TAX FIRM PROJECT

Fourier Mirage

CIS 9340 UWA

Chloe Huang chloe.huang@baruchmail.cuny.edu

Abel Roman Abel.Roman@baruchmail.cuny.edu

Ming Mao Ming.Mao@baruchmail.cuny.edu

Kenny Li kenny.li4@baruchmail.cuny.edu

BUSINESS NARRATIVE

Our Company is a family-run tax firm that has been in business for over five years. Located in midtown Manhattan, our company has grown exponentially over the last couple of years with newer clients from around New York. We need to streamline our day-to-day business operations. We have been using spreadsheets and paper logbooks where we keep track of our customers, appointments, payments, vendors, and supply inventory. We would like to replace this manual method of tracking we currently have in place with a database that will help us become more efficient as we continue to grow as a business.

In our business, we have customers that need a wide range of services. Our services include the following:

1) Filing clients individual tax return
2) Filing client corporation tax
3) Bookkeeping and payroll tax
4) Business Registration
5) All other taxes/problems.

Each service has a unique standard price and specific materials that are used for those services. We need to keep track of those materials used (Tax form, Office supplies, software) and the time it takes to complete each service. Once we are running low on a certain supply, or we need a special order for a client we have a list of vendors we contact who will deliver it to us.

We also keep track of our employee and customer information. For our employees, we keep track of their SSN, address, rate of, and position. We also keep track of each customer's address, telephone number as well as their first and last name.

BUSINESS SCENARIO

Our client operates a small to midsize tax firm for 5 years. They have been using excel spreadsheets and paper logs to track their invoices from customers, vendors, and employees. They came to us to create a database system to replace their manual record-keeping methods.

In their business, they have clients coming to them for a range of services. These services include but are not limited to; filing for their individual tax return, corporation taxes, bookkeeping and payroll, company formation, and other taxes that businesses are responsible for. Our client needs to keep track of the supplies they order. For instance, office supplies and specific forms.

In addition, they also need to keep track of their employee information; name, SSN, address, and rate of pay.

We propose creating an access database to streamline their day to day business activity. We will create forms that will help our client easily input and delete information for new clients, appointments, vendors, and services. We will create a master customers form which will make it easier for our client to keep track of appointments and services. Finally, we will provide our customer the ability to create a customer and supply order report for a quick view.

ENTITIES AND ATTRIBUTES

Employee

ServicePerformed

Appointments

Customers

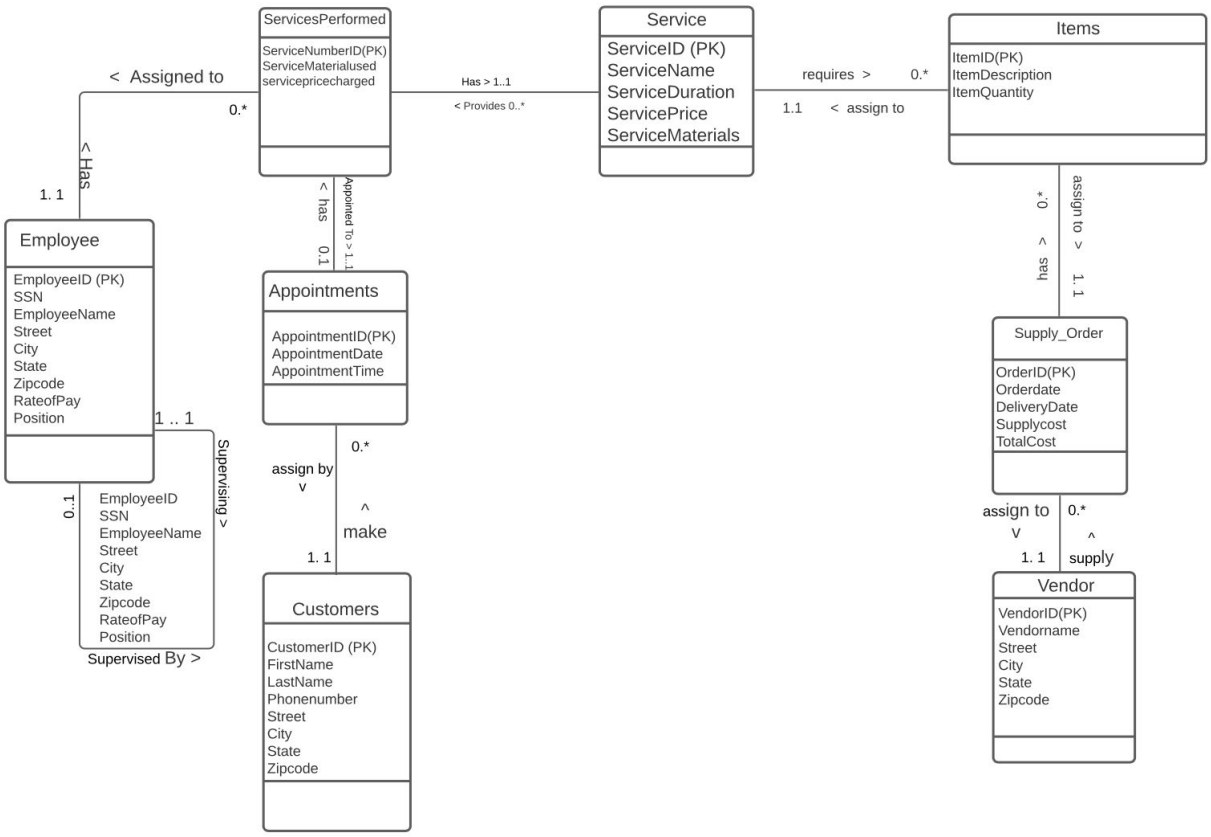
Service

Items

Supply_Order

Vendor

SYSTEMS ANALYSIS



LOGICAL AND PHYSICAL MODELING

Employee to supervising

Zero or One Supervisor assigned to one or more employees

Employees supervised by one and only one (1) Supervisor

Employee to Services Performed

One Employee can take on zero or many services

One Service can be assigned to one and only (1) employee

Services Performed to Appointments

One Appointment can be assigned to one and only (1) service

One Service can take on zero or many appointments

Customers to Appointments

One customer may make (0) one or more (*) appointments

One appointment must be (1) assigned by one and only one (1) customer

Service to Service Performed

One service must be assigned to one and only (1) service

One service performed may take on one service

Vendor to Supply_Order

One vendor may be (0) supply one or more (*) supply orders.

One supply order must (1) assign to one and only one (1) vendor.

Supply_Order to Items

One supply order may be (0) having one or more (*) items.

One item must (1) assign to one and only one (1) supply order.

Service to Items

One Service may (0) require one or more (*) items.

One item must (1) assign to one and only one (1) service.

Employee (EmployeeID (key) , SSN, Employeename, Street, City, State, ZipCode, Rateofpay, Position) key: EmployeeID

FD1: EmployeeID -> SSN, EmployeeName, Street, City, State, ZipCode, Rateofpay, Position

FD2: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: Transitive dependency exists: EmployeeID -> ZipCode, ZipCode -> City, State Solution: Split the relation into two new relations named EmployeeInfo and ZipCode

EmployeeInfo(EmployeeID(key), SSN, EmployeeName, Street, ZipCode(fk), RateOfPay, Position) key: EmployeeID

FD1: EmployeeID -> SSN, EmployeeName, Street, ZipCode(fk), Rateofpay, Position

1NF: Meets the definition of a relation

2NF: No partial Key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

ZipCode(ZipCode(key), City, State)

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

Appointments (AppointmentID (key) , AppointmentDate, AppointmentTime, CustomerID(fk))

key: AppointmentID

FD1: AppointmentID -> AppointmentDate, AppointmentTime, CustomerID(fk) 1NF: Meets the definition of a relation

2NF: No partial key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

Customers (CustomerID (key) , FirstName, LastName, PhoneNumber, Street, City, State, ZipCode)

key: CustomerID

FD1: CustomerID -> FirstName, LastName, PhoneNumber, Street, City, State, ZipCode FD2: ZipCode -> City, State

1NF: Meets the definition of a relation

2NF: No partial key dependencies

3NF: Transitive dependency exists: CustomerID -> ZipCode, ZipCode -> City, State Solution: Split Customer relation into two new relations named CustomerInfo and ZipCode

CustomerInfo(CustomerID(key), FirstName, LastName, PhoneNumber, Street, ZipCode(fk)) Key: CustomerID

FD1:CustomerID -> FirstName, LastName, PhoneNumber, Street, ZipCode(fk)

2NF: No partial key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

Note: we have already set up the ZipCode table, so we will use that relation for customer.

