Insert into packagecomps values ('Headlight Change','Package 1',1);

/\* trigger 1\*/

DELIMITER :

CREATE TRIGGER packagecomps\_before\_insert

BEFORE INSERT ON packagecomps

FOR EACH ROW

BEGIN

IF((SELECT COUNT(maintenance\_name) FROM packagecomps GROUP BY package\_name HAVING package\_name = new.package\_name) >= 5) THEN

signal sqlstate'45000' set message\_text='Package already has max (5 items)!';

END IF;

END :

DELIMITER ;

/\*generic version for update\*/

DELIMITER :

CREATE TRIGGER packagecomps\_before\_update

BEFORE UPDATE ON packagecomps

FOR EACH ROW

BEGIN

IF(OLD.package\_name != NEW.package\_name) THEN

IF((SELECT COUNT(maintenance\_name) FROM packagecomps GROUP BY package\_name HAVING package\_name = new.package\_name) >= 5) THEN

signal sqlstate'45000' set message\_text='Package already has max (5 items)!';

END IF;

END IF;

END :

DELIMITER ;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/\* TRIGGER 2\*/

//generic version for insert

DELIMITER :

CREATE TRIGGER visitOrders\_before\_insert

BEFORE INSERT ON visitOrders

FOR EACH ROW

BEGIN

IF ((SELECT count(logTime) from customers

inner join vehicle using (cust\_id)

inner join visitOrders using (VIN)

WHERE cust\_id = new.cust\_id AND DATE(logTime) = DATE(new.logTime) ) >= 3) THEN

signal sqlstate'45000' set message\_text='Customer reach max car entries(3 times)!';

end if;

END :

DELIMITER ;

//generic version for update

DELIMITER :

CREATE TRIGGER visitOrders\_before\_update

BEFORE UPDATE ON visitOrders

FOR EACH ROW

BEGIN

IF(OLD.logTime != NEW.logTime) THEN

IF ((SELECT count(logTime) from customers

inner join vehicle using (cust\_id)

inner join visitOrders using (VIN)

WHERE cust\_id = new.cust\_id AND DATE(logTime) = DATE(new.logTime) ) >= 3) THEN

signal sqlstate'45000' set message\_text='Customer reach max car entries(3 times)!';

END IF;

end if;

END :

DELIMITER ;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//TRIGGER 3

DELIMITER :

CREATE TRIGGER taken\_mechanic\_before\_insert

BEFORE INSERT ON servicelineitems

FOR EACH ROW

BEGIN

IF((SELECT COUNT(servName) FROM servicelineitems GROUP BY VIN, logTime HAVING employ\_id = NEW.employ\_id AND VIN = NEW.VIN AND logTime = NEW.logTime) >= 1) THEN

signal sqlstate'45000' set message\_text='That mechanic has already been assigned to a maintenance item for that visit!';

END IF;

END :

DELIMITER ;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//TRIGGER 4

//on insert

DELIMITER :

CREATE TRIGGER loyalty\_bal\_after\_insert

AFTER INSERT ON servicelineitems

FOR EACH ROW

BEGIN

DECLARE userid VARCHAR (20);

SELECT cust\_id INTO userid FROM vehicles v INNER JOIN customers c ON (c.VIN = v.VIN) INNER JOIN nonprospectives np ON (np.cust\_id = c.cust\_id) INNER JOIN steadys s ON (np.cust\_id = s.cust\_id) WHERE v.VIN = NEW.VIN;

IF(userid IS NOT NULL)

DECLARE visitTotal DECIMAL (6,2);

SELECT SUM(parts.price \* mp.quantity + mi.laborMarkUp - discount - loyaltyPointUsed \* 2 INTO visitTotal FROM servicelineitems sli INNER JOIN visitOrders vo ON (vo.VIN = sli.VIN AND vo.logTime=sli.logTime) INNER JOIN vehicles v ON (v.VIN = vo.VIN) INNER JOIN customers c ON (c.cust\_id = v.cust\_id) GROUP BY c.cust\_id HAVING c.cust\_id = userid AND sli.logTime = NEW.logTime AND sli.VIN = NEW.VIN;

UPDATE steadys

SET loyaltyBal = LEAST(999, loyaltyBal + visitTotal/10)

WHERE cust\_id = userid;

END IF;

END :

DELIMITER ;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//TRIGGER 5

DELIMITER :

CREATE TRIGGER mechanic\_skill\_before\_insert

BEFORE INSERT ON servicelineitems

FOR EACH ROW

BEGIN

IF (EXISTS(SELECT \* FROM servicelineitems INNER JOIN serviceItems ON (name = servName) INNER JOIN maintenance\_items USING (name) INNER JOIN itemSkills USING (skillName) INNER JOIN employee\_skill USING (skillName) WHERE logTime = NEW.logTime AND VIN = NEW.VIN AND servName = NEW.servName AND employ\_id = NEW.employ\_id) ) THEN

signal sqlstate'45000' set message\_text='That mechanic doesn’t have the skills to perform that service!';

end if;

END :

DELIMITER ;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//TRIGGER 6

DELIMITER :

CREATE TRIGGER mentorRelationships\_before\_insert

BEFORE INSERT ON mentorRelationships

FOR EACH ROW

BEGIN

DECLARE sdate DATE;

SELECT startDate INTO sdate FROM mentorRelationships WHERE mentor\_id = NEW.mentor\_id AND mentee\_id = NEW.mentee\_id AND certName = NEW.certName AND endDate IS NULL;

IF (sdate IS NOT NULL) THEN

signal sqlstate'45000' set message\_text='That mentor is still teaching that mentee that skill!!';

end if;

END :

DELIMITER ;