package main  
  
import (  
 "fmt"  
 "sync"  
)  
  
var lock = &sync.Mutex{}  
  
type single struct {  
}  
  
var singleInstance \*single  
  
func getInstance() \*single {  
 if singleInstance == nil {  
 lock.Lock()  
 defer lock.Unlock()  
 if singleInstance == nil {  
 fmt.Println("Creating single instance now.")  
 singleInstance = &single{}  
 } else {  
 fmt.Println("Single instance already created.")  
 }  
 } else {  
 fmt.Println("Single instance already created.")  
 }  
  
 return singleInstance  
}  
func main() {  
  
 for i := 0; i < 3; i++ {  
  
 go getInstance()  
 }  
 fmt.Scanln()  
}

We use singleton for checking if database recreated or created more than 1 time. So we have a Single class that we are gonna check if it is recreated or not if it is not we will create it.