ServicePerformed (ServiceNumberID(key), ServiceMaterialUsed, ServicePriceCharged, AppointmentID(fk),EmployeeID(fk),ServiceID(fk)) key: ServiceNumberID

FD1: ServiceNumberID -> ServiceMaterialUsed, ServicePriceCharged, AppointmentID (fk), EmployeeID (fk)

1NF: Meets the definition of a relation

2NF: No partial key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

Service(ServiceID(Key), ServiceName, ServiceDuration, ServicePrice, ServiceMaterials, ItemID(fk))

key: ServiceID

FD1: ServiceID -> ServiceName, ServiceDuration, ServicePrice, ServiceMaterials, 1NF: Meets the definition of a relation

2NF: No partial key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

Items(ItemsID(key), ItemDescription, ItemQuantitiy, OrderID(fk),))

key: ItemsID

FD1: ItemsID -> ItemDescription, ItemQuantity, OrderID(fk), ServiceID(fk) 1NF: Meets the definition of a relation

2NF: No partial key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

Supply_Order(OrderID(key), OrderDate, DeliverDate, SupplyCost, TotalCost, VendorID(fk))

key: OrderID

FD1: OrderID -> OrderDate, DeliverDate, SupplyCost, TotalCost, VendorID(fk) 1NF: Meets the definition of a relation

2NF: No partial key dependencies

3NF: No transitive dependencies

BCNF: All determinants are candidate keys

Vendor(VendorID(Key), Vendorname, Street, City, State, Zipcode)

Key: VendorID

FD1: VendorID -> VendorName, Street, City, Zipcode

FD2: Zipcode -> city, state

1NF: Meets the definition of a relation

2NF: No partial key dependencies

3NF: Transitive dependency exists: VendorID -> zipcode and zipcode -> city, state

Solution: Split vendor into two new relations named Vendor_data and Zipcodes

Vendor_data (Vendorname, street, zipcode(fk)) Key: VendorID

FD1: VendorID -> Vendorname, street, Zipcode(fk) 1NF: Meets the definition of a relation

2NF No partial keys dependencies 3NF: No Transitive dependencies

Note: we have already set up the ZipCode table, so we will use that relation for vendor.

DATABASE IMPLEMENTATION

```
CREATE TABLE Customer
(
        CustomerID VARCHAR(10) NOT NULL,
        FirstName VARCHAR(20),
        LastName VARCHAR(20),
```

```
	PhoneNumber VARCHAR(15),
	Street VARCHAR(30),
	ZipCode VARCHAR(10),
	CONSTRAINT pk_customer
		PRIMARY KEY (CustomerID)
)

CREATE TABLE ZipCode
(
	ZipCode VARCHAR(10) NOT NULL,
	City VARCHAR(10),
	State VARCHAR(2),
	CONSTRAINT pk_zipcode
		PRIMARY KEY (ZipCode)
)

CREATE TABLE Employee
(
	EmployeeID VARCHAR(10) NOT NULL,
	SSN VARCHAR(15),
	EmployeeName VARCHAR(20),
	Street VARCHAR(30),
	ZipCode VARCHAR(10),
```

```
	RateOfPay NUMBER,
	Position VARCHAR(20),
	CONSTRAINT pk_employee
		PRIMARY KEY (EmployeeID)
)

CREATE TABLE Appointment
(
	AppointmentID VARCHAR(10) NOT NULL,
	AppointmentDate Date,
	AppointmentTime VARCHAR(10),
	CustomerID VARCHAR(10) NOT NULL,
	CONSTRAINT pk_appointment
		PRIMARY KEY (AppointmentID)
)

CREATE TABLE ServicePerformed
(
	ServicePerformedID VARCHAR(10) NOT NULL,
	AppointmentID VARCHAR(10) NOT NULL,
	ServiceID VARCHAR(10) NOT NULL,
	EmployeeID VARCHAR(10) NOT NULL,
	ServiceMaterialUsed VARCHAR(30),
```

```
	ServicePriceCharged NUMBER,
	CONSTRAINT pk_servicePerformed
		PRIMARY KEY (ServicePerformedID, AppointmentID)
)

CREATE TABLE Service
(
	ServiceID VARCHAR(10) NOT NULL,
	ItemID VARCHAR(10) NOT NULL,
	ServiceName VARCHAR(30),
	ServiceDuration INTEGER,
	ServicePrice NUMBER,
	ServiceMaterials VARCHAR(30),
	CONSTRAINT pk_service
		PRIMARY KEY (ServiceID)
)

CREATE TABLE Item
(
	ItemID VARCHAR(10) NOT NULL,
	ServiceID VARCHAR(10) NOT NULL,
	OrderID VARCHAR(10) NOT NULl,
	ItemDescription VARCHAR(30),
```

```
	ItemQuatity NUMBER,
	CONSTRAINT pk_item
		PRIMARY KEY (ItemID)
)

CREATE TABLE Supply_Order
(
	OrderID VARCHAR(10) NOT NULL,
	VendorID VARCHAR(10) NOT NULL,
	OrderDate DATE,
	DeliverDate DATE,
	SupplyCost NUMBER,
	TotalCost NUMBER,
	CONSTRAINT pk_order
		PRIMARY KEY (OrderID)
)

CREATE TABLE Vendor
(
	VendorID VARCHAR(10) NOT NULL,
	VendorName VARCHAR(40),
	Street VARCHAR(30),
	ZipCode VARCHAR(10),
```

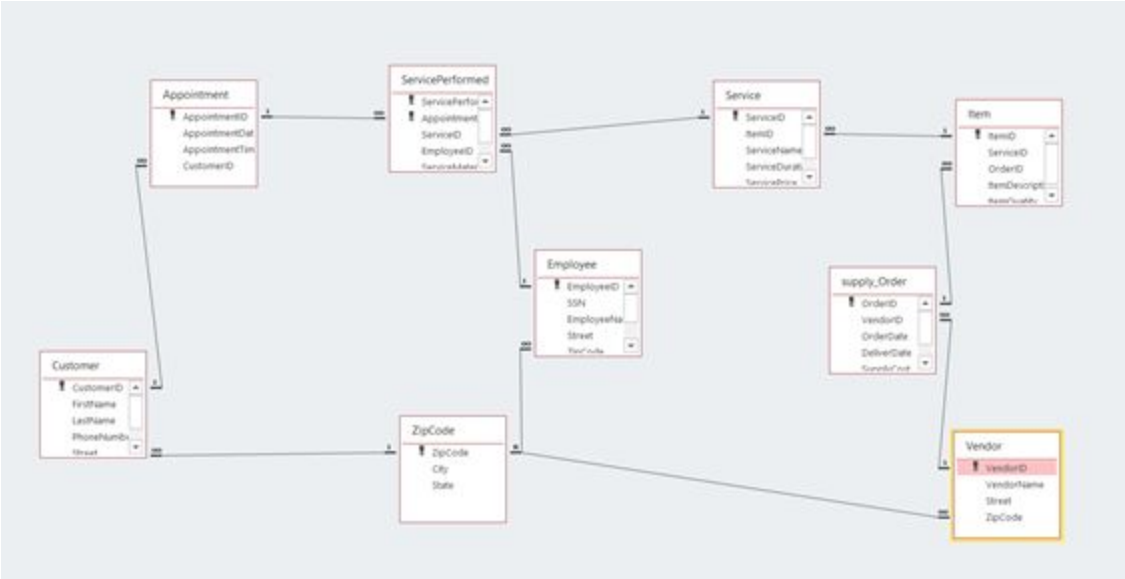
```
    CONSTRAINT pk_vendor
        PRIMARY KEY (VendorID)
)
```

Adding foreign key:

```
ALTER TABLE Customer
    ADD CONSTRAINT fk_customer_zipcodes
        FOREIGN KEY (ZipCode)
            REFERENCES ZipCode (ZipCode)

ALTER TABLE Employee
    ADD CONSTRAINT fk_emplooyee_zipcodes
        FOREIGN KEY (ZipCode)
            REFERENCES ZipCode (ZipCode)

ALTER TABLE Vendor
    ADD CONSTRAINT fk_vendor_zipcodes
        FOREIGN KEY (ZipCode)
            REFERENCES ZipCode (ZipCode)

ALTER TABLE Appointment
    ADD CONSTRAINT fk_customer_appointment
        FOREIGN KEY (CustomerID)
```

```
                REFERENCES Customer (CustomerID)

ALTER TABLE ServicePerformed
    ADD CONSTRAINT fk_ServicePerfomed_employee
        FOREIGN KEY (EmployeeID)
            REFERENCES Employee (EmployeeID)

ALTER TABLE ServicePerformed
    ADD CONSTRAINT fk_ServicePerfomed_appointment
        FOREIGN KEY (AppointmentID)
            REFERENCES Appointment (AppointmentID)

ALTER TABLE ServicePerformed
    ADD CONSTRAINT fk_ServicePerfomed_ServiceID
        FOREIGN KEY (ServiceID)
            REFERENCES Service (ServiceID)

ALTER TABLE Service
    ADD CONSTRAINT fk_Service_ItemID
        FOREIGN KEY (ItemID)
            REFERENCES Item (ItemID)
```

