

Instruções para Configurar o Banco de Dados no Supabase

Problema Identificado

O Manus criou tabelas que **não correspondem** ao schema da aplicação:

Tabelas Criadas pelo Manus (Incompatíveis):

- `suppliers`
- `products`
- `product_aliases`
- `purchases`
- `purchase_items`
- `bills`

Tabelas Necessárias para a Aplicação:

- `categories`
- `products`
- `purchases`
- `purchase_items`
- `receipts`

Solução: Executar Script SQL no Supabase

Passo 1: Acessar o SQL Editor do Supabase

1. Acesse: <https://supabase.com/dashboard/project/ixyxegpijupehxykntck>
2. No menu lateral, clique em **“SQL Editor”**
3. Clique em **“New query”** (ou use uma query existente)

Passo 2: Copiar e Colar o Script SQL

1. Abra o arquivo `supabase_schema.sql` que está na raiz do projeto
2. **Copie TODO o conteúdo** do arquivo
3. **Cole** no SQL Editor do Supabase

Passo 3: Executar o Script

1. Clique no botão **“Run”** (ou pressione `Ctrl+Enter`)
2. Aguarde a execução (deve levar 5-10 segundos)
3. Você verá a mensagem: **“Success. No rows returned”**

Passo 4: Verificar as Tabelas Criadas

1. No menu lateral, clique em **"Table Editor"**

2. Você deve ver **5 tabelas**:

- ☒ categories
- ☒ products
- ☒ purchases
- ☒ purchase_items
- ☒ receipts



Passo 5: Redeployar a Aplicação (Opcional)

Como a **connection string já está configurada** corretamente, a aplicação deve funcionar automaticamente após a execução do script.

Mas para garantir, você pode:

1. Voltar aqui no chat
2. Me avisar que executou o script
3. Eu faço um redeploy para garantir



O Que o Script Faz?

1. Limpeza (Remove tabelas incompatíveis)

```
DROP TABLE IF EXISTS "bills" CASCADE;  
DROP TABLE IF EXISTS "product_aliases" CASCADE;  
DROP TABLE IF EXISTS "suppliers" CASCADE;
```

2. Criação das Tabelas Corretas

- **categories** - Categorias de produtos (Frutas, Carnes, Laticínios, etc)
- **products** - Catálogo de produtos normalizados
- **purchases** - Registro de compras (cupons fiscais processados)
- **purchase_items** - Itens individuais de cada compra
- **receipts** - Histórico de uploads de notas fiscais

3. Índices para Performance

- Índice em `products.normalized_name` (busca rápida de produtos)
- Índice em `purchases.purchase_date` (ordenação por data)
- Índice em `purchases.supplier_name` (busca por fornecedor)
- Índices em `purchase_items` para joins eficientes

4. Foreign Keys (Relacionamentos)

- `products` → `categories` (categoria do produto)
- `purchase_items` → `purchases` (item pertence a uma compra)

- `purchase_items` → `products` (item referencia um produto)
 - `purchase_items` → `categories` (item tem categoria)
 - `receipts` → `purchases` (nota fiscal referencia uma compra)
-

✓ Após Executar o Script

A aplicação estará **100% funcional** e você poderá:

- ✓ **Fazer upload de cupons fiscais**
 - ✓ **Processar OCR automaticamente** (GPT-4o)
 - ✓ **Consultar histórico de compras**
 - ✓ **Buscar produtos** (com autocomplete)
 - ✓ **Ver relatórios e gráficos**
 - ✓ **Categorizar produtos**
 - ✓ **Comparar preços** entre compras
-

SOS Se Tiver Algum Problema

Se der algum erro ao executar o script:

1. **Copie a mensagem de erro completa**
 2. **Cole aqui no chat**
 3. Eu te ajudo a resolver!
-

🎯 Status Atual

- ✓ **Aplicação deployada** em produção
 - ✓ **Connection string** configurada com Session Pooler (IPv4 compatível)
 - ✓ **Código no GitHub** atualizado
 - ⌚ **Aguardando:** Execução do script SQL no Supabase
-

Qualquer dúvida, é só me avisar! 😊