

set echo on

-- Code to add tables.

-- Originally, some columns were modified and some tables were dropped and added again.

-- This code repairs for those errors.

-- DealershipInformation created successfully.

```
CREATE TABLE DealershipInformation(  
    averageRating          NUMBER(2,1),  
    dealerAddressLine1     VARCHAR2(50),  
    dealerAddressLine2     VARCHAR2(50),  
    dealerCity             VARCHAR2(20),  
    dealerEmail            VARCHAR2(25),  
    dealerName             VARCHAR2(25),  
    dealerPhone            NUMBER(10),  
    dealershipID           NUMBER(1) PRIMARY KEY,  
    dealerZipCode          NUMBER(9),  
    inventory              NUMBER(3),  
    managerAddressLine1    VARCHAR2(50),  
    managerAddressLine2    VARCHAR2(50),  
    managerCity            VARCHAR2(20),  
    managerEmail           VARCHAR2(25),  
    managerFirstName       VARCHAR2(15) NOT NULL,  
    managerLastName        VARCHAR2(20) NOT NULL,  
    managerPayWeekly       NUMBER(4,2),  
    managerPayYearly       NUMBER(6,2),  
    managerPhone           NUMBER(10),  
    managerSSN             NUMBER(9),  
    managerZipCode         NUMBER(9),
```

```
salespeopleNumber    NUMBER(2)
);
```

-- Vehicle created successfully.

```
CREATE TABLE Vehicle(
    approval           NUMBER(1),
    bodyStyle          VARCHAR2(11),
    capacityOf         NUMBER(2),
    condition          VARCHAR2(7),
    customization      VARCHAR2(200),
    cylinders           NUMBER(2),
    datePurchased      DATE,
    dateSold           DATE,
    dateDelivered      DATE,
    dealershipID       NUMBER(1),
    doors              NUMBER(2),
    electric            NUMBER(1),
    electricRange       NUMBER(3),
    extendedWarrantyCost  NUMBER(5,2),
    extendedWarranty    NUMBER(1),
    exteriorColor      VARCHAR2(15),
    gasCity            NUMBER(3),
    gasHighway         NUMBER(3),
    interiorColor      VARCHAR2(15),
    manufacturer       VARCHAR2(15),
    mileageCurrent     NUMBER(6),
    mileageDelivery    NUMBER(6),
    mileageSold        NUMBER(6),
    modelName          VARCHAR2(25),
```

```

motor          VARCHAR2(20),
ownerOf        VARCHAR2(10),
purchasePrice  NUMBER(6,2),
sellPrice      NUMBER(6,2),
status         VARCHAR2(11),
stickerPrice   NUMBER(6,2),
vehicleMiscFeatures VARCHAR2(350),
vehicleID      VARCHAR2(17) PRIMARY KEY,
warrantyLength NUMBER(5),
warrantyOther  VARCHAR2(50),
weight         NUMBER(5),
yearMade       NUMBER(4),
CONSTRAINT approvalCheckV CHECK(approval IN('0','1')),
CONSTRAINT bodyCheckV CHECK(bodyStyle
IN('bus','convertible','coupe','hatchback','minivan','motorcycle','other','roadster','RV','sedan','van')),
CONSTRAINT conditionCheckV CHECK(condition IN('new','used','salvage')),
CONSTRAINT electricCheckV CHECK(electric IN('0','1')),
CONSTRAINT extendedCheckV CHECK(extendedWarranty IN('0','1')),
CONSTRAINT ownerCheckV CHECK(ownerOf IN('dealership','customer')),
CONSTRAINT statusCheckV CHECK(status IN('available','customizing','ordered','pending','sold')),
CONSTRAINT Vehicle_dealershipID_fk FOREIGN KEY (dealershipID) REFERENCES
DealershipInformation(dealershipID) ON DELETE SET NULL
);

```

-- Salesperson created successfully.

```

CREATE TABLE Salesperson(
    addressLine1    VARCHAR2(50),
    addressLine2    VARCHAR2(50),
    averageRating   NUMBER(2,1),

