INTRODUCTION

The entities we created are Customer, Employee, Car, and Part. We assumed that the accounting supplement form can get all of the needed values from other forms such as the Sales Invoice and Car tables, and the values can all be calculated and therefore don't need to be stored. We also assumed that options in the sales invoice are the same as optional equipment and accessories in the vehicle inventory record.

NORMALIZATION

SALES INVOICE

0NF

SalesInvoice(**salesInvoiceNo**, date, name, address, city, state, postCode, salesman, serialNo, make, model, year, color, insurance(fireTheft, liability, collision, propertyDamage), {code, desc, price}, {serialNo, make, model, year, allowance}, totalPrice, tradeInAllowance, discount, net, taxes, totalPayable)

1NF

SalesInvoice(**salesInvoiceNo**, date, name, address, city, state, postCode, salesman, serialNo, make, model, year, color, totalPrice, tradeInAllowance, discount, net, taxes, totalPayable)
Insurance(**insuranceID**, fireTheft, liability, collision, propertyDamage)
Option(**code**, desc, price)
TradeIn(**serialNo**, make, model, year, allowance)

2NF

SalesInvoice(**salesInvoiceNo**, date, name, address, city, state, postCode, salesman, serialNo, make, model, year, color, totalPrice, tradeInAllowance, discount, net, taxes, totalPayable)
Insurance(**insuranceID**, fireTheft, liability, collision, propertyDamage)
Option(**code**, desc, price)
TradeIn(**serialNo**, make, model, year, allowance)

3NF

SalesInvoice(**salesInvoiceNo**, date, custID, empID, serialNo)
SalesInvoiceInsurance(**salesInvoiceNo, insuranceID**)
Insurance(**insuraceID**, fireTheft, liability, collision, propertyDamage)
SalesInvoiceOptions(**salesInvoiceNo, code)**
Option(**code**, desc, price)
SalesInvoiceCar(**salesInvoiceNo**, serialNo, allowance)
Car(**serialNo**, make, model year)
Employee(**empID**, firstName, lastName, jobID)
Job(**jobID**, title)
Customer(**custID**, name, address, city, state, postCode, telephone)

VEHICLE INVENTORY RECORD

0NF

VehicleInventoryRecord(**vehicleInventoryRecordNo**, serialNo, make, model, year, exteriorColor, trim, purchasedFrom, purchaseInvoiceNo, date, cost, basePrice, {code, desc, listPrice})

1NF

Option(**code**, desc, listPrice)
VehicleInventoryRecord(**vehicleInventoryRecordNo,** serialNo, make, model, year, exteriorColor, trim, purchasedFrom, purchaseInvoiceNo, date, cost, basePrice)

2NF

Option(**code**, desc, listPrice)
VehicleInventoryRecord(**vehicleInventoryRecordNo,** serialNo, make, model, year, exteriorColor, trim, purchasedFrom, purchaseInvoiceNo, date, cost, basePrice)

3NF

Option(**code**, desc, listPrice)
VehicleInventoryRecord(**vehicleInventoryRecordNo,** serialNo, custID, salesInvoiceNo, date)
Car(**serialNo**, make, model, year, exteriorColor, trim)
Customer(**custID**, name, address, city, state, postCode, telephone)
SalesInvoice(**salesInvoiceNo**, date, custID, empID, serialNo)

SERVICE WORK ORDER

0NF

ServiceWorkOrder(**serviceWorkOrderID**, date, name, address, city, postCode, telephoneWork, telephoneHome, serialNo, make, model, year, color, workToBeDone, partsCost, laborCost, tax, total)

1NF

ServiceWorkOrder(**serviceWorkOrder**, date, name, address, city, postCode, telephoneWork, telephoneHome, serialNo, make, model, year, color, workToBeDone, partsCost, laborCost, tax, total)

2NF

ServiceWorkOrder(**serviceWorkOrder**, date, name, address, city, postCode, telephoneWork, telephoneHome, serialNo, make, model, year, color, workToBeDone, partsCost, laborCost, tax, total)

