READPAST & Furious: Transactions, Locking and Isolation

Mark Broadbent

SQLCloud

Agenda

TRANSACTIONS

- Structure, Scope and Management

LOCKING

- Compatibility, Multi Granularity and Escalation
- NOLOCK, READPAST and UPDLOCK

ISOLATION

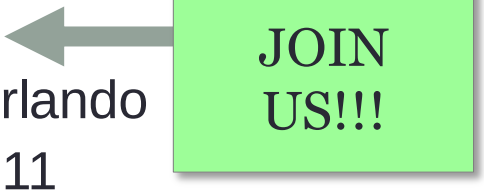
- Snapshot Isolation and Read Committed Snapshot
- Rolling Database Snapshots

A bit more about little old me!

More than 21 years in IT and more than 15 years using SQL Server.
Worked at many large global corporations and SMEs such as Microsoft, Nokia, Hewlett Packard and Encyclopaedia Britannica.

Blog: **tenbulls.co.uk**

Cambridgeshire SQL UG Leader **sqlcambs.org.uk**



Presented at SQLBits 7, 8 (and now 9), SQL Rally Orlando

Awarded Microsoft Community Contributor Award 2011

MCITP DB Development SQL 2008

MCITP DBA SQL 2008/ 2005 and MCDBA SQL 2000

Microsoft Certified Application Developer (C# .net)

Microsoft Certified Systems Engineer + Internet

Participate on #sqlhelp, MSDN Forums

LinkedIn SQL Server Scripting Group **http://linkd.in/pfBkRI**

Transactions basically…

Provide Atomicity (all or nothing) …or does it!

Can be defined as Implicit (evil), Explicit (declarative) or Auto-Commit (default)

Effect duration of locks

Transactions can have batch scope (MARS), be distributed and even be bound across sessions or servers.

Can be really confusing!

Transactions are easy-peasy…

```
BEGIN TRAN transaction_1 WITH MARK 'restorepoint'
    BEGIN TRAN
       --do something
    COMMIT TRAN
    SAVE TRAN savepoint
    BEGIN TRAN transaction_3
       BEGIN TRAN
             --do something else
       COMMIT
       IF {something_wrong} THEN ROLLBACK TRAN savepoint
    COMMIT
COMMIT
```

Transaction Name

Transaction Mark

Savepoint

Nested Transaction

Transactions Demo

Locks, the Lock Manager and Locking

Locks are memory structures only and can be converted or escalated
…have a compatibility or incompatibility with other locks
…are taken depending upon the Isolation level
…can cause blocking or deadlocks
…will wait if they are incompatible