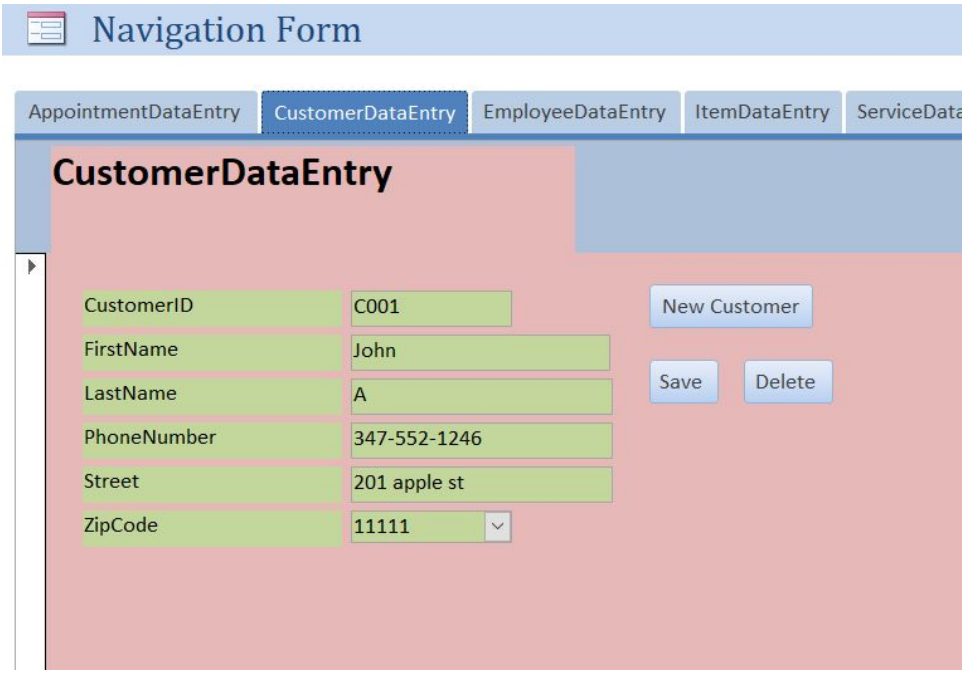
```
ALTER TABLE Item
    ADD CONSTRAINT fk_supply_order_item
        FOREIGN KEY (OrderID)
            REFERENCES supply_Order (OrderID)

ALTER TABLE Supply_Order
    ADD CONSTRAINT fk_supply_vendor
        FOREIGN KEY (VendorID)
            REFERENCES Vendor (VendorID)
```



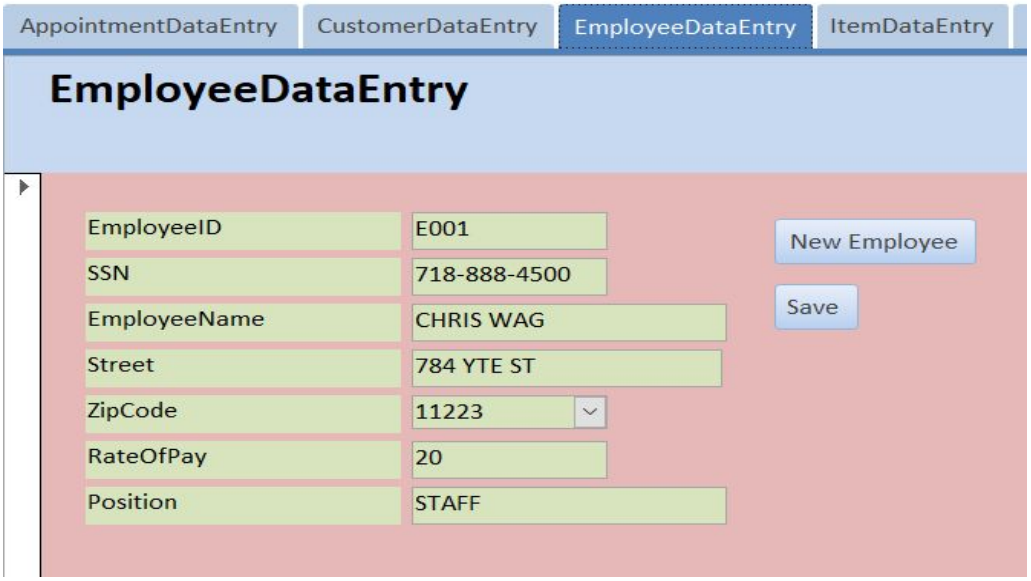
Relationship View

Navigation Form is the first form that appears when the database is opened. Different data entry forms and reports can be displayed by clicking on the selection above.

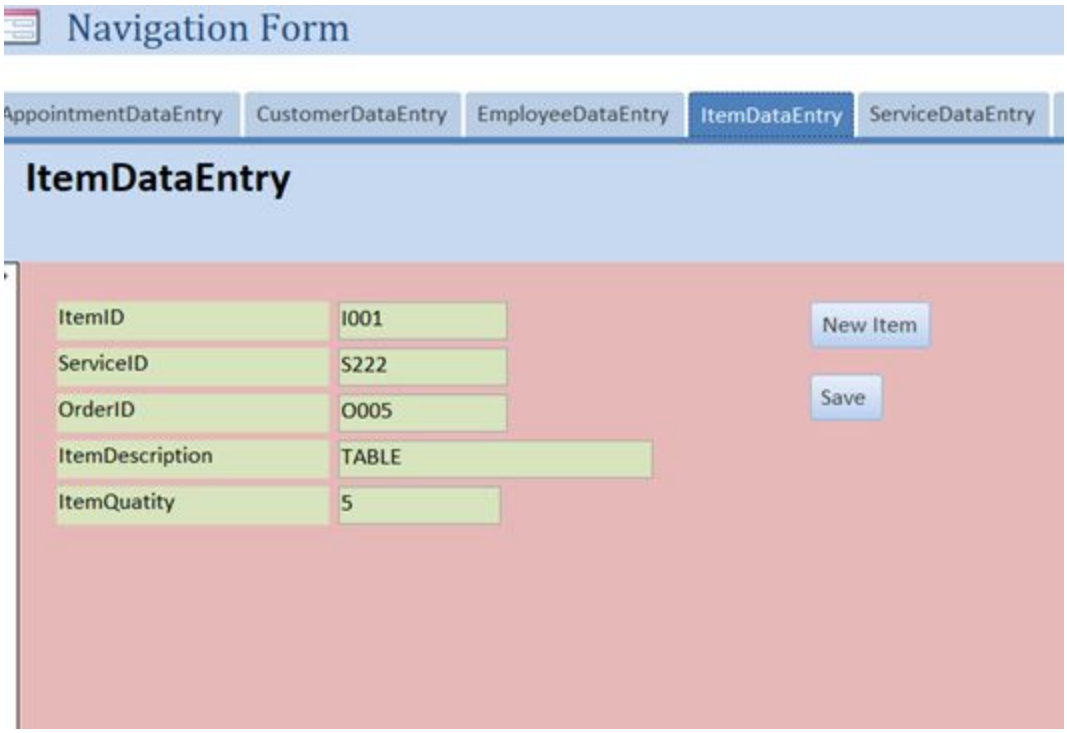
By selecting the New Appointment button, it creates a blank record ready to enter a new appointment. The ApointmentDataEntry is used to look up existing appointments and to input new appointment information. Clicking save, Saves the new or modified record. While the delete button gets rid of the current record



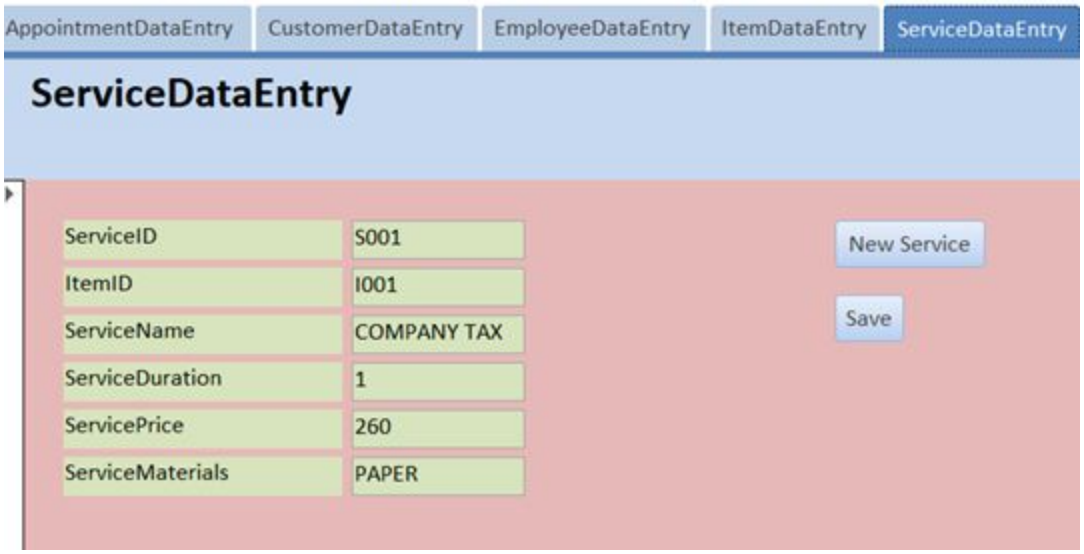
The Customer Data Entry form is used to look up existing customers and to input new customer information. By selecting a new customer button, it creates a blank record ready to enter a new customer. Zipcode field contains a drop down of all existing zipcodes from the zipcode table. Clicking save, Saves the new or modified record.While the delete button gets rid of the current record.