3NF

ServiceWorkOrder(**serviceWorkOrderNo**, date, custNo, serialNo, workToBeDone)

Customer(**custNo**, name, address, city, postCode, telephoneWork, telephoneHome)
Car(**serialNo**, make, model, year, color)
PartsServiceWorkOrder(serviceWorkOrderNo, serialNo**)**
Part(**serialNo**, cost, sale)

SERVICE LOG

0NF

ServiceLog(**serviceLogID**, serviceWorkOrderID, date, serialNo, totalCost)

1NF

ServiceLog(**serviceLogID**, serviceWorkOrderID, date, serialNo, totalCost)

2NF

ServiceLog(**serviceLogID**, serviceWorkOrderID, date, serialNo, totalCost)

3NF

ServiceLog(**serviceLogID**, serviceWorkOrderID, date, serialNo, totalCost)
Car(**serialNo**, make, model, year, color)
ServiceWorkOrder(**serviceWorkOrder**, serviceInvoiceNo, date, custNo, serialNo, workToBeDone)

PROSPECTIVE CUSTOMER LIST

0NF

ProspectiveCustomer(**prospectiveCustNo,** name, want(make, model, year, upholsteryStyle))

1NF

ProspectiveCustomer(**prospectiveCustNo,** name, carTypeID)
CarType(**carTypeID**, model, year, make, exteriorColor, trim, upholsteryStyle)

