```
--Greg Michael Sarah Syed
--DML: Insert statements used to populate the tables.
--Call the DML file "insert.sql"

--DDL for tables:-----------------------------------------------------
/*Beginning of Customer-related tables*/

CREATE TABLE Customer
(
    customerID          INTEGER NOT NULL AUTO_INCREMENT,
    phone               VARCHAR(10) NOT NULL,
    email               VARCHAR(30) NOT NULL,

    CONSTRAINT pk_Customer     PRIMARY KEY (customerID),
    CONSTRAINT ck_Customer     UNIQUE (phone, email)
);

CREATE TABLE Referral
(
    referralID          INTEGER NOT NULL AUTO_INCREMENT,
    customerID          INTEGER NOT NULL,
    dateReferred        DATE NOT NULL,
    timeReferred        TIME NOT NULL,

    CONSTRAINT pk_Referral     PRIMARY KEY (referralID),
    CONSTRAINT ck_Referral     UNIQUE (customerID, dateReferred, timeReferred),
    CONSTRAINT fk_Referral
    FOREIGN KEY (customerID)
    REFERENCES Customer(customerID)
);

CREATE TABLE Steady
(
    customerID          INTEGER NOT NULL,
    firstServiceDate    DATE,

    CONSTRAINT pk_Steady       PRIMARY KEY (customerID),
    CONSTRAINT fk_Steady
    FOREIGN KEY (customerID)
    REFERENCES Customer(customerID)
);

CREATE TABLE Prospective
(
    customerID          INTEGER NOT NULL,
    referralID          INTEGER,
    deadProspectFlag    BOOLEAN,

    CONSTRAINT pk_Prospective     PRIMARY KEY (customerID),
    CONSTRAINT fk_Prospective_0
    FOREIGN KEY (customerID)
    REFERENCES Customer(customerID),
    CONSTRAINT fk_Prospective_1
    FOREIGN KEY (referralID)
    REFERENCES Referral(referralID)
);


CREATE TABLE Contact
(
    customerID          INTEGER NOT NULL,
    dateContacted       DATE NOT NULL,

    CONSTRAINT pk_Contact       PRIMARY KEY (customerID, dateContacted),
    CONSTRAINT fk_Contact
    FOREIGN KEY (customerID)
    REFERENCES Prospective(customerID)
);

CREATE TABLE Individual
(
    customerID          INTEGER NOT NULL,
    firstName           VARCHAR(30),
    lastName            VARCHAR(30),
    mailingAddress      VARCHAR(30),
    mailingCity         VARCHAR(30),
    mailingState        VARCHAR(30),
    zipcode             CHAR(5),
    dateOfBirth         DATE,

    CONSTRAINT pk_Individual    PRIMARY KEY (customerID),
    CONSTRAINT fk_Individual
    FOREIGN KEY (customerID)
    REFERENCES Customer(customerID)
);


CREATE TABLE Corporate
(
    customerID          INTEGER NOT NULL,
    corpName            VARCHAR(30),

    CONSTRAINT pk_Corporate     PRIMARY KEY (customerID),
    CONSTRAINT fk_Corporate
    FOREIGN KEY (customerID)
    REFERENCES Customer(customerID)
);
```

```
CREATE TABLE Address
(
    customerID          INTEGER NOT NULL,
    addressType         VARCHAR(30) NOT NULL,
    address             VARCHAR(30) NOT NULL,
    city                VARCHAR(30) NOT NULL,
    state             VARCHAR(30) NOT NULL,
    zipcode             CHAR(5) NOT NULL,

    CONSTRAINT pk_Address PRIMARY KEY (customerID, addressType, address, city,
state, zipcode),
    CONSTRAINT fk_Address
    FOREIGN KEY (customerID)
    REFERENCES Corporate(customerID)
);

CREATE TABLE Premier
(
    customerID          INTEGER NOT NULL,
    startDate           DATE,
    endDate             DATE,

    CONSTRAINT pk_Premier       PRIMARY KEY (customerID),
    CONSTRAINT fk_Premier
    FOREIGN KEY (customerID)
    REFERENCES Customer(customerID)
);

CREATE TABLE ReferralDiscounts
(
    discountName        VARCHAR(30) NOT NULL,

    CONSTRAINT pk_ReferralDiscounts PRIMARY KEY (discountName)
);


CREATE TABLE SteadyReferral
(
    referralID          INTEGER NOT NULL,
    customerID          INTEGER,
    discountName        VARCHAR(30),

    CONSTRAINT pk_SteadyReferral   PRIMARY KEY (referralID),
    CONSTRAINT fk_SteadyReferral_0
    FOREIGN KEY (customerID)
    REFERENCES Steady(customerID),
    CONSTRAINT fk_SteadyReferral_1
    FOREIGN KEY (discountName)
    REFERENCES ReferralDiscounts(discountName)
);
```

```
CREATE TABLE PremierReferral
(
    referralID          INTEGER NOT NULL,
    customerID          INTEGER,
    discountName        VARCHAR(30),

    CONSTRAINT pk_PremierReferral   PRIMARY KEY (referralID),
    CONSTRAINT fk_PremierReferral_0
    FOREIGN KEY (customerID)
    REFERENCES Premier(customerID),
    CONSTRAINT fk_PremierReferral_1
    FOREIGN KEY (discountName)
    REFERENCES ReferralDiscounts(discountName)
);

CREATE TABLE MonthlyBill
(
    customerID          INTEGER NOT NULL,
    dateDue             DATE NOT NULL,
    referralID          INTEGER,

    CONSTRAINT pk_MonthlyBill PRIMARY KEY (customerID, dateDue),
    CONSTRAINT fk_MonthlyBill_0
    FOREIGN KEY (customerID)
    REFERENCES Premier(customerID),
    CONSTRAINT fk_MonthlyBill_1
    FOREIGN KEY (referralID)
    REFERENCES PremierReferral(referralID)
);

/*End of Customer-related Tables*/



/*Beginning of Vehicle-related Tables*/

CREATE TABLE MakeModel
(
    make                VARCHAR(30) NOT NULL,
    model               VARCHAR(30) NOT NULL,
    yearEdition         CHAR (4) NOT NULL,

    CONSTRAINT pk_MakeModel     PRIMARY KEY (make, model, yearEdition)
);
CREATE TABLE Vehicle
(
    VIN                CHAR(17) NOT NULL,
    licensePlate       CHAR(7) NOT NULL,
    accidentFlag       BOOLEAN,
    estimatedMileage   CHAR(6),
    make               VARCHAR(30),
    model              VARCHAR(30),
    yearEdition        CHAR (4),

    CONSTRAINT pk_Vehicle       PRIMARY KEY (licensePlate),
    CONSTRAINT ck_Vehicle       UNIQUE (VIN),
    CONSTRAINT fk_Vehicle
    FOREIGN KEY (make, model, yearEdition)
    REFERENCES MakeModel(make, model, yearEdition)
);

CREATE TABLE MaintenanceInterval
(
    mileage             CHAR(6) NOT NULL,
    make                VARCHAR(30) NOT NULL,
    model               VARCHAR(30) NOT NULL,
    yearEdition         CHAR(4) NOT NULL,
    cost                VARCHAR(30),

    CONSTRAINT pk_MaintenanceInterval   PRIMARY KEY (mileage, make, model,
yearEdition),
    CONSTRAINT fk_MaintenanceInterval
    FOREIGN KEY (make, model, yearEdition)
    REFERENCES MakeModel(make, model, yearEdition)
);


CREATE TABLE Notification
(
    dateTime            TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    customerID          INTEGER NOT NULL,
    licensePlate        CHAR(7) NOT NULL,
    milageToComeIn CHAR(6),
    mileageInterval     CHAR(6),
    make                VARCHAR(30),
    model               VARCHAR(30),
    yearEdition         CHAR(4),

    CONSTRAINT pk_Notification      PRIMARY KEY (dateTime, customerID,
licensePlate),
    CONSTRAINT fk_Notification_0
    FOREIGN KEY (customerID)
    REFERENCES Steady(customerID),
    CONSTRAINT fk_Notification_1
    FOREIGN KEY (licensePlate)
    REFERENCES Vehicle(licensePlate),
    CONSTRAINT fk_Notification_2
    FOREIGN KEY (mileageInterval, make, model, yearEdition)
    REFERENCES MaintenanceInterval(mileage, make, model, yearEdition)
```

```
);


/*END of Vehicle-related tables*/




/*Beginning of Employee-related tables*/

CREATE TABLE Employee
(
    employeeID          INTEGER NOT NULL AUTO_INCREMENT,
    firstName           VARCHAR(30) NOT NULL,
    lastName            VARCHAR(30) NOT NULL,
    eDateYearEmployed    DATE NOT NULL,
    eDateYearNotEmployed DATE,

    CONSTRAINT pk_Employee PRIMARY KEY (employeeID),
    CONSTRAINT ck_Employee UNIQUE (firstName, lastName, eDateYearEmployed)
);

CREATE TABLE Mechanic
(
    employeeID          INTEGER NOT NULL,
    mDateYearEmployed   DATE,
    mDateYearNotEmployed   DATE,


    CONSTRAINT pk_Mechanic PRIMARY KEY (employeeID),
    CONSTRAINT fk_Mechanic
    FOREIGN KEY (employeeID)
    REFERENCES Employee(employeeID)
);

CREATE TABLE ServiceTechnician
(
    employeeID              INTEGER NOT NULL,
    sDateYearEmployed       DATE,
    sDateYearNotEmployed    DATE,

    CONSTRAINT pk_ServiceTechnitian PRIMARY KEY (employeeID),
    CONSTRAINT fk_ServiceTechnitian
    FOREIGN KEY (employeeID)
    REFERENCES Employee(employeeID)
);

CREATE TABLE Certification
(
    certName        VARCHAR(30) NOT NULL,
    certSchool      VARCHAR(30) NOT NULL,

    CONSTRAINT pk_Certification PRIMARY KEY (certName, certSchool)
);

CREATE TABLE MechanicCertification
(
    employeeID          INTEGER NOT NULL,
    certName            VARCHAR(30) NOT NULL,
    certSchool          VARCHAR(30) NOT NULL,
    dateYear            DATE,

    CONSTRAINT pk_MechanicCertification PRIMARY KEY (employeeID, certName,
certSchool),
    CONSTRAINT fk_MechanicCertification_0
    FOREIGN KEY (employeeID)
    REFERENCES Mechanic(employeeID),
    CONSTRAINT fk_MechanicCertification_1
    FOREIGN KEY (certName, certSchool)
    REFERENCES Certification(certName, certSchool)
);

CREATE TABLE MentorRelation
(
    employeeID          INTEGER NOT NULL,
    certName            VARCHAR(30) NOT NULL,
    certSchool          VARCHAR(30) NOT NULL,
    empMenteeID         INTEGER NOT NULL,
    startDate           DATE NOT NULL,
    endDate             DATE,

    CONSTRAINT pk_MentorRelation PRIMARY KEY (employeeID, certName, certSchool,
empMenteeID, startDate),
    CONSTRAINT fk_MentorRelation_0
    FOREIGN KEY (empMenteeID)
    REFERENCES Mechanic(employeeID),
    CONSTRAINT fk_MentorRelation_1
    FOREIGN KEY (employeeID, certName, certSchool)
    REFERENCES MechanicCertification(employeeID, certName, certSchool)
);


/*END of Employee-related tables*/



/*BEGINNING OF ServiceVisit-related tables*/

CREATE TABLE ServiceVisit
(
    customerID          INTEGER NOT NULL,
    dateMonthYear       DATE NOT NULL,
    timeServiceVisit    TIME NOT NULL,
    licensePlate        CHAR(7) NOT NULL,
    mileageUpdate       VARCHAR(30),

    CONSTRAINT pk_ServiceVisit  PRIMARY KEY (customerID, dateMonthYear,
timeServiceVisit, licensePlate),
    CONSTRAINT ck_ServiceVisit  UNIQUE (customerID, dateMonthYear, licensePlate),
    CONSTRAINT fk_ServiceVisit_0
    FOREIGN KEY (customerID)
    REFERENCES Customer(customerID),
    CONSTRAINT fk_ServiceVisit_1
    FOREIGN KEY (licensePlate)
    REFERENCES Vehicle(licensePlate)
);


CREATE TABLE MaintenanceItem
(
    mItemName               VARCHAR(30) NOT NULL,
    mItemCost               VARCHAR(30),
    laborHourseEstimated    VARCHAR(30),
    certName                VARCHAR(30),
    certSchool              VARCHAR(30),

    CONSTRAINT pk_MaintenanceItem   PRIMARY KEY (mItemName),
    CONSTRAINT fk_Certification
    FOREIGN KEY (certName, certSchool)
    REFERENCES Certification(certName, certSchool)
);