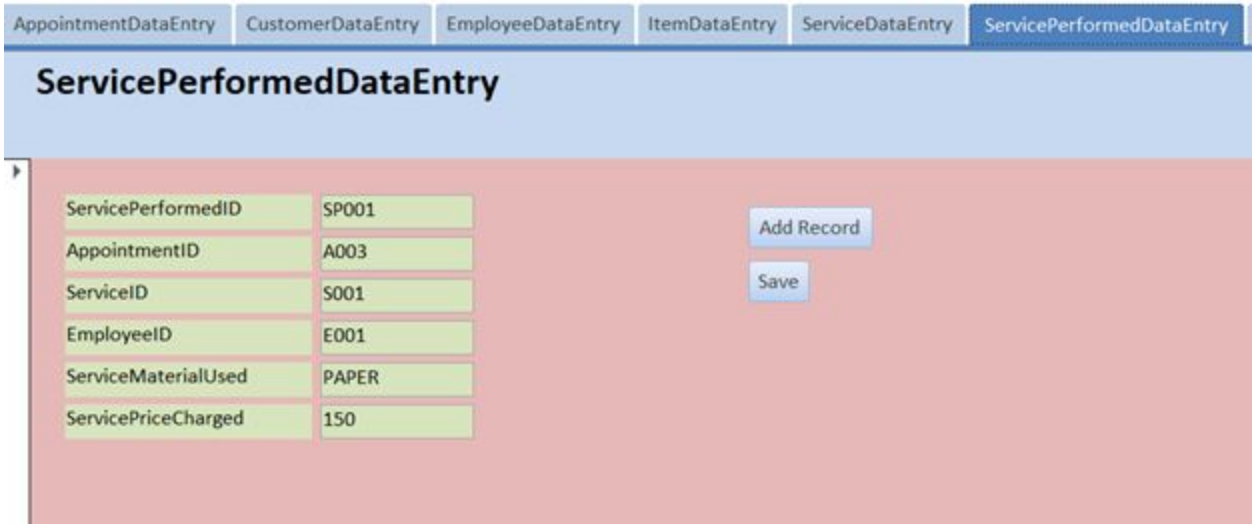
The EmployeeDataEntry form is used to look up employees and to input new employee information. By selecting a new employee button, it creates a blank record ready to enter a new employee. Zipcode field contains a drop down of all existing zipcodes from the zipcode table. Clicking save, Saves the new or modified record.



The ItemData Entry form is used to look up existing Items used and to input new items. By selecting a new item button, it creates a blank record ready to enter a new item.Clicking save, Saves the new or modified record.

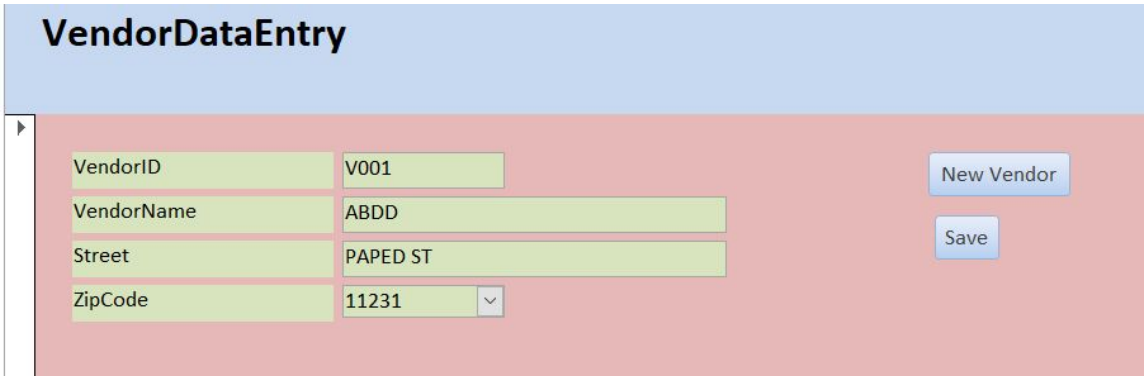


The ServiceData Entry form is used to look up existing Services provided and to input new services. By selecting a new service button, it creates a blank record ready to enter a new service. Clicking save, Saves the new or modified record.



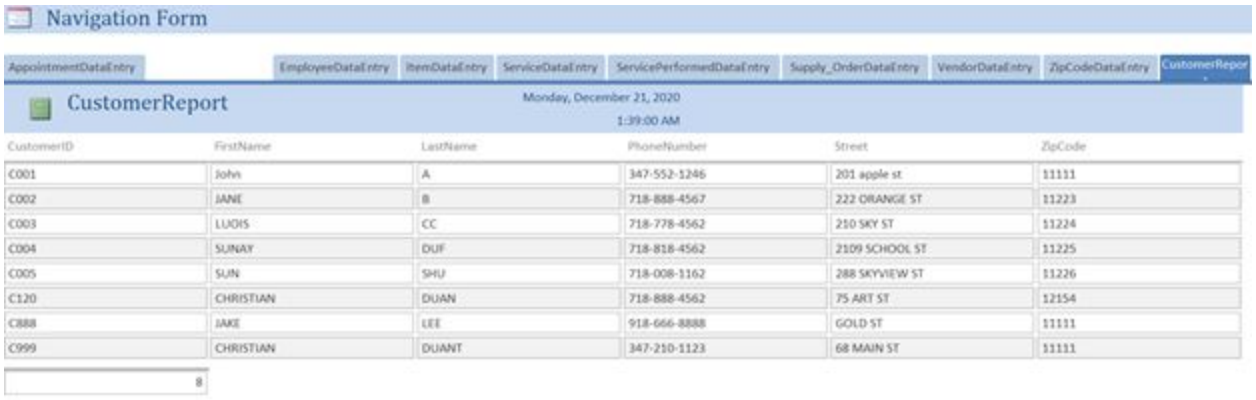
The ServicePerformedDataEntry form is used to look up existing Services provided and to input new services. By selecting a new service button, it creates a blank record ready to enter a new service. Clicking save, Saves the new or modified record.

The Service_OrderDataEntry form is used to look up existing supply orders and to input new supply orders. By selecting a new order button, it creates a blank record ready to enter a new supply order. Clicking save, Saves the new or modified record.

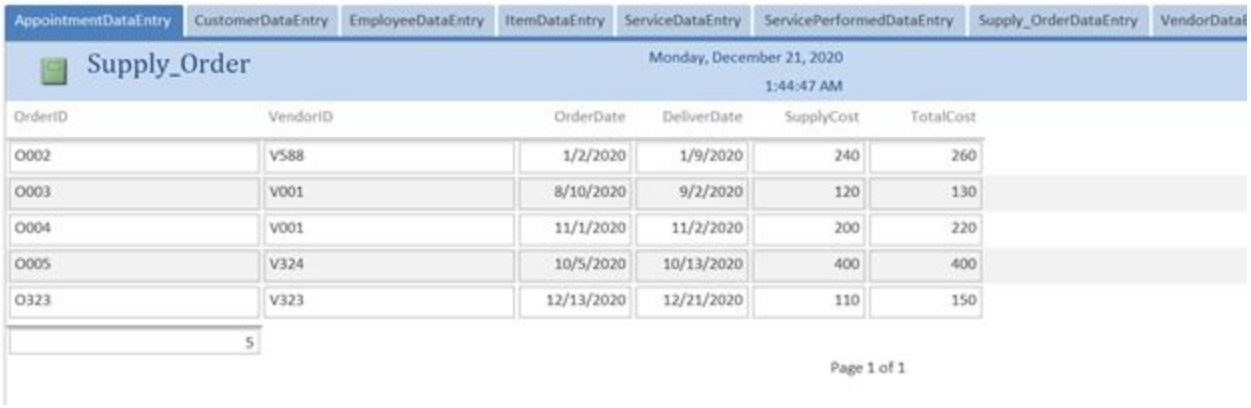


The VendorDataEntry form is used to look up existing Vendors and to input new Vendors. By selecting a new Vendor button, it creates a blank record ready to enter new vendor information. Clicking save, Saves the new or modified record.

