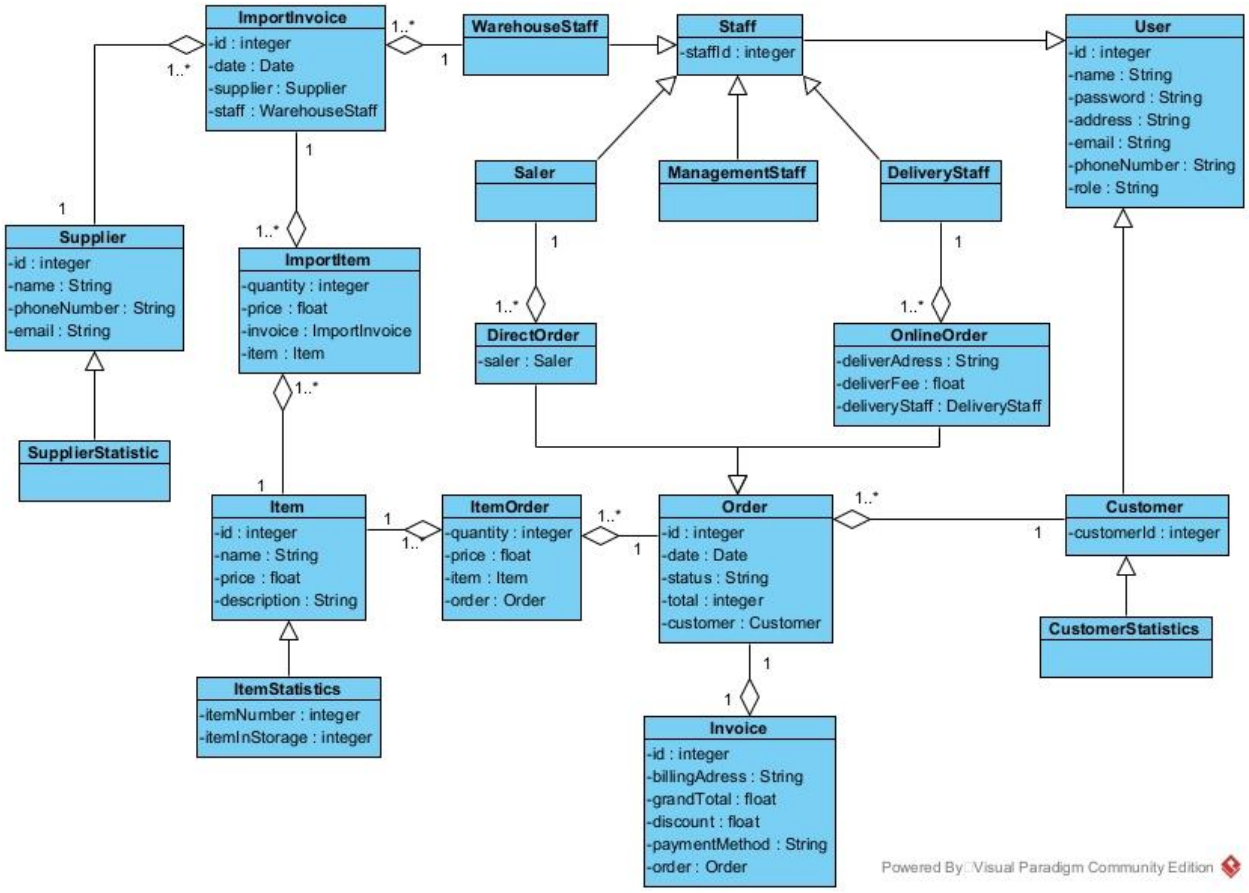
Design 1

Entity class diagram

- Step 1: Add id attribute to all class that is not inherit from other class
- Step 2: Add attribute type for all attribute. Attribute type must compatible with the chosen programming language
- Step 3: Change all association relationship to aggregation/composition
    - ImportInvoice associate Item create ImportItem: Change to ImportItem include ImportInvoice and Item
    - Item associate Order create ItemOrder: Change to ItemOrder include Item and Order
- Step 4: Add class atribute to entity class
    - ImportInvoice has Supplier, WarehouseStaff
    - ImportItem has ImportInvoice, Item
    - DirectOrder has Saler
    - OnlineOrder has DeliveryStaff
    - ItemOrder has Item, Order
    - Order has Customer
    - Invoice has Order