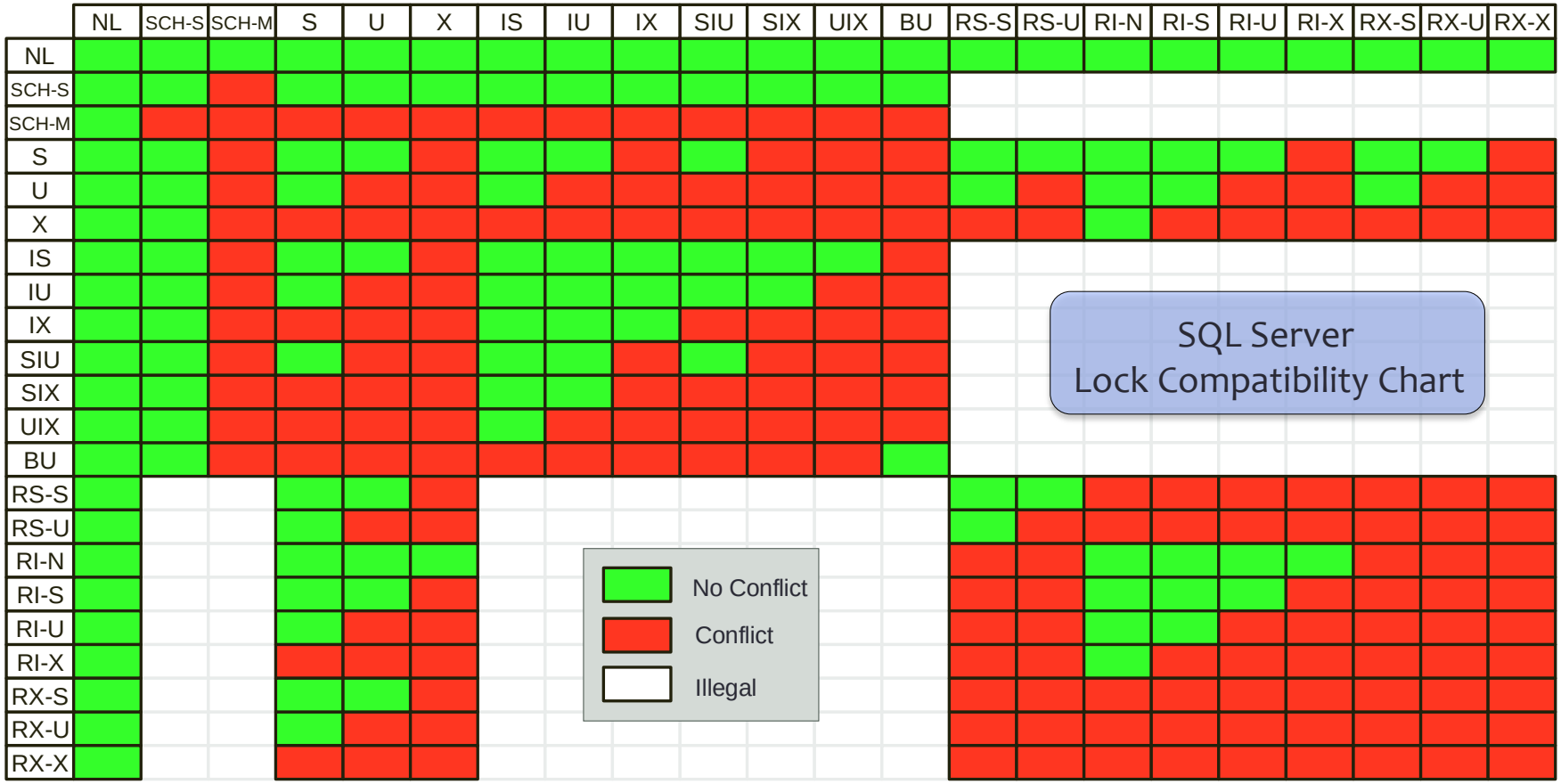
Lock Manager compares locks ONLY on same resource
… and that's why so intent locks are needed (granular)

Lock escalation will occur for performance and memory savings
…and escalate straight to Table by default (not partitions)!!!
…but you can turn off ALL escalation (TF1211), due to # locks (TF1224) or artificially by taking IS on table or by ALTER statement <table> SET (LOCK_ESCALATION = auto|table|disable)

	NL	SCH-S	SCH-M	S	U	X	IS	IU	IX	SIU	SIX	UIX	BU	RS-S	RS-U	RI-N	RI-S	RI-U	RI-X	RX-S	RX-U	RX-X
NL	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
SCH-S	N	N	C	N	N	N	N	N	N	N	N	N	N	I	I	I	I	I	I	I	I	I
SCH-M	N	C	C	C	C	C	C	C	C	C	C	C	C	I	I	I	I	I	I	I	I	I
S	N	N	C	N	N	C	N	N	C	N	C	C	C	N	N	N	N	N	C	N	N	C
U	N	N	C	N	C	C	N	C	C	C	C	C	C	N	C	N	N	C	C	N	C	C
X	N	N	C	C	C	C	C	C	C	C	C	C	C	C	C	N	C	C	C	C	C	C
IS	N	N	C	N	N	C	N	N	N	N	N	N	C	I	I	I	I	I	I	I	I	I
IU	N	N	C	N	C	C	N	N	N	N	N	C	C	I	I	I	I	I	I	I	I	I
IX	N	N	C	C	C	C	N	N	N	C	C	C	C	I	I	I	I	I	I	I	I	I
SIU	N	N	C	N	C	C	N	N	C	N	C	C	C	I	I	I	I	I	I	I	I	I
SIX	N	N	C	C	C	C	N	N	C	C	C	C	C	I	I	I	I	I	I	I	I	I
UIX	N	N	C	C	C	C	N	C	C	C	C	C	C	I	I	I	I	I	I	I	I	I
BU	N	N	C	C	C	C	C	C	C	C	C	C	N	I	I	I	I	I	I	I	I	I
RS-S	N	I	I	N	N	C	I	I	I	I	I	I	I	N	N	C	C	C	C	C	C	C
RS-U	N	I	I	N	C	C	I	I	I	I	I	I	I	N	C	C	C	C	C	C	C	C
RI-N	N	I	I	N	N	N	I	I	I	I	I	I	I	C	C	N	N	N	N	C	C	C
RI-S	N	I	I	N	N	C	I	I	I	I	I	I	I	C	C	N	N	N	C	C	C	C
RI-U	N	I	I	N	C	C	I	I	I	I	I	I	I	C	C	N	N	C	C	C	C	C
RI-X	N	I	I	C	C	C	I	I	I	I	I	I	I	C	C	N	C	C	C	C	C	C
RX-S	N	I	I	N	N	C	I	I	I	I	I	I	I	C	C	C	C	C	C	C	C	C
RX-U	N	I	I	N	C	C	I	I	I	I	I	I	I	C	C	C	C	C	C	C	C	C
RX-X	N	I	I	C	C	C	I	I	I	I	I	I	I	C	C	C	C	C	C	C	C	C

