COEN 280 - Database Systems
Final Project

Stephen Tambussi
Dylan Hoover

StrawberryField Rental Management Inc.

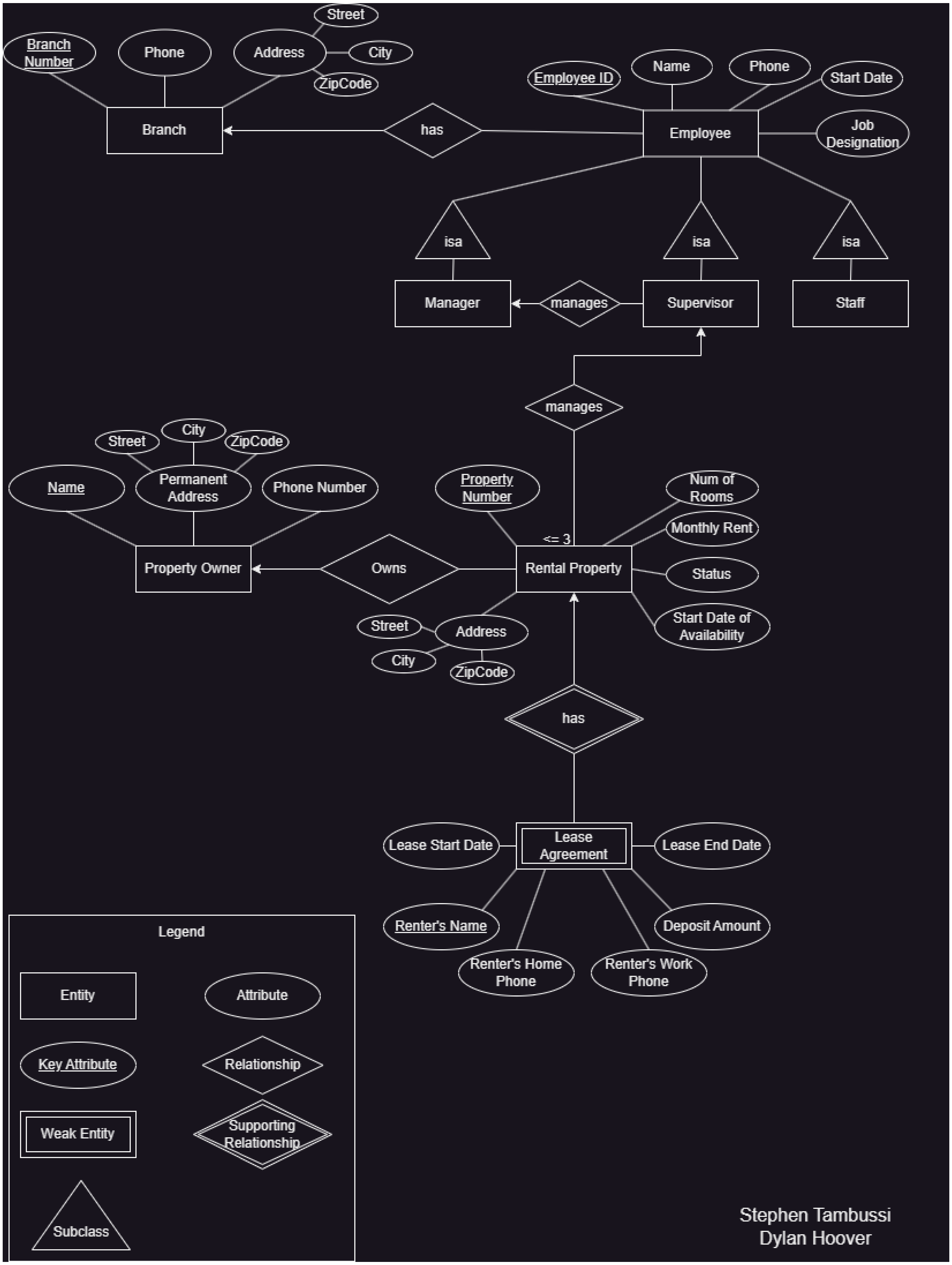
Description

The StrawberryField Rental Management System provides the company with a means of storing and organizing important information about rental properties, leases, and tenants. Built upon the Oracle Database Software, the system manages this information through a sequence of SQL queries. On the technical side, the system consists of 7 database tables and a variety of different SQL scripts for transactions. The schema for these tables may be found in the following sections of this document. The system implements the following transactions:

