CODE:

```
// Code No:-1
 import java.util.*;
import java.util.concurrent.Semaphore;
 public class Main { static int mutex=1; static int database=1;
static int Read_Count=0;
static void Reader() throws Exception { while(true) { mutex=wait(mutex); Read_Count=Read_Count+1;
if(Read_Count==1){ database=signal(database); }
 mutex=signal(mutex);
System.out.println(Read_Count+ " User Reading the Data.....");
mutex=wait(mutex);
Read_Count=Read_Count-1;
 if(Read_Count==0) { database=signal(database); }
mutex=signal(mutex);
 System.out.println("Reading Finished!!!!!!");
break; } }
static int wait(int mutex)
{ while(mutex<=0)
 break ;
mutex=mutex-1;
 return mutex; } static int signal(int database)
{ database=database+1; return database; }
 static void Writer() throws Exception
{ while(true)
{ database=wait(database);
 System.out.println("Writing on the database.....");
database=signal(database);
 System.out.println("Writing Finished!!!!!.");
break; } }
 public static void main(String[] args)throws Exception
 { Writer();
Reader();
 Reader();
} }
 // Code No:-2
import java.util.concurrent.Semaphore;
import java.util.Scanner;
public class Main { static Semaphore mutex = new Semaphore(1);
static Semaphore wrt = new Semaphore(1);
static int readCount = 0;
 static String message = "Hello";
static Scanner SC = new Scanner(System.in);
static class Reader implements Runnable
 { public void run()
 { try { mutex.acquire();
readCount++;
 if (readCount == 1) { wrt.acquire(); }
 mutex.release();
System.out.println("Thread "+Thread.currentThread().getName() + " is READING: " + message);
 Thread.sleep(1500);
 System.out.println("Thread "+Thread.currentThread().getName() + " has FINISHED READING");
mutex.acquire(); readCount--;
 if(readCount == 0) { wrt.release(); }
 mutex.release(); }
catch (InterruptedException e)
{ System.out.println(e.getMessage()); } } }
 static class Writer implements Runnable { public void run()
 { try { wrt.acquire();
 message = "Good Morning";
System.out.println("Thread "+Thread.currentThread().getName() + " is WRITING: " + message);
 Thread.sleep(1500);
System.out.println("Thread "+Thread.currentThread().getName() + " has finished WRITING");
wrt.release(); }
catch (InterruptedException e)
 { System.out.println(e.getMessage()); } } }
public static void main(String[] args) {
Reader read = new Reader();
Writer write = new Writer();
 Thread r1 = new Thread(read);
r1.setName("Reader1");
Thread r2 = new Thread(read);
r2.setName("Reader2");
 Thread r3 = new Thread(read);
r3.setName("Reader3");
 Thread w1 = new Thread(write);
 w1.setName("Writer1");
```

```
Thread w2 = new Thread(write);
 w2.setName("Writer2");
 Thread w3 = new Thread(write);
w3.setName("Writer3");
 w1.start();
r1.start();
w2.start();
 r2.start();
 w3.start();
 r3.start(); } }
```

OUTPUT:

```
java -cp /tmp/Jn1yri5pEq/Main
Writing on the database......
Writing Finished!!!!!.
1 User Reading the Data.........
Reading Finished!!!!!!
1 User Reading the Data.........
Reading Finished!!!!!!

=== Code Execution Successful ===
```

```
java -cp /tmp/2ptxqM6dO8/Main
Thread Writer1 is WRITING: Good Morning
Thread Writer1 has finished WRITING
Thread Writer2 is WRITING: Good Morning
Thread Writer2 has finished WRITING
Thread Writer3 is WRITING: Good Morning
Thread Writer3 has finished WRITING
Thread Reader3 is READING: Good Morning
Thread Reader2 is READING: Good Morning
Thread Reader1 is READING: Good Morning
Thread Reader1 has FINISHED READING
Thread Reader3 has FINISHED READING
Thread Reader2 has FINISHED READING

=== Code Execution Successful ===
```