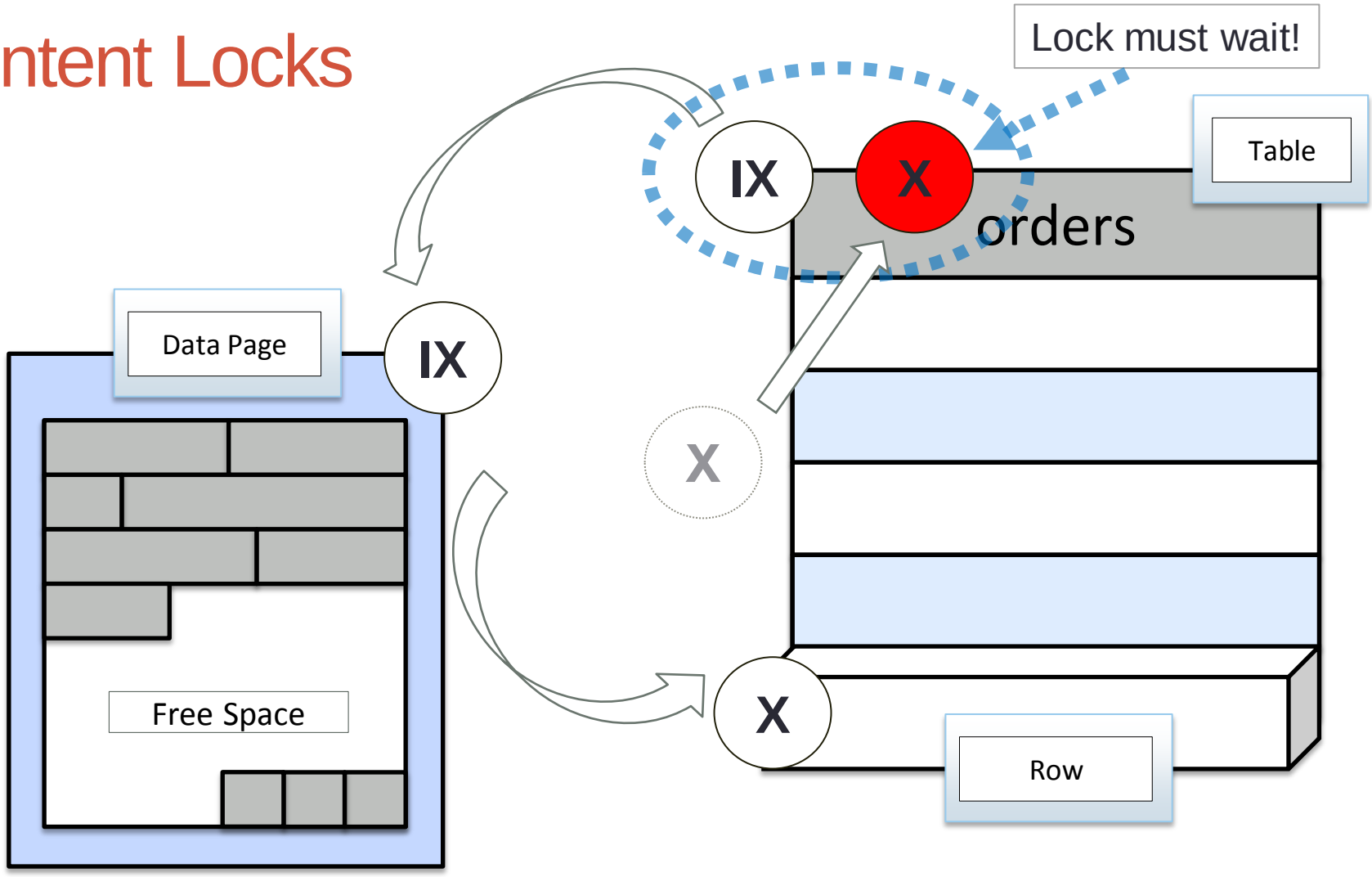
Key

N	No Conflict	SIU	Share with Intent Update
I	Illegal	SIX	Shared with Intent Exclusive
C	Conflict	UIX	Update with Intent Exclusive
		BU	Bulk Update
NL	No Lock	RS-S	Shared Range-Shared
SCH-S	Schema Stability Locks	RS-U	Shared Range-Update
SCH-M	Schema Modification Locks	RI-N	Insert Range-Null
S	Shared	RI-S	Insert Range-Shared
U	Update	RI-U	Insert Range-Update
X	Exclusive	RI-X	Insert Range-Exclusive
IS	Intent Shared	RX-S	Exclusive Range-Shared
IU	Intent Update	RX-U	Exclusive Range-Update
IX	Intent Exclusive	RX-X	Exclusive Range-Exclusive



NL	No Lock	IX	Intent Exclusive	RI-S	Insert Range-Shared
SCH-S	Schema Stability Lock	SIU	Share with Intent Update	RI-U	Insert Range-Update
