

Apex Trigger:

trigger test on Tenant__c (before insert)

```
{  
    if(trigger.isInsert && trigger.isBefore){  
        testHandler.preventInsert(trigger.new);  
    }  
}
```

Test Class:

```
public class testHandler {  
    public static void preventInsert(List<Tenant__c> newList) {  
        Set<Id> existingPropertyIds = new Set<Id>();  
        for (Tenant__c existingTenant : [SELECT Id, Property__c FROM Tenant__c WHERE  
Property__c != null]) {  
            existingPropertyIds.add(existingTenant.Property__c);  
        }  
        for (Tenant__c newTenant : newList) {  
            if (newTenant.Property__c != null &&  
existingPropertyIds.contains(newTenant.Property__c)) {  
                newTenant.addError('A tenant can have only one property');  
            }  
        }  
    }  
}
```

MonthlyEmailScheduler:

```
global      class      MonthlyEmailScheduler      implements      Schedulable      {

    global      void      execute(SchedulableContext      sc)      {

        Integer      currentDay      =      Date.today().day();

        if      (currentDay      ==      1)      {

            sendMonthlyEmails();

        }

    }

}

public      static      void      sendMonthlyEmails()      {

    List<Tenant__c>      tenants      =      [SELECT      Id,      Email__c      FROM      Tenant__c];

    for      (Tenant__c      tenant      :      tenants)      {

        String      recipientEmail      =      tenant.Email__c;

        String emailContent = 'I trust this email finds you well. I am writing to remind you that the monthly rent is due Your timely payment ensures the smooth functioning of our rental arrangement and helps maintain a positive living environment for all.';

        String      emailSubject      =      'Reminder:      Monthly      Rent      Payment      Due';

        Messaging.SingleEmailMessage      email      =      new      Messaging.SingleEmailMessage();

        email.setToAddresses(new      String[]{recipientEmail});

        email.setSubject(emailSubject);

        email.setPlainTextBody(emailContent);

        Messaging.sendEmail(new      Messaging.SingleEmailMessage[]{email});

    }

}
```

}

}

}