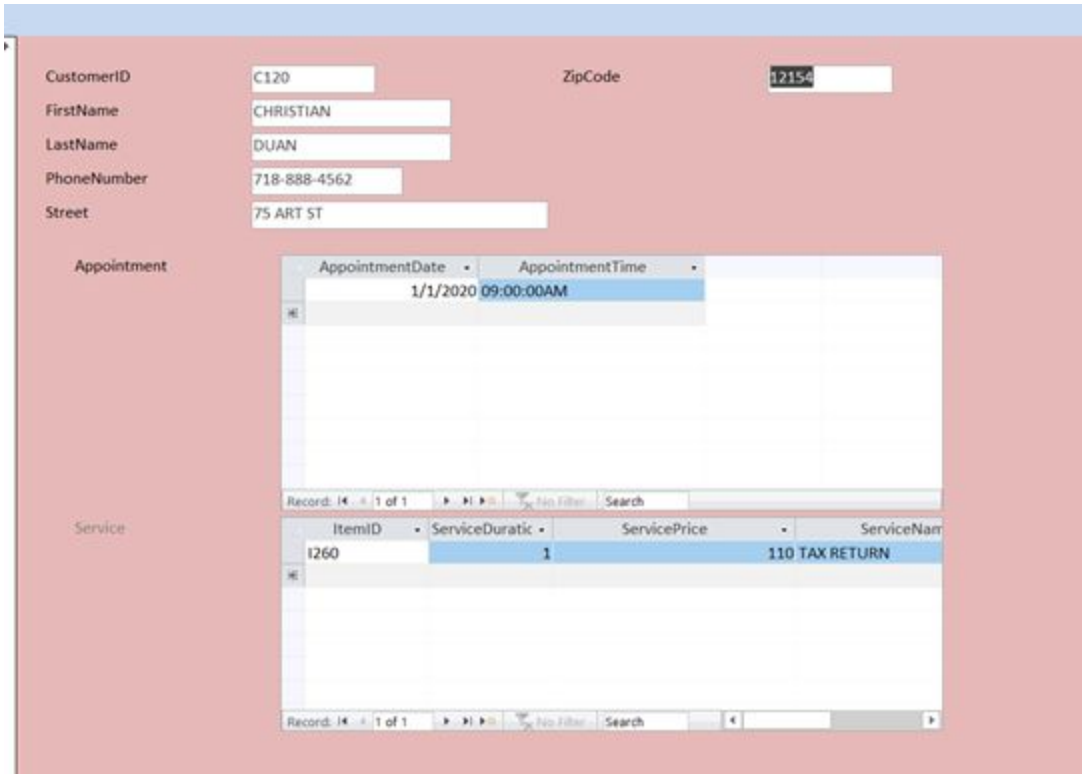
ZipcodeDataEntry form is used to look up existing zip codes which are connected to the other tables that require the zipcode field. By clicking the New Zipcode button you can enter a new zipcode.



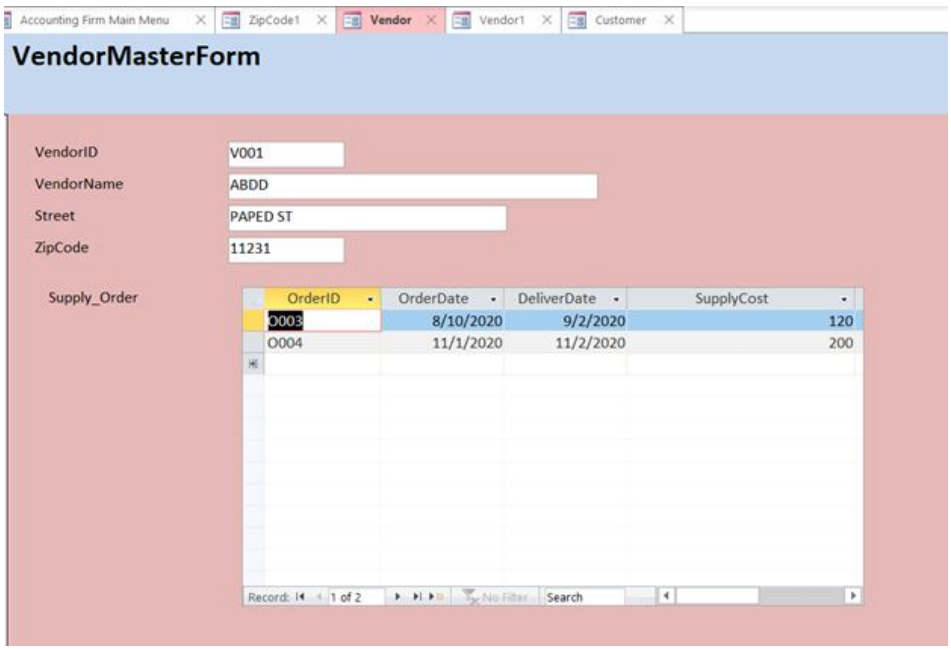
Customer Report shows the complete list of customers that is saved on the database.



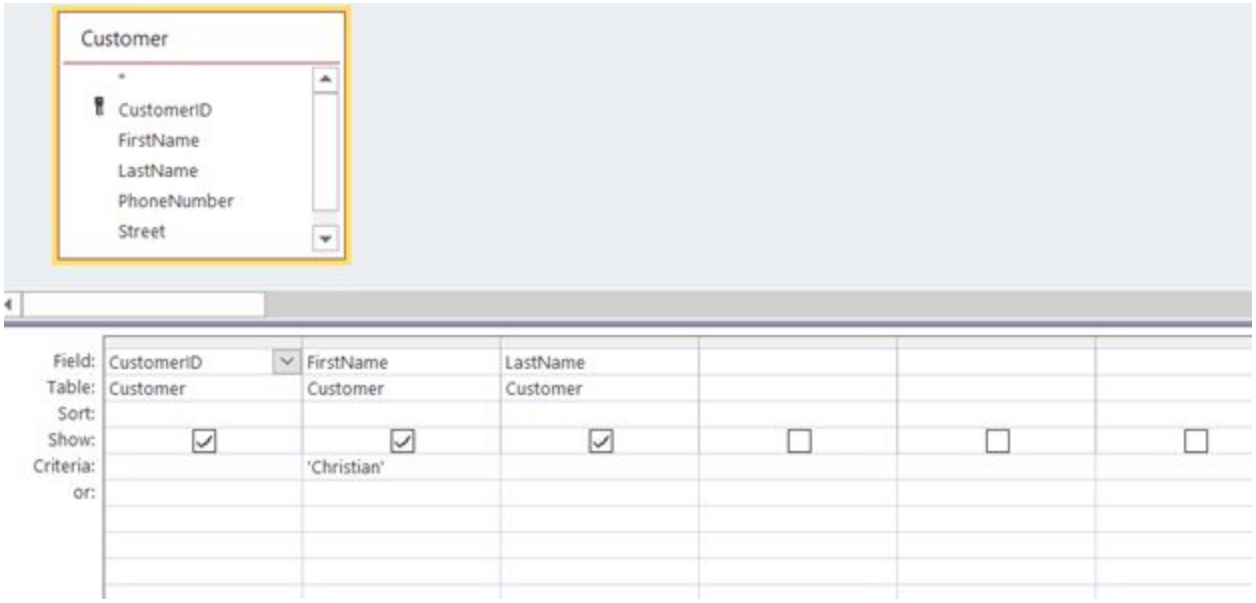
Supply order report shows all the orders that are outstanding with the order date and delivery date. As well with Vendor ID.



CustomerMaster Form Gives more information for each client such as any appointments coming up, the type of service needed.

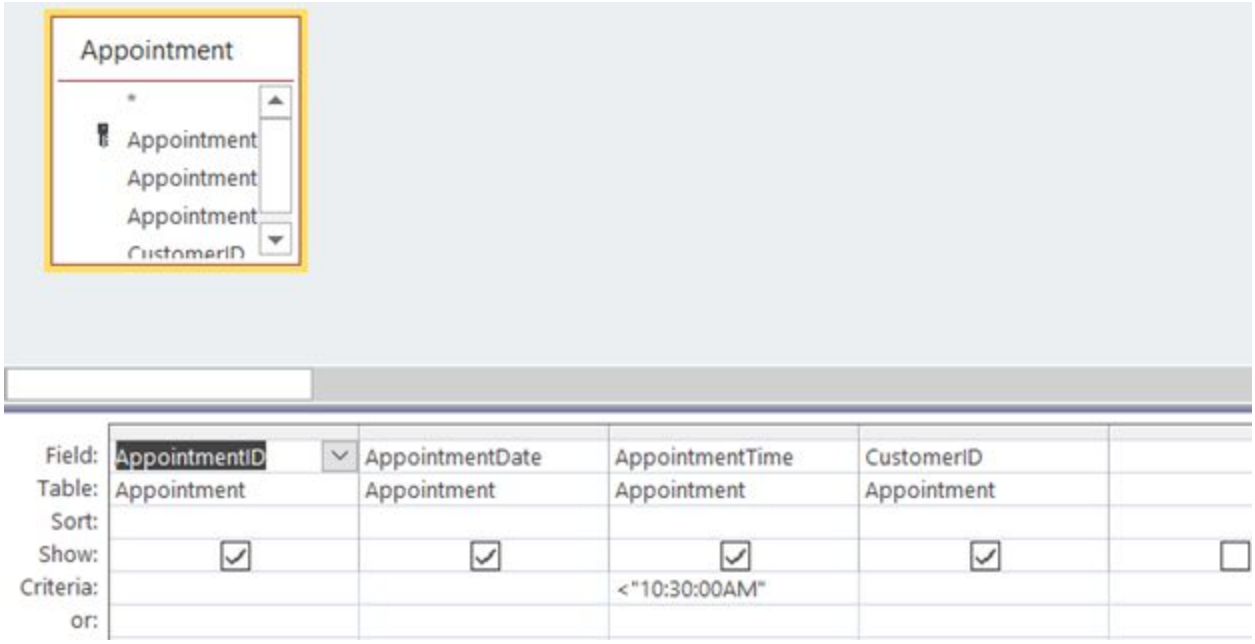


VendorMaster form Provides more information in regards to the vendor by showing all open order items.



