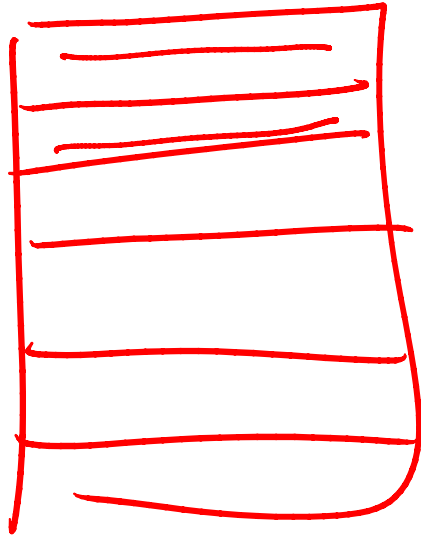


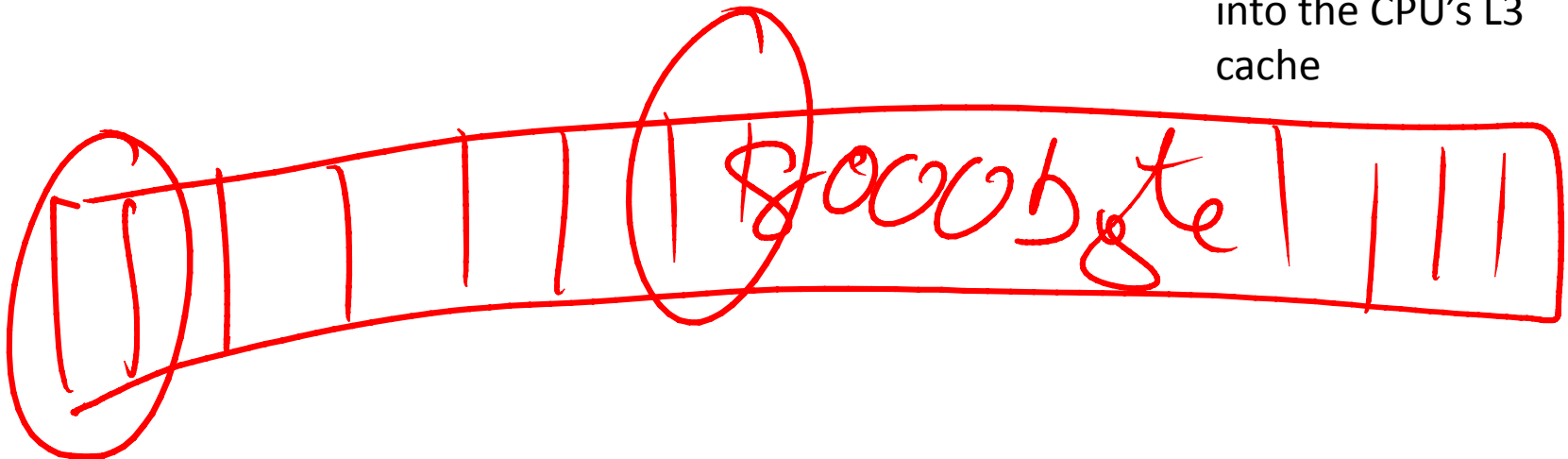
#sqlhelp

CACHE LINE INVALIDATION

CACHE LINES
64b / 128b



M1S06 Discussing
why the NULL
bitmap is a perf
optimization by
avoiding having to
read record parts
into the CPU's L3
cache



