Normalization

Sales Invoice

Assuming a single car can be sold in a single Invoice, only one salesperson is responsible for each sales invoice, and multiple insurance options can be chosen per car. Total Price has been change to Car Price.

0NF:

SalesInvoice(InvNo,DateSold,CustId,CustName,CustAddress,CustCity,CustState,CustPostal, CustPhone,Salesman,SerialNo,CarMake,CarModel,CarYear,CarColor, FireTheft,Collision,Liability,PropertyDamage,[OptionCode,OptionDesc, OptionPrice],[TradeSerialNo,TradeMake,TradeModel,TradeYear,TradeAllowance],CarPrice,TotalAllowance,Discount,NetCost,Taxes,TotalPayable)

1NF:

SalesInvoice(InvNo,DateSold,CustId,CustName,CustAddress,CustCity,CustState,CustPostal, CustPhone,Salesman,SerialNo,CarMake,CarModel,CarYear,CarColor, FireTheft,Collision,Liability,PropertyDamage,CarPrice,TotalAllowance,Discount, NetCost,Taxes,TotalPayable)

SalesInvoiceOptions(InvNo,OptionCode,OptionDesc,OptionPrice)

SalesInvoiceTrade(InvNo,TradeSerialNo,TradeMake,TradeModel,TradeYear,TradeAllowance)

2NF:

OptionPrice was kept in SalesInvoiceOptions and OptionCost was added to options.

SalesInvoice(InvNo,DateSold,CustId,CustName,CustAddress,CustCity,CustState,CustPostal, CustPhone,Salesman,SerialNo,CarMake,CarModel,CarYear,CarColor, FireTheft,Collision,Liability,PropertyDamage,CarPrice,TotalAllowance,Discount, NetCost,Taxes,TotalPayable)

SalesInvoiceOptions(InvNo,OptionCode,OptionPrice)

SalesInvoiceTrade(InvNo,TradeSerialNo, TradeAllowance)

Options(OptionCode,OptionDesc,OptionCost)

Trade(TradeSerialNo,TradeMake,TradeModel,TradeYear)

3NF:

Assuming that car price can change overtime, a new field was created call CarCost for Car. Carprice on sales invoice represents the amount it was sold for while CarPrice represents the price a car will be sold for.

SalesInvoice(InvNo,DateSold,CustId,Salesman,SerialNo,FireTheft,Collision,Liability,PropertyDamage,CarPrice,TotalAllowance,Discount,NetCost,Taxes,TotalPayable)

SalesInvoiceOption(InvNo,OptionCode,OptionPrice)

SalesInvoiceTrade(InvNo,TradeSerialNo, TradeAllowance)

Options(OptionCode,OptionDesc,OptionCost)

Trade(TradeSerialNo,TradeMake,TradeModel,TradeYear)

Car(SerialNo,CarMake,CarModel,CarYear,CarColor)

Customer(CustId,CustName, CustAddress,CustCity,CustState,CustPostal,CustPhone)

Vehicle Inventory Record

0NF: VehicleInvRec(SerialNo,CarMake,CarModel,CarYear,CarTrim,PurchFrom,PurchInvNo,
PurchDate,PurchCost,ListBasePrice,[OptionCode,OptionDesc, OptionPrice])

1NF:

VehicleInvRec(SerialNo,CarMake,CarModel,CarYear,CarTrim,PurchFrom,PurchInvNo,
PurchDate,PurchCost,ListBasePrice)

VehicleInvRecOption(SerialNo,OptionCode,OptionDesc,OptionPrice)

2NF:

VehicleInvRec(SerialNo,CarMake,CarModel,CarYear,CarTrim,PurchFrom,PurchInvNo,
PurchDate,PurchCost,ListBasePrice)

VehicleInvRecOption(SerialNo,OptionCode,OptionPrice)

Options(OptionCode,OptionDesc,OptionCost)

3NF:

VehicleInvRec(SerialNo,CarMake,CarModel,CarYear,CarTrim,PurchInvNo, ListBasePrice)

VehicleInvRecOption(SerialNo,OptionCode,OptionPrice)

Options(OptionCode,OptionDesc,OptionCost)

PurchaseInv(PurchInvNo, PurchFrom, PurchDate, PurchCost)

Vehicle Sales Accounting Supplement

CustId was added as a primary key for customers table.