CREATE TABLE MaintenanceIntervalLine
(
    mileage             CHAR(6) NOT NULL,
    make                VARCHAR(30) NOT NULL,
    model               VARCHAR(30) NOT NULL,
    yearEdition         CHAR(4) NOT NULL,
    mItemName           VARCHAR(30) NOT NULL,

    CONSTRAINT pk_MaintenanceIntervalLine PRIMARY KEY (mileage, make, model,
yearEdition, mItemName),
    CONSTRAINT fk_MaintenanceIntervalLine_0
    FOREIGN KEY (mileage, make, model, yearEdition)
    REFERENCES MaintenanceInterval(mileage, make, model, yearEdition),
    CONSTRAINT fk_MaintenaceIntervalLine_1
    FOREIGN KEY (mItemName)
    REFERENCES MaintenanceItem(mItemName)
);
```

```
CREATE TABLE MaintenanceLineItem
(
    customerID          INTEGER NOT NULL,
    dateMonthYear       DATE NOT NULL,
    timeServiceVisit    TIME NOT NULL,
    licensePlate        CHAR(7) NOT NULL,
    mItemName           VARCHAR(30) NOT NULL,
    quantity            INTEGER,
    laborMarkup         VARCHAR(30),

    CONSTRAINT pk_MaintenanceLineItem PRIMARY KEY (customerID, dateMonthYear,
timeServiceVisit, licensePlate, mItemName),
    CONSTRAINT fk_MaintenanceLineItem_0
    FOREIGN KEY (customerID, dateMonthYear, timeServiceVisit, licensePlate)
    REFERENCES ServiceVisit(customerID, dateMonthYear, timeServiceVisit,
licensePlate),
    CONSTRAINT fk_MaintenanceLineItem_1
    FOREIGN KEY (mItemName)
    REFERENCES MaintenanceItem(mItemName)
);

CREATE TABLE SingleItem
(
    mItemName           VARCHAR(30) NOT NULL,
    singleItemName      VARCHAR(30),

    CONSTRAINT pk_SingleItem PRIMARY KEY (mItemName),
    CONSTRAINT fk_SingleItem
    FOREIGN KEY (mItemName)
    REFERENCES MaintenanceItem(mItemName)
);

CREATE TABLE Package
(
    mItemName           VARCHAR(30) NOT NULL,
    packageName         VARCHAR(30),

    CONSTRAINT pk_Package PRIMARY KEY (mItemName),
    CONSTRAINT fk_Package
    FOREIGN KEY (mItemName)
    REFERENCES MaintenanceItem(mItemName)
);

CREATE TABLE PackageComp
(
    mItemNameP         VARCHAR(30) NOT NULL,
    mItemNameM         VARCHAR(30) NOT NULL,
    CONSTRAINT pk_PackageComp PRIMARY KEY (mItemNameP, mItemNameM),
    CONSTRAINT fk_PackageComp_0
    FOREIGN KEY (mItemNameP)
    REFERENCES Package(mItemName),
    CONSTRAINT fk_PackageComp_1
    FOREIGN KEY (mItemNameM)
    REFERENCES MaintenanceItem(mItemName)
);

CREATE TABLE Supplier
(
    supplierName        VARCHAR(30) NOT NULL,
    supplierAddress     VARCHAR(30) NOT NULL,
    supplierCity        VARCHAR(30) NOT NULL,
    supplierState       VARCHAR(30) NOT NULL,
    supplierZipcode     CHAR(5) NOT NULL,
    supplierPhone       VARCHAR(10) NOT NULL,

    CONSTRAINT pk_Supplier  PRIMARY KEY (supplierName),
    CONSTRAINT ck_Supplier  UNIQUE (supplierAddress, supplierCity, supplierState,
supplierZipcode, supplierPhone)
);

CREATE TABLE Product
(
    prodID              INTEGER NOT NULL AUTO_INCREMENT,
    prodName            VARCHAR(30),
    prodModel           VARCHAR(30),
    prodMake            VARCHAR(30),
    prodCost            VARCHAR(30),

    CONSTRAINT pk_Product PRIMARY KEY (prodID)
);

CREATE TABLE ProductSupplier
(
    supplierName        VARCHAR(30) NOT NULL,
    prodID              INTEGER NOT NULL,
    dateSupplied        DATE,

    CONSTRAINT pk_ProductSupplier  PRIMARY KEY (supplierName, prodID),
    CONSTRAINT fk_ProductSupplier_0
    FOREIGN KEY (supplierName)
    REFERENCES Supplier(supplierName),
    CONSTRAINT fk_ProductSupplier_1
    FOREIGN KEY (prodID)
    REFERENCES Product(prodID)
);

CREATE TABLE OrderLineProduct
```

```
(
    mItemName           VARCHAR(30) NOT NULL,
    supplierName        VARCHAR(30) NOT NULL,
    prodID              INTEGER NOT NULL,

    CONSTRAINT pk_OrderLineProduct PRIMARY KEY (mItemName, supplierName, prodID),
    CONSTRAINT fk_OrderLineProduct_0
    FOREIGN KEY (mItemName)
    REFERENCES MaintenanceItem(mItemName),
    CONSTRAINT fk_OrderLineProduct_1
    FOREIGN KEY (prodID)
    REFERENCES Product(prodID)
);


/*END OF ServiceVisit-related tables*/


--DML for Employee:---------------------------------------------------
insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Karen', 'Vo', '2013-02-13', '2014-05-20');

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('William', 'Lake', ‘2001-06-28', ‘2002-01-04’);

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Lisa', 'Kim', ‘2015-04-21', ‘2015-10-09');

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Helena', 'Reyes', '2012-03-01', NULL);

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Anthony', 'Johnson', '2004-05-15', '2014-05-20');

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Anderson', 'Davis', '2007-03-03', NULL);

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
```

```
values ('Christina', 'Taylor', '2003-12-13', NULL);

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Kristy', 'Song', '2001-05-23', NULL);

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Roberto', 'Martinez', '2012-11-10', '2012-12-26');

insert into Employee (firstName, lastName, eDateYearEmployed,
eDateYearNotEmployed)
values ('Jackson', 'Clark', '2002-08-23', '2005-08-20');


UPDATE Employee
SET Salary=20342.12
WHERE EmployeeID=2;

UPDATE Employee
SET Salary=124320.29
WHERE EmployeeID=3;

UPDATE Employee
SET Salary=23406.03
WHERE EmployeeID=4;

UPDATE Employee
SET Salary=56754.79
WHERE EmployeeID=5;

UPDATE Employee
SET Salary=76644.43
WHERE EmployeeID=6;

UPDATE Employee
SET Salary=93834.66
WHERE EmployeeID=7;

UPDATE Employee
SET Salary=34293.55
WHERE EmployeeID=8;
```

```
UPDATE Employee
SET Salary=43543.34
WHERE EmployeeID=9;


UPDATE Employee
SET Salary=53455.23
WHERE EmployeeID=10;


//DML for Mechanic and ServiceTechnician:----------------------------
insert into Mechanic values (1, '2013-02-13', '2014-05-20');
insert into Mechanic values (2, '2001-06-28', '2005-01-04');
insert into Mechanic values (3, '2015-04-21', '2015-10-09');
insert into Mechanic values (4, '2012-03-01', NULL);
insert into Mechanic values (5, '2004-05-15', '2008-06-02');
insert into Mechanic values (10, '2002-08-23', '2005-08-20');

insert into ServiceTechnician values (6, '2007-03-03', '2009-08-09');
insert into ServiceTechnician values (7, '2003-12-13', '2012-12-15');
insert into ServiceTechnician values (8, '2001-05-23', NULL);
insert into ServiceTechnician values (9, '2012-11-10', '2012-12-26');

insert into Mechanic values (6, '2009-08-09', NULL);
insert into Mechanic values (7, '2013-11-15', NULL);
insert into ServiceTechnician values (5, '2008-07-02', '2014-05-20');

insert into ServiceTechnician values (4, '2015-12-28', NULL);
--this Mechanic is a Mechanic as well as a ServiceTechnician
```

```
INSERT INTO MakeModel VALUES ('Toyota', 'Paseo', '1996');
INSERT INTO MakeModel VALUES ('Toyota', 'Camry', '1996');
INSERT INTO MakeModel VALUES ('Toyota', 'Camry', '2005');
INSERT INTO MakeModel VALUES ('Hyundai', 'Elantra', '2013');
INSERT INTO MakeModel VALUES ('Kia', 'Forte', '2013');
INSERT INTO MakeModel VALUES ('Mercedes', 'c300', '2015');

INSERT INTO Vehicle VALUES ('FLJBB25V324LB5IF1', 'LARL3KW', TRUE,
'000800', 'Toyota', 'Paseo', '1996');
INSERT INTO Vehicle VALUES ('BPN9A6316NNXBV4KF', 'MH81YMA', FALSE,
'003456', 'Toyota', 'Camry', '1996');
INSERT INTO Vehicle VALUES ('N1WV4UTUQB4R5KKCG', 'I60PF11', FALSE,
'000321', 'Toyota', 'Camry', '2005');
INSERT INTO Vehicle VALUES ('62YI54A09X2Y6CFV8', '9PPW6QU', FALSE,
'009370', 'Hyundai', 'Elantra', '2013');
INSERT INTO Vehicle VALUES ('OD0ZSJ7BBVH24DFN2', '0ZNT7W3', FALSE,
'002432', 'Kia', 'Forte', '2013');
INSERT INTO Vehicle VALUES ('PPXGRQFGCG8CE8JRI', 'HDI9IQ7', TRUE,
'000101', 'Mercedes', 'c300', '2015');
INSERT INTO Vehicle VALUES ('C7M2J3BHEY0G3OSQ5', 'ENAM55M', FALSE,
'002002', 'Mercedes', 'c300', '2015');

INSERT INTO MaintenanceInterval VALUES ('010000', 'Toyota', 'Paseo',
'1996', '$300.00');
INSERT INTO MaintenanceInterval VALUES ('020000', 'Mercedes', 'c300',
'2015', '$600.00');
INSERT INTO MaintenanceInterval VALUES ('010000', 'Toyota', 'Camry',
'1996', '$200.00');
INSERT INTO MaintenanceInterval VALUES ('005000', 'Toyota', 'Camry',
'2005', '$200.00');
INSERT INTO MaintenanceInterval VALUES ('005000', 'Hyundai',
'Elantra', '2013', '$600.00');
INSERT INTO MaintenanceInterval VALUES ('010000', 'Kia', 'Forte',
'2013', '$400.00');

INSERT INTO Supplier VALUES ('Mike''s Auto Stuff', '335 Park Ave.',
'Long Beach', 'California', '90803', '5626826174');
```

```
INSERT INTO Supplier VALUES ('Greg''s Big Wheels', '123 Sesame St.',
'Las Vegas', 'Nevada', '12345', '2344562341');
INSERT INTO Supplier VALUES ('Syed''s Da Plug', '666 Hell St.', 'San
Francisco', 'California', '77777', '3452359685');
INSERT INTO Supplier VALUES ('Sarah''s Car Supplies', '745 Orange
St.', 'Lakewood', 'California', '90706', '2345434545');
INSERT INTO cecs323bg1.Supplier VALUES ('Autozone','6786 Vehicle
Dr.','Queens','New York','93122','2093231234');
INSERT INTO cecs323bg1.Supplier VALUES ('PepBoys','7126 Violet
St.','Sacramento','California','91102','4325740123');
INSERT INTO Supplier VALUES ('Autozone','6786 Vehicle
Dr.','Queens','New York','93122','2093231234');
INSERT INTO Supplier VALUES ('PepBoys','7126 Violet
St.','Sacramento','California','91102','4325740123');
```

```
INSERT INTO Product (prodName, prodModel, prodMake, prodCost) VALUES
('Engine Starter', '4 Cylinder', 'Dyson','$80.00');
INSERT INTO Product (prodName, prodModel, prodMake, prodCost) VALUES
('Water Pump', 'X-400', 'Dyson', '$60.00');
INSERT INTO Product (prodName, prodModel, prodMake, prodCost) VALUES
('Timing Belt', null, 'Tom''s', '$20.00');
INSERT INTO Product (prodName, prodModel, prodMake, prodCost) VALUES
('Oil', '500', 'Penzoil', '$7.00');
INSERT INTO Product (prodName, prodModel, prodMake, prodCost) VALUES
('Enginer Starter', '5 Cylinder', 'Viper', '$100.00');
```

```
/*NEED TO ADD IN DATES*\
INSERT INTO ProductSupplier VALUES ('Mike''s Auto Stuff', 1);
INSERT INTO ProductSupplier VALUES ('Mike''s Auto Stuff', 2);
INSERT INTO ProductSupplier VALUES ('Mike''s Auto Stuff', 3);
INSERT INTO ProductSupplier VALUES ('Mike''s Auto Stuff', 4);
INSERT INTO ProductSupplier VALUES ('Mike''s Auto Stuff', 5);
INSERT INTO ProductSupplier VALUES ('Greg''s Big Wheels', 1);
INSERT INTO ProductSupplier VALUES ('Greg''s Big Wheels', 2);
INSERT INTO ProductSupplier VALUES ('Greg''s Big Wheels', 3);
INSERT INTO ProductSupplier VALUES ('Greg''s Big Wheels', 4);
INSERT INTO ProductSupplier VALUES ('Syed''s Da Plug', 1);
INSERT INTO ProductSupplier VALUES ('Syed''s Da Plug', 2);