SCH-M	Schema Modification Lock	SIX	Share with Intent Exclusive	RI-X	Insert Range-Exclusive
S	Shared	UIX	Update with Intent Exclusive	RX-S	Exclusive Range-Shared
U	Update	BU	Bulk Update	RX-U	Exclusive Range-Update
X	Exclusive	RS-S	Shared Range-Shared	RX-X	Exclusive Range-Exclusive
IS	Intent Shared	RS-U	Shared Range-Update		
IU	Intent Update	RI-N	Insert Range-Null		

Intent Locks



NOLOCK (a Wolf in Sheep's clothing?)

Should use READUNCOMMITTED hint instead.

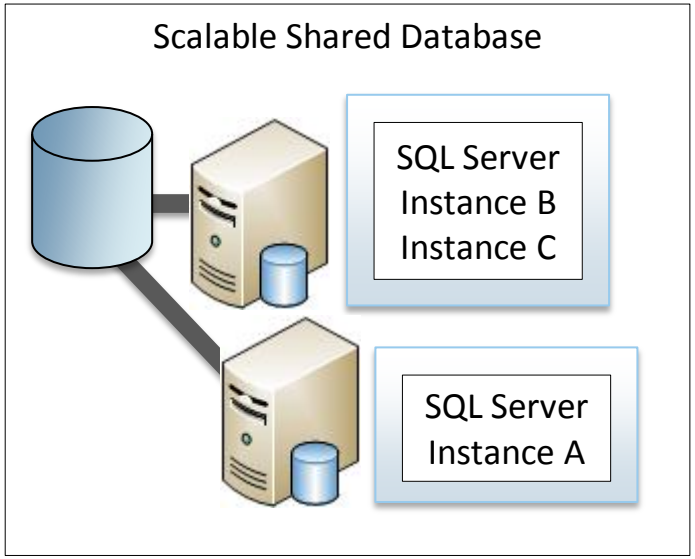
Doesn't acquire Shared locks so is able to perform DIRTY READS, NON-REPEATABLE READS, PHANTOMS

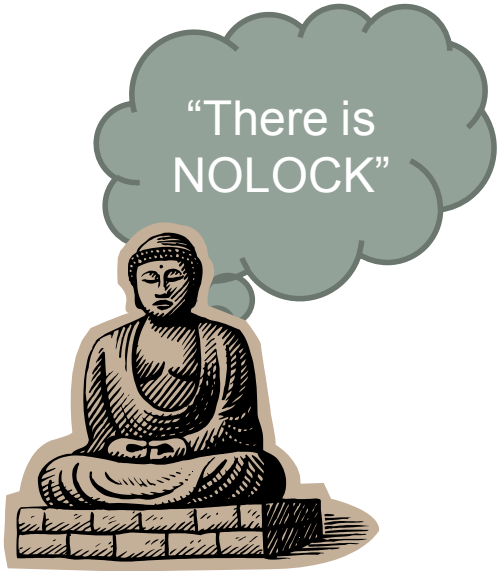
…and I bet you didn't know it can also return DUPLICATE READS!