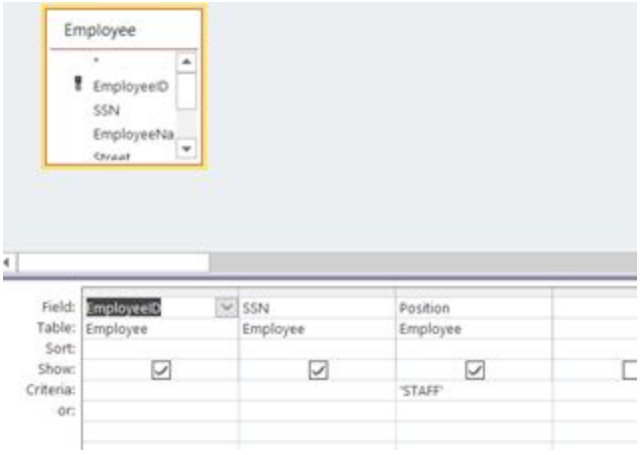
Customer Query

Be used as a search criterion for a specific FirstName from the customer table or LastName.



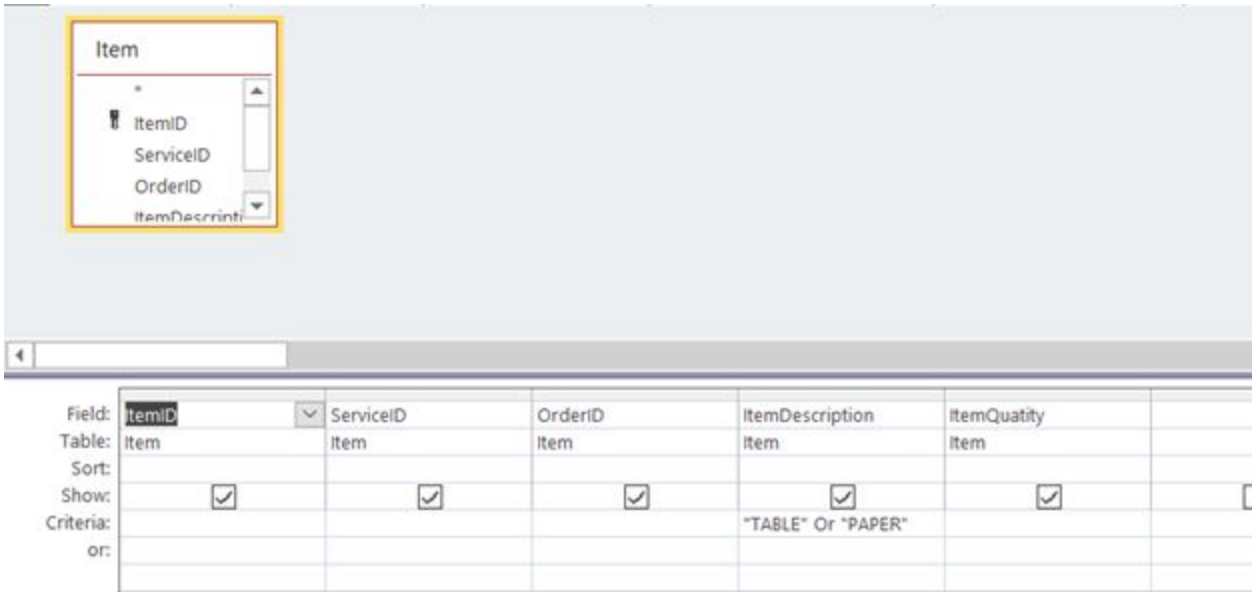
Appointment Query

Be used as a search criterion for a specific appointment by its time, date or appointmentID.



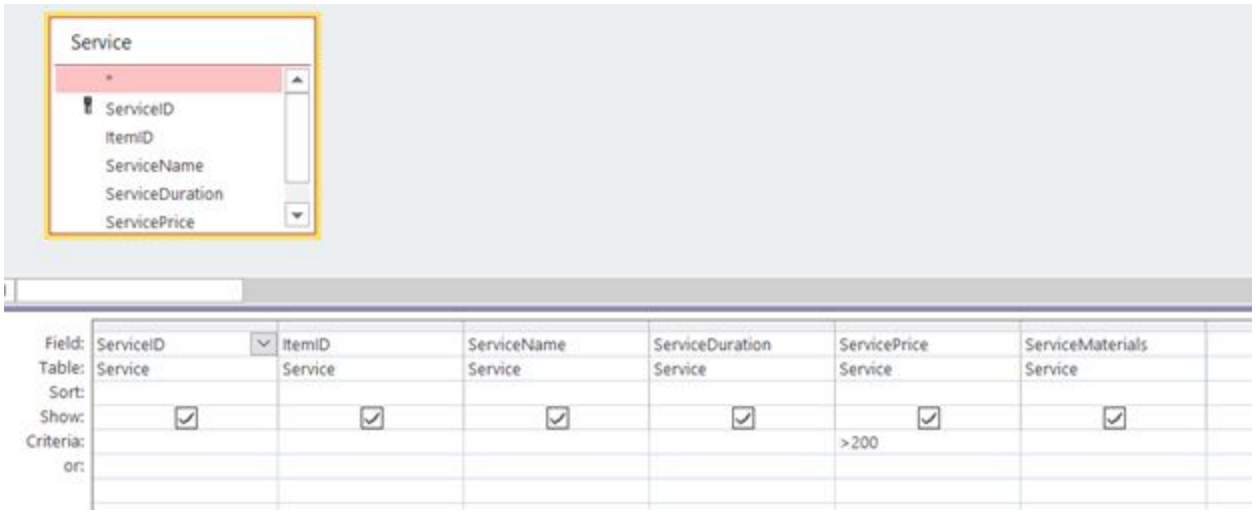
Employee query

Be used as a search criterion for a specific employee by its position, employee is or SSN.



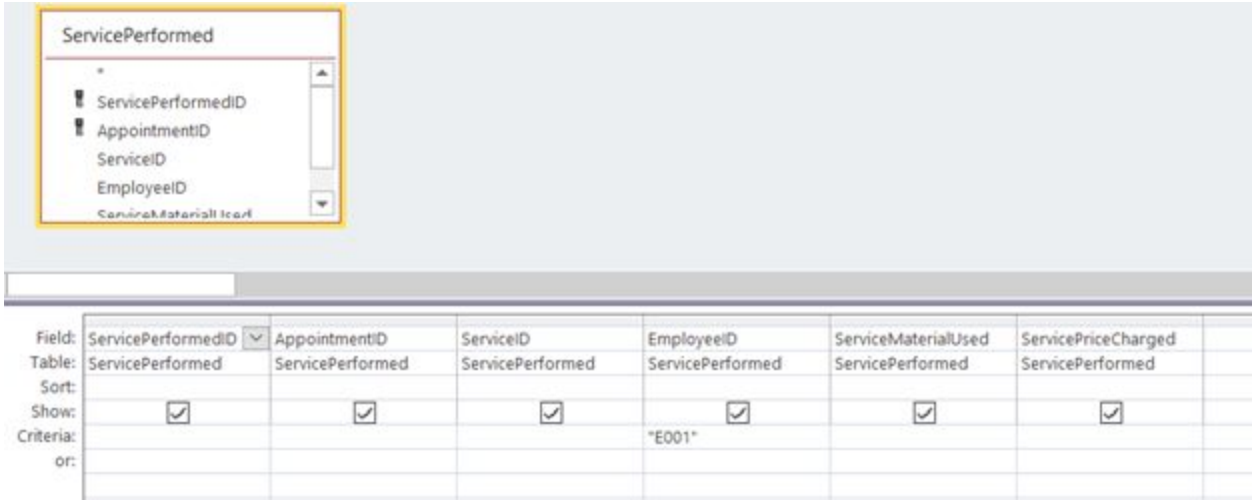
Item Query

Be used as a search criterion for a specific item by its itemDescription, orderID or serviceID.