```

```

city                VARCHAR2(20),
dealershipID        NUMBER(1),
email               VARCHAR2(25),
firstName           VARCHAR2(15) NOT NULL,
lastName            VARCHAR2(20) NOT NULL,
phone               NUMBER(10),
salespersonID       NUMBER(2),
SSN                 NUMBER(9),
zipCode             NUMBER(9),
CONSTRAINT Salesperson_salespersonID_pk PRIMARY KEY (salespersonID),
CONSTRAINT Salesperson_dealershipID_fk FOREIGN KEY (dealershipID) REFERENCES
DealershipInformation(dealershipID) ON DELETE SET NULL
);

```

-- Customer created successfully.

```

CREATE TABLE Customer(
    addressLine1      VARCHAR2(50),
    addressLine2      VARCHAR2(50),
    city              VARCHAR2(20),
    customerID         NUMBER(6) PRIMARY KEY,
    email              VARCHAR2(25),
    firstName          VARCHAR2(15) NOT NULL,
    lastName           VARCHAR2(20) NOT NULL,
    licenseNumber      NUMBER(12),
    licenseState       VARCHAR2(2),
    phone              NUMBER(10),
    salespersonID      NUMBER(2),
    SSN                NUMBER(9),
    stateOfResidence   VARCHAR2(2),

```

```

tradeInVehicleID    VARCHAR2(17),
vehicleID            VARCHAR2(17),
zipCode              NUMBER(9),

CONSTRAINT Customer_salespersonID_fk FOREIGN KEY (salespersonID) REFERENCES
Salesperson(salespersonID) ON DELETE SET NULL,

CONSTRAINT Customer_tradeInVehicleID_fk FOREIGN KEY (tradeInVehicleID) REFERENCES
Vehicle(vehicleID) ON DELETE SET NULL,

CONSTRAINT Customer_vehicleID_fk FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID) ON
DELETE SET NULL

);

```

-- BillOfSale created successfully.

```

CREATE TABLE BillOfSale(
    customerID          NUMBER(6),
    downPayment          NUMBER(6,2),
    financingLocation    VARCHAR2(20),
    insuranceCompany     VARCHAR2(20),
    insuranceNumber      VARCHAR2(20),
    licenseFee           NUMBER(3,2),
    otherDocumentation   VARCHAR2(200),
    salespersonID        NUMBER(2),
    salesTax              NUMBER(5,2),
    titleStatus          VARCHAR2(10),
    vehicleID            VARCHAR2(17),

    CONSTRAINT titleCheck CHECK(titleStatus IN('customer','pending','dealership')),

    CONSTRAINT BillOfSale_vehicleID_pk PRIMARY KEY (vehicleID),

    CONSTRAINT BillOfSale_customerID_fk FOREIGN KEY (customerID) REFERENCES
Customer(customerID) ON DELETE SET NULL,

    CONSTRAINT BillOfSale_salespersonID_fk FOREIGN KEY (salespersonID) REFERENCES
Salesperson(salespersonID) ON DELETE SET NULL

```

);

-- Commissions created successfully.

CREATE TABLE Commissions(

commissionMade NUMBER(5,2),

hours NUMBER(3),

monthOf NUMBER(2),

reportNumber NUMBER(4),

salespersonID NUMBER(2),

soldNumber NUMBER(2),

soldValue NUMBER(6,2),

yearOf NUMBER(4),

CONSTRAINT Commissions_reportNumber_pk PRIMARY KEY (reportNumber),

CONSTRAINT Commissions_salespersonID_fk FOREIGN KEY (salespersonID) REFERENCES
Salesperson(salespersonID) ON DELETE SET NULL

);

-- DealershipFinances created successfully.

CREATE TABLE DealershipFinances(

commissionsAmount NUMBER(6,2),

dealershipID NUMBER(1),

expenses NUMBER(6,2),

monthOf NUMBER(2),

numberPurchased NUMBER(3),

numberSold NUMBER(3),

otherFinanceInfo VARCHAR2(200),

profits NUMBER(7,2),

purchaseValue NUMBER(7,2),

reportNumber NUMBER(4),

```

salesValue          NUMBER(7,2),
totalIncome         NUMBER(7,2),
yearOf              NUMBER(4),
CONSTRAINT DF_reportNumber_pk PRIMARY KEY (reportNumber),
CONSTRAINT DF_dealershipID_fk FOREIGN KEY (dealershipID) REFERENCES
DealershipInformation(dealershipID) ON DELETE SET NULL
);