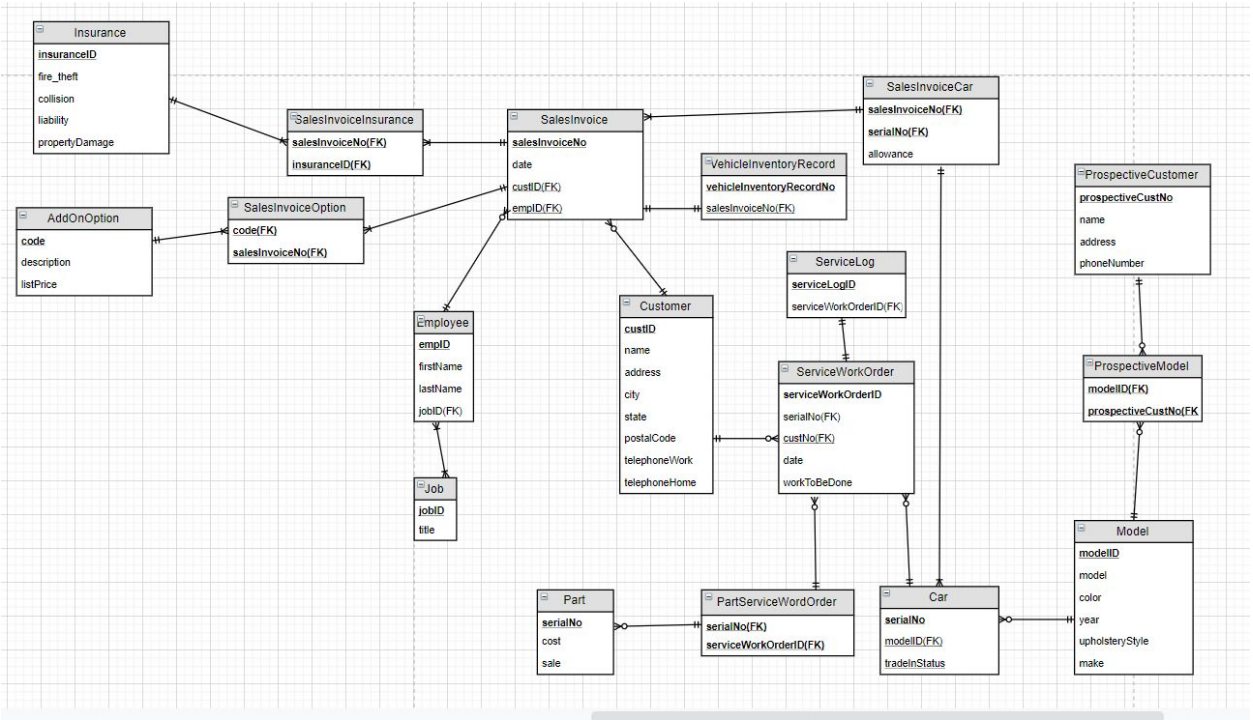
2NF

ProspectiveCustomer(**prospectiveCustNo,** name, carTypeID)
CarType(**carTypeID**, model, year, make, exteriorColor, trim, upholsteryStyle)

3NF

ProspectiveCustomer(**prospectiveCustNo,** name, carTypeID)
CarType(**carTypeID**, model, year, make, exteriorColor, trim, upholsteryStyle)

