

```
if(trigger.isAfter && trigger.isinsert ){  
    TriggerHandler.sendingemail(trigger.new,null);  
}  
if(trigger.isAfter && trigger.isupdate ){  
    TriggerHandler.sendingemail(trigger.new,Trigger.oldmap);  
}
```

```

public static void sendingemail(list<account> acclist ,Map <id,Account> oldMap){
    date thirtyDaysAgo=system.today()-30;
    set<id> ownerids= new set<id>();
    for(account acc: acclist){
        if(acc.AnnualRevenue != null &&
            acc.AnnualRevenue>50000 &&
            acc.CreatedDate >thirtyDaysAgo &&
            (oldMap==null || oldmap.get(acc.id).AnnualRevenue != acc.AnnualRevenue)){
            ownerids.add(acc.OwnerId);
        }
    }
    Map<Id, User> usermap = new Map<Id, User>();
    if (!ownerids.isEmpty()) {
        usermap = new Map<Id, User>([SELECT Id, name, Email FROM User WHERE Id IN :ownerids]);
    }
    list<messaging.SingleEmailMessage> emails= new list<Messaging.SingleEmailMessage>();
    if(usermap.size()>0){
        for(account acc:acclist){
            if(usermap.containsKey(acc.OwnerId)){
                Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();
                string[] toaddress= new string[]{usermap.get(acc.OwnerId).email};
                email.setToAddresses(toaddress);
                email.setSubject('Account Record Id '+ acc.id );
                email.setplaintextbody('Dear '+usermap.get(acc.OwnerId).name + ',\n\n The revenue for the below mentioned Record is' +
                    'above 50K \n Annual Revenue = ' + acc.AnnualRevenue + '\n RecordId = '+ acc.Id + '\n\nBest regards');
                emails.add(email);
                //send email messages
                Messaging.sendEmail(emails);
            }
        }
    }
}

```