```

-- PotentialCustomer created successfully.

```

CREATE TABLE PotentialCustomer(
    addressLine1      VARCHAR2(50),
    addressLine2      VARCHAR2(50),
    city              VARCHAR2(20),
    dateInquired      DATE,
    email             VARCHAR2(25),
    firstName         VARCHAR2(15) NOT NULL,
    lastName          VARCHAR2(20) NOT NULL,
    phone             NUMBER(10),
    potentialCustomerID NUMBER(6),
    stateOfResidence  VARCHAR2(2),
    vehicleOfInterest1 VARCHAR2(17),
    vehicleOfInterest2 VARCHAR2(17),
    vehicleOfInterest3 VARCHAR2(17),
    zipCode           NUMBER(9),
    CONSTRAINT PotentialCustomer_potentialCustomerID_pk PRIMARY KEY (potentialCustomerID),
    CONSTRAINT PotentialCustomer_vehicle1_fk FOREIGN KEY (vehicleOfInterest1) REFERENCES
Vehicle(vehicleID) ON DELETE SET NULL,
    CONSTRAINT PotentialCustomer_vehicle2_fk FOREIGN KEY (vehicleOfInterest2) REFERENCES
Vehicle(vehicleID) ON DELETE SET NULL,

```

```
CONSTRAINT PotentialCustomer_vehicle3_fk FOREIGN KEY (vehicleOfInterest3) REFERENCES
Vehicle(vehicleID) ON DELETE SET NULL
```

```
);
```

```
-- Survey created successfully.
```

```
CREATE TABLE Survey(
```

```
    customerID          NUMBER(6),
```

```
    dealershipID        NUMBER(1),
```

```
    dealershipOpinion    VARCHAR2(200),
```

```
    dealershipRating     NUMBER(2),
```

```
    discoveredDealership VARCHAR2(10),
```

```
    salespersonID        NUMBER(2),
```

```
    salespersonOpinion   VARCHAR2(200),
```

```
    salespersonRating    NUMBER(2),
```

```
    surveyNumber         NUMBER(6),
```

```
    vehicleID            VARCHAR2(17),
```

```
    vehicleOpinion       VARCHAR2(200),
```

```
    vehicleRating        NUMBER(2),
```

```
    CONSTRAINT dealerCheck CHECK(dealershipRating IN ('1','2','3','4','5','6','7','8','9','10')),
```

```
    CONSTRAINT salesCheck CHECK(salespersonRating IN ('1','2','3','4','5','6','7','8','9','10')),
```

```
    CONSTRAINT vehicleCheck CHECK(vehicleRating IN ('1','2','3','4','5','6','7','8','9','10')),
```

```
    CONSTRAINT Survey_surveyNumber_pk PRIMARY KEY (surveyNumber),
```

```
    CONSTRAINT Survey_customerID_fk FOREIGN KEY (customerID) REFERENCES Customer(customerID)
ON DELETE SET NULL,
```

```
    CONSTRAINT Survey_dealershipID_fk FOREIGN KEY (dealershipID) REFERENCES
DealershipInformation(dealershipID) ON DELETE SET NULL,
```

```
    CONSTRAINT Survey_salespersonID_fk FOREIGN KEY (salespersonID) REFERENCES
Salesperson(salespersonID) ON DELETE SET NULL,
```

```
    CONSTRAINT Survey_vehicleID_fk FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID) ON
DELETE SET NULL
```



```
);
```

```
-- Code to insert values into the tables.
```

```
-- Adding dealership:
```

```
-- Adding Friendly Cars Dealership, the table is setup in a way that allows more dealerships to be added in the future.
```

```
INSERT INTO
```

```
DealershipInformation(dealerCity,dealerName,dealershipID,managerFirstName,managerLastName,sales  
peopleNumber)
```

```
VALUES('Asheville','Friendly Cars Dealership',0,'William','Shatner',2);
```

```
-- Adding vehicles:
```

```
-- Pursuit Special (V8 Interceptor)
```

```
INSERT INTO
```

```
Vehicle(cylinders,dealershipID,exteriorColor,interiorColor,manufacturer,modelName,vehicleID,yearMade)
```

```
VALUES(8,'0','black','tan','Ford','Pursuit Special','1',1973);
```

```
UPDATE Vehicle -- Adding some more information.
```

```
SET bodyStyle = 'coupe', capacityOf = 2, customization = 'Outfitted for the wasteland.', motor = '351 cu  
in', ownerOf = 'customer', vehicleMiscFeatures = 'Supercharger, wasteland enhancements', weight =  
3073
```

```
WHERE VehicleID = '1';
```

```
-- Battlestar Galactica
```

```
INSERT INTO
```

```
Vehicle(dealershipID,exteriorColor,interiorColor,manufacturer,modelName,motor,vehicleMiscFeatures,  
vehicleID)
```

```
VALUES('0','metal','metal','Colonial Fleet','Galactica','Sublight','Vipers, Raptors, FTL','75');
```

```
-- USS Enterprise (NCC-1701-D)
```

```
INSERT INTO
```

```
Vehicle(dealershipID,exteriorColor,interiorColor,manufacturer,modelName,motor,ownerOf,vehicleMisc  
Features,vehicleID,yearMade)
```

```
VALUES('0','metal','metal','Utopia Plantia','Enterprise','Impulse','customer','Two warp nacelles,  
Picard','1701D',2363);
```

```
-- Adding salespeople:
```

```
-- Quark
```

```
INSERT INTO Salesperson(averageRating,city,dealershipID,firstName,lastName,salespersonID)
```

```
VALUES(1,'Deep Space Nine',0,'Quark','Unknown',0);
```

```
-- Watto
```

```
INSERT INTO Salesperson(averageRating,city,dealershipID,firstName,lastName,salespersonID)
```

```
VALUES(4,'Mos Espa',0,'Watto','Unknown',1);
```

```
-- Adding customers:
```

```
-- Malcolm Reynolds
```

```
INSERT INTO Customer(customerID,firstName,lastName,salespersonID,vehicleID)
```

```
VALUES(0,'Malcolm','Reynolds',0,'75');
```

```
-- Jack O'Neill
```

```
INSERT INTO Customer(city,customerID,firstName,lastName,licenseState,salespersonID,vehicleID)
```

```
VALUES('Cheyenne Mountain',1,'Jack','ONeill','CO',1,'1701D');
```