DBCC CSINDEX

How to look inside columnstore indexes. See

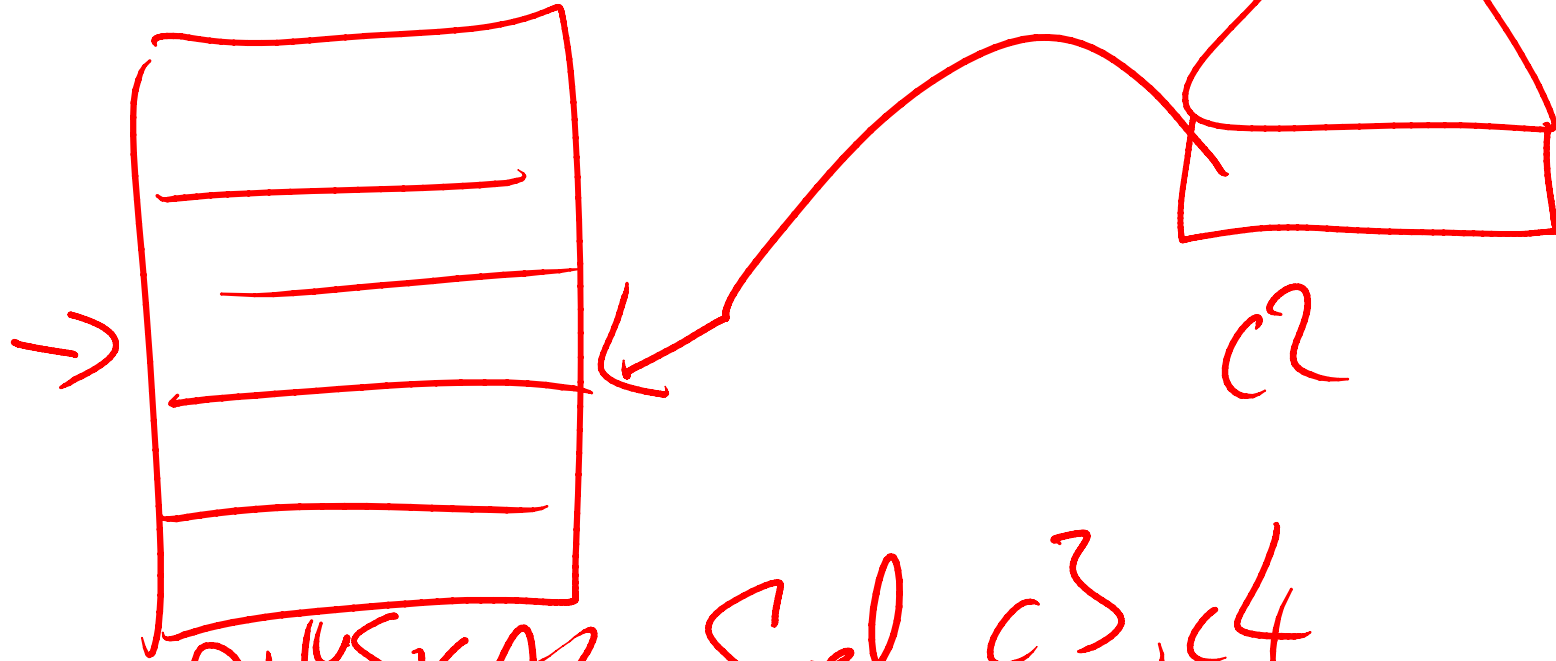
<http://www.nikoport.com/2013/11/07/clustered-columnstore-indexes-part-21-dbcc-csindex/>

IBDB

INS = FREE SPACE
(CACHE)

M1S9 Insert of a new record into a heap depends on the free space cache. A small cache of PFS values (last 32/64 pages seen) that are used to find a page that has enough space. If none is found, a new page is allocated.