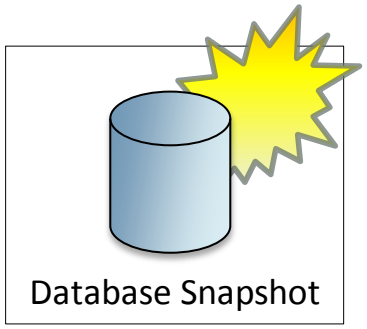
It DOES NOT give "Oracle style" concurrency

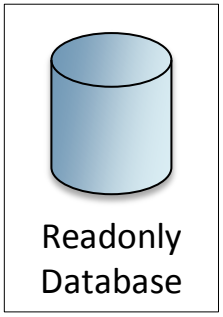
When can this ever be acceptable?

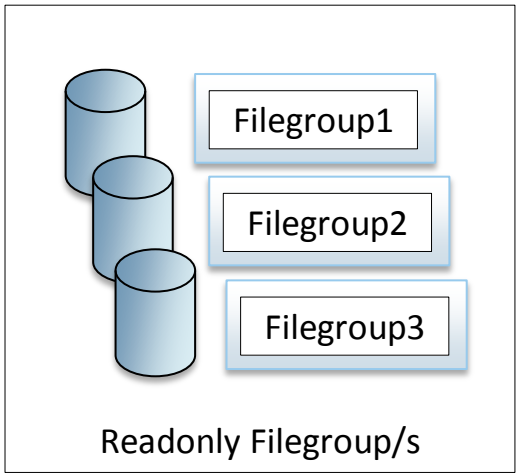
Well since you mention it…





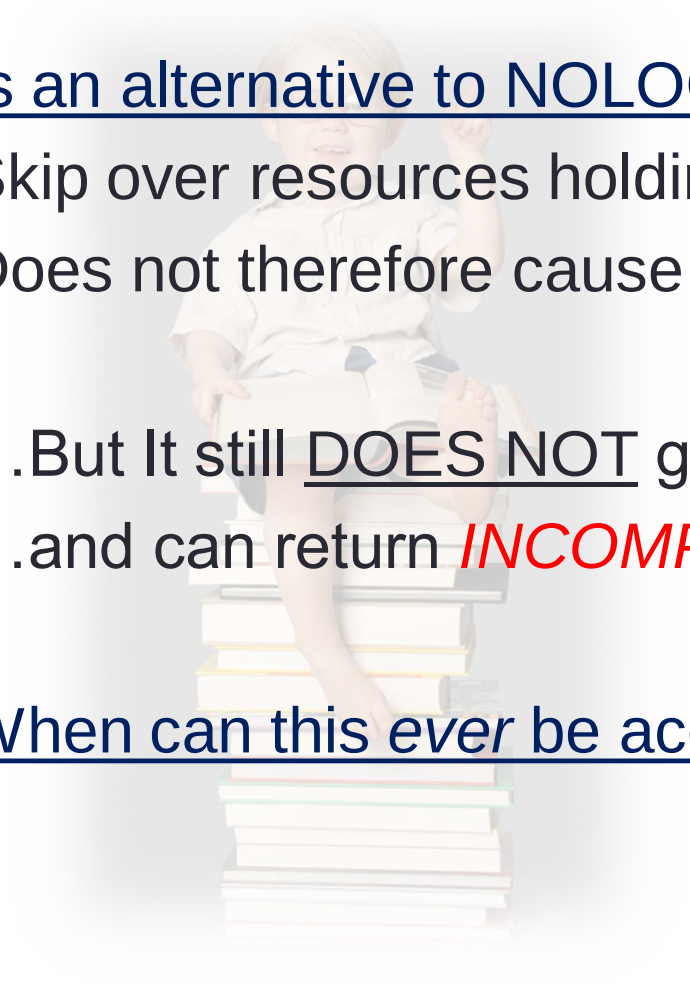






"**When should you use NOLOCK?**" http://bit.ly/rdGzow

READPAST & (your boss) may be Furious!



Is an alternative to NOLOCK/ READUNCOMMITTED.

Skip over resources holding incompatible (to S) locks.