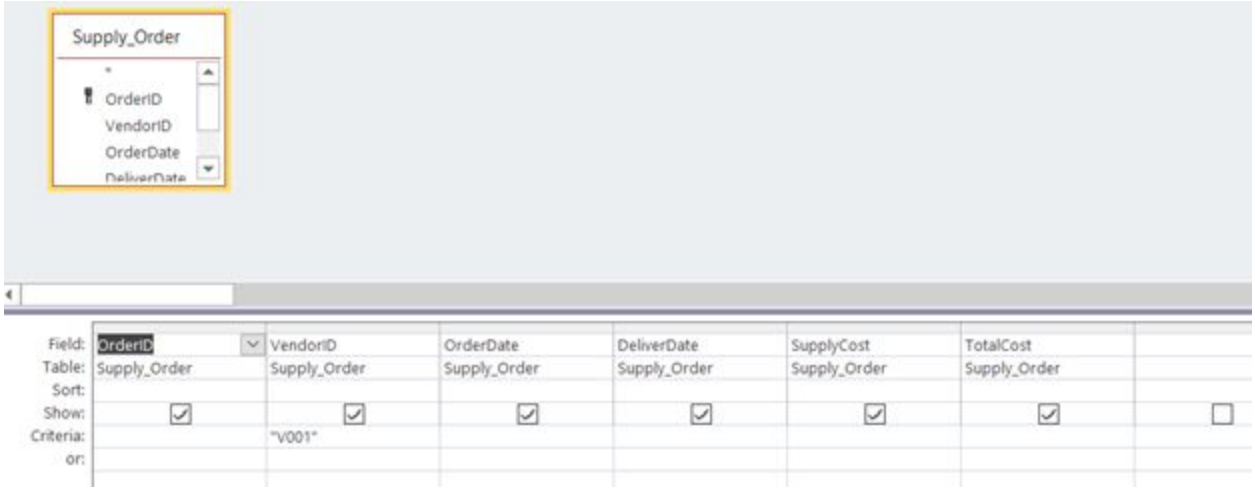
Service Query

Be used as a search criterion for a specific service by its ServicePrice, ItemID or serviceID.



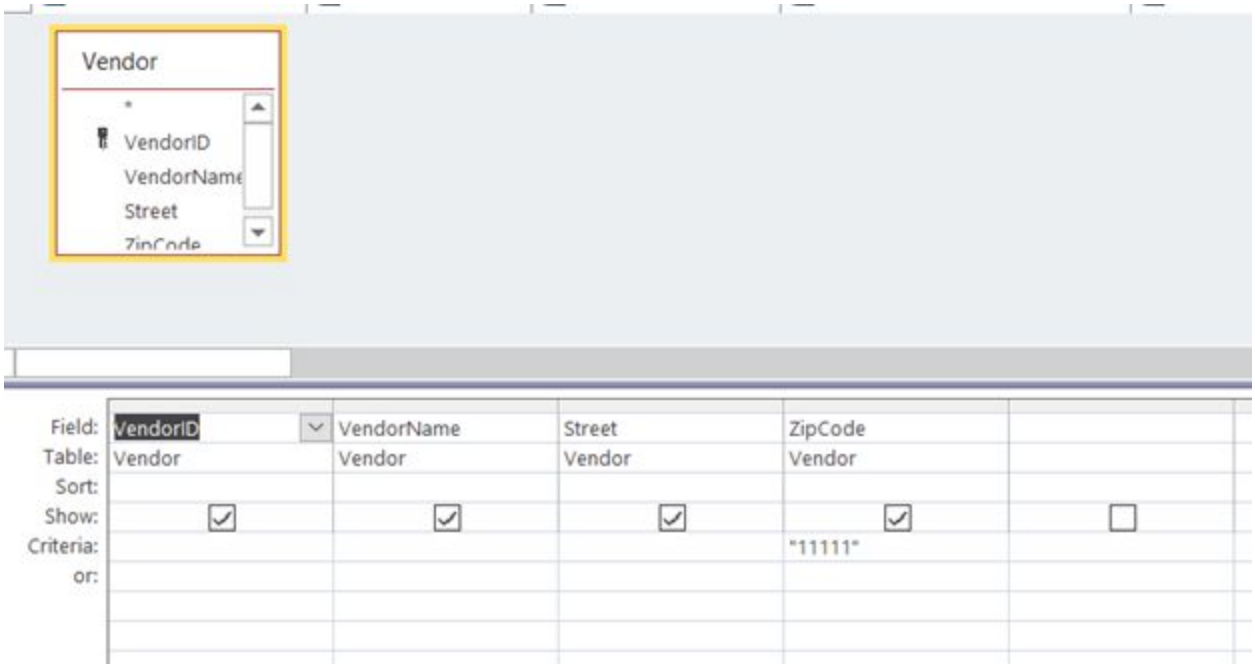
ServicePerformed Query

Be used as a search criterion for a specific service performed by its employeeID who was assigned that specific service, AppointmentID or serviceID.



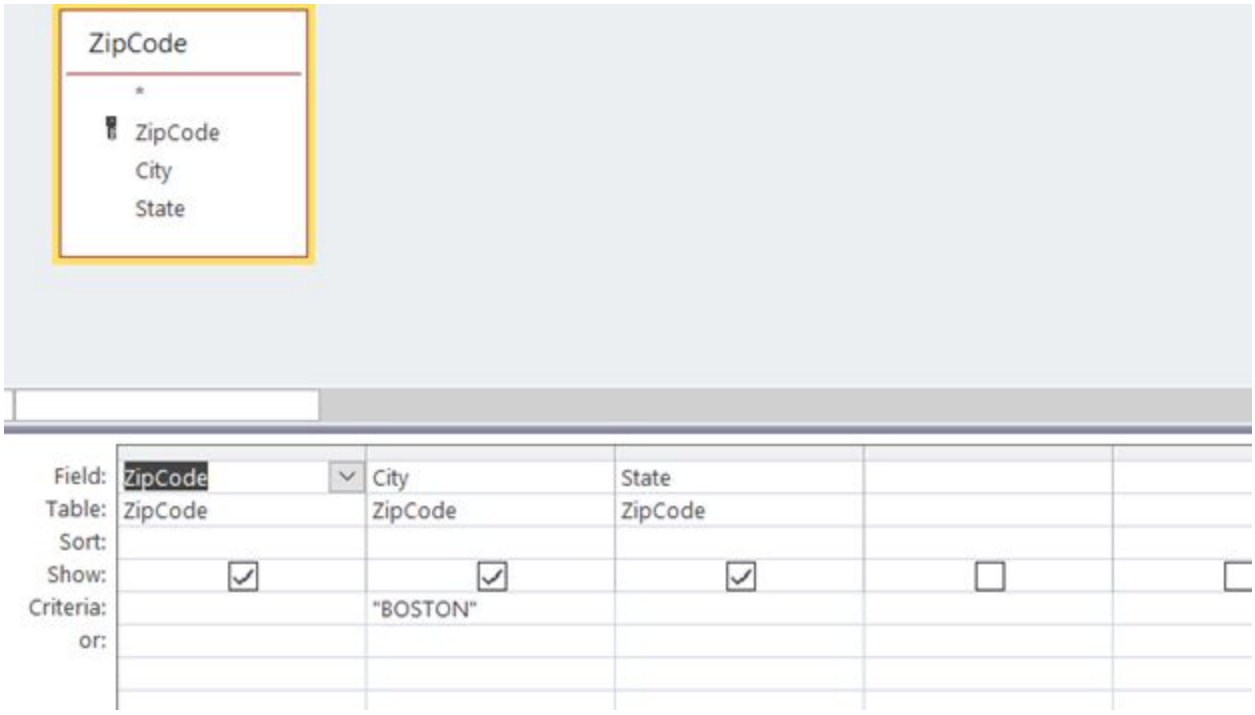
Supply_Order Query

Be used as a search criterion for a specific Supply Ordered by its VendorID who was assigned that specific order.



VendorID Query

Be used as a search criterion for a specific vendor by its Zipcode.