$c_1, c_2, \dots, c_{10}$



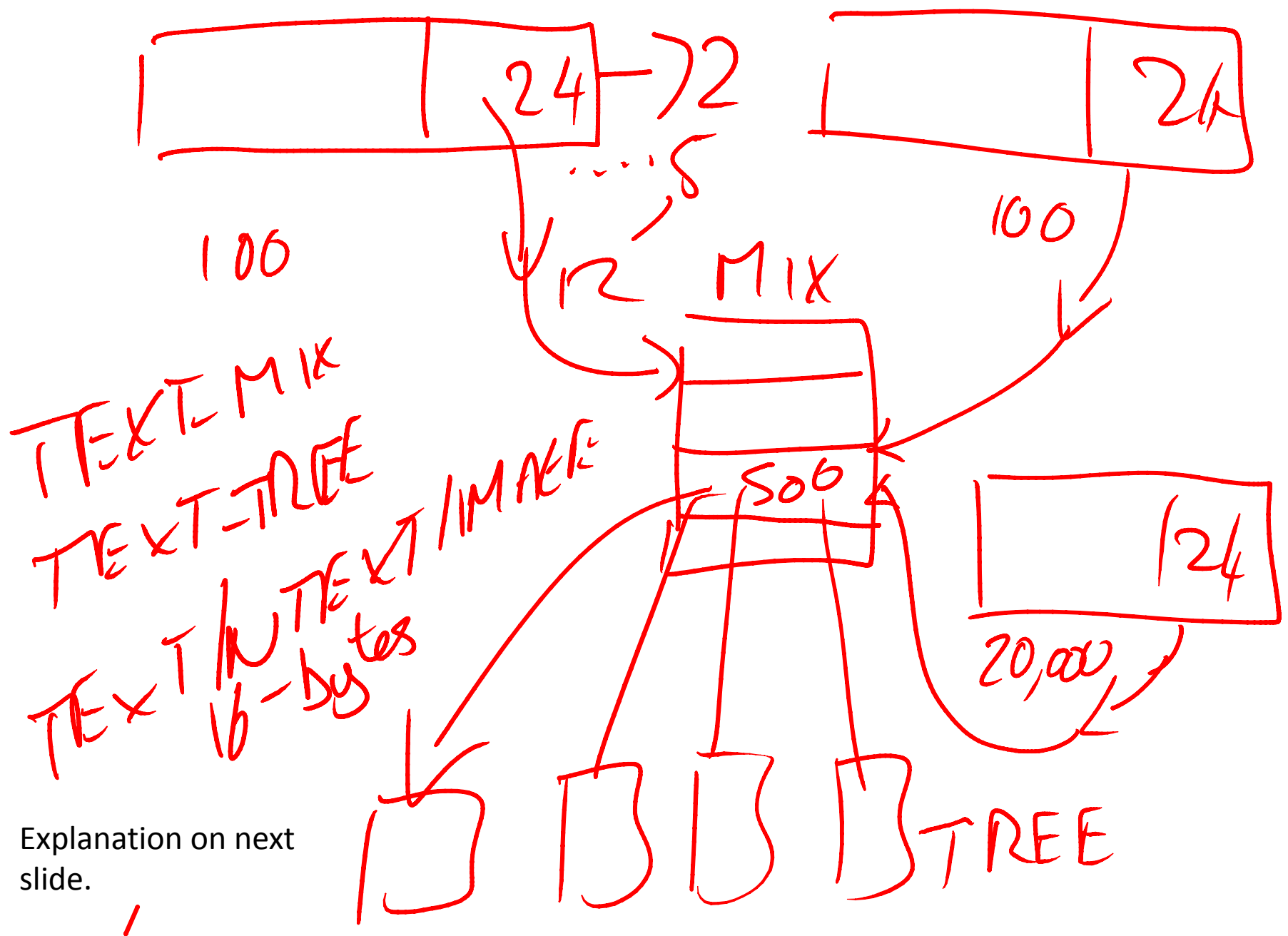
HL: PHYSICAL
RID

CL: LOGICAL
RID

Set c_3, c_4

Where $c_2 = X$

M1S09 Explaining NC index back-links to the data records as part of describing why forwarding/forwarded records exist. Without them, moving a heap row would entail changing all NC index records with a back-link to that row.



100	1900 byte
-----	-----------

This image shows a single sheet of white paper with three horizontal red lines, resembling notebook paper. The lines are evenly spaced and extend across most of the width of the page. There is no handwriting or other markings on the paper.

4

M1S12 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.

A hand-drawn sketch of a spiral-bound notebook. The notebook is oriented vertically. The top page is visible and has the number '128' written on it. The bottom page is also visible and has the number '64' written on it. The spiral binding is on the right side of the notebook. The drawing is done in a simple, sketchy style with black lines.