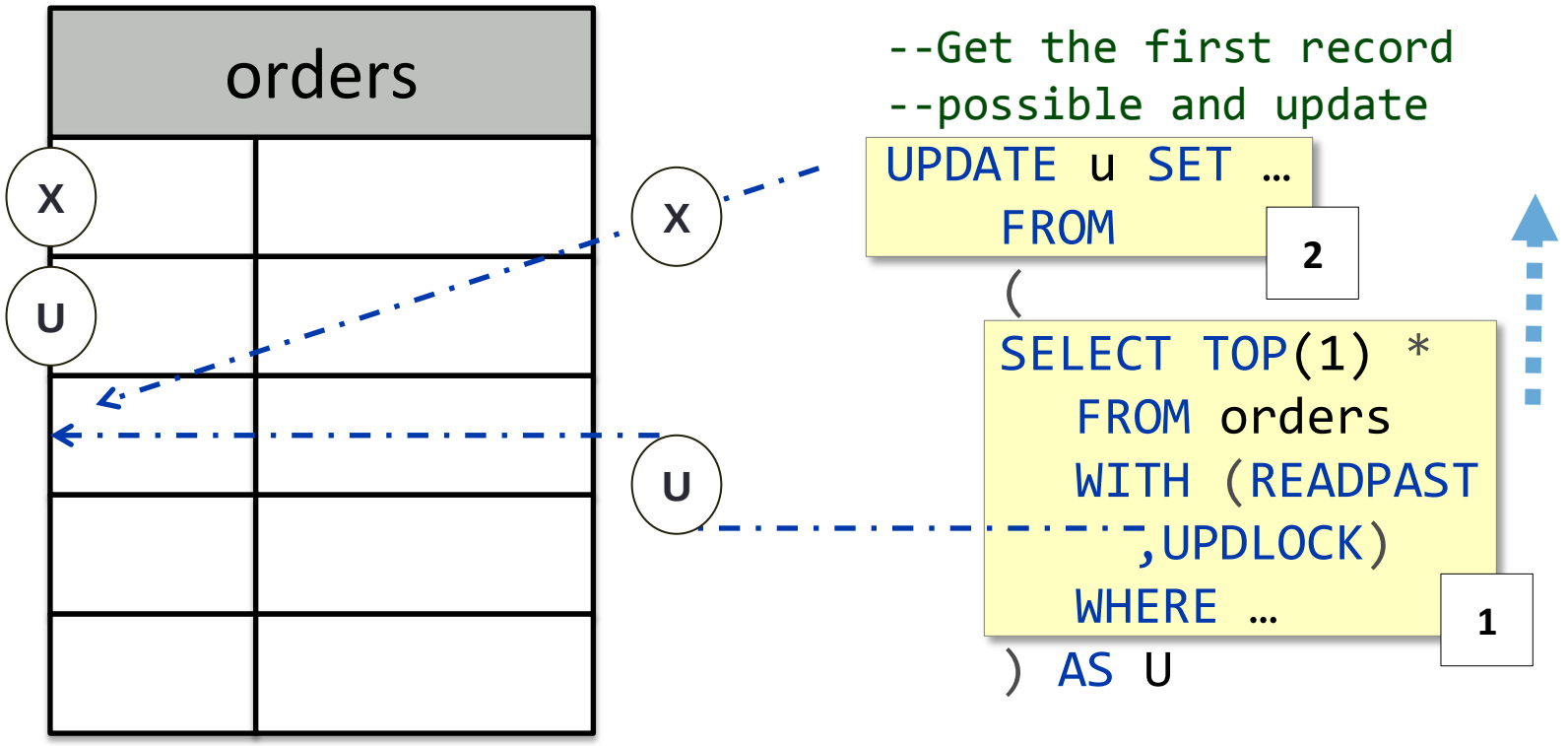
Does not therefore cause dirty or duplicate reads. (yay!)

…But It still DOES NOT give “Oracle style” concurrency

…and can return *INCOMPLETE DATA SETS!*

When can this *ever* be acceptable?

