**ERD**

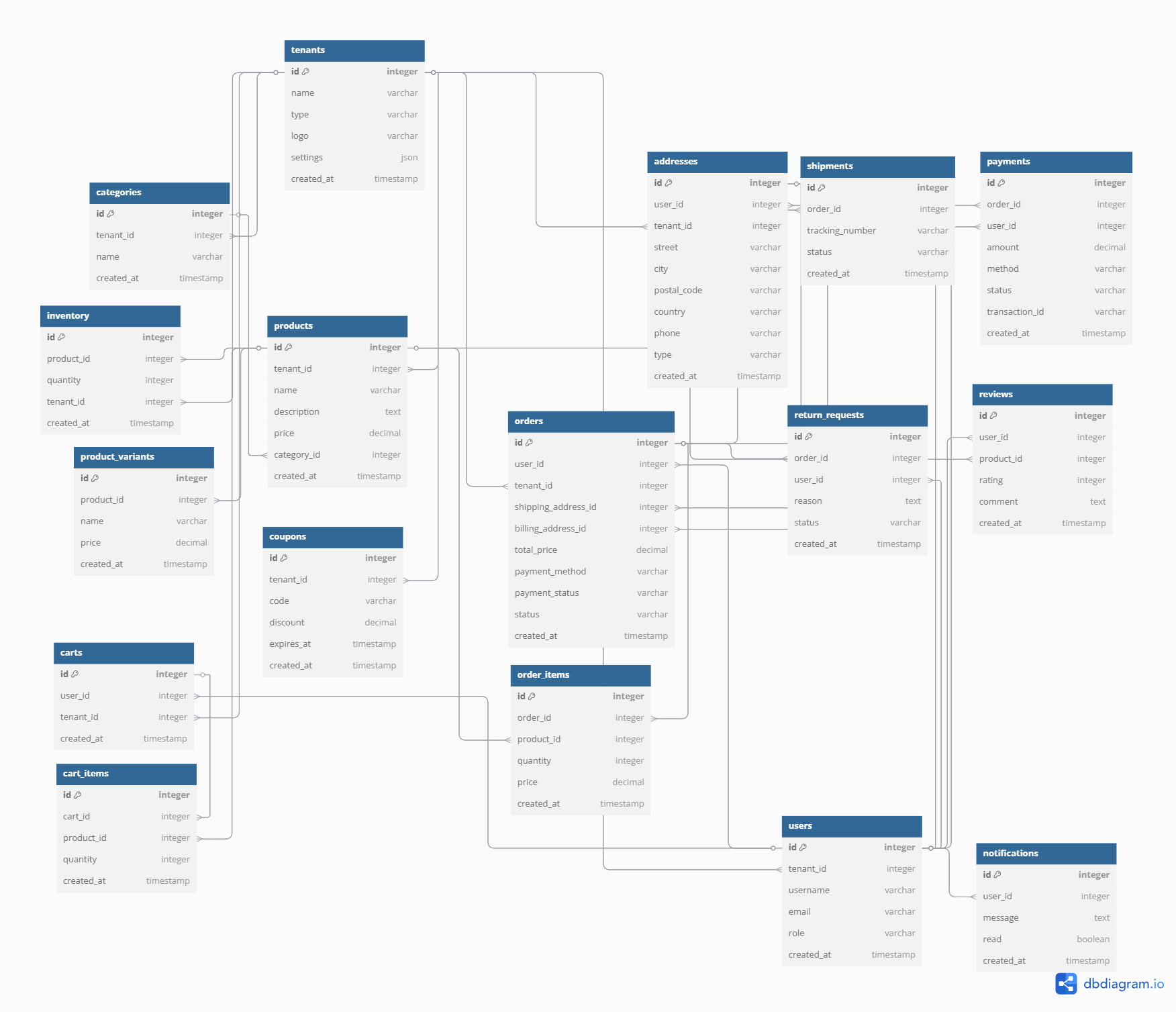


Table tenants {

id integer [primary key]

name varchar [note: 'Restaurant or Store name']

type varchar [note: 'restaurant or store']

logo varchar

settings json

created\_at timestamp

}

// Associations

// Ref: tenants.id < users.tenant\_id // One-to-Many (One Tenant has many Users)

