CIS 3400 S3DA

Group 3 Project

*Rent & Go*

*Business Scenario:*

Our name is “Rent & Go.” We operate as a laptop rental system. Access to technology is a growing concern as our world moves toward a digitalization of daily processes in business and social activities. Not having a usable device or experiencing an unexpected malfunction can put busy people in tight spots. Fortunately, places such as libraries and colleges can confront this issue with our laptop rental system so that students do not have to fall behind in their work or communication due to a lack of personal devices. As information technology and online dependence continuously grow, it is imperative that systems of providing machine access become more efficient as well. Our goal is to make it easier for students, teachers, administrators to rent out computers from their system, while also allowing staff to have easier access to the current status of machines that are currently rented out.

Our system serves library staff, students, teachers, IT staff, and administrative personnel. With each computer, there is a serial ID, computer ID, operating systems, and ways to track its status and condition. We also collect basic information from renters for communication purposes. In addition, we make sure computers are in pristine condition and any in need of repair are properly fixed before continued use. Keeping track of customers, computers, staff, and conditions means happy customers and an organized, quality inventory. Each entity in our system is thoughtfully connected through relationships and accompanying attributes that identify individuals.

Technology continues to be a vital part of our lives, especially as our education is adopting remote and online alternatives. With our system, anyone who needs a computer can get one with ease and keep the momentum in their day-to-day lives. Students can stay on top of their classwork and not be excluded from online meetings. Teachers can grab one to view online platforms and communicate with students. IT Staff will be ready to assist those in need of technological assistance and keep track of where computers are as well as their conditions. With (our system’s name), your online presence is in good hands. Stay connected and keep your digital needs in check!

*Systems Analysis:*

Customers in need of a computer complete a transaction for a rental which is overseen by a vendor. Each transaction assigns a computer to be rented and each computer is linked to a manufacturer. In the case that a computer might need repairs, a repair work order can be requested, which consists of task information and is completed by a technician.

*Set of Relations:*

*\*\*\*All addresses in New York*

**Customer**(Customer\_ID(ID), Occupation, First\_Name, Last\_Name, Address\_1 , Address\_2 , City, State, ZipCode, PhoneNumber, Email, Fee)

This table contains all the information for individual customers. This contains background information and communication options for transactions. They can be informed about the status of their computer whether it is time for them to return the computer or they are waiting for a repair to be finished.

* Customer\_ID(ID): ID unique to a customer. A foreign key to check who the customer was in that specific transaction.
* Occupation: The occupation of the customer
* First\_Name: The first name of the customer
* Last\_Name: The last name of the customer
* Address\_1: The address of the customer, specifying the street number and street name of their address
* Address\_2: Any additional information related to the customer’s address, such as apartment numbers
* City: The city in which the customer resides
* State: The state in which the customer resides
* ZipCode: The zipcode of the customer’s address
* PhoneNumber: The primary phone number of the customer
* Email: The primary email address of the customer
* Fee: Any fees that the customer has to pay, such as for late returns

**Vendor**(Vendor(ID)(FK), Location, Installed\_By\_Self(Y/N))

This table displays information for the vendors in charge of the transactions. Vendor refers to the person, or machine, that the customer communicates with to rent out a computer. Once a customer completes a request, a vendor logs the transaction for the computer to get to the customer.

* Vendor\_ID(ID): Unique record number to specify an individual vendor
* Location: The location of the vendor. In-person would denote a person and online would denote a machine
* Installed\_By\_Self:

**Transaction**(Transaction\_ID(ID), Customer\_ID(FK), Computer\_ID(FK), Date\_Reserved, Date\_Due, Status, Transaction\_Cost, Vendor\_ID(FK))

This table contains all the laptop loan transaction that have been completed. The customer creates the transaction. The transaction shows the details needed from a manager such as who rented the computer, and when it was rented and when it was suppose to return. This is so the library can decide if a customer should be charged or not. Since computers are in multiple vendors, it is expected to know where the laptop was loaned from.

* Transaction\_ID(ID): Unique record number to indicate a specific transaction have occurred.
* Customer\_ID(FK): ID unique to a customer. A foreign key to check who the customer was in that specific transaction.
* Date\_Reserved: The date when the transaction first occurred
* Date\_Due: The date when a laptop is suppose to return
* Status: The status of computer if it has returned on time, still being loaned, or the laptop is late
* Transaction\_Cost: The cost of the late fee
* Vendor\_ID(FK): The vendor where the computer is loaned from

**Computer**(Computer\_ID(ID), Reserved(Y/N), Company\_Name(FK), OS\_System, Cost, Model, Damage\_Status, CPU, RAM)

This table contains the information about each individual computer in the system. Customer complete transactions to rent computers and they can also acquire the practical information regarding the computer they are using and compare them to select the right computer for their use.

