-





­

Assignment Document:

Isolation Level Sql Server

Version: < Sql Server 2008>/ASSIGNMENT/xxxx/x.x

Date: 12/10/2012

Cognizant

500 Glen Pointe Center West

Teaneck, NJ 07666

Ph: 201-801-0233  
[www.cognizant.com](http://www.cognizant.com)

Contents

[Topic: <ISOLATION LEVEL> 3](#_Toc300153669)

[Hands-On Exercises 3](#_Toc300153670)

Hands-On Exercise 1: <ISOLATION LEVEL>…………………………………………………….3

Topic: <ISOLATION LEVEL>

Total Completion Time:9 minutes

### Hands-On Exercises

Hands-On Exercise 1: <ISOLATION LEVEL>

Estimated Completion Time: 3 Minutes

Step 1: Create a sample table and populate it.

Step 2: Run the two SET of queries given below in two different sessions.

Step 3: Understand the concurrency problems in different ISOLATION LEVELs.

-----------------------------------------------------------------------------

--Session 1 Queries

-----------------------------------------------------------------------------

-- Creation of the sample database

CREATE DATABASE TEST\_SAMPLE

USE TEST\_SAMPLE

GO

--creation of the sample tables

CREATE TABLE IsolationTests

(

Id INT IDENTITY,

Col1 INT,

Col2 INT,

Col3 INT

)

CREATE TABLE mytab

(

id INT,

variable VARCHAR(10)

)

--Insertion of data

INSERT INTO IsolationTests(Col1,Col2,Col3)

SELECT 1,2,3

UNION ALL SELECT 1,2,3

UNION ALL SELECT 1,2,3

UNION ALL SELECT 1,2,3

UNION ALL SELECT 1,2,3

UNION ALL SELECT 1,2,3

UNION ALL SELECT 1,2,3

INSERT INTO mytab(id,variable)

SELECT 1,'a'

UNION ALL SELECT 2,'b'

UNION ALL SELECT 3,'c'

UNION ALL SELECT 4,'d'

-----------------------------------------------------------------------------

--Identify the present ISOLATION LEVEL upon which you are working by running:

DBCC UserOptions

-----------------------------------------------------------------------------

--Read Uncommited

--Query1

BEGIN TRAN

UPDATE IsolationTests SET Col1 = 100

--Simulate having some intensive processing here with a wait

WAITFOR DELAY '00:00:10'

ROLLBACK

SELECT \* FROM IsolationTests

-----------------------------------------------------------------------------

--Read Committed

--Query1

BEGIN TRAN

UPDATE IsolationTests SET Col1 = 2

--Simulate having some intensive processing here with a wait

WAITFOR DELAY '00:00:10'

ROLLBACK

SELECT \* FROM IsolationTests

---Showing non REPEATABLE READ problem

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRAN

SELECT \* from mytab

--UPDATE mytab SET variable='c' WHERE id=2

WAITFOR DELAY '00:00:09'

SELECT \* from mytab

COMMIT TRAN

--showing phantom read problem

SET TRANSACTION ISOLATION LEVEL READ COMMITTED

BEGIN TRAN

SELECT \* from mytab

--UPDATE mytab SET variable='c' WHERE id=2

WAITFOR DELAY '00:00:09'

SELECT \* from mytab

COMMIT TRAN

-----------------------------------------------------------------------------

--Reapeatable Read

-----Query1

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRAN

SELECT \* FROM IsolationTests

WAITFOR DELAY '00:00:10'

SELECT \* FROM IsolationTests

ROLLBACK

SELECT \* FROM IsolationTests

--new rows------------(Phantom read)

SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

BEGIN TRAN

SELECT \* from mytab

--UPDATE mytab SET variable='c' WHERE id=2

WAITFOR DELAY '00:00:05'

SELECT \* from mytab

COMMIT TRAN

-----------------------------------------------------------------------------

--Serializable

-----query1

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

BEGIN TRAN

SELECT \* FROM IsolationTests

WAITFOR DELAY '00:00:10'

