Final Project
COEN 178
3 June 2018

Rental Management System for Happy Renter's Rental Management Inc.
Developed By: Andrew Nguyen and Kelly Ouye

Project Documentation

Andrew Nguyen and Kelly Ouye have designed and implemented a rental management system for Happy Renter's Rental Management Inc. This system will be used by branch employees and allows them to access information about employees, properties, tenants, and lease agreements. As a result of using a relational database and transaction management, rental managers will have an easier time keeping track of properties and lease agreements.

In order to implement the relational database we made the following assumptions:
Description

- A branch is identified by a branch number, phone, and address (street, city, and zip)
- A branch has several employees identified by their employee id, name, phone, and job designation
    - One manager who manages the supervisors
    - Several supervisors who manage several properties
- A rental property has a status (available or leased), a start date (if applicable), and a property owner
- A property owner is identified by a name, permanent address, and a phone number
    - An owner may own more than one property
- A lease agreement is identified by a renter, home phone, work phone, start day of lease, end day of lease, deposit amount, rent amount, and supervisor's name
    - A lease is created when a property is rented

Constraints

- A branch is managed by one manager
- A lease agreement should have a duration of six months to a fully year
- The status of the property should be changed to leased when the lease agreement is created
- A rental property should be removed from the supervisor's list when the rental property is no longer rented
- With every new lease, an increase of 10% will be added to the rent amount from the previous lease

Assumptions

- For Transaction 1, the list of Rental properties are generated based on the Branch Number. We are assuming that branches are identified by their Branch Number rather than a Branch Name to have a primary key.
- For Transaction 3, the list of properties is returned as the Property Numbers rather than the addresses
- For Transaction 5, the number of properties available for rent corresponds with the Branch Number
- For Transaction 6, we assume that the renter can deposit any amount before renting the Property
- Instead of creating a Renter's Entity, we put the renter's information in the lease agreement. The Transactions that require the Renter's Name can be parced from the Lease Agreement.
- For the Manager and Supervisor Tables, we included a phone attribute because the Employees Entity has the phone attribute, and Managers and Supervisors belong in the Employees Entity. However, the phone number of the employees are never used in any of the transactions.

In order to avoid redundancy problems and the update and deletion anomalies, we used Boyce Codd Normal Form to deconstruct the tables.