* Computer\_ID(ID): The unique ID for each entity in this table, used to represent each unique computer device
* Reserved(Y/N): The status of whether a computer is reserved or not
* Company\_Name(FK): A foreign key connecting to the Manufacturer table, connecting which manufacturer made the computer
* OS\_System: The operating system of the computer
* Cost: The price value of the computer
* Model: The specific model of the computer brand
* Damage\_Status: The functional condition of the computer
* CPU: The central processing unit of the computer
* RAM: The amount of random-access memory of the computer

**Manufacturer**(Company\_ID, Company\_Name, Address\_1, Address\_2, City, State, ZipCode, PhoneNumber)

This table contains the information about the manufacturer for the computers used in the system. They can be contacted in case there is a specific inquiry about the computer hardware for repairs or warranty.

* Company\_ID(ID): The unique ID for each entity in this table, used to represent each unique manufacturer
* Company\_Name: The company name of the manufacturer
* Address\_1: The address of the manufacturer, specifying the street number and street name of their address
* Address\_2: Any additional information related to the manufacturer’s address, such as apartment numbers
* City: The city in which the manufacturer resides
* State: The state in which the manufacturer resides
* ZipCode: The zipcode of the manufacturer’s address
* PhoneNumber: The primary phone number of the manufacturer

**Technician**(Technician\_ID(ID), First\_Name, Last\_Name, Address\_1 , Address\_2 , City, State, ZipCode, PhoneNumber, Email)

This table contains all of the information about the various technicians what work for the library. They maintain the various computers that customers rent out for use, and record when computers need to be fixed, and what work needs to be done to get them back to being fully repaired.

* Technician\_ID(ID): The Unique ID for each entity in this table, used to represent each unique technician
* First\_Name: The first name of the technician
* Last\_Name: The last name of the technician. If the technician has a middle name, the first initial of the middle name is recorded here.
* Address\_1: The address of the technician, specifying the street number and street name of their address
* Address\_2: Any additional information related to the technician’s address, such as apartment numbers
* City: The city that the technician lives within
* State: The state that the technician lives within
* ZipCode: The zipcode of the technician’s address
* PhoneNumber: The primary phone number that the technician is associated with.
* Email: The email address that the technician is assigned to by the library.

**WorkOrder**( WorkOrderID(ID), TechnicianID(FK), ComputerID(FK), StartDate, CompletionDate, Description)

This table is used to keep track of the various work orders that need to be completed for each of the various computers. When a computer is damaged, a work order is made by a technician, specifying the date with which the work order is placed. The technician notes what needs to be repaired, whether it is a broken component or a software update, and what they need to do to fix the issue. Then, once the technician repairs the computer, that technician updates the respective entry, and

* WorkOrderID(ID): The unique, primary key used to identify each unique work order.
* TechnicianID(FK): A foreign key connecting to the Technician Table, connecting which technician is working on which specific work order
* ComputerID(FK): A foreign key connecting to the Computer Table, connecting which computer is being worked on in this work order, or if the work order is completed, what repairs have been made to fix the computer.
* StartDate: The day the technician receives the computer, and diagnoses the problem the needs to be fixed.
* CompletionDate: The day the technician completes the work order, and fixes the computer. If the computer is still being repaired, this field is left empty, i.e. null
* Description: A brief description of the issue that is affecting the computer, and how it will be fixed. If the problem is solved, the description explains how the computer was fixed.

*Relationship sentences:*

One Transaction must be made by one Customer

One Customer may make none or many Transactions

One Computer must be made by one Manufacturer

One Manufacturer must have manufactured one or many Computers

One Vendor may oversee one or many Transactions

One Transaction must be overseen by one Vendor

One Technician may be assigned to work on one or many Work Orders

One Work Order must be completed by one Technician

One Computer May be repaired through one Work Order

One Work Order must be associated with one Computer

One Transaction must be assigned one Computer

One Computer may be assigned to none or many Transactions

*SQL Database Schema:*

CREATE TABLE Customer

(

CustomerID VARCHAR(10) NOT NULL,

Occupation VARCHAR(35),

FirstName VARCHAR(35),

LastName VARCHAR(35),

Address1 VARCHAR(35),

Address2 VARCHAR(35),

city VARCHAR(35),

state VARCHAR(4),

zipcode VARCHAR(12),

PhoneNumber VARCHAR(15),

Email VARCHAR(35),

Fee NUMBER,

PRIMARY KEY (CustomerID)

)

CREATE TABLE Vendor

(

VendorID VARCHAR(10) NOT NULL,

Location VARCHAR(35)

InstalledBySelf BOOL,

PRIMARY KEY (VendorID)

)

CREATE TABLE Transaction

(

TransactionID VARCHAR(10) NOT NULL,

CustomerID VARCHAR(10)NOT NULL,

ComputerID VARCHAR(10)NOT NULL,

Address1 VARCHAR(35),

Date\_Reserved DATE

Date\_Due DATE

Status VARCHAR(35),

TransactionCost NUMBER,

VendorID VARCHAR(10) NOT NULL

PRIMARY KEY (TransactionID)

