído) - Texto

Itens do Pedido:

- ID do Pedido (Chave Estrangeira referenciando a tabela de Pedidos) - Inteiro
- ID do Produto (Chave Estrangeira referenciando a tabela de Produtos) - Inteiro
- Quantidade - Inteiro

- Preço Unitário - Decimal ou Float

Relatórios:

- ID (Chave Primária) - Inteiro
- Tipo (ex: Produção, Vendas, Estoque) - Texto
- Data de Geração - Data e Hora
- Descrição - Texto
- Arquivo (Armazenamento de Relatório em Formato Digital) - Blob (ou outro tipo de dado binário)

Usuários:

- ID (Chave Primária) - Inteiro
- Nome de Usuário - Texto
- Senha (Criptografada) - Texto
- Nível de Acesso (ex: Administrador, Funcionário) - Texto
- Último Acesso - Data e Hora
- Status de Ativação (Ativo, Inativo) - Texto

Log de Acesso:

- ID (Chave Primária) - Inteiro
- ID do Usuário (Chave Estrangeira referenciando a tabela de Usuários) - Inteiro
- Data e Hora do Acesso - Data e Hora
- Endereço IP - Texto
- Tipo de Acesso (ex: Login bem-sucedido, Tentativa de Login Mal-sucedida) - Texto

Configurações do Sistema:

- ID (Chave Primária) - Inteiro
- Parâmetro (ex: Taxa de Impostos, Limite de Estoque Mínimo) - Texto
- Valor - Dependente do tipo de parâmetro (Decimal, Inteiro, Texto)

Funcionários:

- ID (Chave Primária) - Inteiro
- Nome Completo - Texto
- CPF - Texto
- Cargo - Texto
- Data de Contratação - Data
- Salário - Decimal ou Float
- Departamento - Texto
- Supervisor (Chave Estrangeira referenciando a tabela de Funcionários para gerenciar a hierarquia) - Inteiro

RELACIONAMENTOS

1. Fornecedores e Produtos
 - Relacionamento: Um fornecedor pode fornecer muitos produtos, mas um produto é fornecido por apenas um fornecedor.
 - Tipo de Relacionamento: Um-para-Muitos (1:N)
 2. Clientes e Pedidos
 - Relacionamento: Um cliente pode fazer muitos pedidos, mas um pedido é feito por apenas um cliente.
 - Tipo de Relacionamento: Um-para-Muitos (1:N)
 3. Pedidos e Itens do Pedido
 - Relacionamento: Um pedido pode conter muitos itens, e um item pode estar em apenas um pedido.
 - Tipo de Relacionamento: Um-para-Muitos (1:N)
 4. Produtos e Itens do Pedido
 - Relacionamento: Um produto pode estar em muitos itens de pedidos, e um item de pedido é associado a apenas um produto.
 - Tipo de Relacionamento: Um-para-Muitos (1:N)
 5. Clientes e Usuários
 - Relacionamento: Um cliente pode ser associado a muitos usuários, e um usuário está associado a apenas um cliente.
 - Tipo de Relacionamento: Um-para-Muitos (1:N)
 6. Usuários e Log de Acesso
 - Relacionamento: Um usuário pode ter muitos registros de log de acesso, mas um registro de log de acesso está associado a apenas um usuário.
 - Tipo de Relacionamento: Um-para-Muitos (1:N)
 7. Funcionários e Funcionários (Supervisor)
 - Relacionamento: Um funcionário pode ser supervisor de muitos outros funcionários, e um funcionário tem apenas um supervisor (um supervisor é também um funcionário).
 - Tipo de Relacionamento: Um-para-Muitos (1:N)
 8. Configurações do Sistema e Configurações do Sistema
 - Relacionamento: Cada configuração do sistema pode depender de outras configurações do sistema (por exemplo, um imposto pode depender de uma taxa de câmbio).
 - Tipo de Relacionamento: Auto-relacionamento (Um-para-Muitos, no caso de uma configuração depender de outra)
-

MODELO INICIAL DO TEXTO:

Entidades e Atributos:

1. Entidade Cliente:
 - Atributos:
 - ID (int)
 - Nome (string)
 - Email (string)

- Telefone (string)
- Endereço (string)
- 2. Entidade Fornecedor:
 - Atributos:
 - ID (int)
 - Nome (string)
 - CNPJ (string)
 - Email (string)
 - Telefone (string)
 - Endereço (string)
- 3. Entidade Produto:
 - Atributos:
 - ID (int)
 - Nome (string)
 - Descrição (string)
 - Preço (float)
 - Estoque (int)
- 4. Entidade Pedido:
 - Atributos:
 - ID (int)
 - ClienteID (int - Chave estrangeira referenciando a entidade Cliente)
 - DataPedido (date)
 - Status (string)
 - Total (float)
- 5. Entidade DetalhePedido:
 - Atributos:
 - ID (int)
 - PedidoID (int - Chave estrangeira referenciando a entidade Pedido)
 - ProdutoID (int - Chave estrangeira referenciando a entidade Produto)
 - Quantidade (int)
 - PreçoUnitário (float)