ER Diagram



MySQL - Create Tables

```
create table Model (
        modelID int identity(1,1),
   make varchar(50),
   model varchar(50),
   color varchar(50),
   year int,
   upholsteryStyle varchar(50),
   constraint pk_modelID primary key(modelID)
);
insert into Model (make, model, color, year) values ('Jaguar', 'UX', 'Black', 1997);
insert into Model (make, model, color, year, upholsteryStyle) values ('Mercedes Benz', 'JL',
'Red', 1997, 'Leather');

create table ProspectiveCustomer (
        prospectiveCustomerNo int identity(1,1),
   name varchar(50),
   address varchar(50),
   phoneNumber numeric,
   constraint pk_prospectiveCustomerNo primary key(prospectiveCustomerNo)
);
```

```
insert into ProspectiveCustomer (name, address, phoneNumber) values ('Same Molare', '45
Main Street', 4128564510);
insert into ProspectiveCustomer (name, address, phoneNumber) values ('Daffy Duck', '12
Hunter Street', 4531028460);

create table Customer (
        custID int identity(1,1),
        name varchar(50),
        address varchar(50),
        city varchar(50),
        state varchar(50),
        postCode varchar(50),
        workPhone varchar(50),
        homePhone varchar(50),
        constraint pk_custID primary key(custID)
);
insert into Customer (name, address, city, state, postCode, workPhone) values ('Bugs Bunny',
'24 Carrot Lane', 'Looneyville', 'California', '555555', 8904535421);
insert into Customer (name, address, city, state, postCode, workPhone, homePhone) values
('Elmer Fudd', '319 Befuddle Street', 'Looneyville', 'California', '00000', 891456445, 489651648);

create table Insurance (
        insuranceID int identity(1,1),
        fireTheft bit,
        collision bit,
        liability bit,
        propertyDamage bit,
        constraint pk_insuranceID primary key(insuranceID)
);
insert into Insurance (fireTheft, collision, liability, propertyDamage) values (1, 0, 0, 1);
insert into Insurance (fireTheft, collision, liability, propertyDamage) values (1, 1, 0, 0);

create table Job (
        jobID int identity(1,1),
   title varchar(50),
   constraint pk_jobID primary key(jobID)
);
insert into Job (title) values ('General Manager');
insert into Job (title) values ('Assistant Manager');

create table Part (
   serialNo varchar(50),
   cost numeric,
```

```
    sale numeric,
    constraint pk_serialNo primary key(serialNo)
);
insert into Part (serialNo, cost, sale) values ('R63', 120.00, 140.00);
insert into Part (serialNo, cost, sale) values ('S24', 320.00, 349.00);

create table Employee (
        empID int identity(1,1),
    firstName varchar(50),
    lastName varchar(50),
    jobID int,
    constraint pk_empID primary key(empID),
    constraint fk_jobID foreign key(jobID) references Job(jobID)
);
insert into Employee(firstName, lastName, jobID) values ('Maurice', 'Risner', 1);
insert into Employee(firstName, lastName, jobID) values ('Artur', 'Hrytsenko', 2);

create table SalesInvoice (
        salesInvoiceNo varchar(50),
        date date,
        custID int,
        empID int,
        constraint pk_salesInvoiceNo primary key(salesInvoiceNo),
        constraint fk_custID foreign key(custID) references Customer(custID),
        constraint fk_empID foreign key(empID) references Employee(empID)
);
insert into SalesInvoice (salesInvoiceNo, date, custID, empID) values ('S1239', '1997-01-14', 1,
2);
insert into SalesInvoice (salesInvoiceNo, date, custID, empID) values ('S1252', '1997-01-28', 2,
1);

create table SalesInvoiceInsurance (
        insuranceID int,
        salesInvoiceNo varchar(50),
    constraint pk primary key(salesInvoiceNo, insuranceID),
        constraint fk_salesInvoiceNo foreign key(salesInvoiceNo) references
SalesInvoice(salesInvoiceNo),
    constraint fk_insuranceID foreign key(insuranceID) references Insurance(insuranceID)
);
insert into SalesInvoiceInsurance (insuranceID, salesInvoiceNo) values (1, 'S1239');
insert into SalesInvoiceInsurance (insuranceID, salesInvoiceNo) values (2, 'S1252');

create table Car (
```

```
        serialNo varchar(50),
    modelID int,
    tradeInStatus bit,
    constraint pk_serialNo2 primary key(serialNo),
    constraint fk_modelID foreign key(modelID) references Model(modelID)
);
insert into Car (serialNo, modelID, tradeInStatus) VALUES ('J97UX301', 1, 0);
insert into Car(serialNo, modelID, tradeInStatus) VALUES ('M97CL701', 2, 1);

create table SalesInvoiceCar (
        salesInvoiceNo varchar(50),
    serialNo varchar(50),
    allowance numeric,
        constraint pk2 primary key(salesInvoiceNo, serialNo),
        constraint fk_salesInvoiceNo2 foreign key(salesInvoiceNo) references
SalesInvoice(salesInvoiceNo),
    constraint fk_serialNo foreign key(serialNo) references Car(serialNo)
);
insert into SalesInvoiceCar (salesInvoiceNo, serialNo, allowance) values ('S1239', 'J97UX301',
13000.00);
insert into SalesInvoiceCar (salesInvoiceNo, serialNo, allowance) values ('S1252', 'M97CL701',
5000.00);

create table AddOnOption (
        code varchar(50),
    description varchar(200),
    listPrice numeric,
    constraint pk_code primary key(code)
);
insert into AddOnOption(code, description, listPrice) values ('S24', 'Sunroof', 349.00);
insert into AddOnOption(code, description, listPrice) values ('R63', 'Stereo Radio', 140.00);

create table SalesInvoiceOption (
        code varchar(50),
    salesInvoiceNo varchar(50),
    constraint pk3 primary key(code, salesInvoiceNo),
    constraint fk_code foreign key(code) references AddOnOption(code),
    constraint fk_salesInvoiceNo3 foreign key(salesInvoiceNo) references
SalesInvoice(salesInvoiceNo)
);
insert into SalesInvoiceOption (code, salesInvoiceNo) values ('S24', 'S1239');
insert into SalesInvoiceOption (code, salesInvoiceNo) values ('S24', 'S1252');
```

```
create table VehicleInventoryRecord (
        vehicleInventoryRecordNo int identity(1,1),
   salesInvoiceNo varchar(50),
   constraint pk_vehicleInventoryRecordNo primary key(vehicleInventoryRecordNo),
   constraint fk_salesInvoiceNo4 foreign key(salesInvoiceNo) references
SalesInvoice(salesInvoiceNo)
);
insert into VehicleInventoryRecord (salesInvoiceNo) values ('S1239');
insert into VehicleInventoryRecord (salesInvoiceNo) values ('S1252');

create table ServiceWorkOrder (
        serviceWorkOrderID varchar(50),
   serialNo varchar(50),
   custID int,
   date date,
   workToBeDone varchar(250),
   constraint pk_serviceWorkOrderID primary key(serviceWorkOrderID),
   constraint fk_carSerialNo foreign key(serialNo) references Car(serialNo),
   constraint fk_custID2 foreign key(custID) references Customer(custID)
);
insert into ServiceWorkOrder (serviceWorkOrderID, serialNo, custID, date, workToBeDone)
values ('W22772', 'J97UX301', 1, '1997-05-11', 'oil change');
insert into ServiceWorkOrder (serviceWorkOrderID, serialNo, custID, date, workToBeDone)
values ('W21642', 'M97CL701', 1, '1997-02-11', 'Find squeak');

create table PartsServiceWorkOrder (
        serviceWorkOrderID varchar(50), -- to ServiceWorkORder
        serialNo varchar(50),-- to parts
        constraint fk_serialNo5 foreign key(serialNo) references Part(serialNo),
        constraint fk_serviceWorkOrderID foreign key(serviceWorkOrderID) references
ServiceWorkOrder(serviceWorkOrderID)
);

insert into PartsServiceWorkOrder (serviceWorkOrderID, serialNo) values ('W22772', 'R63');
insert into PartsServiceWorkOrder (serviceWorkOrderID, serialNo) values ('W21642', 'S24');

create table ServiceLog (
        serviceLogID int identity(1,1),
   serviceWorkOrderID varchar(50),
   constraint pk_serviceLogID primary key(serviceLogID),
   constraint fk_serviceWorkOrderID2 foreign key(serviceWorkOrderID) references
ServiceWorkOrder(serviceWorkOrderID)
);
```

```
insert into ServiceLog (serviceWorkOrderID) values ('W22772');
insert into ServiceLog (serviceWorkOrderID) values ('W21642');

create table ProspectiveCustomerModel (
            modelID int,
      prospectiveCustomerNo int,
      constraint pk4 primary key(modelID, prospectiveCustomerNo),
      constraint fk_modelID foreign key(modelID) references Model(modelID),
      constraint fk_prospectiveCustomerNo foreign key(prospectiveCustomerNo) references
ProspectiveCustomer(prospectiveCustomerNo)
);
insert into ProspectiveCustomerModel (modelID, prospectiveCustomerNo) values (1, 2);
insert into ProspectiveCustomerModel (modelID, prospectiveCustomerNo) values (2, 1);
```