)

CREATE TABLE Computer

(

ComputerID VARCHAR(10) NOT NULL,

FirstName VARCHAR(35),

LastName VARCHAR(35),

Address1 VARCHAR(35),

Address2 VARCHAR(35),

city VARCHAR(35),

state VARCHAR(4),

zipcode VARCHAR(12),

PhoneNumber VARCHAR(15),

Email VARCHAR(35),

PRIMARY KEY (ComputerID)

)

CREATE TABLE Manufacturer

(

ManufacturerID VARCHAR(10) NOT NULL,

ComputerID VARCHAR(10) NOT NULL,

FirstName VARCHAR(35),

LastName VARCHAR(35),

Address1 VARCHAR(35),

Address2 VARCHAR(35),

city VARCHAR(35),

state VARCHAR(4),

zipcode VARCHAR(12),

PhoneNumber VARCHAR(15),

Email VARCHAR(35),

PRIMARY KEY (ManufacturerID)

)

CREATE TABLE Technician

(

TechnicianID VARCHAR(10) NOT NULL,

FirstName VARCHAR(35),

LastName VARCHAR(35),

Address1 VARCHAR(35),

Address2 VARCHAR(35),

city VARCHAR(35),

state VARCHAR(4),

zipcode VARCHAR(12),

PhoneNumber VARCHAR(15),

Email VARCHAR(35),

PRIMARY KEY (TechnicianID)

)

CREATE TABLE WorkOrder

(

WorkOrderID VARCHAR(10) NOT NULL,

TechnicianID VARCHAR(10) NOT NULL,

ComputerID VARCHAR(10) NOT NULL,

StartDate DATE,

CompletionDate DATE,

Description VARCHAR(256),

PRIMARY KEY (WorkOrderID)

)

CREATE TABLE Technician

(

TaskID VARCHAR(10) NOT NULL,

WorkOrderID VARCHAR(10) NOT NULL,

NextTaskID VARCHAR(10) NOT NULL,

CompletionStatus VARCHAR(35),

TaskDescription VARCHAR(256),

PRIMARY KEY (TaskID)

*Relational model:*

**Customer**(Customer\_ID(ID), Occupation, First\_Name, Last\_Name, Address\_1 , Address\_2 , ZipCode(FK), PhoneNumber, Email, Fee)

**Vendor**(Vendor(ID)(PK), Location, Installed\_By\_Self(Y/N))

**Transaction**(Transaction\_ID(ID), Customer\_ID(FK), Computer\_ID(FK), Date\_Reserved, Date\_Due, Status, Transaction\_Cost, Vendor\_ID(FK))

**Computer**(Computer\_ID(ID), Reserved(Y/N), Company\_Name, OS\_System, Cost, Model, Damage\_Status,CPU, RAM)

**Manufacturer**(Company\_ID(ID), Company\_Name, Address\_1, Address\_2, City, State, ZipCode(FK), PhoneNumber)

**Technician**(Technician\_ID(ID), First\_Name, Last\_Name, Address\_1 , Address\_2 , ZipCode(FK), PhoneNumber, Email)

**WorkOrder**( WorkOrderID(ID), TechnicianID(FK), ComputerID(FK), StartDate, CompletionDate, Description)

**Zipcode**(ZipCode(ID), City, State)

*Normalization:*

**Customer relation**

**Customer**(Customer\_ID(ID), Occupation, First\_Name, Last\_Name, Address\_1 , Address\_2 , City, State, ZipCode, PhoneNumber, Email, Fee)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Customer\_ID | Occupation | First\_Name | Last\_Name | Address\_1 | Address\_2 | City | State | ZipCode | Phone\_Number | Email | Fee |
| 1 | Professional Mourner | Huggy | Weiners | 9 Front Dr. |  | New York | NY | 10034 | (467) 871-4292 | weiners@gmail.com | 0 |
| 2 | Dog food taster | Crapps | Johnson | 74 Gartner St. |  | Jackson Heights | NY | 11372 | (565) 739-6820 | crapps@gmail.com | 0 |
| 3 | Professional sleeper | TeeTee | Lazy | 52 Green Hill Street |  | Brooklyn | NY | 11223 | (222) 331-0857 | lazy@motivation.com | 95 |

Key: Customer\_ID

FD1: Customer\_ID -> Occupation, First\_Name, Last\_Name, Address\_1, Address\_2, City, State, ZipCode, Phone\_Number, Email, Fee

FD2: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency: Customer\_ID -> ZipCode and ZipCode -> City, State

The Customer table is in 2NF.

