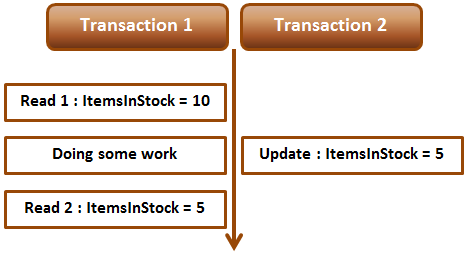
In this video we will discuss **non repeatable read concurrency problem with an example**.   
  
   
  
Non repeatable read problem happens when one transaction reads the same data twice and another transaction updates that data in between the first and second read of transaction one.    
  
We will use the following table **tblInventory** in this demo   
sql server non-repeatable read   
  
**The following diagram explains the problem :** Transaction 1 starts first. Reads ItemsInStock. Gets a value of 10 for first read. Transaction 1 is doing some work and at this point Transaction 2 starts and UpdatesItemsInStock to 5. Transaction 1 then makes a second read. At this point Transaction 1 gets a value of 5, reulting in non-repeatable read problem.   
  
   
  
**Non-repeatable read example :** Open 2 instances of SQL Server Management studio. From the first window execute Transaction 1 code and from the second window, execute Transaction 2 code. Notice that when Transaction 1 completes, it gets different values for read 1 and read 2, resulting in non-repeatable read. 

-- Transaction 1

Begin Transaction

Select ItemsInStock from tblInventory where Id = 1

-- Do Some work

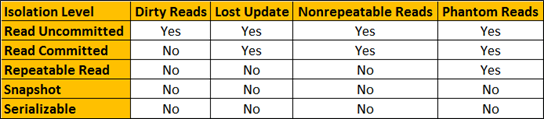
waitfor delay '00:00:10'

Select ItemsInStock from tblInventory where Id = 1

Commit Transaction

-- Transaction 2

Update tblInventory set ItemsInStock = 5 where Id = 1

Repeatable read or any other higher isolation level should solve the non-repeatable read problem.  
   
  
**Fixing non repeatable read concurrency problem :** To fix the non-repeatable read problem, set transaction isolation level of Transaction 1 to repeatable read. This will ensure that the data that Transaction 1 has read, will be prevented from being updated or deleted elsewhere. This solves the non-repeatable read problem.    
  
When you execute Transaction 1 and 2 from 2 different instances of SQL Server management studio, Transaction 2 is blocked until Transaction 1 completes and at the end of Transaction 1, both the reads get the same value for ItemsInStock.

-- Transaction 1

Set transaction isolation level repeatable read

Begin Transaction

Select ItemsInStock from tblInventory where Id = 1

-- Do Some work

waitfor delay '00:00:10'

Select ItemsInStock from tblInventory where Id = 1

Commit Transaction

-- Transaction 2

Update tblInventory set ItemsInStock = 5 where Id = 1