0NF: AccSupplement(InvNo,CustName,Date,SerialNo, CarCost, CarSalePrice,[OptionCode, OptionCost, OptionSale], Frieght,Tax, LicenseFees,Other,Commision,TotalCost,TotalSale)

1NF:

AccSupplement(InvNo,CustName,Date,SerialNo,CarCost,CarSalePrice,Frieght,Tax,LicenseFees ,Other,Commision, TotalCost,TotalSale)

AccSupplementOptions(InvNo,OptionCode,OptionCost,OptionSale)

2NF:

AccSupplement(InvNo,CustName,Date,SerialNo, CarCost,CarSalePrice, Frieght,Tax,LicenseFees,Other,Commision, TotalCost,TotalSale)

AccSupplementOptions(InvNo,OptionCode,OptionSale)

Options(OptionCode,OptionCost)

3NF:

AccSupplement(InvNo,CustName,Date,SerialNo,Frieght,Tax,LicenseFees,Other,Commision, TotalCost,TotalSale)

AccSupplementOptions(InvNo,OptionCode,OptionSale)

Customer(CustName)

Options(OptionCode,OptionCost)

Car(SerialNo,CarCost,CarSalePrice)

Specialty Imports Service Work Order

0NF:

ServiceWorkOrder(ServiceInvNo,ServiceDate,CustId,CustName,CustAddress,CustCity,CustState,CustPostal, CustWorkPhone,CustHomePhone,SerialNo,CarMake,CarModel,CarYear, CarColor, [WorkToDo],PartCost,LaborCost,Tax,TotalCost)

1NF:

ServiceWorkOrder(ServiceInvNo,ServiceDate,CustId,CustName,CustAddress,CustCity,CustState,CustPostal, CustWorkPhone,CustHomePhone,SerialNo,CarMake,CarModel,CarYear,CarColor,PartCost,LaborCost,Tax,TotalCost)

ServiceWorkOrderWorkToDo(ServiceInvNo, WorkToDo)

2NF: No Partial Dependencies

3NF:

ServiceWorkOrder(ServiceInvNo,ServiceDate,CustId,SerialNo,PartCost,LaborCost,Tax, TotalCost)

ServiceWorkOrderWorkToDo(ServiceInvNo, WorkToDo)

Customer(CustId, CustName,CustAddress,CustCity,CustState,CustPostal,CustWorkPhone, CustHomePhone)

Car(SerialNo,CarMake,CarModel,CarYear,CarColor)

Service Log

0NF:

ServiceLog(InvNo,Date,SerialNo,TotalCost)

1NF:

ServiceLog(InvNo,Date,SerialNo,TotalCost)

2NF: No partial dependancies

3NF: No transitive dependancies

Prospect List

0NF:

ProspectList(CustId, CustName, [YearWant, [ColorWant, [MakeWant ,[ModelWant, [TrimWant, [OptionCode,OptionDescription]]]]]])

1NF:

ProspectList(CustId,CustName)

ProspectListYearWant(CustId, YearWant)

ProspectListYearColorWant(CustId, YearWant, ColorWant)

ProspectListYearColorMakeWant(CustId, YearWant, ColorWant, MakeWant)

ProspectListYearColorMakeModelWant(CustId, YearWant, ColorWant, MakeWant, ModelWant)

ProspectListYearColorMakeModelTrimWant(CustId, YearWant, ColorWant, MakeWant, ModelWant, TrimWant)

ProspectListYearColorMakeModelTrimOptionWant(CustId, YearWant, ColorWant, MakeWant, ModelWant, TrimWant, OptionCode, OptionDescription)

2NF:

ProspectList(CustId,CustName)

ProspectListYearWant(CustId, YearWant)

ProspectListYearColorWant(CustId, YearWant, ColorWant)