1. View a list of available rental properties for a specific branch
2. View a list of managers, supervisors, and properties associated with the supervisors
3. View a list of rental properties owned by a specific person in a particular branch
4. Show a list of available properties meeting the specified criteria (city, number of rooms, rent range)
5. Show the number of properties available for rent by branch
6. Create a lease agreement
7. Show a lease agreement for a renter
8. Display the renters who rented more than one rental property
9. Show the average rent for managed properties in each city
10. Display the properties with leases set to expire in the next two months from the current date
11. Show the money that is earned by the rental agency per month

In addition to the above transactions, the system provides a graphical user interface (GUI) via a web application that allows viewing available rental properties, creating a lease agreement, and displaying the lease agreement.

Entity-Relationship (ER) Diagram



ER to Relations

Functional Dependencies

Branch
BranchNumber → Phone, Street, City, ZipCode

Manager
EmployeeID → EmpName, Phone, StartDate, JobDesignation, BranchNumber

Supervisor
EmployeeID → EmpName, Phone, StartDate, JobDesignation, BranchNumber, ManagerID

Staff
EmployeeID → EmpName, Phone, StartDate, JobDesignation, BranchNumber

PropertyOwner
OwnerName → Street, City, ZipCode, PhoneNumber

RentalProperty
PropertyNumber → Street, City, ZipCode, NumOfRooms, MonthlyRent, Status, DateAvailable, SupervisorID, OwnerName

LeaseAgreement
(PropertyNumber, RenterName) → LeaseStart, LeaseEnd, RenterHomePhone, RenterWorkPhone, DepositAmount

Tables

Underlined = primary key
Red = foreign key

Branch

BranchNumber	Phone	Street	City	ZipCode

Manager

EmployeeID	EmpName	Phone	StartDate	JobDesignation	BranchNumber

Supervisor

EmployeeID	EmpName	Phone	StartDate	JobDesignation	BranchNumber	ManagerID

Staff

EmployeeID	EmpName	Phone	StartDate	JobDesignation	BranchNumber

PropertyOwner

OwnerName	Street	City	ZipCode	PhoneNumber

RentalProperty

Property Number	Street	City	ZipCode	NumOfRooms	MonthlyRent	Status	DateAvailable	Supervisor ID	Owner Name

LeaseAgreement

PropertyNumber, RenterName	LeaseStart	LeaseEnd	RenterHomePhone	RenterWorkPhone	DepositAmount

Additional Notes

- Some of the given constraints are implemented using triggers. They are as follows:
  - The supervisor limit on how many properties they can manage at once is implemented using a trigger.
  - When a lease is created for a property, a trigger updates the property's Status attribute to "unavailable".
  - A trigger handles the 10% rent increase with every new lease on a property. Furthermore, if the lease is for the minimum allowable duration (6 months), then the rent is increased by an additional 10%
- The GUI provides three transactions: show properties available, create a lease agreement, and show a lease agreement.
  - These transactions are additionally implemented using SQL scripts.
- Address is a composite attribute so only its simple component attributes are included in the tables (Street, City, ZipCode).
- LeaseAgreement is a weak entity as seen in the above ER diagram so it has two primary keys (PropertyNumber and RenterName).
- Manager, Supervisor, and Staff entities are implemented as subclasses of Employee, with each entity having its own associated table.
- The MonthlyRent attribute is only included in the RentalProperty entity to limit redundancy. The LeaseAgreement entity gets this attribute from RentalProperty for associated queries.
- The following scripts are included in this project to facilitate its functionality:
  - *insert_data.sql* - populates the tables with data
  - *run_transactions.sql* - runs all the scripts and captures their output into a text file called **project_output.txt**