Solution: Split Customer relation into two new relations named Customer and ZipCodes:

**Customer**(Customer\_ID(PK), Occupation, First\_Name, Last\_Name, Address\_1 , Address\_2 , ZipCode(fk), PhoneNumber, Email, Fee)

Key: Customer\_ID

FD1: CustomerID -> Occupation, First\_Name, Last\_Name, Address\_1 , Address\_2 , ZipCode, PhoneNumber, Email, Fee

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Zipcode¹**(ZipCode(PK), City, State)

Key: ZipCode

FD1: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Vendor Relation**

**Vendor**(Vendor(ID)(PK), Location, Installed\_By\_Self(Y/N))

|  |  |  |
| --- | --- | --- |
| Vendor\_ID (ID) | Installed\_By\_Self(Y/N) | Location |
| 1121 | N | Online |
| 1122 | N | Online |
| 1123 | Y | In-person |

Key: Vendor\_ID

FD1: Vendor\_ID (ID) -> Installed\_By\_Self(Y/N), Location

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Transaction Relation**

**Transaction**(Transaction\_ID(ID), Customer\_ID(FK), Computer\_ID(FK), Date\_Reserved, Date\_Due, Status, Transaction\_Cost, Vendor\_ID(FK))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Transaction\_ID | Computer\_ID | Customer\_ID | Date\_Reserved | Date\_Due | Status | Transaction\_Cost | Vendor\_ID |
| 1 | 11 | 1 | 8/18/2015 | 10/18/2015 | Late | $ 3.12 | 1121 |
| 2 | 16 | 2 | 1/18/2016 | 3/18/2016 | In-progress | $ 1.82 | 1122 |
| 3 | 5 | 3 | 3/1/2016 | 5/1/2016 | Late | $ 4.64 | 1123 |

Key: Transaction\_ID

FD1: Transaction\_ID -> Customer\_ID, Computer\_ID, Date\_Reserved, Date\_Due, Status, Transaction\_Cost, Vendor\_ID

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Computer Relation**

**Computer**(Computer\_ID(ID) ,Reserved(Y/N), Company\_Name(FK), OS\_System, Cost, Model, Damage\_Status,CPU, RAM)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Computer\_ID | Reserved(Y/N) | Company\_Name(FK) | OS\_System | Cost | Model | Damage\_Status | CPU | RAM |
| 1 | Y | Lenovo | Windows | $750.00 | Thinkpad | Operational | Intel | 8GB |
| 2 | N | Apple | MacOS | $800.00 | Air | Operational | Intel | 16GB |
| 3 | N | Lenovo | Windows | $750.00 | Thinkpad | Repairs Needed | Intel | 8GB |

Key: Computer\_ID

FD1: Computer\_ID -> Reserved(Y/N), Company\_Name, OS\_System, Cost, Model, Damage\_Status,CPU, RAM

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Technician Relation**

**Technician**(Technician\_ID(ID), First\_Name, Last\_Name, Address\_1, Address\_2, City. State, ZipCode, PhoneNumber, Email)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Technician\_ID | First\_Name | Last\_Name | Address\_1 | Address\_2 | City | State | ZipCode | PhoneNumber | Email |
| 1 | Yaseen | Beattie | 2154 West Virginia Avenue | 2C | Colonie | NY | 12205 | 518-702-1225 | BeattleY@library.edu |
| 2 | Karina | Solis | 4165 Shinn Street |  | New York | NY | 10018 | 212-092-0162 | SolisK@library.edu |
| 3 | Safah | Greene | 941 Turkey Pen Road | 14B | New York | NY | 10016 | 212-555-0122 | GreeneS@library.edu |

Key: Technician\_ID

FD1: Technician\_ID-> First\_Name, Last\_Name, Address\_1, Address\_2, City. State, ZipCode, PhoneNumber, Email

FD2: ZipCode -> City. State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency: Technician\_ID -> ZipCode and ZipCode -> City. State

The Technician table is in 2NF.

The solution to get 3NF: Split Technician relation into two new relations named Customer and ZipCodes.

**Technician**(Technician\_ID(ID), First\_Name, Last\_Name, Address\_1, Address\_2, ZipCode(FK), PhoneNumber, Email)

Key: Technician\_ID

FD1: Technician\_ID-> First\_Name, Last\_Name, Address\_1, Address\_2, ZipCode(FK), PhoneNumber, Email

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Zipcode¹**(ZipCode(PK), City, State)

Key: ZipCode

FD1: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**WorkOrder relation**

**WorkOrder**(WorkOrderID(ID), TechnicianID(FK), ComputerID(FK), StartDate, CompletionDate, Description)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| WorkOrder\_ID | Technician\_ID | Computer\_ID | Start\_Date | Completion\_Date | Description |
| 1 | 1 | 1 | 6/6/2016 | 6/20/2016 | Hardrive error, needed to reinstall OS |
| 2 | 5 | 4 | 6/8/2016 | 6/22/2016 | Software Update failed, reinstalled |
| 4 | 6 | 3 | 6/14/2016 | 6/16/2016 | Software Update failed, reinstalled |