ProspectListYearColorMakeWant(CustId, YearWant, ColorWant, MakeWant)

ProspectListYearColorMakeModelWant(CustId, YearWant, ColorWant, MakeWant, ModelWant)

ProspectListYearColorMakeModelTrimWant(CustId, YearWant, ColorWant, MakeWant, ModelWant, TrimWant)

ProspectListYearColorMakeModelTrimOptionWant(CustId, YearWant, ColorWant, MakeWant, ModelWant, TrimWant)

Options(OptionCode, OptionDescription)

3NF:

No transitive dependancies

Data Structure Diagram

SalesInvoice(InvNo,DateSold,CustId,Salesman,SerialNo,FireTheft,Collision,Liability, PropertyDamage,CarPrice,TotalAllowance,Freight,LicenseFees, OtherCosts,Commision,Discount,Taxes)

SalesInvoiceOption(InvNo,OptionCode,OptionPrice,OptionSale)

SalesInvoiceTrade(InvNo,TradeSerialNo, TradeAllowance)

Options(OptionCode,OptionDesc,OptionCost)

Car(SerialNo,CarMake,CarModel,CarYear,CarColor,PurchInvNo, ListBasePrice, CustId)

Customer(CustId,CustName, CustAddress,CustCity,CustState,CustPostal,CustPhone)

VehicleInvRecOption(SerialNo,OptionCode)

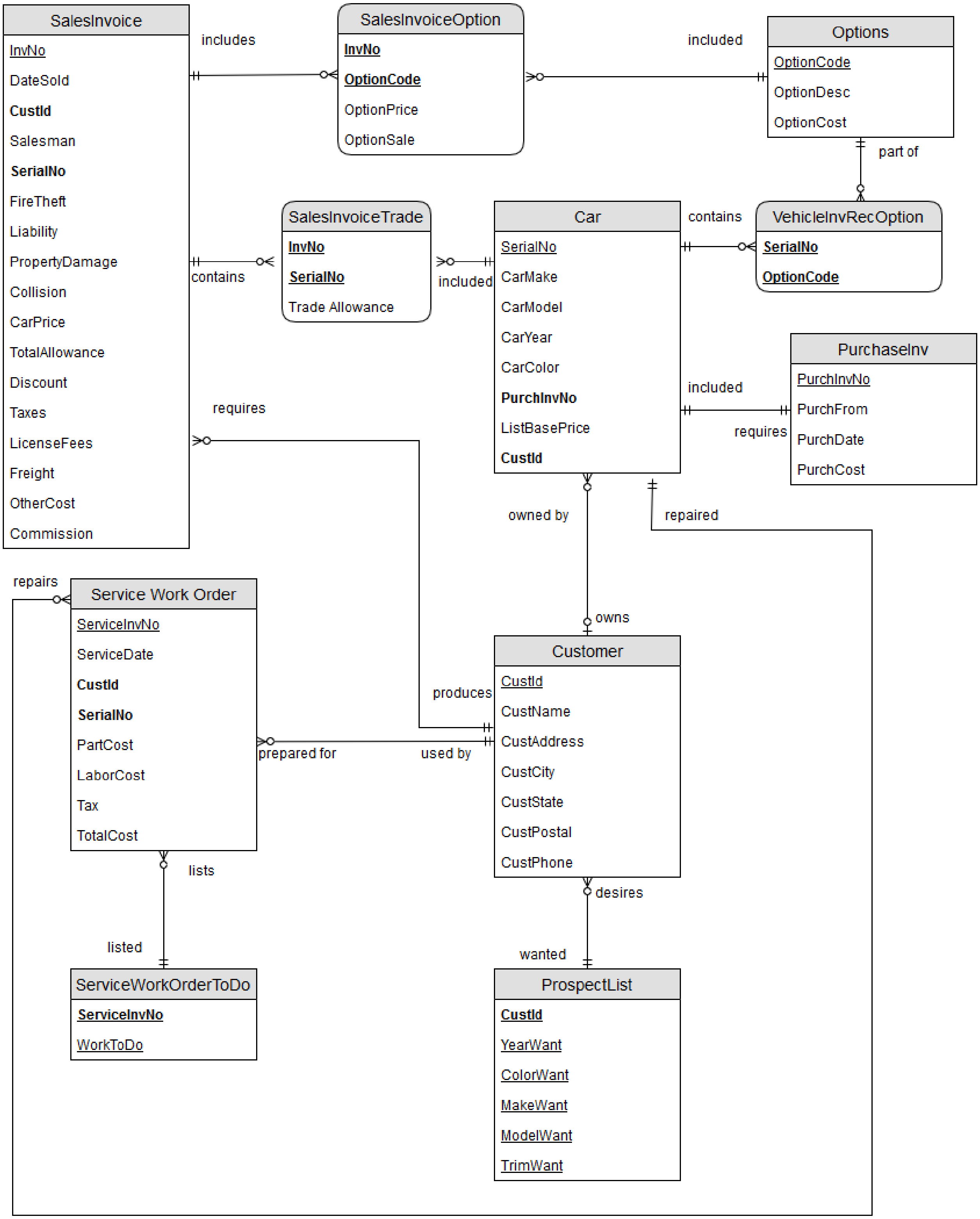
PurchaseInv(PurchInvNo, PurchFrom, PurchDate, PurchCost)