INSERT INTO ProductSupplier VALUES ('Syed''s Da Plug', 3);
INSERT INTO ProductSupplier VALUES ('Sarah''s Car Supplies', 1);
INSERT INTO ProductSupplier VALUES ('Sarah''s Car Supplies', 4);
INSERT INTO ProductSupplier VALUES ('Sarah''s Car Supplies', 5);
INSERT INTO ProductSupplier VALUES ('Autozone',7,'2016-10-01');
INSERT INTO ProductSupplier VALUES ('Autozone',6,'2015-12-30');
INSERT INTO ProductSupplier VALUES ('PepBoys',5,'2016-01-23');
```

```
//DML FOR Customer

INSERT INTO Customer(phone,email) VALUES
('7140122345','syedhaider@gmail.com');
INSERT INTO Customer(phone,email) VALUES
('5621234567','michaelblack@gmail.com');
INSERT INTO Customer(phone,email) VALUES
('3232345678','sarahhan@gmail.com');
INSERT INTO Customer(phone,email) VALUES
('3103456789','gregviolan@gmail.com');

INSERT INTO Customer(phone,email) VALUES
('1234567890','davidbrown@gmail.com');
INSERT INTO Customer(phone,email) VALUES
('2345677890','johndoe@gmail.com');
INSERT INTO Customer(phone,email) VALUES
('3457890123','janedoe@gmail.com');
INSERT INTO Customer(phone,email) VALUES
('5678901234','mikesmith@gmail.com');
```

```
INSERT INTO Customer(phone,email) VALUES
('3102343425','bobsaget@gmail.com');
INSERT INTO Customer(phone,email) VALUES
('3423421234','mikewowski@gmail.com');

INSERT INTO Referral (customerID,dateReferred,timeReferred) VALUES
(1,'2016-02-16','12:12:12');
INSERT INTO Referral (customerID,dateReferred,timeReferred) VALUES
(2,'2015-03-15','11:11:01');

INSERT INTO Steady (CustomerID,firstServiceDate) VALUES
(1,'2015/07/29');
INSERT INTO Steady (customerID,firstServiceDate) VALUES
(3,'2013/10/03');
INSERT INTO Steady (customerID,firstServiceDate) VALUES
(4,'2014/08/11');

INSERT INTO Contact VALUES (5,'2014/09/04');
INSERT INTO Contact VALUES (6,'2012/05/07');

INSERT INTO Individual VALUES(1,'Syed','Haider','8162 4th St.','Buena
Park','California','90621','1996-04-22');
INSERT INTO Individual VALUES(2,'Michael','Black','1250 Bellflower
Blvd','Long Beach','California','90540','1996-05-19');
INSERT INTO Individual VALUES(3,'Sarah','Han','1227 Burnett
St','Signal Hill','California','90210','1994-02-14');
INSERT INTO Individual VALUES(5,'David','Brown','5344 Westwood
St','Westwood','California','94302','1985-11-23');
INSERT INTO Individual VALUES(7,'Jane','Doe','3424 Dobbey
Dr','Cerritos','California','91024','1990-05-12');

INSERT INTO Corporate VALUES (4,'Progressive Insurance');
INSERT INTO Corporate VALUES (6,'Geico');
INSERT INTO Corporate VALUES (8,'State Farm');

INSERT INTO Premier VALUES (2,'2013-12-31',NULL);
INSERT INTO Premier VALUES (7,'2010-04-12',NULL);
INSERT INTO Premier VALUES (8,'2013-05-18','2015-03-16');

INSERT INTO ReferralDiscounts VALUES ('Free Oil Change');
INSERT INTO ReferralDiscounts VALUES('50$OffMonthlyBill');
INSERT INTO SteadyReferral VALUES (3,1,'Free Oil Change');

INSERT INTO PremierReferral VALUES (4,2,'50$OffMonthlyBill');

INSERT INTO MonthlyBill VALUES (2,'2016-03-31',4);
INSERT INTO MonthlyBill VALUES (2,'2016-04-30',NULL);
INSERT INTO MonthlyBill VALUES (2,'2016-05-31',NULL);
UPDATE MonthlyBill
SET dateDue = '2016-05-30'
WHERE dateDue = '2016-04-30';
INSERT INTO MonthlyBill VALUES (7,'2016-11-22',NULL);
INSERT INTO MonthlyBill VALUES (7,'2016-12-22',NULL);
INSERT INTO MonthlyBill VALUES (7,'2017-01-22',NULL);
```

```
INSERT INTO Certification VALUES ('Proficient in Brakes', 'University
of Brakes');
INSERT INTO Certification VALUES ('Proficient in Tires', 'University
of Tires');
INSERT INTO Certification VALUES ('Proficient in Engines', 'Auto
University');
INSERT INTO Certification VALUES ('Oil Changing Ability', 'Auto
University');
INSERT INTO Certification VALUES ('General Auto Proficiency',
'CSULB');

INSERT INTO MechanicCertification VALUES (1, 'General Auto
Proficiency', 'CSULB', '2000-02-02');
INSERT INTO MechanicCertification VALUES (2, 'General Auto
Proficiency', 'CSULB', '2001-01-01');
INSERT INTO MechanicCertification VALUES (3, 'General Auto
Proficiency', 'CSULB', '2002-02-02');
INSERT INTO MechanicCertification VALUES (4, 'General Auto
Proficiency', 'CSULB', '2004-04-04');
INSERT INTO MechanicCertification VALUES (5, 'General Auto
Proficiency', 'CSULB', '2005-05-05');
```

```
INSERT INTO MechanicCertification VALUES (6, 'General Auto
Proficiency', 'CSULB', '2006-06-06');
INSERT INTO MechanicCertification VALUES (7, 'General Auto
Proficiency', 'CSULB', '2007-07-07');
INSERT INTO MechanicCertification VALUES (10, 'General Auto
Proficiency', 'CSULB', '2010-10-10');
INSERT INTO MechanicCertification VALUES (1, 'Proficient in Brakes',
'University of Brakes', '2010-10-10');
INSERT INTO MechanicCertification VALUES (1, 'Proficient in Tires',
'University of Tires', '2009-09-09');
INSERT INTO MechanicCertification VALUES (1, 'Proficient in Engines',
'Auto University', '2008-08-08');
INSERT INTO MechanicCertification VALUES (1, 'Oil Changing Ability',
'Auto University', '2007-07-07');
INSERT INTO MechanicCertification VALUES (2, 'Proficient in Tires',
'University of Tires', '2009-09-09');
INSERT INTO MechanicCertification VALUES (2, 'Proficient in Engines',
'Auto University', '2008-08-08');
INSERT INTO MechanicCertification VALUES (2, 'Oil Changing Ability',
'Auto University', '2007-07-07');
INSERT INTO MechanicCertification VALUES (3, 'Proficient in Engines',
'Auto University', '2008-08-08');
INSERT INTO MechanicCertification VALUES (3, 'Oil Changing Ability',
'Auto University', '2007-07-07');
INSERT INTO MechanicCertification VALUES (4, 'Oil Changing Ability',
'Auto University', '2007-07-07');

INSERT INTO MentorRelation VALUES (1, 'Proficient in Brakes',
'University of Brakes', 10, '2016-11-28', null);
INSERT INTO MentorRelation VALUES (2, 'Proficient in Tires',
'University of Tires', 10, '2015-10-31', '2016-10-31');
INSERT INTO MentorRelation VALUES (2, 'Proficient in Engines', 'Auto
University', 6, '2016-09-24', null);
INSERT INTO MentorRelation VALUES (1, 'Proficient in Engines', 'Auto
University', 6, '2016-09-24', null);
INSERT INTO MentorRelation VALUES (3, 'Oil Changing Ability', 'Auto
University', 6, '2014-09-24', '2015-09-24');
INSERT INTO MentorRelation VALUES (3, 'Oil Changing Ability', 'Auto
University', 10, '2014-09-24', '2015-09-24');

INSERT INTO ServiceVisit VALUES (1, '2014-09-24', '10:29:07',
'MH81YMA', '123456');
```

```
INSERT INTO ServiceVisit VALUES (2, '2016-07-05', '11:35:15',
'LARL3KW', '197800');
INSERT INTO ServiceVisit VALUES (3, '2016-08-24', '12:00:00',
'9PPW6QU', '019370');
INSERT INTO ServiceVisit VALUES (6, '2016-10-27', '10:45:00',
'0ZNT7W3', '012432');
INSERT INTO ServiceVisit VALUES (1, '2016-11-28', '15:30:00',
'MH81YMA', '129964');

INSERT INTO MaintenanceItem VALUES ('Change Oil', '$60.00', '0.5
hrs', 'Oil Changing Ability', 'Auto University');
INSERT INTO MaintenanceItem VALUES ('Change Breaks', '$300.00', '2
hrs', 'Proficient in Brakes', 'University of Brakes');
INSERT INTO MaintenanceItem VALUES ('Install Engine Starter',
'$80.00', '1 hrs', 'Proficient in Engines', 'Auto University');
INSERT INTO MaintenanceItem VALUES ('Rotate Tire', '$50.00', '0.25
hrs', 'Proficient in Tires', 'University of Tires');
INSERT INTO MaintenanceItem VALUES ('Replace Driver''s Door',
'$600.00', '2 hrs', 'General Auto Proficiency', 'CSULB');
INSERT INTO MaintenanceItem VALUES ('Replace Water Pump', '$350.00',
'1.5 hrs', 'General Auto Proficiency', 'CSULB');

INSERT INTO MaintenanceIntervalLine VALUES ('010000', 'Toyota',
'Camry', '1996', 'Change Oil');
INSERT INTO MaintenanceIntervalLine VALUES ('010000', 'Toyota',
'Paseo', '1996', 'Install Engine Starter');
INSERT INTO MaintenanceIntervalLine VALUES ('010000', 'Toyota',
'Paseo', '1996', 'Replace Water Pump');
INSERT INTO MaintenanceIntervalLine VALUES ('005000', 'Hyundai',
'Elantra', '2013', 'Change Breaks');
INSERT INTO MaintenanceIntervalLine VALUES ('005000', 'Hyundai',
'Elantra', '2013', 'Rotate Tire');
INSERT INTO MaintenanceIntervalLine VALUES ('010000', 'Kia', 'Forte',
'2013', 'Replace Driver''s Door');

INSERT INTO MaintenanceLineItem VALUES (1, '2014-09-24', '10:29:07',
'MH81YMA', 'Change Oil', 1, '$10.00');
INSERT INTO MaintenanceLineItem VALUES (1, '2016-11-28', '15:30:00',
'MH81YMA', 'Change Oil', 1, '$10.00');
INSERT INTO MaintenanceLineItem VALUES (2, '2016-07-05', '11:35:15',
'LARL3KW', 'Install Engine Starter', 1, '$40.00');
INSERT INTO MaintenanceLineItem VALUES (2, '2016-07-05', '11:35:15',
'LARL3KW', 'Replace Water Pump', 1, '$40.00');
```

```
INSERT INTO MaintenanceLineItem VALUES (3, '2016-08-24', '12:00:00',
'9PPW6QU', 'Rotate Tire', 4, '$10.00');
INSERT INTO MaintenanceLineItem VALUES (6, '2016-10-27', '10:45:00',
'0ZNT7W3', 'Replace Driver''s Door', 1, '$50.00');

INSERT INTO Notification VALUES (NOW(), 1, 'MH81YMA', '133456',
'010000', 'Toyota', 'Camry', '1996');
INSERT INTO Notification VALUES (NOW(), 3, '9PPW6QU', '024370',
'005000', 'Hyundai', 'Elantra', '2013');
INSERT INTO Notification VALUES (NOW(), 4, 'I60PF11', null, '005000',
'Toyota', 'Camry', '2005');

INSERT INTO OrderLineProduct VALUES ('Change Oil', 'Mike''s Auto
Stuff', 4, 1);
INSERT INTO OrderLineProduct VALUES ('Install Engine Starter',
'Sarah''s Car Supplies', 1, 1);
INSERT INTO OrderLineProduct VALUES ('Replace Water Pump', 'Mike''s
Auto Stuff', 2, 1);
INSERT INTO OrderLineProduct VALUES ('Rotate Tire', 'Greg''s Big
Wheels', 6, 4);
INSERT INTO OrderLineProduct VALUES ('Replace Driver''s Door',
'Mike''s Auto Stuff', 7, 1);



INSERT INTO SingleItem VALUES ('Change Oil', 'Dave''s Oil Change');
INSERT INTO SingleItem VALUES ('Replace Driver''s Door', 'Car Door
Replacement');
INSERT INTO SingleItem VALUES ('Rotate Tire', 'Rotate Tires');

INSERT INTO Package VALUES ('Paseo 10000 Mile Checkup', 'Paseo Yo
Oponents');

INSERT INTO PackageComp VALUES ('Paseo 10000 Mile Checkup', 'Replace
Water Pump');
INSERT INTO PackageComp VALUES ('Paseo 10000 Mile Checkup', 'Install
Engine Starter');

INSERT INTO Address VALUES (4, ‘Mailing’, '455 Conch St.', 'Long
Beach', 'California', '90803');
INSERT INTO Address VALUES (4, ‘Billing’, '345 Computer St.',
'Austin', 'Texas', '34523');
```

```
INSERT INTO Address VALUES (6, ‘Mailing’, '543 Orange Ave.', 'Xenia',
'Ohio', '23423');
INSERT INTO Address VALUES (6, ‘Billing’, '758 Beach St.', 'Long
Beach', 'California', '90803');
INSERT INTO Address VALUES (8, ‘Mailing’, '865 Bellflower',
'Lakewood', 'Califonia', '90706');
INSERT INTO Address VALUES (8, ‘Billing’, '987 Lakewood',
'Bellflower', 'California', '90814');
```

