

# SQL Server Concurrency--Non-repeatable Reads

By Christian Bolton (<https://logicalread.com/author/christian-bolton/>), Justin Langford (<https://logicalread.com/author/justin-langford/>), Glenn Berry (<https://logicalread.com/author/glenn-berry/>), Gavin Payne (<https://logicalread.com/author/gavin-payne/>), Rob Farley (<https://logicalread.com/author/rob-farley/>) on February 9, 2013

Database concurrency ensures that when multiple operations are occurring at once, the final result is still in agreement — that they concur. The problems described in this three-part article only occur when multiple sessions are occurring at once in your database, including lost updates (Part 1) (<http://logicalread.com/sql-server-concurrency-lost-updates-w01/>), dirty reads (Part 2) (<http://logicalread.com/sql-server-concurrency-dirty-reads-w01/>) and non-repeatable reads.

**NOTE:**

*You can run the examples in this article against the AdventureWorks2012 database ([http://msdn.microsoft.com/en-us/library/ms124501\(v=sql.100\).aspx](http://msdn.microsoft.com/en-us/library/ms124501(v=sql.100).aspx)).*

What is a Non-repeatable Read?

A non-repeatable read is one in which data read twice inside the same transaction cannot be guaranteed to contain the same value. Depending on the isolation level, another transaction could have nipped in and updated the value between the two reads.

Non-repeatable reads occur because at lower isolation levels reading data only locks the data for the duration of the read, rather than for the duration of the transaction. Sometimes this behavior might be completely desirable. Some applications may want to know the absolute, real-time value, even mid transaction, whereas other types of transactions might need to read the same value multiple times.

Consider the following example. In Session 1 the transaction reads the data for the top five people from `Person.Person` and then waits for five seconds before repeating the step.

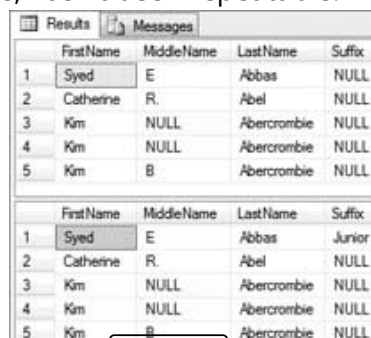
Execute the code in Session 1 before flipping to a second session and executing the code in Session 2 (code file `Ch6NonRepeatableReads.sql`):

```

/*SESSION 1*/
USE AdventureWorks2012;
SET TRANSACTION ISOLATION LEVEL
READ COMMITTED;
--REPEATABLE READ;
BEGIN TRANSACTION;
SELECT TOP 5
FirstName
,MiddleName
,LastName
,Suffix
FROM Person.Person
ORDER BY LastName;
WAITFOR DELAY '00:00:05.000';
SELECT TOP 5
FirstName
,MiddleName
,LastName
,Suffix
FROM Person.Person
ORDER BY LastName;
COMMIT TRANSACTION;
/*SESSION 2*/
USE AdventureWorks2012;
BEGIN TRANSACTION;
UPDATE Person.Person
SET Suffix = 'Junior'
WHERE LastName = 'Abbas'
AND FirstName = 'Syed';
COMMIT TRANSACTION;
/*
UPDATE Person.Person
SET Suffix = NULL
WHERE LastName = 'Abbas'
AND FirstName = 'Syed';
*/

```

Providing you execute the update in Session 2 in time, your results will match Figure 1. The first read from Session 1, Syed Abbas, had no suffix; but in the second read he's now Syed Abbas Junior. The first read, therefore, hasn't been repeatable.



	FirstName	MiddleName	LastName	Suffix
1	Syed	E	Abbas	NULL
2	Catherine	R.	Abel	NULL
3	Km	NULL	Abercrombie	NULL
4	Km	NULL	Abercrombie	NULL
5	Km	B	Abercrombie	NULL

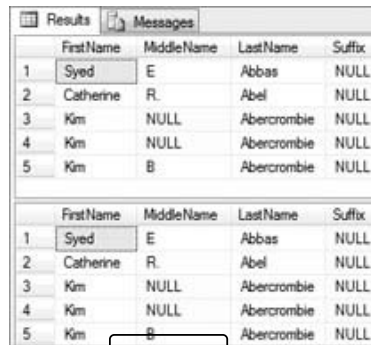
  

	FirstName	MiddleName	LastName	Suffix
1	Syed	E	Abbas	Junior
2	Catherine	R.	Abel	NULL
3	Km	NULL	Abercrombie	NULL
4	Km	NULL	Abercrombie	NULL
5	Km	B	Abercrombie	NULL

Figure 1.