ServiceWorkOrder(ServiceInvNo,ServiceDate,CustId,SerialNo,PartCost,LaborCost,Tax)

ServiceWorkOrderWorkToDo(ServiceInvNo, WorkToDo)

ProspectList (CustId, YearWant, ColorWant, MakeWant, ModelWant, TrimWant)

EER Diagram



TABLES SQL CODE

```
DROP TABLE SalesInvoiceTrade;
DROP TABLE SalesInvoiceOption;
DROP TABLE VehicleInvRecOption;
DROP TABLE Options;
DROP TABLE ProspectList;
DROP TABLE ServiceWorkOrderWorkToDo;
DROP TABLE ServiceWorkOrder;
DROP TABLE SalesInvoice;
DROP TABLE Car;
DROP TABLE Customer;
DROP TABLE PurchaseInv;

CREATE TABLE Options
(
   optioncode CHAR(3),
   optiondesc VARCHAR(70),
   optioncost NUMBER(8,2),
   CONSTRAINT pkoptions PRIMARY KEY (optioncode)
);

CREATE TABLE Customer
(
   custid CHAR(6),
   custname VARCHAR(30),
   custaddress VARCHAR(50),
   custcity VARCHAR(30),
   custstate VARCHAR(30),
   custpostal VARCHAR(6),
   custphone NUMBER(10),
   CONSTRAINT pkcustomer PRIMARY KEY (custid)
);

CREATE TABLE PurchaseInv
(
   purchinvno CHAR(6),
   purchfrom VARCHAR(30),
   purchdate DATE,
   purchcost NUMBER(10,2),
   CONSTRAINT pkpurchinvno PRIMARY KEY (purchinvno)
);




CREATE TABLE Car
(
   serialno CHAR(17),
   carmake VARCHAR(30),
   carmodel VARCHAR(15),
   caryear NUMBER(4),
   carcolor VARCHAR(15),
   carcost NUMBER(10,2),
   purchinvno CHAR(6),
   custid CHAR(6),
   CONSTRAINT pk1car PRIMARY KEY (serialno),
   CONSTRAINT fk1car FOREIGN KEY (purchinvno)
       REFERENCES PurchaseInv(purchinvno),
   CONSTRAINT fk2car FOREIGN KEY (custid)
       REFERENCES Customer(custid)
);
```

```
CREATE TABLE SalesInvoice
(
    invno CHAR(8),
    datesold DATE,
    custid CHAR(6),
    salesman VARCHAR(30),
    serialno CHAR(17),
    firetheft CHAR(1), /*CHAR(1) is used for boolean (Y for yes and N for no)*/
    liability CHAR(1),
    propertydamage CHAR(1),
    collision CHAR(1),
    carprice NUMBER(10,2),
    totalallowance NUMBER(9,2),
    freight NUMBER (9,2),
    licensefees NUMBER (9,2),
    othercosts NUMBER (9,2),
    commission NUMBER (8,2),
    discount NUMBER (8,2),
    netcost NUMBER (11,2),
    taxes NUMBER (9,2),
    CONSTRAINT pksalesinvoice PRIMARY KEY (invno),
    CONSTRAINT fk1salesinvoice FOREIGN KEY (custid)
        REFERENCES Customer(custid),
    CONSTRAINT fk2salesinvoice FOREIGN KEY (serialno)
        REFERENCES Car(serialno)
);


CREATE TABLE SalesInvoiceOption
(
    invno CHAR(8),
    optioncode CHAR(3),
    optionprice NUMBER(8,2),