VIEWS:

```
--VIEW 1

CREATE VIEW customer_v AS
SELECT distinct Individual.customerID, Individual.lastName as
LASTNAME, Individual.firstName as FIRSTNAME, 'PREMIER' AS "TYPE",
IF(endDate = null, floor(datediff(endDate,startDate) / 365),
floor(datediff(curdate(),startDate) / 365)) as YEARS
FROM Premier
inner join Individual using(customerID)
UNION ALL
SELECT distinct Individual.customerID, Individual.lastName as
LASTNAME, Individual.firstName as FIRSTNAME, 'STEADY' AS "TYPE",
floor(datediff(curdate(),firstServiceDate) / 365) AS YEARS
FROM Steady
inner join Individual using(customerID)
UNION ALL
SELECT distinct Individual.customerID, Individual.lastName as
LASTNAME, Individual.firstName as FIRSTNAME, 'PROSPECTIVE' AS "TYPE",
null AS YEARS
FROM Prospective
inner join Individual using(customerID)


UNION ALL
/*Corporate*/
SELECT distinct Premier.customerID, Corporate.corpName AS LASTNAME,
"" AS FIRSTNAME, 'PREMIER' AS "TYPE",
floor(datediff(endDate,startDate) / 365) AS YEARS
FROM Premier
inner join Corporate using(customerID)
UNION ALL
SELECT distinct Steady.customerID,  Corporate.corpName AS LASTNAME,
"" AS FIRSTNAME, 'STEADY' AS "TYPE",
floor(datediff(curdate(),firstServiceDate) / 365) AS YEARS
FROM Steady
inner join Corporate using(customerID)
UNION ALL
SELECT distinct Prospective.customerID, Corporate.corpName AS
LASTNAME, "" AS FIRSTNAME, 'PROSPECTIVE' AS "TYPE",
null AS YEARS
FROM Prospective
inner join Corporate using(customerID);

--VIEW 2
CREATE VIEW Customer_addresses_v AS
    SELECT lastName AS name, firstName, 'Individual' AS customerType,
'Mailing' AS addressType,
    mailingAddress AS address, mailingCity AS city, mailingState as
state, zipcode
    FROM Individual

    UNION ALL

    SELECT corpName AS name, '' AS firstName, 'Corporate' AS
customerType, addressType, address, city, state, zipcode
    FROM Corporate INNER JOIN Address USING (customerID) ORDER BY
customerType DESC, name;

--VIEW 3
CREATE VIEW Mechanic_mentor_v AS
    SELECT employeeID.lastName AS mentorLastName,
employeeID.firstName AS mentorFirstName,
    empMenteeID.lastName AS menteeLastName, empMenteeID.firstName AS
menteeFirstName
    FROM MentorRelation
    INNER JOIN Employee employeeID ON MentorRelation.employeeID =
employeeID.employeeID
    INNER JOIN Employee empMenteeID ON MentorRelation.empMenteeID =
empMenteeID.employeeID
    ORDER BY mentorLastName, menteeLastName;
```