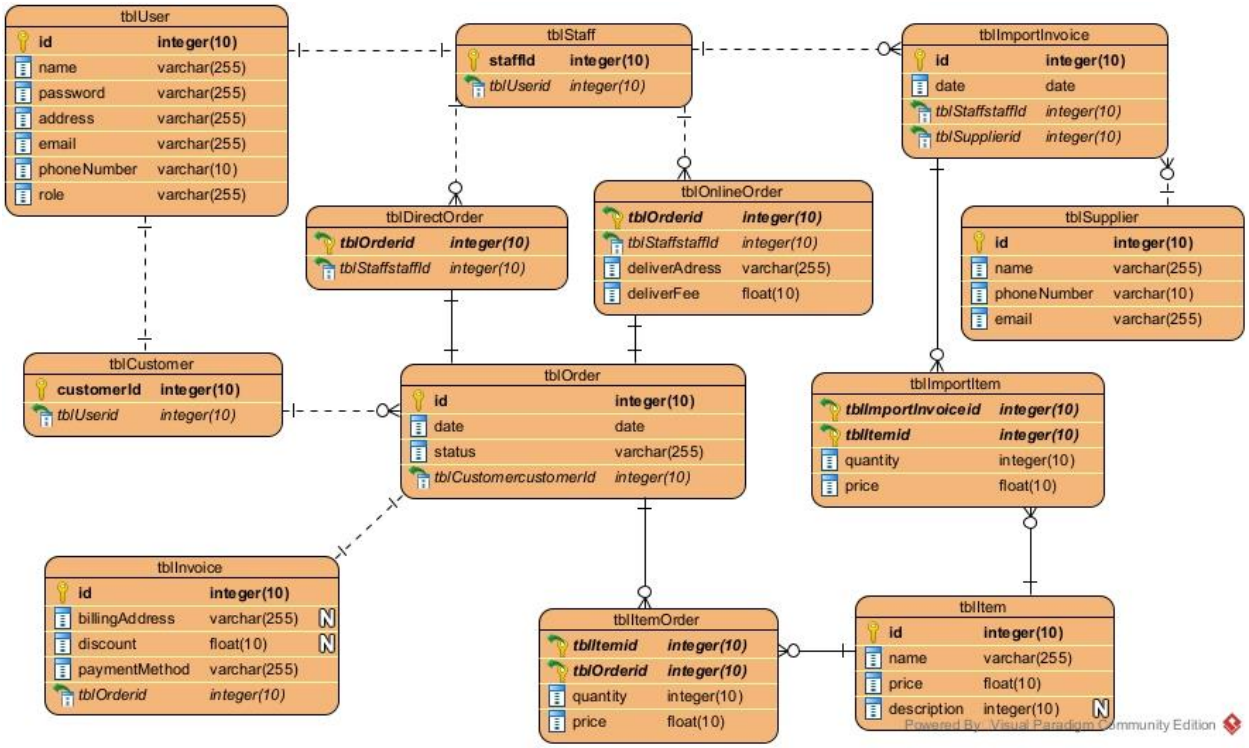
Database

- Step 1: For each class entity, create a table
    - User: tblUser
    - Staff: tblStaff
    - Customer: tblCustomer
    - Order: tblOrder
    - DirectOrder: tblDirectOrder
    - OnlineOrder: tblOnlineOrder
    - ImportInvoice: tblImportInvoice
    - Invoice: tblInvoice
    - Item: tblItem
    - Supplier: tblSupplier
    - ImportItem: tblImportItem
- Step 2: For each entity class, take all attribute type except entity attribute
    - tblUser:
        - id: integer(10)
        - name: varchar(255)
        - password: varchar(255)

        - address: varchar(255)
        - email: varchar(255)
        - phoneNumber: varchar(255)
        - role: varchar(255)
    - tblStaff:
        - staffId: integer(10)
        - tblUserid: integer(10)
    - tblCustomer:
        - customerId: integer(10)
        - tblUserid: integer(10)
    - tblOrder:
        - id: integer(10)
        - date: date
        - status: varchar(255)
        - total: float(10)
        - tblCustomercustomerId: integer(10)
    - tblInvoice:
        - id: integer(10)
        - billingAddress: varchar(255)
        - discount: float(10)
        - grandTotal: float(10)
        - paymentMethod: varchar(255)
        - tblOrderid: integer(10)
    - tblItemOrder
        - quantity: integer(10)
        - price: float(10)
        - tblItemid: integer(10)
        - tblOrderid: integer(10)
    - tblItem
        - id: integer(10)
        - name: varchar(255)
        - price: float(10)
        - description: varchar(255)
    - tblDirectOrder
        - tblStaffstaffId: integer(10)
        - tblOrderid: integer(10)
    - tblOnlineOrder
        - tblStaffstaffId: integer(10)
        - tblOrderid: integer(10)
        - deliverAdress: varchar(255)
        - deliverFee: integer(10)
    - tblImportInvoice
        - id: integer(10)
        - date: date