    optionsale NUMBER(8,2),
    CONSTRAINT pk1salesinvoiceoption PRIMARY KEY (invno,optioncode),
    CONSTRAINT fk1salesinvoiceoption FOREIGN KEY (invno)
        REFERENCES salesinvoice(invno),
    CONSTRAINT fk2salesinvoiceoption FOREIGN KEY (optioncode)
        REFERENCES options(optioncode)
);

CREATE TABLE SalesInvoiceTrade
(
    invno CHAR(8),
    serialno CHAR(17),
    tradeallowance NUMBER(8,2),
    CONSTRAINT pk1salesinvoicetrade PRIMARY KEY (invno,serialno),
    CONSTRAINT fk1salesinvoicetrade FOREIGN KEY (invno)
        REFERENCES salesinvoice(invno),
    CONSTRAINT fk2salesinvoicetrade FOREIGN KEY (serialno)
        REFERENCES car(serialno)
);



CREATE TABLE VehicleInvRecOption
(
    serialno CHAR(17),
    optioncode CHAR(3),
    CONSTRAINT pk1vehicleinv PRIMARY KEY (serialno,optioncode),
    CONSTRAINT fk1vehicleinv FOREIGN KEY (serialno)
        REFERENCES Car(serialno),
    CONSTRAINT fk2vehicleinv FOREIGN KEY (optioncode)
        REFERENCES options(optioncode)
);


CREATE TABLE ServiceWorkOrder
(
    serviceinvno CHAR(6),
    servicedate DATE,
    custid CHAR(6),
    serialno CHAR(17),
    partcost NUMBER(8,2),
    laborcost NUMBER(8,2),
    tax NUMBER(7,2),
    CONSTRAINT pkserviceworkorder PRIMARY KEY (serviceinvno),
    CONSTRAINT fk1serviceworkorder FOREIGN KEY (custid)
        REFERENCES customer(custid),
    CONSTRAINT fk2serviceworkorder FOREIGN KEY (serialno)
        REFERENCES car(serialno)
);

CREATE TABLE ServiceWorkOrderWorkToDo
(
    serviceinvno CHAR(6),
    worktodo VARCHAR(50),
    CONSTRAINT pk1serviceworkordertodo PRIMARY KEY (serviceinvno, worktodo),
    CONSTRAINT fkserviceworkordertodo FOREIGN KEY (serviceinvno)
        REFERENCES ServiceWorkOrder(serviceinvno)
);

CREATE TABLE ProspectList
(
    custid CHAR(6),
    yearwant NUMBER(4),
    colorwant VARCHAR(30),
    makewant VARCHAR(30),
    modelwant VARCHAR(30),
    trimwant VARCHAR(30),
    CONSTRAINT pk1prospectlist PRIMARY
KEY(custid,yearwant,colorwant,makewant,modelwant,trimwant),
    CONSTRAINT fkprospectlist FOREIGN KEY(custid)
        REFERENCES Customer(custid)
);
```

NOSQL Inserts

In this report, we have chosen to create collections for the OPTIONS, CARS, and CUSTOMER. Since NoSQL has limited functionality in checking the integrity of the relations between entities, which is a vital part of any database. We chose these tables to show how an entity will reference another entity by inserting object ids.

