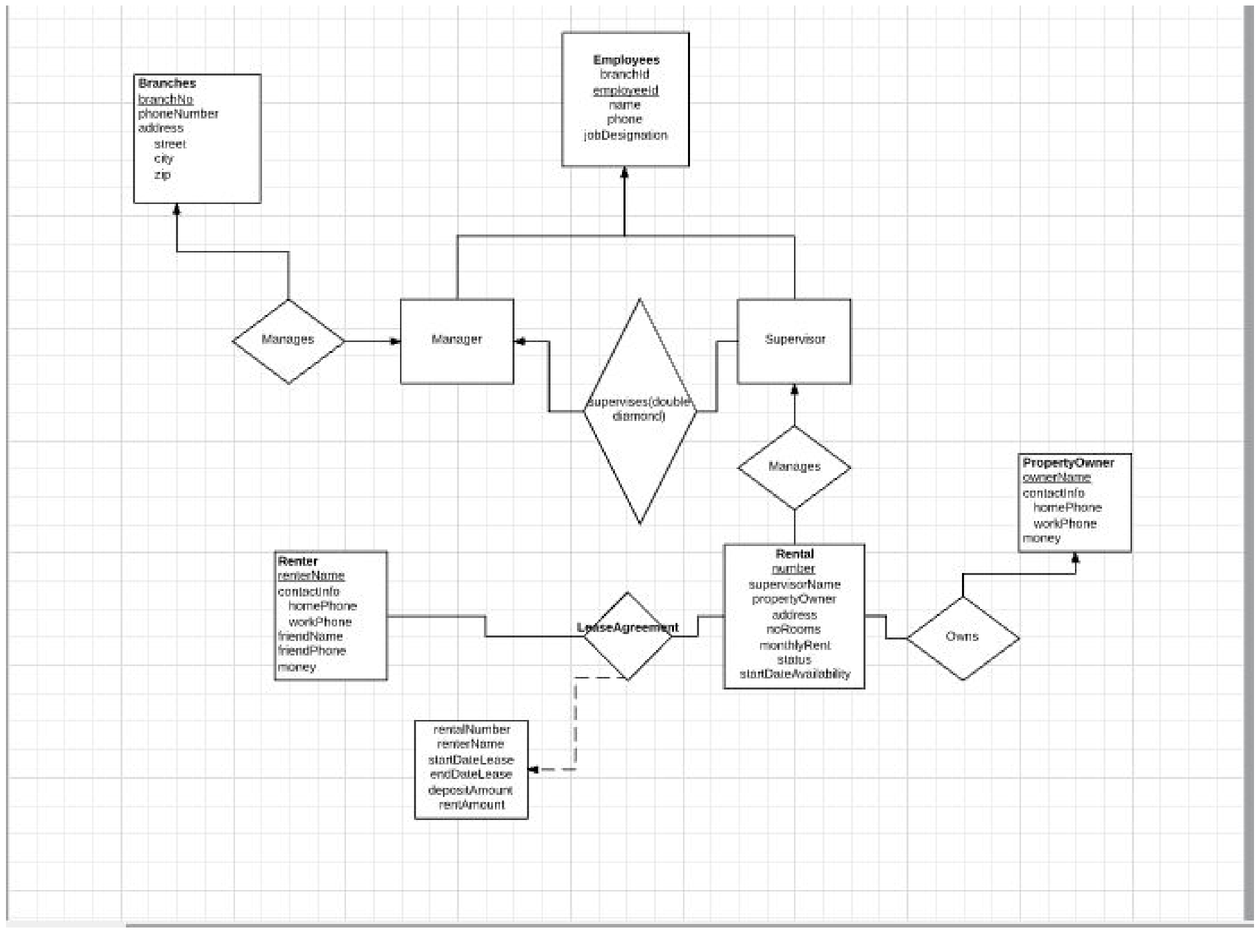
Alma Niu
Coen 178 Final Project Documentation

Description

In this system, we implemented a rental management system called Greenfield, which has several branches located in California. Within each branch, there exists a manager and several supervisors. The supervisors keep track of rental properties. Meanwhile, the owners can own one or more rental properties. When a renter want to rent a property, a lease agreement is created. Overall, our system keep track of transactions to ensure proper organization for rental properties.

ER Diagram



Functional Dependencies Used:

To find a functional dependencies in my table, I look for constraints that are generally true for all possible data.

<u>Branches Relation</u>
branchNo->phoneNo, street, city,zip

<u>Employees Relation</u>
EmployeesId->JobDesignation, Phone, Name,branchNo

<u>Manager Relation</u>
branchId->employeeId
Employeeid->branchId,name,phone,jobDesignation

<u>Supervisor Relation</u>
EmployeeId->branchId,name,phone,joDesignation

<u>PropertyOwner Relation</u>

OwnerName->homePhone,WorkPhone, Money

<u>Rental Property Relation</u>
RentNum,supervisorId,OwnName->city_Address,noRooms, montlyrent, status, startDateAvail

<u>Renter Relation</u>

rentername->homephone, workphone,friendName,friendPhone,money

<u>LeaseAgreement Relation</u>
rentalNumber->supervisorId
renternName ,rentalNumber,startDateLease->enddateLease, depositAmount, rentAmount

<u>Tables with Primary and Foreign Keys</u>
create table Branches (branchNo varchar (5) **primary key,** phoneNo Integer, street varchar(20), city varchar(5), zip integer);

create table Employees (employeeId varchar(5) **primary key**, branchId varchar(5), name varchar(20), phone integer,jobDesignation varchar(10) CHECK (jobDesignation in ('Manager','Supervisor')),
foreign key (branchId) references Branches(branchNo));

```
 create table Manager(employeeId varchar(5), startDate Date, foreign key (employeeId)
references Employees(employeeId));


create table Supervisor (employeeId varchar(5) primary key, startDate Date, foreign key
(employeeId) references Employees(employeeId));


create table PropertyOwner(ownerName varchar(20) primary key, homePhone integer,
workPhone integer, money decimal(10,2));

create table Rental(RentNum varchar (5) primary key, supervisorId varchar(5), OwnName
varchar(20), city_Address varchar(20), noRooms integer,
monthlyRent DECIMAL(7,2), status varchar(20) CHECK (status
in('available','leased')),startDateAvail date,
foreign key (supervisorId) references Employees(employeeId),foreign key (OwnName)
references PropertyOwner(ownerName));


create table Renter (renterName varchar(20) primary key, homePhone integer, workPhone
integer,friendName varchar(20), friendPhone varchar(20), money DECIMAL(7,2));

create table LeaseAgreement( renterName varchar(20), startDateLease date, endDateLease
date, depositAmount DECIMAL(7,2), rentAmount DECIMAL(7,2) , supId varchar (5),
foreign key (renterName) references Renter(renterName),
foreign key (supId) references Supervisor(employeeId));
```

Normalization Process Applied

To prevent redundancy, I first implemented an ER diagram such that a table consists of data only directly related to the primary keys. For instance, the PropertyOwner table consists of all attributes directly related to the primary key (ownername) of the table. We do not have, for instance, rentAmount included in the PropertyOwner table, because it will be redundant and rentAmount is saved in the Rental Property table. In addition, I made sure all the tables are in BCNF. Furthermore, I created a table for managers and supervisors. There is an Employees table where their employeeId's are displayed.

Queries And Results

-My results/output are shown in the QueryResults.txt. My code is located and labeled in separate files. I did not need to create a trigger for 1.2 d, I just did it using sql.

Additional Attributes

I added a startdate attribute for both my Manager and Supervisor table.