**PROGRAM CODES**

1. **Create apex class “wordorderclass”**

public class WorkOrderClass {

public static void workOrder(List<WorkOrder\_\_C> newListWorkOrder){

Map<Integer, List<String>> maptotech = new map<Integer,List<String>>();

integer num = 0;

List<WorkOrder\_\_c> properWo = new List<WorkOrder\_\_c>();

List<Assignment\_\_c> lstAssignment = new List<Assignment\_\_c>();

List<Technician\_\_c> techniciantoAssignment = new List<Technician\_\_c>();

for(WorkOrder\_\_c iter : newListWorkOrder){

List<String> lststring = new List<string>();

If(iter.Service\_Type\_\_c != null && iter.Location\_\_c != null ){

num = num+1;

properWo.add(iter);

lststring.add(iter.Service\_Type\_\_c);

lststring.add(iter.Location\_\_c);

maptotech.put(num,lststring);

}

}

Map<integer,Id> techId = new Map<integer,Id>();

Map<Id,Technician\_\_c> allTechnician = new Map<Id,Technician\_\_c>([SELECT Id, Name, Phone\_\_c, Location\_\_c, Skills\_\_c, Availibility\_\_c, Name\_\_c, Email\_\_c FROM Technician\_\_c]);

integer num2 = 0;

For(Technician\_\_c T : allTechnician.values()){

num2 = num2+1;

if(maptotech.get(num2) != null){

List<string> valofmap = maptotech.get(num2);

system.debug('error 1 ----> the maptotech is empty ---> ' + maptotech.get(num2));

if(valofMap.contains(t.Skills\_\_c) && ValofMap.contains(t.Location\_\_c) && t.Availibility\_\_c == 'Available'){

techid.put(num2,t.Id);

}

}

}

integer num3 = 0;

For(WorkOrder\_\_c W : properWo){

num3 = num3 + 1;

Assignment\_\_c A = new Assignment\_\_c();

A.WorkOrder\_ID\_\_c = W.Id;

A.Technician\_ID\_\_c = techid.get(num3);

lstAssignment.add(A);

}

If(!lstAssignment.IsEmpty()){

insert lstAssignment;

}

}

}

1. **Create the Apex Trigger name as “WorkOrderTrigger”, and select “WorkOrder\_\_c” from the dropdown for sObject.**

trigger WorkOrderTrigger on WorkOrder\_\_c (after insert) {

if(trigger.isafter && trigger.isinsert){

WorkOrderClass.workOrder(trigger.new);

}

}

1. **Give the Apex Class name as “AssigningEmail”.**

public class AssigningEmail {

public static void sendEmailmsg(List<Assignment\_\_c> assRec){

List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();

Map<id,Technician\_\_c> tecnicians = new Map<id,Technician\_\_c>([SELECT Id, Phone\_\_c, Location\_\_c, Skills\_\_c, Name\_\_c, Email\_\_c, Availibility\_\_c, Name FROM Technician\_\_c]);

try{

for(Assignment\_\_c con : assRec){

if(con.Technician\_ID\_\_c != null){

messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();

List<String> sendTo = new List<String>();

sendTo.add(tecnicians.Get(con.Technician\_ID\_\_c).Email\_\_c);

mail.setToAddresses(sendTo);

string subject = 'WorkOrder Assignment ';

mail.setSubject(subject);

string body = 'The following WorkOrder has been assigned to you ';

mail.setHTMLbody(body);

myVar.add(mail);

}

}

Messaging.sendEmail(myvar);

}

catch(exception e){

system.debug('Error -----> ' + e.getMessage());

}

}

}

1. **Give the Apex Trigger name as “AssignmentTrigger”, and select “Assignment\_\_c” from the dropdown for sObject.**

trigger AssignmentTrigger on Assignment\_\_c (after insert) {

if(Trigger.IsAfter && Trigger.IsInsert){

AssigningEmail.sendEmailmsg(Trigger.New);

}

}

1. **Give the Apex Class name as “CompletionMail”.**

public class CompletionMail {

public static void sendEmailMsg(List<WorkOrder\_\_c> workOrderList){

List<messaging.SingleEmailMessage> myVar = new List<messaging.SingleEmailMessage>();

for(WorkOrder\_\_c con : workOrderList){

if(con.Status\_\_c == 'Resolved'){

messaging.SingleEmailMessage mail = new messaging.SingleEmailMessage();

List<String> sendTo = new List<String>();

sendTo.add(con.Email\_\_c);

mail.setToAddresses(sendTo);

string subject = 'Status Updated';

mail.setSubject(subject);

string body = 'email body ';

mail.setHTMLbody(body);

myVar.add(mail);

}

}

Messaging.sendEmail(myvar);

}

}

1. **Open the Apex Trigger “WorkOrderTrigger”**

trigger WorkOrderTrigger on WorkOrder\_\_c (after insert,after update) {

if(Trigger.IsAfter && Trigger.IsUpdate){

CompletionMail.sendEmailMsg(Trigger.New);

}

if(Trigger.IsAfter && Trigger.IsUpdate){

CompletionMail.sendEmailMsg(Trigger.New);

}

}

1. **Give the Apex Class name as “RecordDeletion”.**

public class RecordDeletions Implements Database.Batchable<Sobject>{

public Database.QueryLocator start(Database.BatchableContext bc) {

string query = 'SELECT Id, Name, WorkOrder\_ID\_\_c, Technician\_ID\_\_c, Assignment\_Date\_\_c, Completion\_Date\_\_c FROM Assignment\_\_c WHERE Completion\_Date\_\_c = LAST\_N\_DAYS:30';

return database.GetQueryLocator(query);

}

public void execute(Database.BatchableContext bc, List<Assignment\_\_c> query){

if(!Query.IsEmpty()){

Delete Query;

}

}

public void finish(Database.BatchableContext bc){

}

}

1. **Give the Apex Class name as “ScheduleClass”.**

global class ScheduleClass implements Schedulable {

global void execute(SchedulableContext SC) {

RecordDeletions delrec = new RecordDeletions();

database.executeBatch(delrec, 200);

}

}