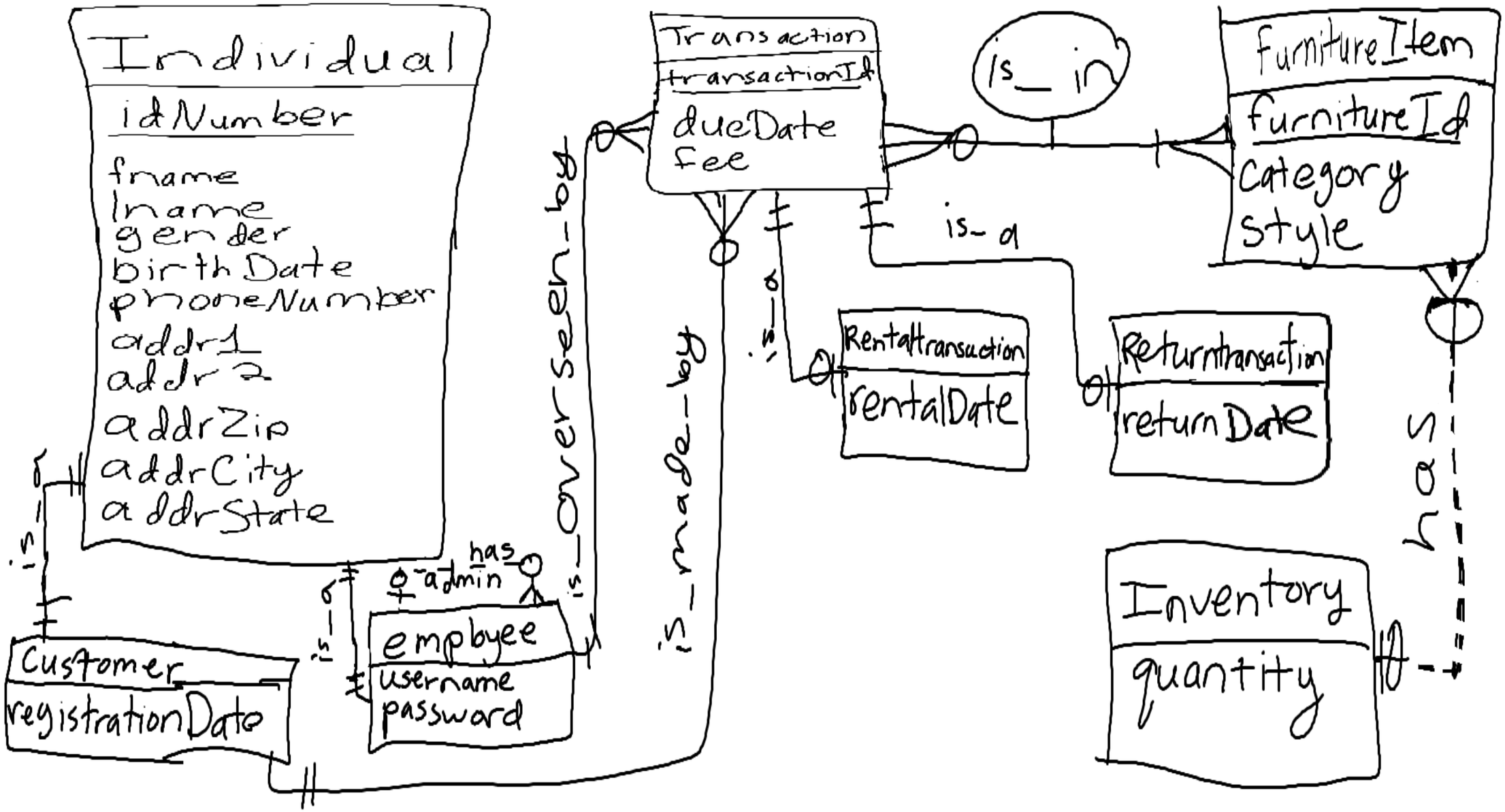
Individual
idNumber
fname
lname
gender
birthDate
phoneNumber
addr1
addr2
addrZip
addrCity
addrState
Transaction
transactionId
dueDate
fee
is_a
FurnitureItem
furnitureId
category
style
RentalTransaction
rentalDate
ReturnTransaction
returnDate
is_overseen_by
is_made_by
has
admin
employee
username
password
Customer
registrationDate
Inventory
quantity

Assumptions:

There is only 1 store thus a store table is not necessary and there is only 1 inventory. I'm not sure if this is a chain of stores or just 1.

Employee is a self-referencing table.

The top admin refers to themselves for their adminId.

Individual has 2 subtypes that are overlapping partial. Because an employee may rent from the store but an individual does not have to be a customer nor an employee.

A member can make multiple transactions on the same day with the same due date and fee. So, I gave transaction a surrogate key.

Fee is either the rental cost or the late fee depending on the type of transaction, it has a default value of $0.00 thus it is never null.

Any given transaction can be either a rental or a return but not both at one. Thus, transaction subtypes are disjoint total.

Any transaction must have at least 1 furniture.

Furniture does not need to be in a transaction to be part of the inventory.

The inventory can be empty. The key for each row would be (furnitureId). The inventory can be empty because an empty inventory will either contain 0 rows thus no row's key would be null, or the furnitureItems contained in it would have quantities of 0. It is functionally a collection.

Individual(idNumber, fname, lname, gender, birthdate, phoneNumber, addr1, addr2, addrZip, addrCity, addrState)

Customer(idNumber, registrationDate)

Employee(idNumber, adminId, username, password)

Transaction(transactionId, dueDate, fee)

RentalTransaction(transactionId, rentalDate)

ReturnTransaction(transactionId, returnDate)

furnitureItem(furnitureId, category, style)

Inventory(furnitureId, quantity)