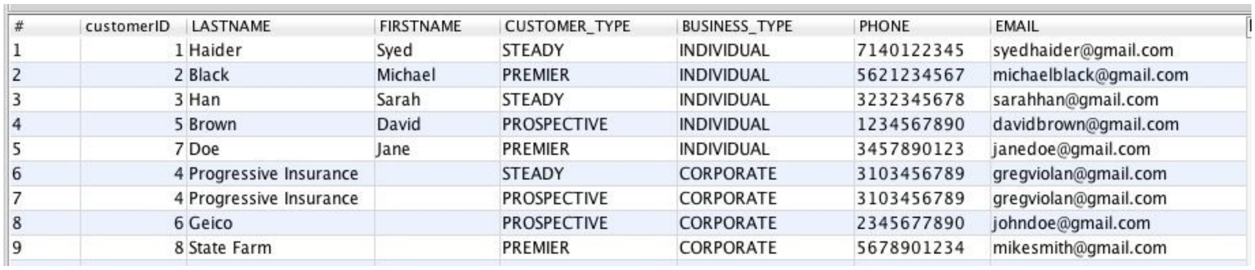
```
--VIEW 4
CREATE VIEW Premier_profits_v AS

SELECT customerID, IF(accidentFlag = 1, (estimatedMileage/100 * 10) +
200,
estimatedMileage/100 * 10) AS PremierPayment,
(SUM(CAST(SUBSTRING(laborMarkup,2) AS DECIMAL (10,2))) +
SUM(CAST(SUBSTRING(mItemCost,2) AS DECIMAL (10,2)))) AS SteadyPayment
FROM Vehicle
INNER JOIN ServiceVisit USING(licensePlate)
INNER JOIN Premier USING(customerID)
INNER JOIN MaintenanceLineItem USING(customerID)
INNER JOIN MaintenanceItem USING(mItemName)
GROUP BY customerID;



--5 view
CREATE VIEW Prospective_resurrection_v AS
    SELECT lastName as name, firstName, 'individual' as customerType
FROM Individual ind WHERE 3 <=
        (SELECT COUNT(*) FROM Contact con1 WHERE ind.customerID =
con1.customerID)
            AND customerID IN
        (SELECT customerID FROM Contact c1 WHERE
            (SELECT MAX(dateContacted) FROM Contact c2 WHERE
c1.customerID = c2.customerID)
        < DATE_SUB(NOW(), INTERVAL 1 YEAR))

    UNION ALL

    SELECT corpName as name, '' as firstName, 'corporate' as
customerType FROM Corporate cor WHERE 3 <=
        (SELECT COUNT(*) FROM Contact con2 WHERE cor.customerID =
con2.customerID)
            AND customerID IN
        (SELECT customerID FROM Contact c3 WHERE
            (SELECT MAX(dateContacted) FROM Contact c4 WHERE
c3.customerID = c4.customerID)
        < DATE_SUB(NOW(),INTERVAL 1 YEAR));
```