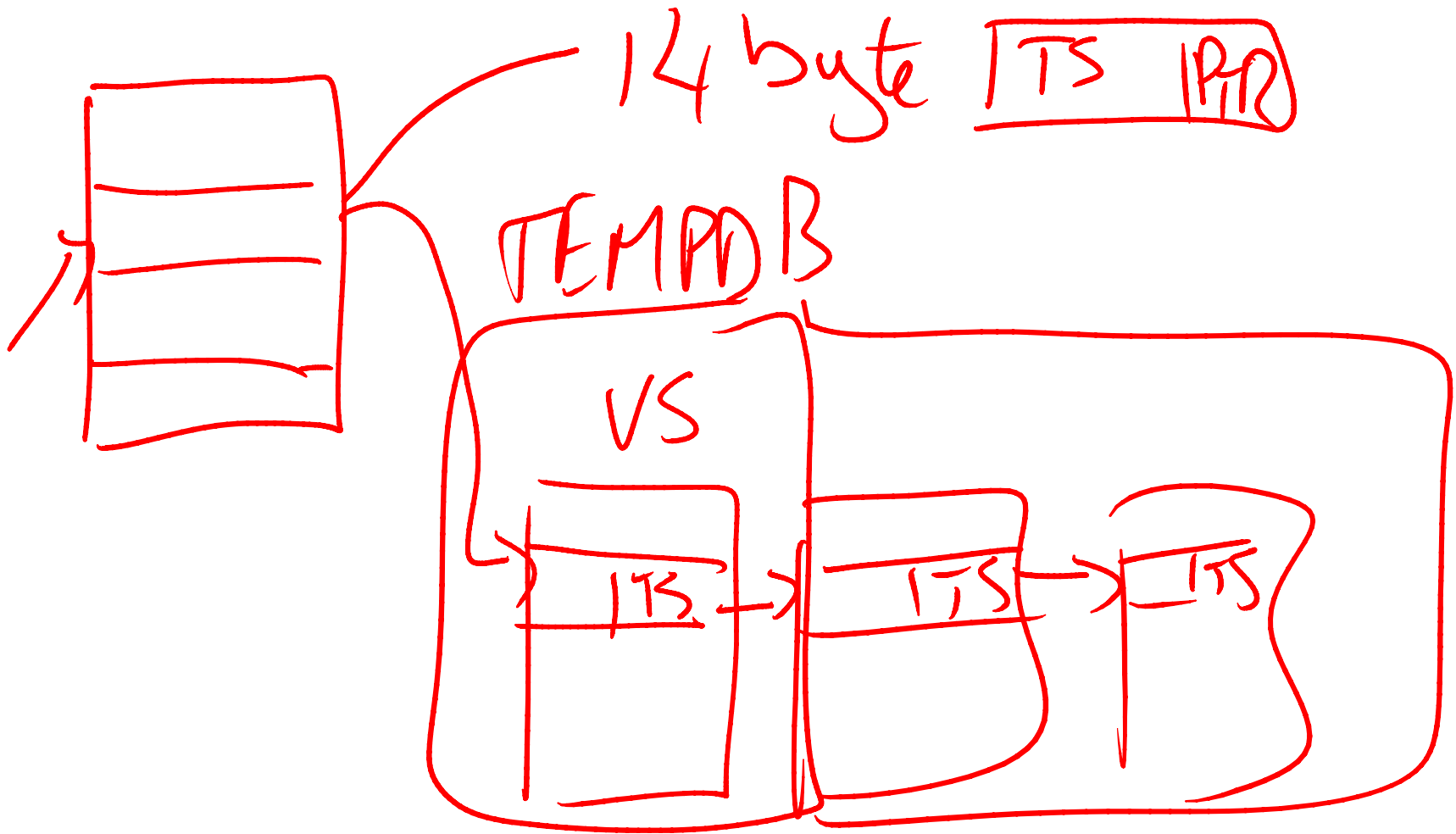
M1S11 Row overflow allows records
to be longer than the in-row
minimum of 8060 bytes

8060

c1 vc(5000), c2 vc(5000)