Looking at the CARS collection, since cars can be manufactured in different ways. There are options available that can be added to the vehicle when it is manufactured. Therefore, having multiple OPTIONS object documented in the CARS collection is understandable because the OPTIONS collection will be a separate entity that will contain the options description and the price. Similarly, with the CUSTOMER collection and the CARS collection. A customer can purchase multiple cars so by entering CARS object in the CUSTOMERS collection it will retain referential integrity between the two entities.

```
use SpecialImports;
db.createCollection.('OPTIONS');
db.createCollection.('CARS');
db.createCollection.('CUSTOMER');

db.getCollection('OPTIONS').insertMany([{
    _id : "J23",
    "optionDesc" : "Lighting Group",
    "optionCost" : 699.0
},{
    _id : "R87",
    "optionDesc" : "Rustproofing",
    "optionCost" : 898.0
},{
    _id : "A23",
    "optionDesc" : "Total Comfort Air",
    "optionCost" : 788.0
},{
    _id : "W31",
    "optionDesc" : "Aluminum Wheels",
    "optionCost" : 440.0
},{
    _id : "L12",
    "optionDesc" : "Lighting Group",
    "optionCost" : 390.0
}
])
```

```
db.getCollection('CARS').insertMany([{
    _id : "J97UX301",
    "carMake" : "Jaguar",
    "carModel" : "UX",
    "carYear" : "1997",
    "carColor" : "black",
    "carCost" : 59000.0,
    "purchInvNo" : "J43294",
    "listBasePrice" : 69999.0,
    "option_id" : [
        {_id : "J23"},
        {_id : "R87"},
        {_id : "A23"}
 ]},
{
    id : "M97CL701",
    "carMake" : "Mercedes Benz",
    "carModel" : "JL",
    "carYear" : "1997",
    "carColor" : "red",
    "carCost" : 34000.0,
    "purchInvNo" : "J43294",
    "listBasePrice" : 38900.0,
    "option_id" : [
        {_id : "W31"},
        {_id : "L12"}
    ]
}])
```

```
db.getCollection('CUSTOMER').insertMany([{
    _id : "000001",
    custName : "Bugs Bunny",
    custAddress : "24 Carrot Line",
    custCity : "Looneysville",
    custState : "California",
    custPostal : "55555",
    custPhone : "890-453-5421",
    "car_id" : [{
        _id : "J97UX301"
}]},
{
    _id : "000002",
    custName : "Elmer Befudd",
    custAddress : "319 Befuddled Street",
    custCity : "Looneysville",
    custState : "California",
    custPostal : "000000",
    custPhone : "891-454-6445",
    "car_id" : [{
        _id : "M97CL701"
    }]
}])
```

```
db.getCollection('CARS').insertMany([{
    _id : "J97UX301",
    "carMake" : "Jaguar",
    "carModel" : "UX",
    "carYear" : "1997",
    "carColor" : "black",
    "carCost" : 59000.0,
    "purchInvNo" : "J43294",
    "listBasePrice" : 69999.0,
    "option_id" : [
        {_id : "J23"},
        {_id : "R87"},
        {_id : "A23"}
]},
{
    id : "M97CL701",
    "carMake" : "Mercedes Benz",
    "carModel" : "JL",
    "carYear" : "1997",
    "carColor" : "red",
    "carCost" : 34000.0,
    "purchInvNo" : "J43294",
    "listBasePrice" : 38900.0,
    "option_id" : [
        {_id : "W31"},
        {_id : "L12"}
    ]
}])
```

```
db.getCollection('CUSTOMER').insertMany([{
    _id : "000001",
    custName : "Bugs Bunny",
    custAddress : "24 Carrot Line",
    custCity : "Looneysville",
    custState : "California",
    custPostal : "55555",
    custPhone : "890-453-5421",
    "car_id" : [{
        _id : "J97UX301"
}]},
{
    _id : "000002",
    custName : "Elmer Befudd",
    custAddress : "319 Befuddled Street",
    custCity : "Looneysville",
    custState : "California",
    custPostal : "000000",
    custPhone : "891-454-6445",
    "car_id" : [{
        _id : "M97CL701"
    }]
}])
```