// Ref: tenants.id < products.tenant\_id // One-to-Many (One Tenant has many Products)

// Ref: tenants.id < categories.tenant\_id // One-to-Many (One Tenant has many Categories)

// Ref: tenants.id < orders.tenant\_id // One-to-Many (One Tenant has many Orders)

// Ref: tenants.id < addresses.tenant\_id // One-to-Many (One Tenant has many Addresses)

// Ref: tenants.id < carts.tenant\_id // One-to-Many (One Tenant has many Carts)

// Ref: tenants.id < inventory.tenant\_id // One-to-Many (One Tenant has many Inventory items)

// Ref: tenants.id < coupons.tenant\_id // One-to-Many (One Tenant has many Coupons)

// Ref: tenants.id < support\_tickets.tenant\_id // One-to-Many (One Tenant has many Support Tickets)

Table users {

id integer [primary key]

tenant\_id integer

username varchar

email varchar

role varchar [note: 'admin, customer, seller, staff']

created\_at timestamp

}

// Associations

Ref: users.tenant\_id > tenants.id // Many-to-One (Many Users belong to One Tenant)

// Ref: users.id < orders.user\_id // One-to-Many (One User has many Orders)

// Ref: users.id < addresses.user\_id // One-to-Many (One User has many Addresses)

// Ref: users.id < carts.user\_id // One-to-One (One User has One Cart)

// Ref: users.id < reviews.user\_id // One-to-Many (One User can write many Reviews)

// Ref: users.id < payments.user\_id // One-to-Many (One User can make many Payments)

// Ref: users.id < notifications.user\_id // One-to-Many (One User can receive many Notifications)

// Ref: users.id < support\_tickets.user\_id // One-to-Many (One User can open many Support Tickets)

// Ref: users.id < return\_requests.user\_id // One-to-Many (One User can make many Return Requests)

Table products {

id integer [primary key]

tenant\_id integer

name varchar

description text

price decimal

category\_id integer

created\_at timestamp

}

// Associations

Ref: products.tenant\_id > tenants.id // Many-to-One (Many Products belong to One Tenant)

Ref: products.category\_id > categories.id // Many-to-One (Many Products belong to One Category)

// Ref: products.id < product\_variants.product\_id // One-to-Many (One Product has many Variants)

// Ref: products.id < order\_items.product\_id // One-to-Many (One Product can be in many Order Items)

// Ref: products.id < cart\_items.product\_id // One-to-Many (One Product can be in many Cart Items)

// Ref: products.id < reviews.product\_id // One-to-Many (One Product can have many Reviews)

// Ref: products.id < inventory.product\_id // One-to-One (One Product has One Inventory record)

Table product\_variants {

id integer [primary key]

product\_id integer

name varchar [note: 'Size, Color, etc.']

price decimal

created\_at timestamp

}

// Associations

Ref: product\_variants.product\_id > products.id // Many-to-One (Many Variants belong to One Product)

Table categories {

id integer [primary key]

tenant\_id integer

name varchar

created\_at timestamp

}

// Associations

Ref: categories.tenant\_id > tenants.id // Many-to-One (Many Categories belong to One Tenant)

// Ref: categories.id < products.category\_id // One-to-Many (One Category has many Products)

Table orders {

id integer [primary key]

user\_id integer

tenant\_id integer

shipping\_address\_id integer

billing\_address\_id integer

total\_price decimal

payment\_method varchar [note: 'cod, advance, card']

payment\_status varchar [note: 'pending, paid, failed']

status varchar [note: 'pending, completed, canceled']

created\_at timestamp

}

// Associations

Ref: orders.user\_id > users.id // Many-to-One (Many Orders belong to One User)

Ref: orders.tenant\_id > tenants.id // Many-to-One (Many Orders belong to One Tenant)

Ref: orders.shipping\_address\_id > addresses.id // One-to-One (One Order has One Shipping Address)

Ref: orders.billing\_address\_id > addresses.id // One-to-One (One Order has One Billing Address)

// Ref: orders.id < order\_items.order\_id // One-to-Many (One Order has many Order Items)

