

SQL Server Tutorial

Part 72 - SQL Server lost update problem

Venkat

PRAGIM Technologies

kudvenkat@gmail.com

<http://csharp-video-tutorials.blogspot.com>

PRAGIM Technologies | 9900113931 | www.pragimtech.com | www.facebook.com/pragimtech

In this session we will learn

- The lost update problem

Link to Dot Net Basics, ASP.NET, C#, ADO.NET and SQL Server video series

<http://www.youtube.com/user/kudvenkat/playlists>

Suggested Videos

Part 69 - Merge in SQL Server

Part 70 - SQL Server concurrent transactions

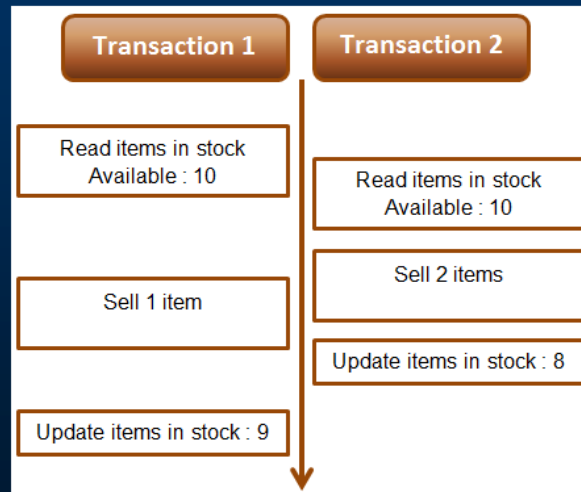
Part 71 - SQL Server dirty read example

SQL Server Lost Update Problem

If you are in need of the DVD with all the videos and PPT's, please visit
<http://pragimtech.com/order.aspx>

Lost update problem happens when 2 transactions read and update the same data

Id	Product	ItemsInStock
1	iPhone	10



SQL Server Lost Update Problem

```
-- Transaction 1
Begin Tran
Declare @ItemsInStock int

Select @ItemsInStock = ItemsInStock
from tblInventory where Id=1

-- Transaction takes 10 seconds
Waitfor Delay '00:00:10'
Set @ItemsInStock = @ItemsInStock - 1

Update tblInventory
Set ItemsInStock = @ItemsInStock
where Id=1

Print @ItemsInStock
Commit Transaction

-- Transaction 2
Begin Tran
Declare @ItemsInStock int

Select @ItemsInStock = ItemsInStock
from tblInventory where Id=1

-- Transaction takes 1 second
Waitfor Delay '00:00:1'
Set @ItemsInStock = @ItemsInStock - 2

Update tblInventory
Set ItemsInStock = @ItemsInStock
where Id=1

Print @ItemsInStock
Commit Transaction
```

Id	Product	ItemsInStock
1	iPhone	10

Transaction 1 is processing an order for 1 iPhone, while Transaction 2 is processing an order for 2 iPhones. At the end of both the transactions ItemsInStock must be 7, but we have a value of 9. This is because Transaction 1 silently overwrites the update of Transaction 2. This is called the lost update problem

SQL Server Lost Update Problem

Isolation Level	Dirty Reads	Lost Update	Nonrepeatable Reads	Phantom Reads
Read Uncommitted	Yes	Yes	Yes	Yes
Read Committed	No	Yes	Yes	Yes
Repeatable Read	No	No	No	Yes
Snapshot	No	No	No	No
Serializable	No	No	No	No

Both Read Uncommitted and Read Committed transaction isolation levels have the lost update side effect.

Repeatable Read, Snapshot, and Serializable isolation levels does not have this side effect.

The repeatable read isolation level uses additional locking on rows that are read by the current transaction, and prevents them from being updated or deleted elsewhere. This solves the lost update problem.

Additional Resources

PRAGIM Home Page:

www.PragimTech.com

Resources:

C#, ADO.NET, ASP.NET, SQL Server & MVC youtube Playlists

<http://www.youtube.com/user/kudvenkat/playlists>

Code samples and text version of all the videos on my blog

<http://www.csharp-video-tutorials.blogspot.com>

To receive email alerts when new videos are uploaded, please subscribe to my YOUTUBE channel

www.YouTube.com/kudvenkat