QUERIES:

--#1

```
SELECT customerID, customer_v.LASTNAME, customer_v.FIRSTNAME, TYPE AS
CUSTOMER_TYPE, 'INDIVIDUAL' AS BUSINESS_TYPE, phone as PHONE, email
as EMAIL
FROM customer_v INNER JOIN Customer USING(customerID)
INNER JOIN Individual USING(customerID)
UNION ALL
SELECT customerID, customer_v.LASTNAME, customer_v.FIRSTNAME, TYPE,
'CORPORATE' AS BUSINESS_TYPE, phone, email
FROM customer_v INNER JOIN Customer USING(customerID)
INNER JOIN Corporate USING(customerID);
```

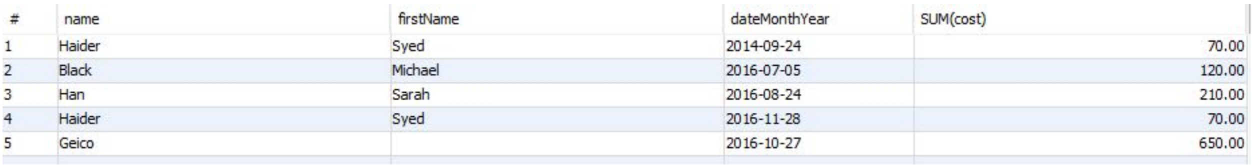
--#1 Sample Output:



#	customerID	LASTNAME	FIRSTNAME	CUSTOMER_TYPE	BUSINESS_TYPE	PHONE	EMAIL
1	1	Haider	Syed	STEADY	INDIVIDUAL	7140122345	syedhaider@gmail.com
2	2	Black	Michael	PREMIER	INDIVIDUAL	5621234567	michaelblack@gmail.com
3	3	Han	Sarah	STEADY	INDIVIDUAL	3232345678	sarahhan@gmail.com
4	5	Brown	David	PROSPECTIVE	INDIVIDUAL	1234567890	davidbrown@gmail.com
5	7	Doe	Jane	PREMIER	INDIVIDUAL	3457890123	janedoe@gmail.com
6	4	Progressive Insurance		STEADY	CORPORATE	3103456789	gregviolan@gmail.com
7	4	Progressive Insurance		PROSPECTIVE	CORPORATE	3103456789	gregviolan@gmail.com
8	6	Geico		PROSPECTIVE	CORPORATE	2345677890	johndoe@gmail.com
9	8	State Farm		PREMIER	CORPORATE	5678901234	mikesmith@gmail.com

```
--#2
SELECT lastName as name, firstName, dateMonthYear, SUM(cost) FROM
    (SELECT lastName, firstName, dateMonthYear, (SELECT
CAST(SUBSTRING(mItemCost, 2) AS DECIMAL(10, 2))
    * (CAST(quantity AS DECIMAL(10,0))) +
(CAST(SUBSTRING(laborMarkup, 2) AS DECIMAL(10,2))))
```

```
    AS cost FROM
    Individual INNER JOIN ServiceVisit USING (customerID) INNER JOIN
    MaintenanceLineItem USING (customerID, dateMonthYear,
timeServiceVisit) INNER JOIN
    MaintenanceItem USING (mItemName)group by lastName,
dateMonthYear)
    T group by dateMonthYear

UNION ALL

SELECT corpName as name, '' as firstName, dateMonthYear, SUM(cost)
FROM
    (SELECT corpName, dateMonthYear, (SELECT
CAST(SUBSTRING(mItemCost, 2) AS DECIMAL(10, 2))
    * (CAST(quantity AS DECIMAL(10,0))) +
(CAST(SUBSTRING(laborMarkup, 2) AS DECIMAL(10,2))))
    AS cost FROM
    Corporate INNER JOIN ServiceVisit USING (customerID) INNER JOIN
    MaintenanceLineItem USING (customerID, dateMonthYear,
timeServiceVisit) INNER JOIN
    MaintenanceItem USING (mItemName)group by corpName,
dateMonthYear)
    T group by dateMonthYear;
```

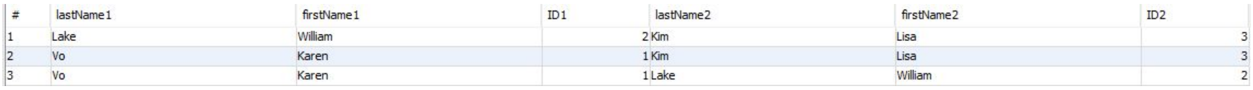


#	name	firstName	dateMonthYear	SUM(cost)
1	Haider	Syed	2014-09-24	70.00
2	Black	Michael	2016-07-05	120.00
3	Han	Sarah	2016-08-24	210.00
4	Haider	Syed	2016-11-28	70.00
5	Geico		2016-10-27	650.00

```
--3. List the top three customers in terms of their net spending for
the past two years, and the total that they have spent in that
period.
SELECT DISTINCT customer_v.customerID, customer_v.LASTNAME,
customer_v.FIRSTNAME, customer_v.TYPE,
CASE TYPE
    WHEN 'PREMIER' THEN IF(accidentFlag = 1, (estimatedMileage/100 *
10) + 200,
estimatedMileage/100 * 10)*2
    WHEN 'STEADY' THEN IF((ServiceVisit.timeServiceVisit >=
CURRENT_DATE - INTERVAL 2 YEAR), (SUM(CAST(SUBSTRING(laborMarkup,2)
AS DECIMAL (10,2))) +
SUM(CAST(SUBSTRING(mItemCost,2) AS DECIMAL (10,2)))), 0)
    ELSE 0
```

```
END AS NET_SPENDING
FROM customer_v, Premier
INNER JOIN ServiceVisit USING(customerID)
INNER JOIN Vehicle USING(licensePlate)
INNER JOIN MaintenanceLineItem USING(customerID)
INNER JOIN MaintenanceItem USING(mItemName)
GROUP BY customerID
ORDER BY NET_SPENDING DESC;
```