SELECT \* FROM IsolationTests

ROLLBACK

SELECT \* FROM IsolationTests

-----------------------------------------------------------------------------

--Snapshot

ALTER DATABASE IsolationTests

SET ALLOW\_SNAPSHOT\_ISOLATION ON

--SELECT \* from IsolationTests

BEGIN TRAN

UPDATE IsolationTests SET Col1=10 WHERE Id=8

WAITFOR DELAY '00:00:10'

COMMIT TRAN

SELECT \* FROM IsolationTests

--------------------------------------------------------------------

--Lost UPDATE

--Query1

BEGIN TRAN

SELECT \* FROM IsolationTests

UPDATE IsolationTests SET Col1=20 WHERE Id=7

SELECT \* FROM IsolationTests

WAITFOR DELAY '00:00:10'

COMMIT TRAN

DBCC useroptions

------------------------------------------------------------------

------------------------------------------------------------------

--SNAPSHOT Alternative

--Query1 for SNAPSHOT isolation

alter database master

SET allow\_SNAPSHOT\_isolation on

SET TRANSACTION ISOLATION LEVEL SNAPSHOT

BEGIN TRANSACTION

SELECT \* from isolationtests

UPDATE IsolationTests SET Col1=99 WHERE Id=7

WAITFOR DELAY '00:00:10'

SELECT \* from IsolationTests

COMMIT TRANSACTION

-----------------------------------------------------------------------------

--Session 2 Queries

-----------------------------------------------------------------------------

--Read Uncommited

--Query2

SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED

SELECT \* FROM IsolationTests

-----------------------------------------------------------------------------

--Read Committed

--Query2

SELECT \* FROM IsolationTests

--showing non repeatbale read problem

BEGIN TRAN

SELECT \* from mytab

--INSERT into mytab(id,variable) VALUES(4,'y')

UPDATE mytab SET variable='c' WHERE id=2

SELECT \* from mytab

COMMIT TRAN

--showing phantom read problem

BEGIN TRAN

SELECT \* from mytab

INSERT into mytab(id,variable) VALUES(4,'y')

--UPDATE mytab SET variable='c' WHERE id=2

SELECT \* from mytab

COMMIT TRAN

-----------------------------------------------------------------------------

--Reapeatable Read

--Query2

UPDATE IsolationTests SET Col1 = -1

---new rows------showing phantom read problem

BEGIN TRAN

SELECT \* from mytab

INSERT into mytab(id,variable) VALUES(3,'z')

--UPDATE mytab SET variable='d' WHERE id=2

SELECT \* from mytab

COMMIT TRAN

-----------------------------------------------------------------------------

--Serializable

--Query2

INSERT INTO IsolationTests(Col1,Col2,Col3)

VALUES (100,100,100)

-----------------------------------------------------------------------------

--Snapshot

ALTER DATABASE IsolationTests

SET ALLOW\_SNAPSHOT\_ISOLATION ON

--SELECT \* from IsolationTests

BEGIN TRAN

UPDATE IsolationTests SET Col1=10 WHERE Id=8

WAITFOR DELAY '00:00:15'

COMMIT TRAN

-------------------------------------------------------

--Lost Update

--Query 2

BEGIN TRAN

SELECT \* FROM IsolationTests

UPDATE IsolationTests SET Col1=10 WHERE Id=7

SELECT \* FROM IsolationTests

--WAITFOR DELAY '00:00:10'

COMMIT TRAN

SELECT \* FROM IsolationTests

------------------------------------------------------------------

------------------------------------------------------------------

--SNAPSHOT Alternative

--Query2

UPDATE IsolationTests SET Col1=150 WHERE Id=7

--Query3

SET TRANSACTION ISOLATION LEVEL SNAPSHOT

BEGIN TRANSACTION

SELECT \* from isolationtests

UPDATE IsolationTests SET Col1=199 WHERE Id=7

--WAITFOR DELAY '00:00:10'

SELECT \* from IsolationTests

COMMIT TRANSACTION

Summary

You have just learnt:

* The different concurrency problems that occur.
* How the different ISOLATION LEVELs work to overcome those problems.