Using READPAST and Table Queues



Locking Demo

Read Committed Snapshot vs Snapshot Isolation

SI provides isolation at the transaction level, RCS provides isolation at the statement level

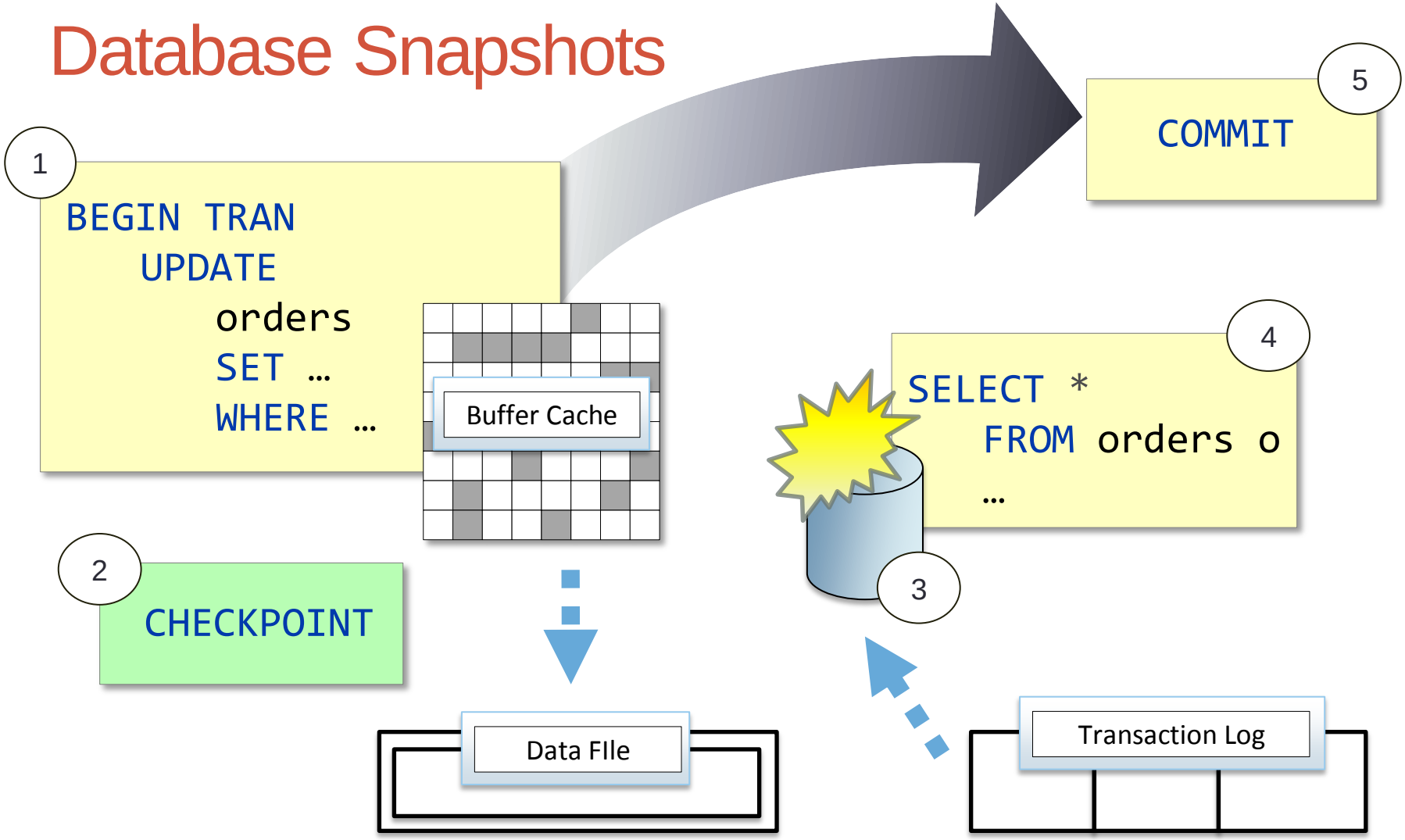
SI must be explicitly SET in each connection, for RCS it becomes the new default

Enabling SI level requires no active transactions in order to transition. Enabling RCS requires Exclusive Transaction Workspace lock (and therefore no other connections to DB).

RCS not allowed on *master*, *tempdb* and *msdb*, SI is allowed

SI implements automatic update conflict detection

Database Snapshots



Snapshots Demo

References and Thanks

Professional SQL Server 2008 Internals and Troubleshooting – Locking and Latches Chapter - James Rowland Jones

Kalen Delaney, SQLPASS Summit 2011 DBA301P - Locking and Blocking and Row Versions, Oh My! – DVD

Jose Barreto's Blog
http://blogs.technet.com/b/josebda/archive/2009/03/19/sql-server-2008-locking.aspx

A special thanks to :-

Paul White @sql_kiwi http://sqlblog.com/blogs/paul_white

Benjamin Neverez @BenjaminNeverez http://www.benjaminnevarez.com/