Key: WorkOrderID

FD1: WorkOrderID -> TechnicianID, ComputerID, StartDate, CompletionDate, Description

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Manafacturer relation**

**Manufacturer**(Company\_ID(ID) Company\_Name, Address\_1, Address\_2, City, State, ZipCode, PhoneNumber)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Company\_ID | Company\_Name | Address\_1 | Address\_2 | City | State | ZipCode | Phone Number |
| 1 | Lenovo | 452A Queen's Rd W |  | Shek Tong Tsui | Hong Kong | 54986 | (855) 253-6686 |
| 2 | Apple | 896 Alexander Avenue |  | Concord | California | 94520 | (800) 692-7753 |

Key: Company\_ID

FD1: Company\_ID -> Company\_Name, Address\_1, Address\_2, City, State, ZipCode, Phone Number

FD2: ZipCode -> City. State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency: Company\_ID -> ZipCode and ZipCode -> City. State

The Manufacturer table is in 2NF.

The solution to get 3NF: Split Manufacturer relation into two new relations named Manufacturer and ZipCodes

**Manufacturer**(Computer\_ID(ID) FirstName, LastName, Address\_1. Address\_2, ZipCode(fk), PhoneNumber, Email)

Key: Computer\_ID

FD1: Computer\_ID(ID) -> FirstName, LastName, Address\_1. Address\_2, ZipCode(fk), PhoneNumber, Email

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Zipcode¹**(ZipCode(PK), City, State)

Key: ZipCode

FD1: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No Transitive dependencies

BCNF: All determinants are candidate keys

**Normalization Remark**¹

During the implementation, the ZipCode table was dropped off.

Sometimes, ZipCodes cross states borders. As an example, let’s consider a ZipCode1 that cross state borders between City1, State1 and City2, State2