You can use the commented-out code in Session 2 to reset the data. Execute this code now. To get a repeatable read, change the transaction isolation level in Session 1 as indicated here:

```
SET TRANSACTION ISOLATION LEVEL  
--READ COMMITTED;  
REPEATABLE READ;
```



The screenshot shows a SQL Server Enterprise Manager interface with a 'Results' pane. It contains two identical tables of employee data. The first table has columns: First Name, Middle Name, Last Name, and Suffix. The data rows are: 1 Syed E Abbas NULL, 2 Catherine R. Abel NULL, 3 Kim NULL Abercrombie NULL, 4 Kim NULL Abercrombie NULL, 5 Kim B Abercrombie NULL. The second table is identical.

	First Name	Middle Name	Last Name	Suffix
1	Syed	E	Abbas	NULL
2	Catherine	R.	Abel	NULL
3	Kim	NULL	Abercrombie	NULL
4	Kim	NULL	Abercrombie	NULL
5	Kim	B	Abercrombie	NULL

	First Name	Middle Name	Last Name	Suffix
1	Syed	E	Abbas	NULL
2	Catherine	R.	Abel	NULL
3	Kim	NULL	Abercrombie	NULL
4	Kim	NULL	Abercrombie	NULL
5	Kim	B	Abercrombie	NULL

**Figure 2.**

Now rerun Session 1 and Session 2 as before. You should notice that Session 2 has been blocked from performing its update until after the transaction has been completed. The first read in Session 1 is now repeatable. Your results from Session 1 should now match those in Figure 2.

Read Part 1 of this article about lost updates (<http://logicalread.com/sql-server-concurrency-lost-updates-w01/>) and Part 2 about dirty reads (<http://logicalread.com/sql-server-concurrency-dirty-reads-w01/>).

## Related Posts

SQL Server Concurrency--Dirty Reads (<https://logicalread.com/sql-server-concurrency-dirty-reads-w01/>)

SQL Server Concurrency--Lost Updates (<https://logicalread.com/sql-server-concurrency-lost-updates-w01/>)

Concurrency Models in SQL Server 2012 (<https://logicalread.com/sql-server-concurrency-models-mc03/>)

## Leave a Reply

### Comment

Name \*

Email \*

Website

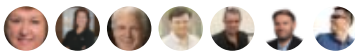
☐ Save my name, email, and website in this browser for the next time I comment.

## The Experts

### Featured Contributor

Janis Griffin (<https://logicalread.com/author/janis-griffin/>)

Oracle ACE; MySQL, SQL Server, Sybase and DB2 expert



## Follow SolarWinds



(<https://twitter.com/SolarWinds>)



(<https://www.facebook.com/SolarWinds>)



(<https://www.linkedin.com/company/solarwinds/>)



(<https://www.youtube.com/user/solarwindsinc>)

## RSS Feeds



All (<http://logicalread.com/feed>)



Databases (<http://logicalread.com/category/databases/feed>)



Systems (<http://logicalread.com/category/systems/feed>)



Security (<http://logicalread.com/category/security/feed>)



Cloud (<http://logicalread.com/category/cloud/feed>)



ITSM (<http://logicalread.com/category/itsm/feed>)

Subscribe to  
LogicalRead

Sponsored By



  
(https://twitter.com/SolarWinds)  
  
(https://www.facebook.com/SolarWinds)  
  
(https://www.linkedin.com/company/solarwinds/)  
  
(https://www.youtube.com/user/solarwindsinc)

Contribute articles  
(mailto:logicalread@solarwinds.com?subject=I'd%20like%20to%20contribute%20content!)

Give feedback  
(mailto:logicalread@solarwinds.com?subject=I'd%20like%20to%20share%20feedback%20on%20LogicalRead)

Contact us  
(mailto:logicalread@solarwinds.com)

Home  
(https://logicalread.com/)

SQL Server  
(https://logicalread.com/category/databases/sql-server/)

Oracle  
(https://logicalread.com/category/databases/oracle/)

SAP ASE  
(https://logicalread.com/category/databases/sybase/)

About SolarWinds  
(http://www.solarwinds.com/com)

Privacy Statement  
(https://www.solarwinds.com/legal/privacy-statement/)

Terms of Use  
(https://www.solarwinds.com/legal/terms-of-use/)

Facebook

SolarWinds uses cookies on its websites to make your online experience easier and better. By using our website, you consent to our use of cookies. For more information on cookies, see our cookie policy. (https://www.solarwinds.com/legal/privacy)