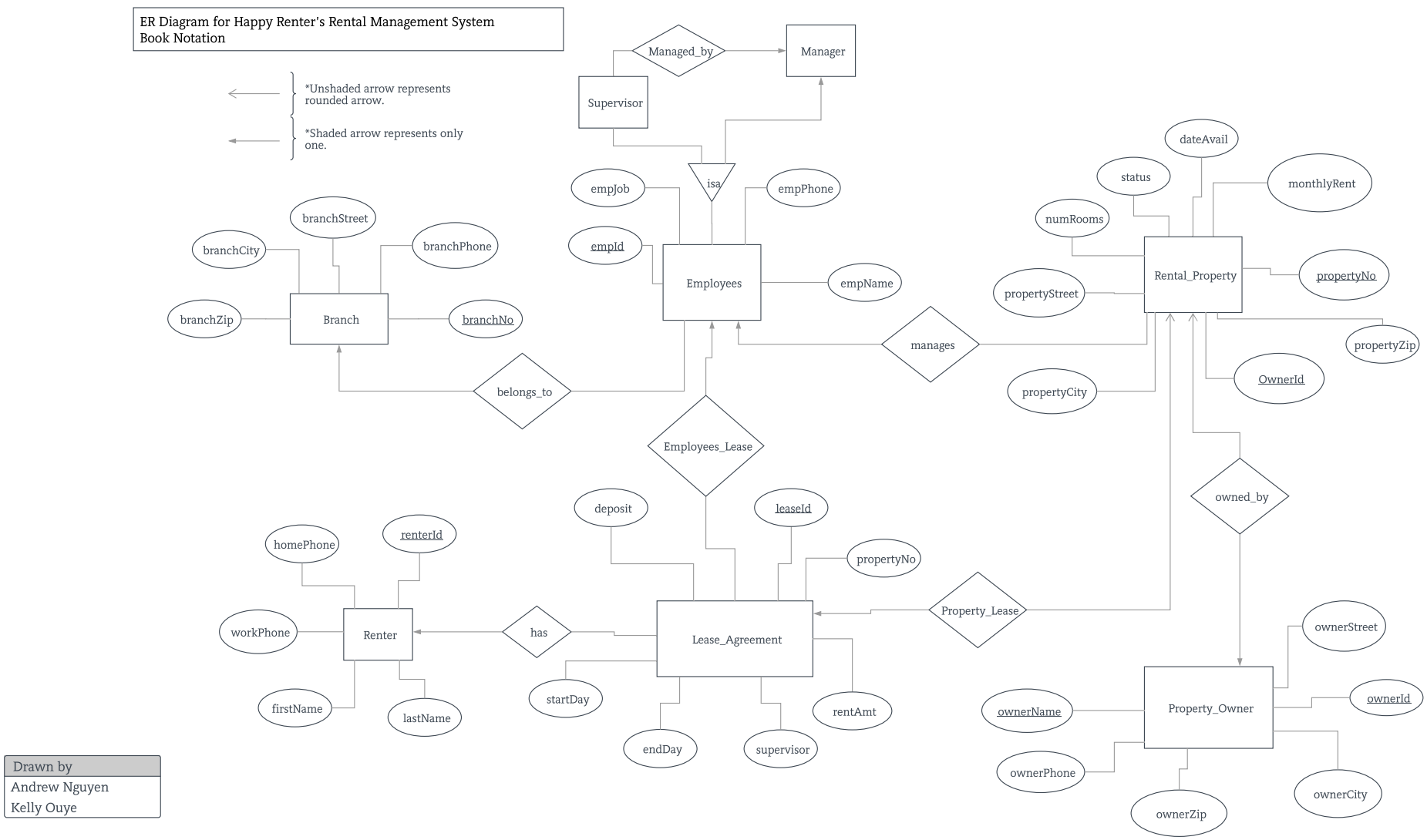
Functional Dependencies

Happy_Renter's(Manager, Supervisor, Branch, Employees, Property_Owner, Rental_Property, Lease_Agreement)

1. Each branch has exactly one manager
    a. Manager → Branch
2. Each supervisor is managed by one manager
    a. Supervisor → Manager
3. Each property is rented by one renter, owned by one property owner, has one lease agreement and is supervised by one supervisor
    a. Rental_Property → Renter, Property_Owner, Lease_Agreement, Supervisor
4. Each lease agreement has one rental property
    a. Lease_Agreement → Rental_Property
5. Each employee works at one branch
    a. Employees → Branch

Schemas and FDs

1. Manager(managerId, firstName, lastName, phone)
    a. managerId → firstName, lastName, phone
2. Branch(branchNo, managerId, branchPhone, streetNum, branchStreet, branchCity, branchZip)
    a. branchNo → managerId, branchPhone, streetNum, branchStreet, branchCity, branchZip
3. Supervisor(supervisorId, managerId, firstName, lastName, supervisorPhone)
    a. supervisorId → managerId, firstName, lastName, supervisorPhone
4. Employees(empId, branchId, firstName, lastName, empPhone, empJob)
    a. empId → branchId, firstName, lastName, empPhone, empJob
5. Property_Owner(ownerId, firstname, lastName, streetNo, ownerStreet, ownerCity, ownerZip, ownerPhone)
    a. ownerId → firstname, lastName, streetNo, ownerStreet, ownerCity, ownerZip, ownerPhone
6. Rental_Property(propertyNo, ownerId, supervisorId, streetNo, streetName, city, propertyZip, numRooms, monthlyRent, status, dateAvail)
    a. propertyNo → ownerId, supervisorId, streetNo, streetName, city, propertyZip, numRooms, monthlyRent, status, dateAvail
7. Lease_Agreement(leaseId, propertyNo, firstName, lastName, startDate, endDate, homePhone, workPhone, rentAmount, deposit)
    a. leaseId → propertyNo, firstName, lastName, startDate, endDate, homePhone, workPhone, rentAmount, deposit


ER Diagram for Happy Renter's Rental Management System
Book Notation
*Unshaded arrow represents rounded arrow.
*Shaded arrow represents only one.
Managed_by
Manager
Supervisor
isa
empJob
empPhone
empId
Employees
empName
branchStreet
branchCity
branchPhone
branchZip
Branch
branchNo
belongs_to
manages
dateAvail
status
monthlyRent
numRooms
Rental_Property
propertyNo
propertyStreet
propertyZip
propertyCity
OwnerId
Employees_Lease
owned_by
deposit
leaseId
propertyNo
homePhone
renterId
workPhone
Renter
has
Lease_Agreement
Property_Lease
ownerStreet
ownerName
Property_Owner
ownerId
startDay
rentAmt
firstName
lastName
endDay
supervisor
ownerPhone
ownerZip
ownerCity
Drawn by
Andrew Nguyen
Kelly Ouye

ER Diagram for Happy Renter's Rental Management System
Crow's Feet Notation