MONGODB CODE

```
db.customer.insertMany([
{   custID: 1
   name:"Artur",
   address: "1234 someStreet",
   city: "Oakville",
   state: "Ontario",
   postalCode: "G6S9F4",
   telephoneWork: 4123412341,
   telephoneHome: 8926352842
   },
   {custID: 2,
   name:"Brianna",
   address: "6578 otherRoad",
   city: "Oakville",
   state: "Ontario",
   postalCode: "H9G6D4",
   telephoneWork: 6381548273,
   telephoneHome: 2816493092,
      }])

db.Car.insertMany([
   {serialNo:4123,
    tradeInStatus:"false",
    model: 871},
   {serialNo:5623,
    tradeInStatus:"true",
     model:231
```

```
    }])
db.Model.insertMany([
    {id:871,
     make:"BMW",
     model:"328i",
     color: "white",
     upholsteryStyle: "leather",
     year: "2015"
     },{
     id:231,
     make:"Chevrolet",
     model:"Malibu",
     color: "blue",
     upholsteryStyle: "regular",
     year: "2013"}])
```

MongoDB Compared to SQL

MongoDB does not have to use foreign keys to connect tables, and has no primary keys. You can use the relational model to create the database, but the connections are made by referencing values that exist in the table as opposed to creating additional keys. Creating large databases with may relations is more difficult in MongoDB because of the amount of references that have to be manually created, and if you decide to nest all the tables it can become overwhelming. It also does not support join queries. However, MongoDB supports much larger data structures than SQL and is faster at searching results. It also does not need predefined schema and you can create fields as needed immediately. The model also allows you to store complex structures such as arrays.