Changelog:

Added dateContacted

Add loyalty points into steady customer

Add deadprospect and numofContacts into prospective customer

Add maintenanceVisit.Actualmileage and yearlyMileageEstimate

Change RecommendedInterval recdate and recMilage to recYearAmount(***data type integer not date anymore***) and recMileageAmount

Add maintenanceVisitCost into maintenanceVisit table

I really think we should change the uml. The email notification should connect from the vehicle SINCE the mileage is different among the vehicle not the model/type/year since we can’t store the accumlated milage and year in there

I haven’t change the uml yet so we can work together. If you guys work together make sure to write the ***changelog***

Triggers needed

Deadprospect and numOfContacts

drop table if exists dateContacted;

drop table  if exists prospectiveCustomer;

create table prospectiveCustomer(

customerID int not null,

referredByID int not null,

deadProspect boolean not null default false ,

numContacted int ,

constraint prospectiveCustomer\_fk FOREIGN KEY (customerID)

REFERENCES customer(customerID) ,

    constraint prospectiveCustomer\_fk2 FOREIGN KEY (referredByID)

   REFERENCES currentCustomer(customerID));

--------------------------------------------------------------------------------------

create table dateContacted(

customerID int not null,

dateContacted date not null,

constraint d\_fk foreign key (customerID)

references prospectiveCustomer(customerID)

);

--------------------------------------------------------------------------------------

drop trigger if exists update\_numContacted;

delimiter |

CREATE TRIGGER update\_numContacted

after INSERT ON dateContacted

FOR EACH ROW

BEGIN

   update prospectiveCustomer

     set numContacted = numContacted +1

where customerID = new.customerID;

end |

delimiter;

--------------------------------------------

drop trigger if exists update\_deadProspect;

delimiter |

CREATE TRIGGER update\_deadProspect

before INSERT ON dateContacted

FOR EACH ROW

BEGIN

   declare n int;

   select numContacted into n

   from prospectiveCustomer

   where customerID = new.customerID;

   IF (n >= 2)  THEN

       update prospectiveCustomer

        set deadProspect = true

       where customerID = new.customerID;

        --SIGNAL SQLSTATE '02000' SET MESSAGE\_TEXT = 'Warning: c > 100!';

   end if;

end |

delimiter;

------------------------------------

INSERT INTO prospectiveCustomer(customerID, referredByID, numContacted) values

(10001, 10002,  0),

(10002, 10004,  0),

(10003, 10005,  0)

;

INSERT INTO dateContacted(customerID, dateContacted) values

(10001,'2017-02-02'),

(10001,'2017-02-03'),

(10001,'2017-02-04'),

(10002,'2017-02-02'),

(10002,'2017-02-03'),

(10002,'2017-02-04'),

(10002,'2017-02-05'),

(10002,'2017-02-06'),

(10002,'2017-02-07'),

(10003,'2017-02-02'),

(10003,'2017-02-03')

;

-- MIGHT NEED TO ADD MORE dateContacted table because i cant trigger one table (need at least 2 table to use trigger) :/

estMileage

-------------------------------------------It works but still need to add the amount suggested from the RecommendedInterval table . You can check it your self. If you need to quick check, replace the keyword insert with update. That way you can just change the attribute data and the related table will change also. You don’t need to insert and delete anymore. Save time

drop trigger if exists update\_estimatedMileage ;

delimiter |

CREATE TRIGGER update\_estimatedMileage

after insert ON maintenanceVisit

FOR EACH ROW

BEGIN

UPDATE vehicle

SET yearlyMileageEstimate = 50000 + (select maintenanceVisit.Actualmileage

   from maintenanceVisit

   where maintenanceVisit.vin = new.vin

)

WHERE vin = new.vin;

end|

delimiter;

estDate

-------------------------------------------Same, work now but suppose to take the value from RecommendedInterval table not just a number 1 like INTERVAL  (1) YEAR)

drop trigger if exists update\_estimatedDate ;

delimiter |

CREATE TRIGGER update\_estimatedDate

after insert ON maintenanceVisit

FOR EACH ROW

BEGIN

UPDATE vehicle

SET dateEstimate =  DATE\_ADD(

(select maintenanceVisit.visitDate

   from maintenanceVisit

   where maintenanceVisit.vin = new.vin

),

INTERVAL  (1) YEAR)

WHERE vin = new.vin;

end|

delimiter;

estCost

It’s in emailNotification

Total cost -- NEED TO loyalties point collumn in customer table into the create table google doc:/

-- add the maintenanceVisit fee into maintenanceVisit

-ADDS loyaltypoints to steadycustomer

--when ever an maintenanceAction

--is added to visitService

--we calculate the totalFee

--and add it to maintenanceVisit

DELIMITER|

CREATE TRIGGER

AFTER INSERT ON maintenanceAction

FOR EACH ROW

BEGIN

IF

UPDATE

END

DELIMITER;

------------------------------------------------------------------------------------------------------------------------

--whenever maintenanceVisit got update

--we update the loyalty point

drop trigger if exists update\_maintenanceVisitFee;

delimiter |

CREATE TRIGGER update\_maintenanceVisitFee

before INSERT ON visitService

FOR EACH ROW

BEGIN

update maintenanceVisit

set maintenanceVisitCost = maintenanceVisitCost+(select maintenanceAction.actionCost

   from maintenanceAction

   where maintenanceAction.actionID = new.actionID)

where visitID=new.visitID;

end |

delimiter;

----------------------------------------------------------------------------------------------

INSERT INTO visitService(visitID, listID,actionID) values

(80011,  90027,60001),

(80011,  90027,60002),

(80011,  90027,60003),

(80011,  90027,60004),

(80011,  90027,60005)

;

Loyalty points

DELIMITER|

CREATE TRIGGER add\_loyalty\_points

AFTER INSERT ON ServiceVisit

FOR EACH ROW

BEGIN

IF

UPDATE steadyCustomer

SET loyaltyPoints = loyaltyPoints + (actionCost/10)

WHERE

END IF;

END|

DELIMITER;

--Added loyaltyPoints in steadyCustomer table

HAVENT TESTED

drop trigger if exists update\_loyaltyPoints;

delimiter |

CREATE TRIGGER update\_loyaltyPoints

After update ON maintenanceVisit

FOR EACH ROW

BEGIN

update steadyCustomer

set loyaltyPoints = loyaltyPoints+((select maintenanceVisit.maintenanceVisitCost

   from maintenanceVisit

inner join vehicle

on maintenanceVisit.vin = vehicle.vin

inner join customer

on vehicle.customerID = customer.customerID

   where maintenanceVisit.visitID = new.visitID)/10)

;

end |

delimiter;