Schema Overview
This schema models a sophisticated e-commerce platform with support for multi-warehouse inventory, product variations, hierarchical categories, promotions, and user-generated content. It emphasizes flexibility, scalability, and complex business logic.

Core Components

1. **User Management**
- **`users`**: Stores user credentials and metadata (email, hashed password, timestamps).
- **`addresses`**: Manages multiple shipping addresses per user with a `is_primary` flag for preferred addresses.

2. **Product Catalog**
- **`categories`**: Hierarchical product categorization using a **nested set model** (`lft`, `rgt`, `depth` columns) for efficient tree queries.
- **`products`**: Base product definitions linked to categories.
- **`product_variations`**: Supports multiple SKUs per product with:
  - JSONB `attributes` (e.g., size/color variations)
  - Dynamic pricing per variation
  - Date-bound availability

3. **Inventory System**
- **`warehouses`**: Geospatial-enabled storage locations (`GEOGRAPHY(POINT)` type).
- **`inventory`**: Tracks stock quantities per variation/warehouse with low-stock alerts.

4. **Order Pipeline**
- **`orders`**: Manages order lifecycle with status tracking (`order_status` ENUM).
- **`order_items`**: Captures purchased items with price-at-purchase snapshots.
- **`payments`**: Flexible payment tracking with multiple methods (`payment_method` ENUM) and statuses.

5. **User Engagement**
- **`reviews`**: Enforces one review per user/product with rating constraints (1-5 stars).

6. **Supplier Relationships**
- **`suppliers`** + **`product_suppliers`**: Tracks multiple suppliers per product with cost prices and lead times.

7. **Promotions Engine**
- **`discounts`**: Time-bound promotions with usage limits and type flexibility (% or fixed).
- **`product_discounts`**: Many-to-many relationship between products and discounts.

8. **Metadata & Relationships**
- **`product_tags`**: Flexible tagging system for products.
- **Triggers**: Auto-update `updated_at` timestamps for key tables.

Key Relationships
1. **User → Orders** (1:M): A user can place multiple orders.
2. **Product → Variations** (1:M): Multiple SKUs per product.
3. **Variation → Inventory** (1:M): Stock tracked across multiple warehouses.
4. **Order → Payments** (1:M): Multiple payment attempts per order.
5. **Category Hierarchy** (Self-referential): Parent/child relationships for nested categorization.

Advanced PostgreSQL Features
1. **ENUM Types**:
   - `order_status`, `payment_method`, `payment_status`
2. **Spatial Data**:
   - Warehouse locations stored as `GEOGRAPHY(POINT)`
3. **JSONB**:
   - Flexible storage of product variation attributes
4. **Arrays**:
   - `image_urls` in product variations
5. **Automated Timestamps**:
   - `created_at`/`updated_at` managed by triggers

Implicit Indexes
While explicit indexes were removed, PostgreSQL automatically creates:
- **Primary Key Indexes** (e.g., `user_id`, `order_id`)
- **Unique Constraints** (e.g., `username`, `email`, `sku`)

Use Cases Supported
1. Multi-warehouse inventory management
2. Complex product variants (e.g., clothing sizes/colors)
3. Hierarchical catalog navigation
4. Discount/promotion campaigns
5. User review moderation
6. Supplier cost analysis
7. Order fulfillment tracking

Extensibility
The schema can be enhanced with:
- Full-text search for product discovery
- Partitioning for large tables (orders, inventory)

- Materialized views for analytics
- Geospatial queries for delivery optimization

This design balances normalization for data integrity with practical denormalization (e.g., `price_at_purchase` snapshots) for performance and auditability.