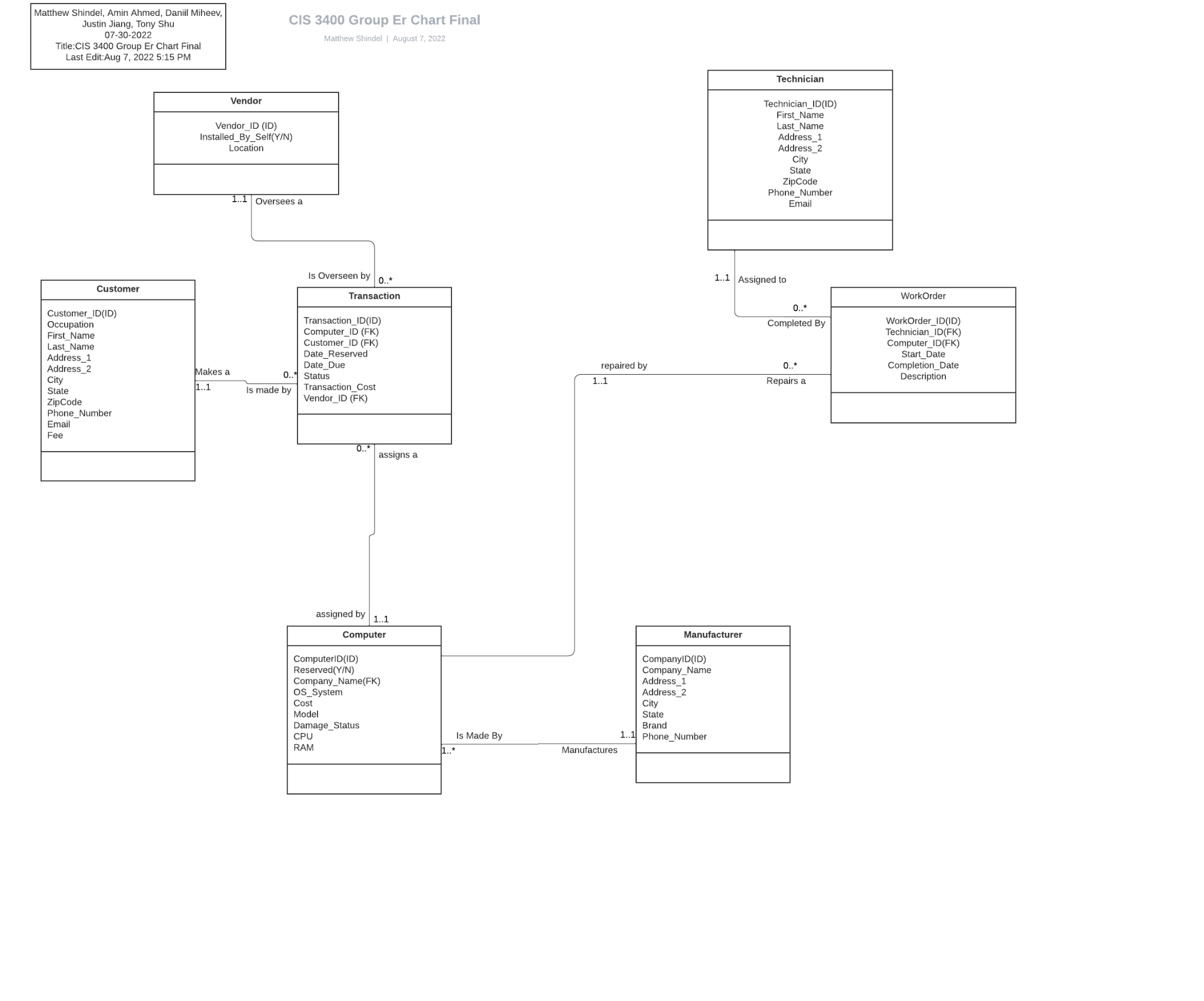
ZipCode1 -> City1, State1

ZipCode1 -> City2, State2

However, ZipCode is our PK and by definition it’s unique

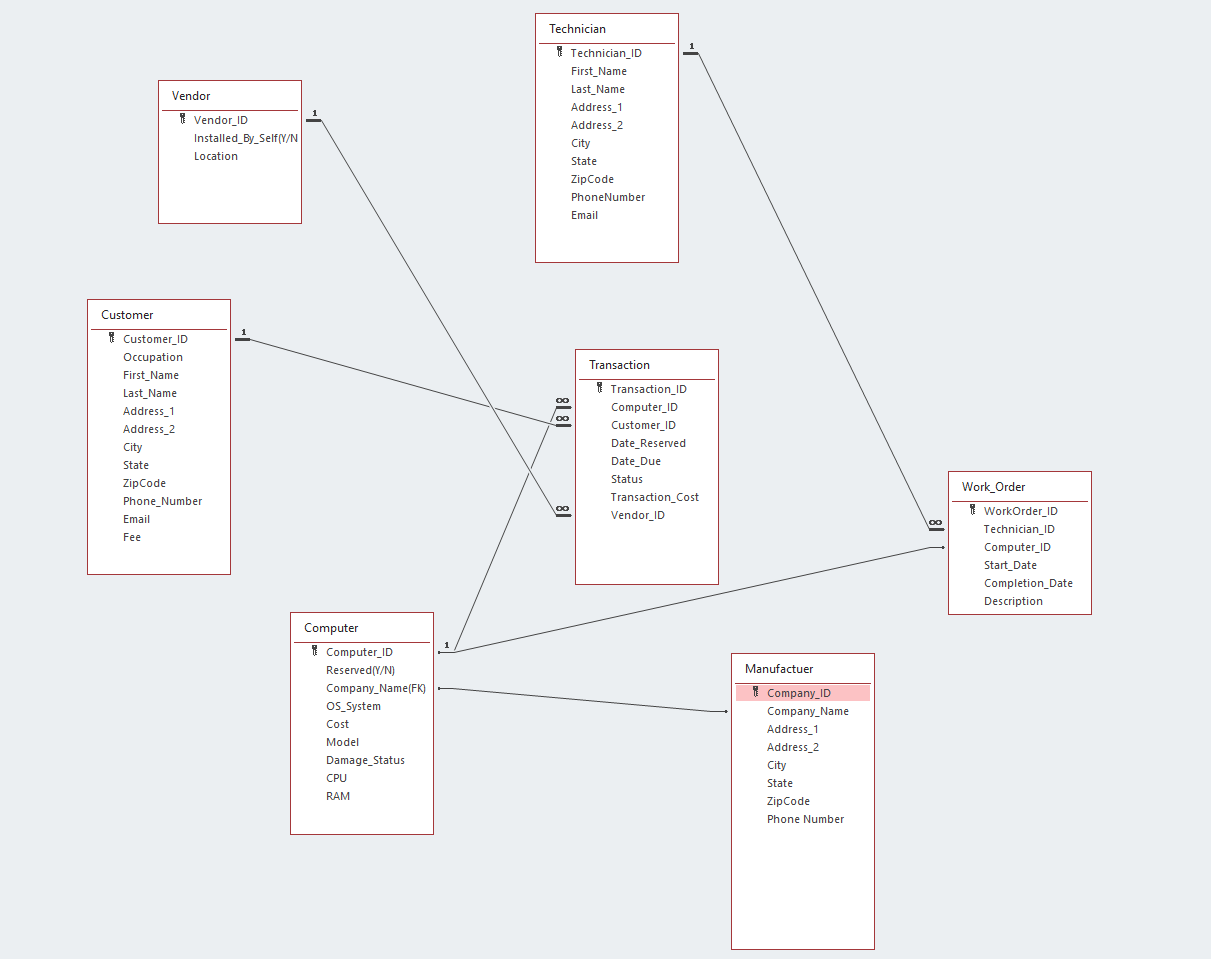
Therefore, the **Zipcode**(ZipCode(PK), City, State) is not a valid table and we will keep ZipCode, City, State columns in the tables.