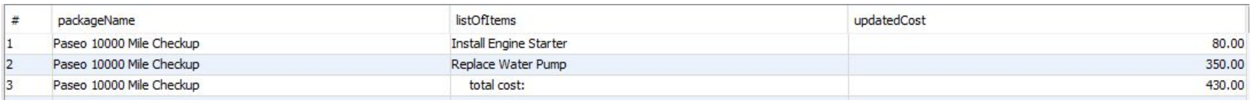
```
--4. Find all of the mechanics who have three or more skills.

SELECT firstName,lastName,COUNT(certName) AS "NumberOfSkills"
FROM cecs323bg1.MechanicCertification INNER JOIN cecs323bg1.Employee
using (employeeID)
GROUP BY firstName,lastName
HAVING COUNT(certName)>=3;

--#5
select a.lastName as lastName1, a.firstName as firstName1,
a.employeeID as ID1,
    b.lastName as lastName2, b.firstName as firstName2, b.employeeID
as ID2,
    count(*) as count from (Employee a inner join
    MechanicCertification ma on (a.employeeID = ma.employeeID)) inner
join (Employee b
    inner join MechanicCertification mb on (b.employeeID =
mb.employeeID)) on (ma.certName = mb.certName)
    where (a.employeeID != b.employeeID) and (a.employeeID <
b.employeeID) group by a.lastName, b.lastName having count > 2;
```



#	lastName1	firstName1	ID1	lastName2	firstName2	ID2
1	Lake	William	2	Kim	Lisa	3
2	Vo	Karen	1	Kim	Lisa	3
3	Vo	Karen	1	Lake	William	2

```
--#6
SELECT mItemNameP AS packageName, mItemNameM AS listOfItems, SUM(cost)
AS updatedCost
FROM (select mItemNameP, mItemNameM, (SELECT CAST(SUBSTRING(mItemCost,
2) AS DECIMAL(10, 2)))
AS cost FROM PackageComp p INNER JOIN MaintenanceItem m ON
(p.mItemNameM = m.mItemName)) T GROUP BY mItemNameM
```

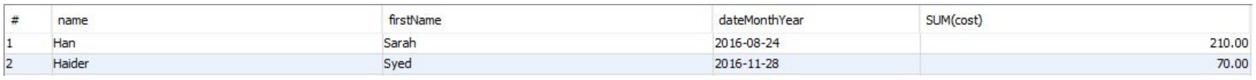
```
UNION ALL

SELECT mItemNameP AS packageName, '     total cost:' AS listOfItems,
SUM(cost) AS updatedCost
FROM (SELECT mItemNameP, mItemNameM, (SELECT CAST(SUBSTRING(mItemCost,
2) AS DECIMAL(10, 2)))
AS cost FROM PackageComp p INNER JOIN MaintenanceItem m ON
(p.mItemNameM = m.mItemName) GROUP BY mItemNameP, mItemNameM) T;
```



#	packageName	listOfItems	updatedCost
1	Paseo 10000 Mile Checkup	Install Engine Starter	80.00
2	Paseo 10000 Mile Checkup	Replace Water Pump	350.00
3	Paseo 10000 Mile Checkup	total cost:	430.00

```
--#9 The premier customers and the difference between what they have
paid in the past year, versus the services that they actually used
during that same time. List from the customers with the largest
difference to the smallest.

select customerID, PremierPayment - IF(ServiceVisit.timeServiceVisit
> year(curdate()) - 1, (SUM(CAST(SUBSTRING(laborMarkup,2) AS DECIMAL
(10,2))) +
SUM(CAST(SUBSTRING(mItemCost,2) AS DECIMAL (10,2)))), 0) as
Difference
from Premier_profits_v
INNER JOIN ServiceVisit USING(customerID)
INNER JOIN MaintenanceLineItem USING(customerID)
INNER JOIN MaintenanceItem USING(mItemName)
order by Difference desc;




--#10
SELECT lastName as name, firstName, dateMonthYear, SUM(cost) FROM
    (SELECT lastName, firstName, dateMonthYear, (SELECT
CAST(SUBSTRING(mItemCost, 2) AS DECIMAL(10, 2))
    * (CAST(quantity AS DECIMAL(10,0))) + (CAST(SUBSTRING(laborMarkup,
2) AS DECIMAL(10,2))))
    AS cost FROM
    Steady INNER JOIN Individual USING (customerID) INNER JOIN
ServiceVisit USING (customerID) INNER JOIN
    MaintenanceLineItem USING (customerID, dateMonthYear,
timeServiceVisit) INNER JOIN
```

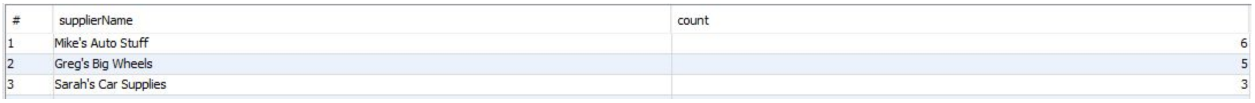
```
    MaintenanceItem USING (mItemName)group by lastName, dateMonthYear)
    T group by dateMonthYear having dateMonthYear >=
DATE_SUB(NOW(),INTERVAL 1 YEAR)

UNION ALL

SELECT corpName as name, '' as firstName, dateMonthYear, SUM(cost)
FROM
    (SELECT corpName, dateMonthYear, (SELECT CAST(SUBSTRING(mItemCost,
2) AS DECIMAL(10, 2))
    * (CAST(quantity AS DECIMAL(10,0))) + (CAST(SUBSTRING(laborMarkup,
2) AS DECIMAL(10,2))))
    AS cost FROM
    Steady INNER JOIN Corporate USING (customerID) INNER JOIN
ServiceVisit USING (customerID) INNER JOIN
    MaintenanceLineItem USING (customerID, dateMonthYear,
timeServiceVisit) INNER JOIN
    MaintenanceItem USING (mItemName)group by corpName, dateMonthYear)
    T group by dateMonthYear having dateMonthYear >=
DATE_SUB(NOW(),INTERVAL 1 YEAR);
```



#	name	firstName	dateMonthYear	SUM(cost)
1	Han	Sarah	2016-08-24	210.00
2	Haider	Syed	2016-11-28	70.00

```
--#11
SELECT supplierName, dateSupplied, COUNT(prodID) AS count
FROM ProductSupplier
WHERE dateSupplied >= DATE_SUB(NOW(),INTERVAL 1 YEAR)
GROUP BY supplierName
ORDER BY COUNT(prodID) DESC LIMIT 3;
```

ALTERED THE SUPPLIER PRODUCT TABLE TO ADD A DATE SUPPLIED FIELD****

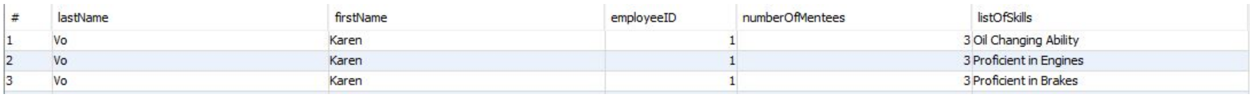


#	supplierName	count
1	Mike's Auto Stuff	6
2	Greg's Big Wheels	5
3	Sarah's Car Supplies	3

--12. List the five suppliers who have supplied us the largest dollar value of parts in the past year.