        - tblStaffstaffId: integer(10)
        - tblSupplierid: integer(10)
    - tblImportItem
        - quantity: integer(10)
        - price: float (10)
        - tblItemid: integer(10)
        - tblImportInvoiceid: integer(10)
    - tblSupplier
        - id: integer(10)
        - name: varchar(255)
        - phoneNumber: varchar(10)
        - email: varchar(255)
- Step 3: Consider quantity relationship between classes:
    - tblUser – tblStaff: 1 – 1
    - tblUser – tblCustomer: 1 – 1
    - tblStaff – tblDirectOrder: 1 – n
    - tblStaff – tblOnlineOrder: 1 – n
    - tblDirectOrder – tblOrder: 1 – 1
    - tblOnlineOrder – tblOrder: 1 – 1
    - tblCustomer – tblOrder: 1 – n
    - tblOrder – tblInvoice: 1 – 1
    - tblOrder – tblItemOrder: 1 – n
    - tblItem – tblItemOrder: 1 – n
    - tblStaff – tblImportInvoice: 1 – n
    - tblSupplier – tblImportInvoice: 1 – n
    - tblImportInvoice – tblImportItem: 1 – n
    - tblItem – tblImportItem: 1 -n
- Step 4: Add primary key and foreign key for all table
    - tblUser:
        - Primary Key: id
    - tblStaff:
        - Primary Key: staffId
        - Foreign Key: tblUserid → tblUser.id
    - tblCustomer:
        - Primary Key: customerId
        - Foreign Key: tblUserid → tblUser.id
    - tblOrder:
        - Primary Key: id
        - Foreign Key: tblCustomercustomerId → tblCustomer.customerId
    - tblInvoice:
        - Primary Key: id

        - Foreign Key: tblOrderid → tblOrder.id
    - tblItemOrder:
        - Primary Key: tblItemid, tblOrderid
            - Foreign Keys: tblItemid → tblItem.id,t blOrderid → tblOrder.id
    - tblItem:
        - Primary Key: id
    - tblDirectOrder:
        - Primary Key: tblOrderid
        - Foreign Keys: tblStaffstaffId → tblStaff.staffId, tblOrderid → tblOrder.id
    - tblOnlineOrder:
        - Primary Key: tblOrderid
        - Foreign Keys: tblStaffstaffId → tblStaff.staffId, tblOrderid → tblOrder.id
    - tblImportInvoice:
        - Primary Key: id
        - Foreign Keys: tblStaffstaffId → tblStaff.staffId, tblSupplierid → tblSupplier.id
    - tblImportItem:
        - Primary Key: tblItemid, tblImportInvoiceid
        - Foreign Keys: tblItemid → tblItem.id, tblImportInvoiceid → tblImportInvoice.id
    - tblSupplier
        - Primary Key: id
- Step 5: Remove abundant attribute
    - Attribute total in order
    - Attribute grandTotal in invoice


tblUser
id integer(10)
name varchar(255)
password varchar(255)
address varchar(255)
email varchar(255)
phoneNumber varchar(10)
role varchar(255)
tblStaff
staffId integer(10)
tblUserid integer(10)
tblImportInvoice
id integer(10)
date date
tblStaffstaffId integer(10)
tblSupplierid integer(10)
tblDirectOrder
tblOrderid integer(10)
tblStaffstaffId integer(10)
tblOnlineOrder
tblOrderid integer(10)
tblStaffstaffId integer(10)
deliverAdress varchar(255)
deliverFee float(10)
tblSupplier
id integer(10)
name varchar(255)
phoneNumber varchar(10)
email varchar(255)
tblCustomer
customerId integer(10)
tblUserid integer(10)
tblOrder
id integer(10)
date date
status varchar(255)
tblCustomercustomerId integer(10)
tblImportItem
tblImportInvoiceid integer(10)
tblItemid integer(10)
quantity integer(10)
price float(10)
tblInvoice
id integer(10)
billingAddress varchar(255) N
discount float(10) N
paymentMethod varchar(255)
tblOrderid integer(10)
tblItemOrder
tblItemid integer(10)
tblOrderid integer(10)
quantity integer(10)
price float(10)
tblItem
id integer(10)
name varchar(255)
price float(10)
description integer(10) N
Powered By Visual Paradigm Community Edition