UPDATE Customer

SET tradeInVehicleID = '1'

WHERE customerID = 1;

-- Adding bills of sale for customers:

-- Malcolm Reynolds and Quark

INSERT INTO

BillOfSale(customerID,financingLocation,insuranceCompany,salespersonID,titleStatus,vehicleID)

VALUES(0,'Starfleet','Weyland-Yutani',0,'customer','1701D');

-- Jack O'Neill and Watto

INSERT INTO

BillOfSale(customerID,financingLocation,insuranceCompany,salespersonID,titleStatus,vehicleID)

VALUES(1,'INGen','SkyNet','1','customer','75');

-- Adding commissions for salespeople:

-- October 2016 for Quark

INSERT INTO

Commissions(commissionMade,hours,monthOf,reportNumber,salespersonID,soldNumber,yearOf)

VALUES(400,200,10,1,0,1,2016);

-- October 2016 for Watto

INSERT INTO

Commissions(commissionMade,hours,monthOf,reportNumber,salespersonID,soldNumber,yearOf)

VALUES(900,165,10,2,1,3,2016);

-- Adding finances for dealerships:

-- September 2016.

```
INSERT INTO
DealershipFinances(dealershipID,monthOf,numberPurchased,numberSold,reportNumber,yearOf)
VALUES(0,9,5,5,0,2016);
```

-- October 2016.

```
INSERT INTO
DealershipFinances(dealershipID,monthOf,numberPurchased,numberSold,reportNumber,yearOf)
VALUES(0,10,3,4,1,2016);
```

-- Adding potential customer:

-- Fox Mulder

```
INSERT INTO
PotentialCustomer(city,firstName,lastName,potentialCustomerID,stateOfResidence,vehicleOfInterest1,v
ehicleOfInterest2,vehicleOfInterest3)
VALUES('Washington D.C.','Fox','Mulder',3,'DC','1701D','75','1');
```

-- Adding survey:

-- Malcolm Reynolds

```
INSERT INTO Survey
VALUES(0,0,'Okay, I guess',5,'flew by',0,'Very pressuring.',3,1,'1701D','It is not Serenity',7);
```

-- Jack O'Neill

INSERT INTO Survey

VALUES(1,0,'Never, in the history of boredom, has anyone been as bored as I am, right now.',6,'internet',0,'Still bored.',3,1,'75','It isn't the ONeill.',7);

-- Code to select objects and create a view.