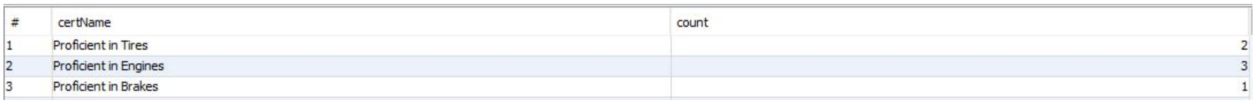
```
SELECT supplierName,dateSupplied,SUM(CAST(SUBSTRING(prodCost, 2) AS
DECIMAL(10, 2))) As "CostofTotalParts"
FROM cecs323bg1.ProductSupplier
INNER JOIN cecs323bg1.Product using (prodID)
WHERE dateSupplied >= DATE_SUB(NOW(),INTERVAL 1 YEAR)
GROUP BY supplierName
ORDER BY MAX(CAST(SUBSTRING(prodCost, 2) AS DECIMAL(10, 2))) DESC;
```

```
--#13
SELECT T.lastName, T.firstName, T.employeeID, MAX(T.count) AS
numberOfMentees, M.certName AS listOfSkills
FROM (SELECT lastName, firstName, employeeID, COUNT(empMenteeID) as
count
FROM (SELECT lastName, firstName, E.employeeID, empMenteeID, endDate
FROM Employee E INNER JOIN MentorRelation M ON (E.employeeID =
M.employeeID) WHERE endDate IS NULL GROUP BY certName ) U) T

INNER JOIN MentorRelation M ON (T.employeeID = M.employeeID) GROUP BY
M.empMenteeID;
```



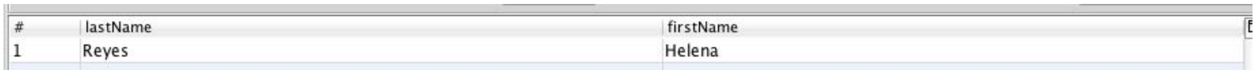
#	lastName	firstName	employeeID	numberOfMentees	listOfSkills
1	Vo	Karen	1	3	Oil Changing Ability
2	Vo	Karen	1	3	Proficient in Engines
3	Vo	Karen	1	3	Proficient in Brakes

```
--#14
SELECT certName, count(certName) AS count
FROM MechanicCertification
GROUP BY certName ORDER BY certName DESC LIMIT 3;
```



#	certName	count
1	Proficient in Tires	2
2	Proficient in Engines	3
3	Proficient in Brakes	1

```
--#15
use cecs323bg1;
select lastName, firstName from Employee
inner join Mechanic m using (employeeID)
inner join ServiceTechnician s using (employeeID)
where m.employeeID = s.employeeID
and s.sDateYearNotEmployed is null
and m.mDateYearNotEmployed is null;

--#15 Sample Output
```



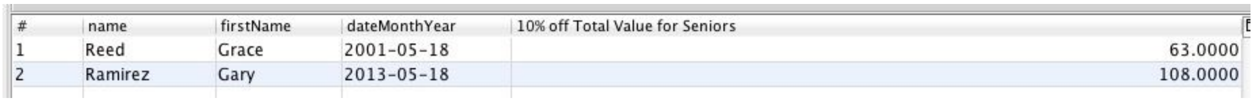
#	lastName	firstName
1	Reyes	Helena

--#16

```
SELECT lastName as name, firstName, dateMonthYear,timeServiceVisit,
SUM(cost -(cost * 0.1)) AS "Total Cost" FROM
        (SELECT lastName, firstName, dateMonthYear,
timeServiceVisit,(SELECT CAST(SUBSTRING(mItemCost, 2) AS DECIMAL(10,
2))
        * (CAST(quantity AS DECIMAL(10,0)))+
(CAST(SUBSTRING(laborMarkup, 2) AS DECIMAL(10,2))))
        AS cost FROM Individual
        INNER JOIN ServiceVisit USING (customerID)
        INNER JOIN MaintenanceLineItem USING (customerID,
dateMonthYear, timeServiceVisit)
        INNER JOIN MaintenanceItem USING (mItemName)
        WHERE timeServiceVisit > CAST('07:00:00' AS time) AND
timeServiceVisit < CAST('11:00:00' AS time)
        group by lastName, dateMonthYear)
        T group by dateMonthYear,timeServiceVisit

    UNION ALL

    SELECT corpName as name, '' as firstName,
dateMonthYear,timeServiceVisit, SUM(cost -(cost * 0.10)) AS "Total
Cost" FROM
        (SELECT corpName, dateMonthYear,timeServiceVisit, (SELECT
CAST(SUBSTRING(mItemCost, 2) AS DECIMAL(10, 2))
        * (CAST(quantity AS DECIMAL(10,0))) +
(CAST(SUBSTRING(laborMarkup, 2) AS DECIMAL(10,2))))
        AS cost FROM
        Corporate INNER JOIN ServiceVisit USING (customerID)
        INNER JOIN MaintenanceLineItem USING (customerID,
dateMonthYear, timeServiceVisit)
        INNER JOIN MaintenanceItem USING (mItemName)
        WHERE timeServiceVisit > CAST('07:00:00' AS time) AND
timeServiceVisit < CAST('11:00:00' AS time)
        group by corpName, dateMonthYear,timeServiceVisit)
        T group by dateMonthYear,timeServiceVisit;
```

```
—seniors
INSERT INTO Individual VALUES
(11,'Grace','Reed','45628 Peppermint St.','Santa
Barbara','California','90161','1946-04-22');

INSERT INTO Customer(phone,email) VALUES
('9514728394','gracereed@gmail.com’);

INSERT INTO Individual VALUES
(12,’Gary’,’Ramirez’,'1291 Avenida
Ave.’,'Temecula','California’,'92592','1936-04-22');

INSERT INTO Customer(phone,email) VALUES
('5622344421','gary.ramirez@gmail.com’);

INSERT INTO Premier VALUES (11,'2001-05-18', NULL);
INSERT INTO Premier VALUES (12,'2013-05-18', NULL);

INSERT INTO MakeModel VALUES ('Kia', 'Optima', '2013’);

INSERT INTO MakeModel VALUES ('Ford', 'Mustang', '1998’);

INSERT INTO Vehicle VALUES ('3GTP1TEA1DG264618', 'MUSIC8', FALSE,
'003800', 'Ford', 'Mustang', '1998');

INSERT INTO Vehicle VALUES ('1HGCP2F49AA123114', '683MWK', FALSE,
'004800', 'Kia', 'Optima', '2013’);

INSERT INTO ServiceVisit VALUES (11, '2001-05-18', '15:30:00',
'683MWK', '045890'); -- Grace
INSERT INTO ServiceVisit VALUES (12, '2013-05-18', '07:30:00',
'MUSIC8', '023040'); -- Gary

INSERT INTO MaintenanceLineItem VALUES (11, '2001-05-18', '15:30:00',
'683MWK', 'Change Oil', 1, '$10.00');
```

```
INSERT INTO MaintenanceLineItem VALUES (12, '2013-05-18', '07:30:00',
'MUSIC8', 'Install Engine Starter', 1, '$40.00');


—business rule #1: seniors gets 10% off
SELECT lastName as name, firstName, dateMonthYear,
ROUND(SUM(cost-(cost*0.10)),2) as '10% off Total Value for Seniors'
FROM
    (SELECT lastName, firstName, dateMonthYear, (SELECT
CAST(SUBSTRING(mItemCost, 2) AS DECIMAL(10, 2))
    * (CAST(quantity AS DECIMAL(10,0))) +
(CAST(SUBSTRING(laborMarkup, 2) AS DECIMAL(10,2))))
    AS cost FROM
    Individual INNER JOIN ServiceVisit USING (customerID) INNER JOIN
    MaintenanceLineItem USING (customerID, dateMonthYear,
timeServiceVisit) INNER JOIN
    MaintenanceItem USING (mItemName)
    where TIMESTAMPDIFF(YEAR, dateOfBirth, CURDATE()) >=65
    group by lastName, dateMonthYear)
    T group by dateMonthYear;
```



#	name	firstName	dateMonthYear	10% off Total Value for Seniors
1	Reed	Grace	2001-05-18	63.0000
2	Ramirez	Gary	2013-05-18	108.0000

```
--business rule #2
SELECT firstName, lastName, salary, ROUND(salary*(1.15),2) AS "Raised
Salary"
FROM Employee
WHERE TIMESTAMPDIFF(YEAR,eDateYearEmployed, NOW()) >=5 AND
eDateYearNotEmployed is NULL;
```

```
--not complete yet, business rule #4
select customerID from ServiceVisit
where (2016-year(dateMonthYear))>=5;
--finds customer who have been customer for 5 years

select mItemName, mItemCost from MaintenanceItem
```

```
where (SELECT CAST(SUBSTRING(mItemCost, 2) AS DECIMAL(10, 2))) <
150.00;
—finds maintenanceItems costs that are less than 150.00
```

5 year premier anniversary customer special: free service that is worth $150 or less

```
--4 BUSINESS REQUIREMENT
use cecs323bg1;
select customer_v.customerID, customer_v.FIRSTNAME,
customer_v.LASTNAME
from  Customer, customer_v
INNER JOIN Premier USING(CustomerID)
WHERE (year(current_date) - year(startDate)) >= 5
UNION
select customer_v.customerID, customer_v.FIRSTNAME,
customer_v.LASTNAME
from Customer, customer_v
INNER JOIN ServiceVisit USING(CustomerID)
NATURAL JOIN (
    SELECT   customerID, MIN(dateMonthYear) AS entry_date
    FROM     ServiceVisit
    GROUP BY customerID
) as tmin
WHERE (year(current_date) - year(entry_date)) >= 5;
```