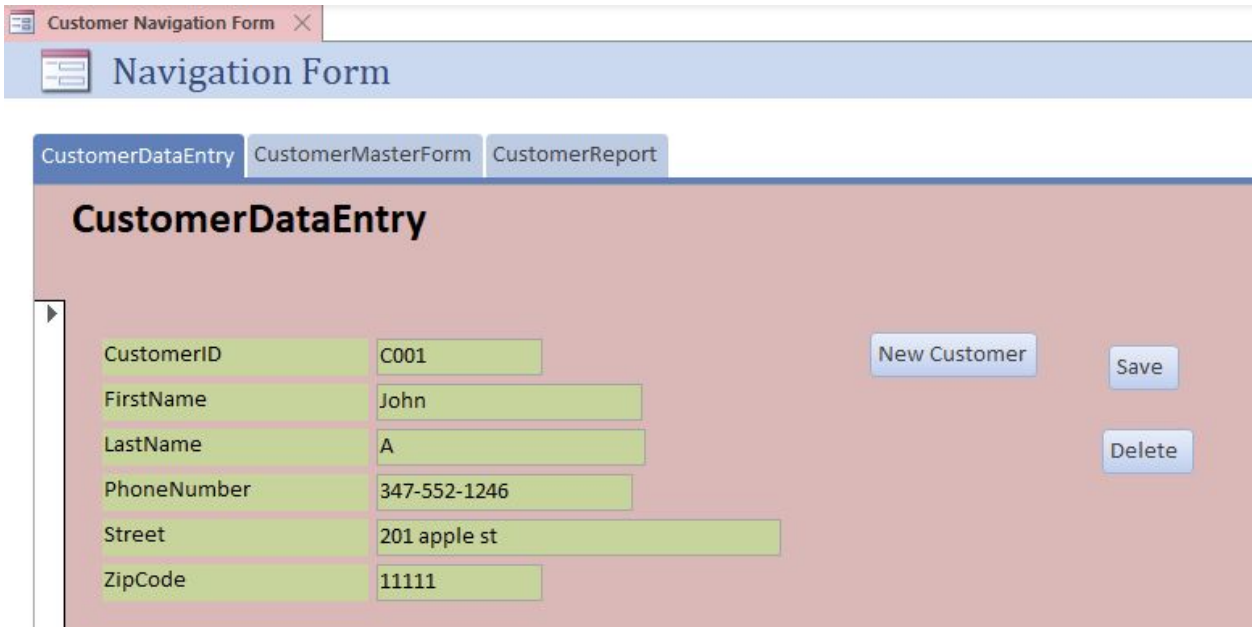


ZipCode Query

Be used as a search criterion for a specific Zipcode by its City name.

```
(General)
Option Compare Database

Sub Form_Current()

    Dim ParentDocName As String

    On Error Resume Next
    ParentDocName = Me.Parent.Name

    If Err <> 0 Then
        GoTo Form_Current_Exit
    Else
        On Error GoTo Form_Current_Err
        Me.Parent![Service Subform].Requery
    End If

Form_Current_Exit:
    Exit Sub

Form_Current_Err:
    MsgBox Error$
    Resume Form_Current_Exit

End Sub
```

VBA Coding for form



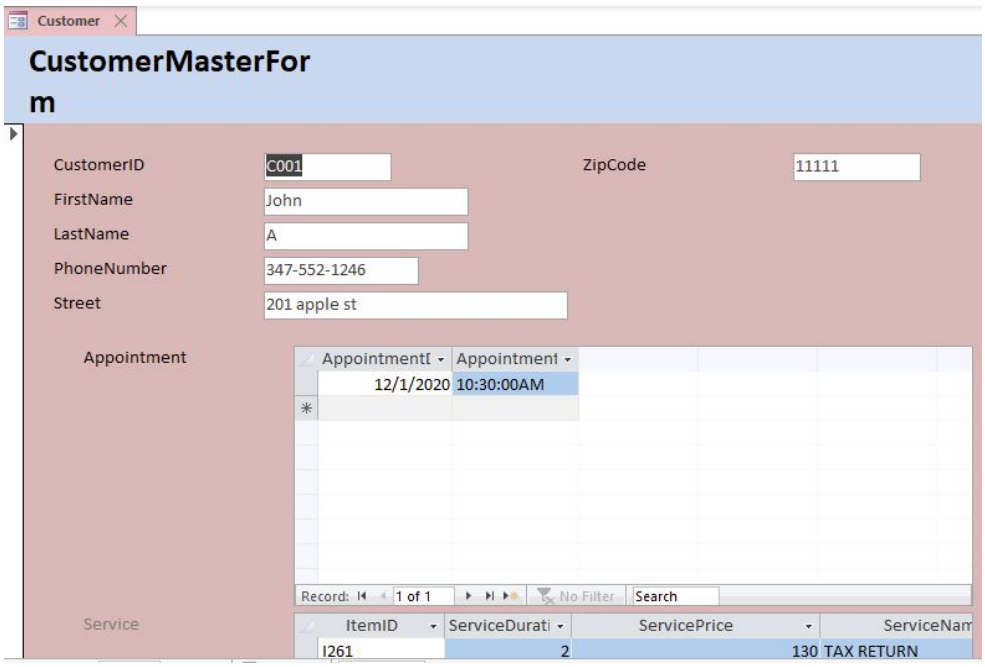
CustomerReport

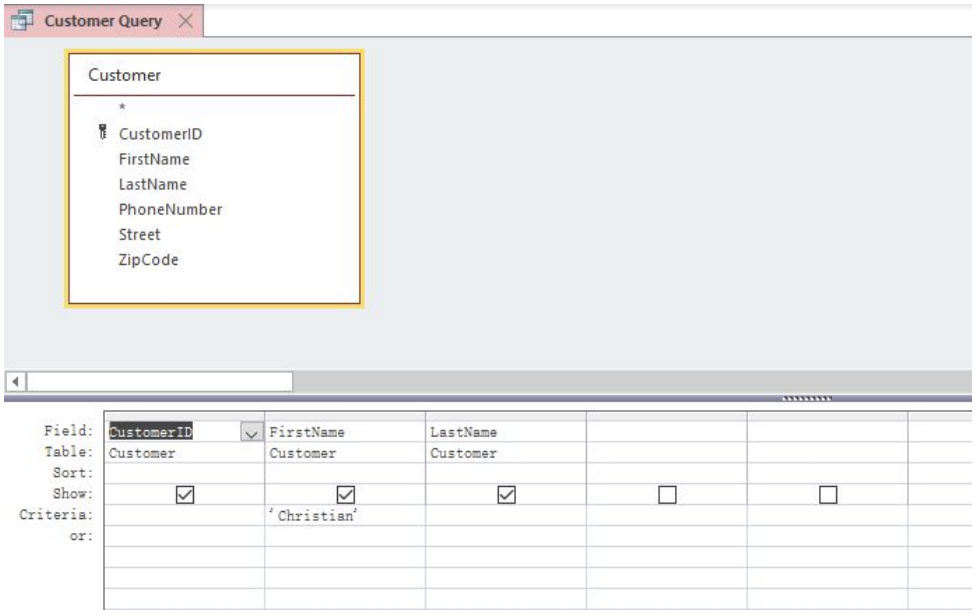
CustomerReport

Sunday, December 20, 2020
9:01:00 PM

CustomerID	FirstName	LastName	PhoneNumber	Street
C120	CHRISTIAN	DUAN	718-888-4562	75 ART ST
C999	CHRISTIAN	DUANT	347-210-1123	68 MAIN ST
C001	John	A	347-552-1246	201 apple st
C002	JANE	B	718-888-4567	222 ORANGE ST
C003	LUOIS	CC	718-778-4562	210 SKY ST
C004	SUNAY	DUF	718-818-4562	2109 SCHOOL ST
C005	SUN	SHU	718-008-1162	288 SKYVIEW ST
C888	JAKE	LEE	918-666-8888	GOLD ST

8


Customer
CustomerMasterFor
m
CustomerID
C001
ZipCode
11111
FirstName
John
LastName
A
PhoneNumber
347-552-1246
Street
201 apple st
Appointment
AppointmentI
Appointment
12/1/2020
10:30:00AM
Record:
1 of 1
No Filter
Search
Service
ItemID
ServiceDurati
ServicePrice
ServiceNam
I261
2
130
TAX RETURN


Customer Query
Customer
*
CustomerID
FirstName
LastName
PhoneNumber
Street
ZipCode
Field:
Table:
Sort:
Show:
Criteria:
or:
CustomerID
FirstName
LastName
Customer
Customer
Customer
'Christian'

Tables were being implemented from the normalised set of relations. In each of the tables, data entry form and functional keys located in the tables were created. To filter the information for easier access, query tables were made. We then created two reports for Customer and Supply_Order, one customer navigation form (including customerDataEntry, CustomerMasterForm, CustomerReport), and one Accounting Firm Main Menu navigation form.

In conclusion, we utilized both lucidcharts to draft the initial sql database and used microsoft access to put the database management system together. First we brainstormed ideas on what database to make and created an outline from start to finish. We followed the example provided by Professor Holowczak with his Hair Salon Database Project as a reference on how to execute our database. The most difficult part was coming together and finding what kind of database to create but in the end, we chose to create an accounting firm database. It was a delight to create the relationship model in addition to producing the syntax for the sql database, both were difficult to perform without prior knowledge. As the weekly lecture continued, it was becoming more and more fluent on creating the syntax for our database.