trigger OpportunityTrigger on Opportunity (before update, After Update) {

if(trigger.isbefore && trigger.isUpdate){

OpportunityHandlerClass.opportunityAutomobileQuantity(trigger.new, trigger.oldMap);

}

}

public class OpportunityHandlerClass {

public static void opportunityAutomobileQuantity(List<Opportunity> LstOpportunity, Map<Id,Opportunity> OldMapOpportunity){

set<Id> opportunityIds = new set<Id>();

for(Opportunity opp : LstOpportunity){

if(opp.StageName =='Closed Won' ){

opportunityIds.add(opp.Id);

}

}

Map<Id,Opportunity\_Automobile\_c> lstOpportunityAutomobile =new Map<Id,Opportunity\_Automobilec>([SELECT Id, Opportunityc, Automobilec, Quantityc, Unit\_Pricec, Total\_Pricec FROM Opportunity\_Automobilec Where Opportunity\_c IN: opportunityIds]);

set<Id> AutoInformationIds = new set<Id>();

for(Opportunity\_Automobile\_\_c OppAuto: lstOpportunityAutomobile.values()){

if(OppAuto.Automobile\_\_c != null){

AutoInformationIds.add(OppAuto.Automobile\_\_c);

}

}

List<Automobile\_Information\_c> lstAutomobileInfomation = new List<Automobile\_Information\_c>();

Map<Id,Automobile\_Information\_c> MapAutomobileInformation = New Map<Id,Automobile\_Informationc>([SELECT Quantityc, Pricec, Name, Id FROM Automobile\_Information\_c WHERE Id IN: AutoInformationIds]);

For(Opportunity\_Automobile\_\_c AutoOpp : lstOpportunityAutomobile.Values()){

decimal num = 0;

if(AutoOpp.Automobile\_c == MapAutomobileInformation.get(AutoOpp.Automobilec).Id && OldMapOpportunity.get(AutoOpp.Opportunity\_c).stagename != 'Closed Won'){

num = MapAutomobileInformation.get(AutoOpp.Automobile\_c).Quantityc- AutoOpp.Quantity\_c;

MapAutomobileInformation.get(AutoOpp.Automobile\_c).quantity\_c = num;

lstAutomobileInfomation.add(MapAutomobileInformation.get(AutoOpp.Automobile\_\_c));

}

}

If(!lstAutomobileInfomation.IsEmpty()){

update lstAutomobileInfomation;

}

}

}