Using LucidChart, our final ER diagram formed this design:

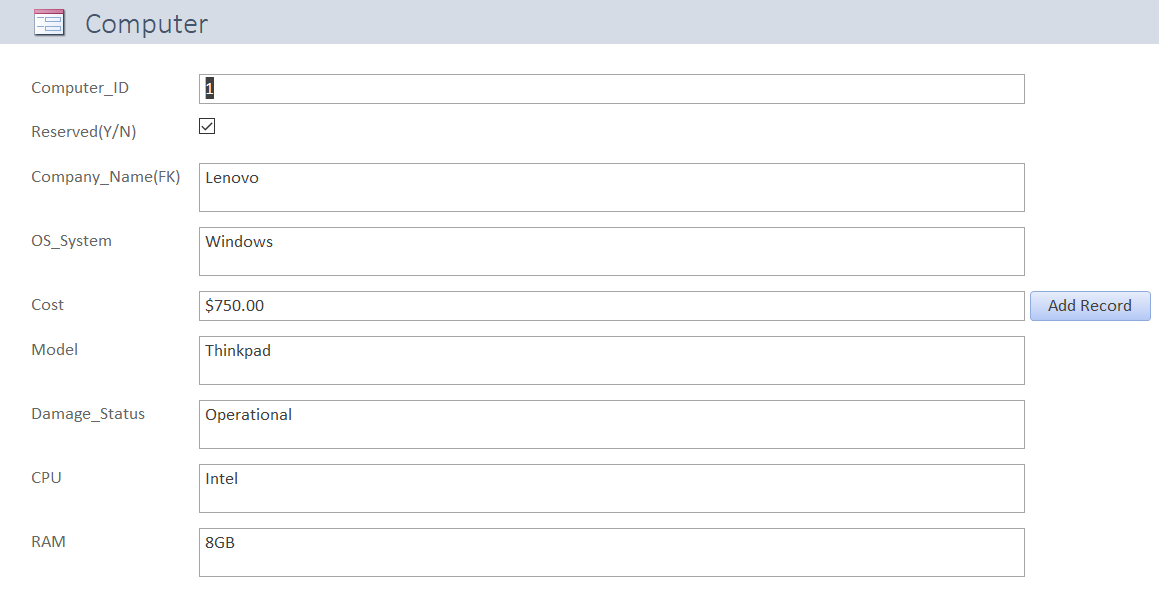


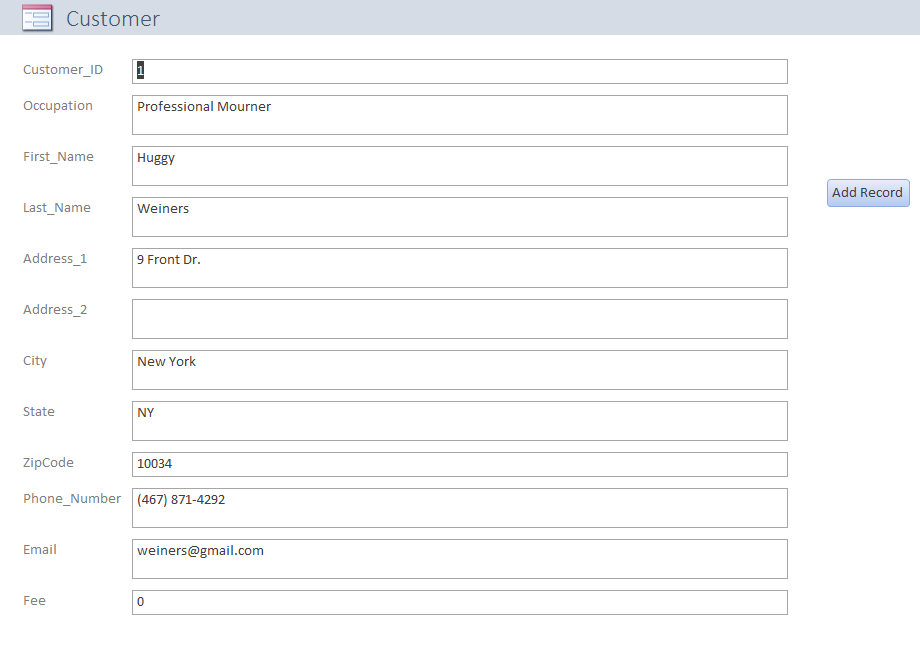
*Physical Database implementation:*

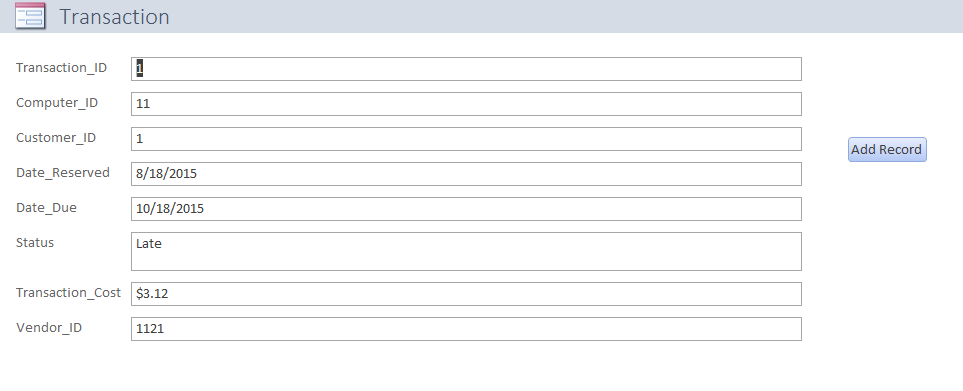
Using Access, we imported the data that we created within Excel, and set up the various relationships between each of the tables following the relationship sentences. Below is an image of the current relationships within the database.

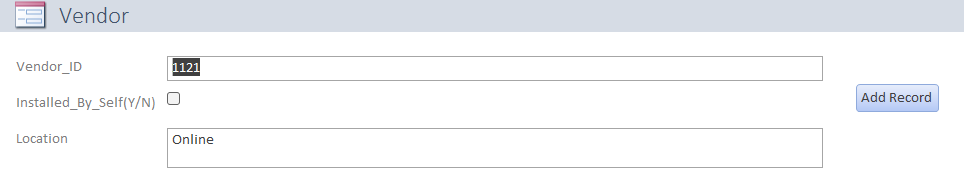


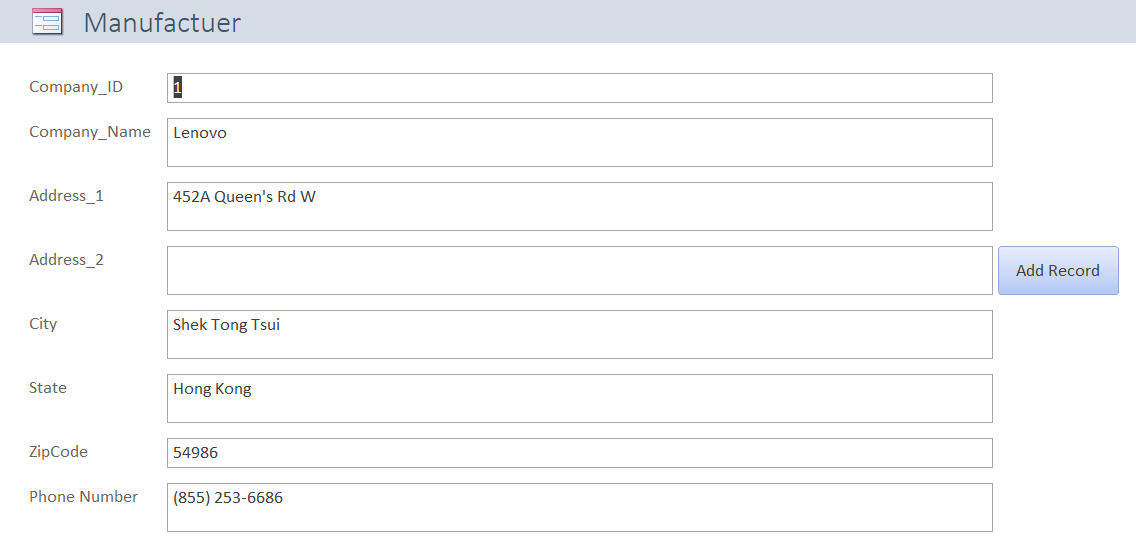
Below are screenshots of the various forms with our database. These forms help us add records/edit records for Computers, Customers, Manufacturers, Transactions, Vendors, and Work Orders:

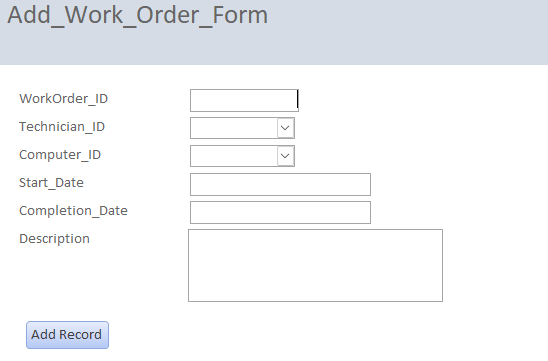












*Conclusion:*

Our group project, *Rent & Go*, illustrates all of the steps involved in planning, creating, and implementing a database within Access. Some of the steps of creating the database, such as importing the data from excel, and creating the forms for the various tables were not covered within this report. Additionally, as our data was manually generated, actual data from a real world environment may differ from what information we have, but the ending implementation of the database would remain the same. The main difference between our data and a real world dataset are the actual scope of data contained within the database.