-- Creating view for the entirety of information needed for a bill of sale.

CREATE VIEW

sale(lastName,firstName,city,licenseNumber,licenseState,financingLocation,insuranceCompany,insuranceNumber,vehicleID,mileageSold,customization,warrantyLength,extendedWarranty,warrantyOther,sellPrice,salespersonID)

AS SELECT

Customer.lastName,Customer.firstName,Customer.city,Customer.licenseNumber,Customer.licenseState,BillOfSale.financingLocation,BillOfSale.insuranceCompany,BillOfSale.insuranceNumber,Vehicle.vehicleID,Vehicle.mileageSold,Vehicle.customization,Vehicle.warrantyLength,Vehicle.extendedWarranty,Vehicle.warrantyOther,Vehicle.sellPrice,Salesperson.salespersonID

FROM BillOfSale,Customer,Vehicle,Salesperson

WHERE BillOfSale.customerID = Customer.customerID

AND BillOfSale.salespersonID = Salesperson.salespersonID

AND BillOfSale.vehicleID = Vehicle.vehicleID;

-- Select from the view for a complete bill of sale.

SELECT lastName,firstName,city,financingLocation,vehicleID,salespersonID

FROM sale

WHERE salespersonID IS NOT NULL;

-- Selecting information about the dealership and monthly sales for October.

SELECT

dealerName,managerFirstName,managerLastName,monthOf,YearOf,numberPurchased,numberSold

FROM DealershipInformation,DealershipFinances

```
WHERE monthOf = 10;
```

```
-- Selecting vehicle with vehicleID of 1.
```

```
SELECT manufacturer,modelName,cylinders,motor,doors,exteriorColor,interiorColor  
FROM Vehicle  
WHERE VehicleID = '1';
```

```
-- Selecting vehicle with modelName of Enterprise.
```

```
SELECT VehicleID,vehicleMiscFeatures  
FROM Vehicle  
WHERE modelName = 'Enterprise';
```

```
-- Selecting all salespeople.
```

```
SELECT city,firstName,salespersonID  
FROM Salesperson;
```

```
-- Selecting customer from Cheyenne Mountain.
```

```
SELECT  
Customer.customerID,Customer.lastName,Customer.firstName,Customer.city,Customer.licenseState,Salesperson.firstName AS Salesperson  
FROM Customer,Salesperson  
WHERE Customer.city = 'Cheyenne Mountain'  
AND Salesperson.salespersonID = Customer.salespersonID;
```

```
-- Getting the number of sales for salespeople for October 2016
```

```
SELECT Salesperson.firstName,Commissions.hours,Commissions.soldNumber  
FROM Salesperson,Commissions  
WHERE monthOf = 10  
AND yearOf = 2016
```

AND Salesperson.salespersonID = Commissions.salespersonID;

-- Getting potential customer information.

```
SELECT
PotentialCustomer.lastName,PotentialCustomer.firstName,PotentialCustomer.stateOfResidence,Vehicle
.manufacturer,Vehicle.modelName,Vehicle.vehicleID
FROM PotentialCustomer,Vehicle
WHERE PotentialCustomer.vehicleOfInterest1 = Vehicle.vehicleID;
```

-- Getting survey information on the lowest rating for a salesperson.

```
SELECT Customer.lastName AS customerLastName,Customer.firstName AS
customerFirstName,Salesperson.firstName AS salesperson,Survey.salespersonRating AS rating,
Survey.salespersonOpinion AS opinion
FROM Customer,Salesperson,Survey
WHERE Customer.customerID = Survey.customerID
AND Salesperson.salespersonID = Survey.salespersonID
AND salespersonRating = (SELECT MIN(salespersonRating) FROM Survey);
```