



## Entities

- user
- employee
- owner
- customer
- shoppingCart
- products
- orders
- orderinfo
- shippinginfo
- billinginfo
- orderproducts

**KEY**  
— = Primary Key  
Yellow Box = Foreign Key

## Relational schema

- users(userID, username, pass, email, isEmployee, isOwner)
- shoppingCart(userID, prodID, qty)
  - userID from users
  - prodID from products
- products(prodID, prodName, descr, qtyAvailable, price, imageLink)
- orders(orderID, userID, billingID, shippingID, total, notes, shippingNumber, orderStatus)
  - userID from users
  - billingID from billing Info
  - shippingID from shipping Info
- orderinfo(infoID, recipientName, street, city, stateAbbr, zip, isBilling, cardNumber, cvc, expMon, expYear)
- orderproducts(orderID, prodID, qty)
  - orderID from orders
  - prodID from products

## Relationship

- User to Customer = (0, many)
- User to Employee = (0, 1)
- User to Owner = (1, many)
- Customer to Shopping Cart = (0, many)
- Customer to Product = (0, many)
- Shopping Cart to Order = (0, many)
- Owner to Product = (1, many)
- Employee to Order = (1, many)
- Order to orderInfo = (1, many)
  - orderInfo to Shipping Info = (many, 1)
  - orderInfo to Billing Info (many, 1)
- orders to orderproducts(1, m)

## Entities description:

- user → Is used for the client to have personal accounts
- employee → Is used to show which worker will fulfill the order for the client
- owner → Is used to check and update the products
- customer → Is the user who will place the order by checking out products and adding them to the shopping cart
- shoppingcart → Is used to store products to eventually buy
- products → stores the information on the products
- orders → Stores the clients final order to be made
- orderinfo → Stores clients info to where the products are going
- shippinginfo → to be able to update tracking
- billinginfo → to charge the user for the order
- orderproducts → Stores the orders made

## Relationship description

- users to customer will be  $(0, m)$  since it can be anywhere from 0 clients to how many will become a user
- user to Employee is  $(0, 1)$  since 1 employee will take care of preparing the order
- User to Owner is  $(1, m)$  since there may be 1 or more owners the user is placing an order from
- customer to shoppingcart is  $(0, m)$  as there may be from 0 customers, to many containing a filled shoppingcart
- customer to products is  $(0, m)$  as there may be out of stock products so 0 to many to choose from
- shoppingcart to Order is  $(0, m)$  since from 0 shoppingcarts can be to many depending on amount of orders
- Owner to products is  $(0, m)$  as the owners may be out of stock to having plenty of products depending on how they restock
- Employee to order is  $(1, m)$  as there may be 1 worker preparing the order to many others helping due to type of order
- Order to orderinfo is  $(1, m)$  as their will be always be orderinfo upon an order
- orderinfo to shippinginfo is  $(m, 1)$  as there will be same amount of info for either one

- Orderinfo to billinginfo is (m,1) as there will always be billing upon an order
- Orders to orderproducts is (1,m) as per order will show the order-products

### Attributes Description:

- users → userID to create a primary key for account
- users → username, pass, email private info to keep account secure
- users → isEmployee, isOwner to identify difference between work accounts and clients accounts
- shoppingcart → userID, prodID is made to be able to identify the user buying, and the product being bought
- shoppingcart → qty is for showing the amount of the product in the shopping cart
- products → prodID is primary for products to have their own identification ID
- products → prodName, descr, qtyAvailable, price is all of the info on the product to show the clients what they may be looking for.
- products → imageLink is to link an image to the products
- orders → orderID is to track the orders that have been placed
- orders → userID, billingID, shippingID are the identifications for the user and the order made
- orders → total, notes, shippingNumber, orderstatus is for information on the order being made and its status
- orderinfo → infoID is an ID for identification of order information ,
- orderinfo → recipientName, street, city, stateAbbr, zip is the info that is used to get the order to the right place
- orderinfo → isBilling, cardNumber, CVC, expMon, expYear is to use payment information to make sure transaction is okay
- orderproducts → orderID, prodID is for identification of the products in the certain order
- orderproducts → qty is to show the amount ordered