Функция возвращает системе обращение пользователя. Это обращение попадает в работу сотруднику поддержки.

Случилась ошибка: одно и то же обращение пользователя попало в работу 2-м сотрудникам поддержки. Измените код так, чтобы этого больше не повторилось.

def set\_assignee(login) // логин сотрудника поддержки  
  {  
              // request\_data - интерфейс управления БД. Считать, что не может исполняться несколько операций с БД параллельно  
              requests = request\_data.get\_all() // получаем все обращения пользователей из БД  
              request = requests.groupBy{  
                          it.id  
              }.collect{ key, value ->  
                          value.max{ it.version }  
              }.findAll{ req ->  
                          req.status == 'opened' &&  
                          req.state == 'ready'  
              }.max{  
                          it.priority  
              }  
              if (request == null)  
                          return null  
                
              request.state = 'inprogress'  
              request.assignee = login  
              request.version += 1  
              request\_data.update(request) // загружаем новую версию обращения в БД (старая не удаляется)  
              // Я бы после получения запроса сохранял бы запрос в переменную, а не заносил обратно в БД  
              // Отсюда и ошибка задваивания. Думаю, стоит удалять полученный запрос из БД.  
              return request   
  }

**Мысли по поводу Groovy кода я прописал внутри кода.**

Groovy я не знаю, поэтому попробовал реализовать свою идею решения проблемы на языке Python

БД к сожалению на рука не было, поэтому пришлось делать как придумалось

# Проблема - Задваивание запроса.  
# Код на питоне. Моя Идея Реализации - +1) Подключение к БД. +2) Выбрать секцию с запросами пользователей.  
# +- 3) Вывести запрос (имя пользователя(Возможно стоит добавить почту для обратного ответа пользователю))  
# +4) Сохранить запрос в переменную/словарь для возможности запоминания запроса  
# +5) Удалить запрос из БД, чтобы не происходило задваивания (двум админам должны приходить разные запросы)  
  
def sq\_request(login):  
 try:  
 conn = sqlite3.connect("БД.sqlite") # Возможно будет другой вид БД  
 # Объект запросов и результатов  
 cursor = conn.cursor()  
 # Возможно у нас идет запросы по порядку, новый появляется на самом верху, тогда мы и будем получать  
 # самый свежий запрос  
 cursor.execute('SQL\_statement where id = ?', 'id') # Выбор запросов в БД через SQL\_запросы по id.  
 # Также возможен вариант выбора запроса по индексу  
  
 result = cursor.fetchone()  
 user = cursor.execute('SQL\_statement where ...') # Возможно у нас есть целый раздел БД, где есть только имена  
 # пользователей и их запрос, тогда стоит удалять и пользователя из этой вкладки, вместе с его запросом  
  
 sql\_delete\_request = """DELETE from ... where ... = ..."""  
 cursor.execute(sql\_delete\_request)  
 sql\_delete\_user = """DELETE from ... where ... = ..."""  
 cursor.execute(sql\_delete\_user)  
 except sqlite3.DatabaseError as err:  
 print("Error", err)  
 else:  
 conn.commit()  
 # Обязательно закрываем соединение с БД  
 conn.close()  
 print(user, result) # Выведем на экран содержимое  
 # Вернем нужные нам данные для дальнейшей работы с ними  
 return user, result