Relacionamentos:

1. Cliente - Pedido:
 - Um cliente pode fazer vários pedidos, mas um pedido pertence a apenas um cliente. (Relacionamento um para muitos)
2. Pedido - DetalhePedido:
 - Um pedido pode ter vários detalhes de pedido, enquanto um detalhe de pedido pertence a apenas um pedido. (Relacionamento um para muitos)
3. Produto - DetalhePedido:
 - Um produto pode estar presente em vários detalhes de pedidos, mas um detalhe de pedido pertence a apenas um produto. (Relacionamento um para muitos)

Integridade Referencial:

- Garantir que cada Pedido e DetalhePedido tenham referências válidas para seus respectivos Clientes e Produtos, garantindo assim a consistência dos dados.

Normalização:

- Organizar os dados de acordo com as formas normais para evitar redundância e garantir a integridade dos dados.

Índices:

- Criar índices em colunas frequentemente consultadas, como ID do Cliente, ID do Pedido e ID do Produto, para otimizar a velocidade de recuperação de dados.

Transações:

- Implementar transações para garantir a atomicidade, consistência, isolamento e durabilidade das operações no banco de dados.

Segurança de Dados:

- Aplicar medidas de segurança, como criptografia, controle de acesso baseado em funções e auditoria de dados, para proteger os dados contra acesso não autorizado.

Backup e Recuperação:

- Estabelecer políticas de backup regulares e procedimentos de recuperação para proteger os dados contra perda e corrupção.

Desempenho:

- Otimizar consultas e operações de banco de dados por meio de técnicas como indexação adequada, análise de plano de execução e ajuste de parâmetros de configuração.

Escalabilidade:

- Projetar o banco de dados com escalabilidade em mente, permitindo que ele cresça conforme necessário e mantenha o desempenho mesmo com um aumento no volume de dados e carga de trabalho.

MODELO MELHOR

1. Introdução

Este relatório técnico descreve o projeto e a implementação do banco de dados para a startup de uma fazenda urbana com foco em segurança alimentar. O banco de dados foi desenvolvido para atender aos requisitos de controle de produção, fornecedores, vendas, relatórios e conformidade com a LGPD (Lei Geral de Proteção de Dados).

2. Objetivo

O objetivo principal do banco de dados é fornecer uma estrutura robusta e eficiente para armazenar e gerenciar dados relacionados às operações da fazenda urbana. Isso inclui

informações sobre clientes, fornecedores, produtos, pedidos e detalhes de pedidos.

3. Entidades e Atributos

O banco de dados é composto pelas seguintes entidades principais:

- Cliente
- Fornecedor
- Produto
- Pedido
- DetalhePedido

Cada entidade possui atributos específicos para armazenar informações relevantes. Os tipos de dados dos atributos foram cuidadosamente selecionados para garantir a precisão e a integridade dos dados.

4. Relacionamentos

Foram definidos os seguintes relacionamentos entre as entidades:

- Cliente - Pedido: Relacionamento um para muitos, onde um cliente pode ter vários pedidos.
- Pedido - DetalhePedido: Relacionamento um para muitos, onde um pedido pode ter vários detalhes de pedido.
- Produto - DetalhePedido: Relacionamento um para muitos, onde um produto pode estar presente em vários detalhes de pedidos.

Esses relacionamentos garantem a consistência dos dados e facilitam a recuperação de informações relacionadas.

5. Integridade Referencial

A integridade referencial foi implementada para garantir que cada Pedido e DetalhePedido tenham referências válidas para seus respectivos Clientes e Produtos. Isso é fundamental para manter a consistência dos dados e evitar inconsistências.

6. Normalização

O banco de dados foi projetado de acordo com as formas normais para evitar redundância e garantir a integridade dos dados. Isso inclui a eliminação de dependências transitivas e a minimização de anomalias de atualização, inserção e exclusão.

7. Segurança de Dados

Foram implementadas medidas de segurança para proteger os dados contra acesso não autorizado, incluindo criptografia de dados sensíveis, controle de acesso baseado em funções e auditoria de acesso.

8. Desempenho

Foram adotadas várias estratégias para otimizar o desempenho do banco de dados, como indexação adequada, análise de plano de execução de consultas e ajuste de parâmetros de configuração.

9. Escalabilidade

O banco de dados foi projetado para ser escalável, permitindo que ele cresça conforme necessário e mantenha o desempenho mesmo com um aumento no volume de dados e carga de trabalho.

10. Conclusão

O banco de dados foi desenvolvido com base em práticas recomendadas de modelagem de dados e gerenciamento de banco de dados. Ele fornece uma base sólida e confiável para as operações da fazenda urbana, garantindo a integridade, segurança e desempenho dos dados.