// Ref: orders.id < payments.order\_id // One-to-One (One Order has One Payment)

// Ref: orders.id < shipments.order\_id // One-to-One (One Order has One Shipment)

// Ref: orders.id < return\_requests.order\_id // One-to-Many (One Order can have many Return Requests)

Table order\_items {

id integer [primary key]

order\_id integer

product\_id integer

quantity integer

price decimal

created\_at timestamp

}

// Associations

Ref: order\_items.order\_id > orders.id // Many-to-One (Many Order Items belong to One Order)

Ref: order\_items.product\_id > products.id // Many-to-One (Many Order Items belong to One Product)

Table addresses {

id integer [primary key]

user\_id integer

tenant\_id integer

street varchar

city varchar

postal\_code varchar

country varchar

phone varchar

type varchar [note: 'shipping, billing']

created\_at timestamp

}

// Associations

Ref: addresses.user\_id > users.id // Many-to-One (Many Addresses belong to One User)

Ref: addresses.tenant\_id > tenants.id // Many-to-One (Many Addresses belong to One Tenant)

Table payments {

id integer [primary key]

order\_id integer

user\_id integer

amount decimal

method varchar [note: 'cod, advance, card, bank\_transfer']

status varchar [note: 'pending, paid, failed']

transaction\_id varchar [note: 'For advance payments']

created\_at timestamp

}

// Associations

Ref: payments.order\_id > orders.id // One-to-One (One Payment belongs to One Order)

Ref: payments.user\_id > users.id // Many-to-One (Many Payments belong to One User)

Table carts {

id integer [primary key]

user\_id integer

tenant\_id integer

created\_at timestamp

}

// Associations

Ref: carts.user\_id > users.id // One-to-One (One Cart belongs to One User)

Ref: carts.tenant\_id > tenants.id // Many-to-One (Many Carts belong to One Tenant)

// Ref: carts.id < cart\_items.cart\_id // One-to-Many (One Cart has many Cart Items)

Table cart\_items {

id integer [primary key]

cart\_id integer

product\_id integer

quantity integer

created\_at timestamp

}

// Associations

Ref: cart\_items.cart\_id > carts.id // Many-to-One (Many Cart Items belong to One Cart)

Ref: cart\_items.product\_id > products.id // Many-to-One (Many Cart Items belong to One Product)

Table reviews {

id integer [primary key]

user\_id integer

product\_id integer

rating integer [note: '1 to 5']

comment text

created\_at timestamp

}

// Associations

Ref: reviews.user\_id > users.id // Many-to-One (Many Reviews belong to One User)

Ref: reviews.product\_id > products.id // Many-to-One (Many Reviews belong to One Product)

Table inventory {

id integer [primary key]

product\_id integer

quantity integer

tenant\_id integer

created\_at timestamp

}

// Associations

Ref: inventory.product\_id > products.id // One-to-One (One Inventory belongs to One Product)

Ref: inventory.tenant\_id > tenants.id // Many-to-One (Many Inventory items belong to One Tenant)

Table coupons {

id integer [primary key]

tenant\_id integer

code varchar

discount decimal

expires\_at timestamp

created\_at timestamp

}

// Associations

Ref: coupons.tenant\_id > tenants.id // Many-to-One (Many Coupons belong to One Tenant)

Table shipments {

id integer [primary key]

order\_id integer

tracking\_number varchar

status varchar [note: 'pending, shipped, delivered']

created\_at timestamp

}

// Associations

Ref: shipments.order\_id > orders.id // One-to-One (One Shipment belongs to One Order)

Table notifications {

id integer [primary key]

user\_id integer

message text

read boolean

created\_at timestamp

}

// Associations

Ref: notifications.user\_id > users.id // Many-to-One (Many Notifications belong to One User)

Table return\_requests {

id integer [primary key]

order\_id integer

user\_id integer

reason text

status varchar [note: 'pending, approved, rejected']

created\_at timestamp

}

// Associations

Ref: return\_requests.order\_id > orders.id // Many-to-One (Many Return Requests belong to One Order)

Ref: return\_requests.user\_id > users.id // Many-to-One (Many Return Requests belong to One User)