

Documentação de Modelagem de Dados

1. Padrão de Modelagem e Justificativa

O esquema apresentado segue o **Modelo Relacional** (SQL), que é o padrão da indústria para sistemas que exigem **integridade transacional, consistência de dados e relações bem definidas** entre as entidades. Este modelo é ideal para aplicações de e-commerce ou sistemas de pedidos, onde a precisão de informações como pedidos, inventário e clientes é crítica.

O modelo está **parcialmente normalizado** (parecendo estar na Terceira Forma Normal - 3NF, ou próximo), com tabelas dedicadas para entidades como Customers, Products, Categories e as transacionais Orders e Order_Items.

2. Escolha do Banco de Dados: PostgreSQL

Justificativa:

A escolha do **PostgreSQL** é altamente recomendada para este esquema por vários motivos que impactam diretamente a performance e a escalabilidade:

1. **Conformidade SQL e Integridade:** É um dos bancos de dados relacionais mais avançados, garantindo a **ACID** (Atomicidade, Consistência, Isolamento, Durabilidade), crucial para operações financeiras e de pedidos.
2. **Performance em Leitura e Escrita:** O PostgreSQL é conhecido por seu robusto otimizador de consultas e eficiente gerenciamento de concorrência (via **MVCC** - Multi-Version Concurrency Control), o que é vital para sistemas com alto volume de pedidos (escritas) e consultas (leituras) simultâneas.
3. **Tipos de Dados Avançados:** Suporte nativo para tipos como UUID (usado nas chaves primárias), que ajuda na geração de IDs únicos e distribuídos, e bigserial/int8 para a tabela Orders, garantindo capacidade para um grande volume de transações.
4. **Recursos Avançados de Performance:**
 - **Índices Flexíveis:** Suporta diversos tipos de índices (B-Tree, GIN, GiST, etc.).
 - **Restrições CHECK:** As restrições CHECK em orders para status e status_payment garantem a **integridade dos dados** na camada do banco, reduzindo erros de aplicação.

- **Foreign Keys:** Gerenciamento eficiente de chaves estrangeiras, mantendo a consistência referencial.

3. Análise de Tabelas e Chaves

Comentários sobre as Tabelas:

- **Uso de UUID:** O uso de UUID para chaves primárias é uma excelente prática para sistemas distribuídos, pois permite a geração de IDs únicos sem depender de um único servidor, facilitando a escalabilidade horizontal.
- **orders.id como BIGSERIAL:** Manter a PK de orders como BIGSERIAL (int8) é adequado. Como é a tabela com maior potencial de crescimento e mais acessada por um auto-incremento, usar um tipo de 64 bits garante capacidade para bilhões de registros.
- **Restrições de Domínio:** As cláusulas CHECK em orders.status e orders.status_payment são vitais para a **integridade dos dados**, garantindo que apenas valores predefinidos sejam inseridos.

4. Análise de Índices Criados e Performance

O esquema já define índices automaticamente para todas as **Chaves Primárias (PKs)** e **Restrições de Unicidade (UNIQUE)**, além de criar índices para as **Chaves Estrangeiras (FKs)**, o que é fundamental para a performance.

A criação automática desses índices garante que as operações de busca (leitura) e junção (JOIN) baseadas nas relações primárias e estrangeiras sejam realizadas com complexidade logarítmica ($O(\log n)$), e não linear ($O(n)$), resultando em **alta performance**, especialmente à medida que o volume de dados cresce.

Índice Criado (Automático)	Colunas	Tipo	Contribuição para a Performance
PK em catego... ▾	id ▾	B-Tree ▾	Busca rápida por ID de categoria.
UNIQUE em c... ▾	name ▾	B-Tree ▾	Garante a unicidade e permite busca rápida por nome.
PK em custom... ▾	id ▾	B-Tree ▾	Busca rápida por ID de cliente.
UNIQUE em c... ▾	cpf ▾	B-Tree ▾	Crucial para consultas que

			buscam clientes por CPF, garantindo rapidez.
PK em orders ▾	id ▾	B-Tree ▾	Busca rápida e ordenação natural dos pedidos.
PK em products ▾	id ▾	B-Tree ▾	Busca rápida por ID de produto.
FK em products ▾	category_id ▾	B-Tree ▾	Melhora a performance ao listar produtos por categoria (consulta comum).
PK em order_i... ▾	id ▾	B-Tree ▾	Busca rápida por ID do item do pedido.
FK em order_i... ▾	order_id ▾	B-Tree ▾	Crucial para buscar todos os itens de um pedido específico (principal consulta de detalhe do pedido).
FK em order_i... ▾	product_id ▾	B-Tree ▾	Ajuda em consultas de inventário ou análise de vendas por produto.
FK em orders ▾	customer_id ▾	B-Tree ▾	Essencial para buscar o histórico de pedidos de um cliente de forma eficiente.