

Ejemplo: Cuando compras, tu pedido se guarda aquí.

Redis: Maneja el carrito de compras temporal.

Ejemplo: Los productos que añades al carrito se guardan aquí por 48 horas.

MongoDB: Registra logs de pagos (respuestas de Stripe, Paypal, etc.).

Ejemplo: Si un pago falla, aquí vemos el error exacto.

## Tablas Principales y Relaciones

### Productos

products: Info básica (nombre, precio, descripción).

product\_variants: Variantes (colores, tamaños) de cada producto.

Relación: Un producto tiene muchas variantes.

categories: Categorías (ej: "Sofás", "Mesas") con opción de subcategorías.

### Carrito (Redis)

No es una tabla SQL, sino un almacén temporal en Redis.

Estructura:

```
{
  "user123": [
    {"variant_id": "abc", "quantity": 2},
    {"variant_id": "def", "quantity": 1}
  ]
}
```

### Pedidos & Pagos

orders: Guarda el resumen del pedido (total, dirección).

order\_items: Los productos comprados (relacionados con products).

payments: Datos del pago (éxito/fallo, método usado).

### Kits de Diseño

design\_templates: Combos prearmados (ej: "Kit Oficina").

design\_template\_products: Los productos que pertenecen a cada kit.

Productos → Variantes:

Un sofá puede venir en 3 colores (cada color es una variante).  
Así el stock se maneja por color/tamaño.

Kits → Productos:

Un kit "Oficina" puede incluir 1 mesa + 1 silla.

Se guarda en design\_template\_products.

Pedidos → Productos:

Al comprar, se registra qué variante se compró (order\_items).

DBML:

DBML:

// Esquema DBML para Ecommerce de Muebles

// Usuarios y Autenticación

```
Table users {
  user_id UUID [pk]
  email varchar(255) [not null, unique]
  password_hash varchar(255) [not null]
  full_name varchar(100)
  phone varchar(20)
  created_at timestamptz [default: `now()`]
  updated_at timestamptz [default: `now()`, note: "Se actualiza con trigger"]
}
```

```
Table user_roles {
  user_id UUID [ref: > users.user_id] // ON DELETE CASCADE
  role varchar(20)
  permissions jsonb
}
```

// Catálogo de Productos

```
Table categories {
  category_id serial [pk]
  name varchar(100) [not null]
  slug varchar(100) [not null, unique]
  parent_id int [ref: > categories.category_id]
  created_at timestamptz [default: `now()`]
  updated_at timestamptz [default: `now()`, note: "Se actualiza con trigger"]
}
```

```
Table products {
  product_id UUID [pk]
  sku varchar(50) [not null, unique]
  slug varchar(255) [not null, unique]
  type varchar(20) [not null, note: "physical, bundle, template"]
  attributes jsonb
  is_active boolean
  created_at timestamptz [default: `now()`]
```

```
    updated_at timestamptz [default: `now()`, note: "Se actualiza con trigger"]
}
```

```
Table product_variants {
  variant_id UUID [pk]
  product_id UUID [ref: > products.product_id] // ON DELETE CASCADE
  sku_suffix varchar(10)
  stock int
  price_modifier decimal(10,2) [default: 0, note: "Ajuste al precio base"]
  min_stock int [default: 5, note: "Alerta de reposición"]
  created_at timestamptz [default: `now()`]
  updated_at timestamptz [default: `now()`, note: "Se actualiza con trigger"]
  Indexes {
    (product_id, sku_suffix)
  }
}
```

// Gestión de Pedidos y Pagos

```
Table orders {
  order_id UUID [pk]
  user_id UUID [ref: > users.user_id]
  status varchar(20) [not null, note: "pending, paid, shipped, delivered, cancelled"]
  total decimal(12,2)
  created_at timestamptz [default: `now()`]
  updated_at timestamptz [default: `now()`, note: "Se actualiza con trigger"]
  Indexes {
    (user_id, status)
  }
}
```

```
Table order_items {
  item_id UUID [pk]
  order_id UUID [ref: > orders.order_id] // ON DELETE CASCADE
  variant_id UUID [ref: > product_variants.variant_id] // ON DELETE CASCADE
  quantity int
}
```

```
Table payments {
  payment_id UUID [pk]
  order_id UUID [ref: > orders.order_id, not null]
  gateway varchar(20)
  metadata jsonb
  amount decimal(12,2) [not null]
  currency char(3) [default: 'PEN']
  gateway_id varchar(255) [note: "ID de transacción en pasarela"]
  created_at timestamptz [default: `now()`]
  updated_at timestamptz [default: `now()`, note: "Se actualiza con trigger"]
  Indexes {
    (gateway, order_id)
  }
}
```

```
// Kits de diseño
Table design_templates {
  template_id UUID [pk]
  name varchar(255)
  slug varchar(255) [not null, unique]
  description text
  cover_image_url varchar(255)
  discount decimal(5,2)
  room_type varchar(50)
  style_tags jsonb
  created_at timestamptz [default: `now()`]
  updated_at timestamptz [default: `now()`, note: "Se actualiza con trigger"]
}

Table design_template_products {
  template_id UUID [ref: > design_templates.template_id] // ON DELETE CASCADE
  product_id UUID [ref: > products.product_id] // ON DELETE CASCADE
  quantity int
  is_optional boolean
}
```