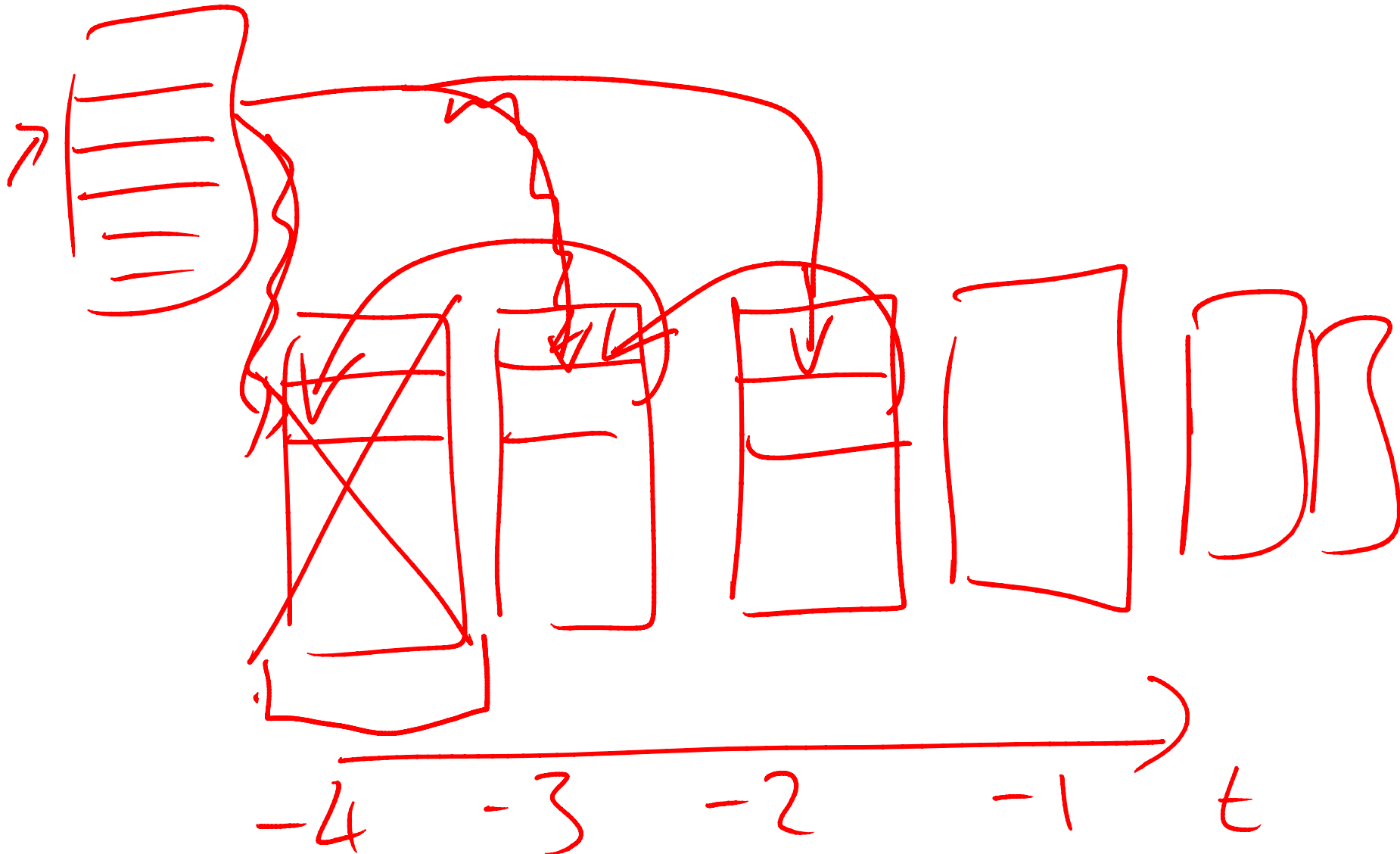
Row OVERFLOW
MARC RASMUSSEN
BLOB

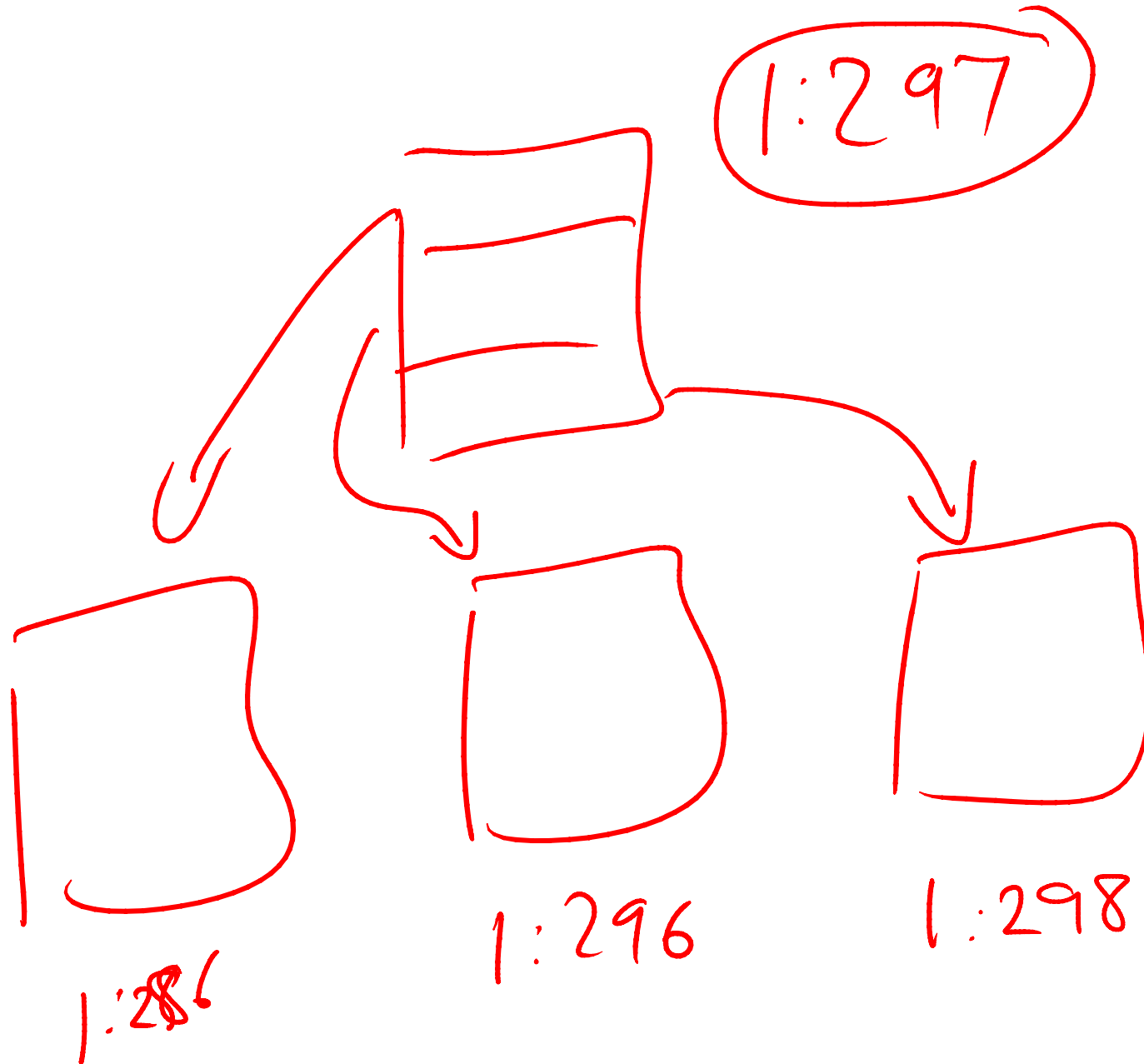
Google Marc Rasmussen for his blog. He's blogged text
tree structures.



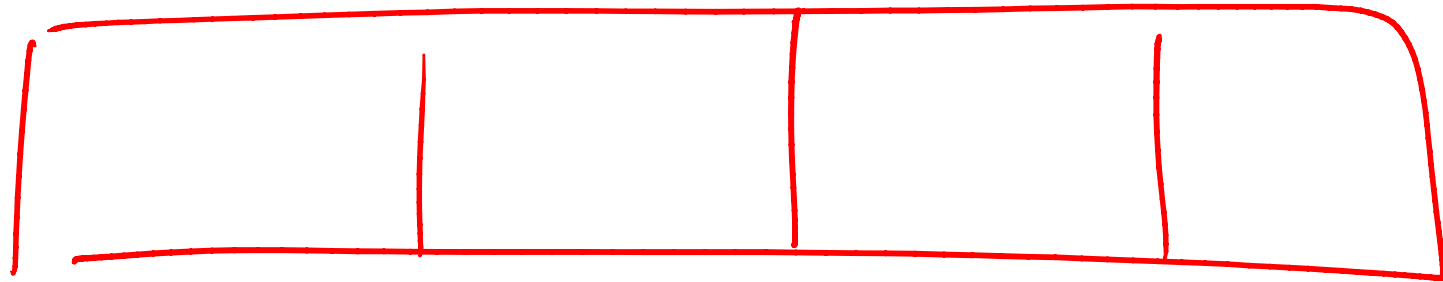
M1S14 When an update occurs to a record and snapshot isolation is enabled, the pre-change record is copied into the version store in tempdb. The changed record has a 14-byte tag on the end with the timestamp and pointer to previous record version. A chain of previous versions is possible.

M1S14 & M2S35 Explaining how there's a new portion of the version store created every minute.





M1S22 Showing the clustered index structure generated by the page/record demo.



0 — 7 8 — 15 16 — 23

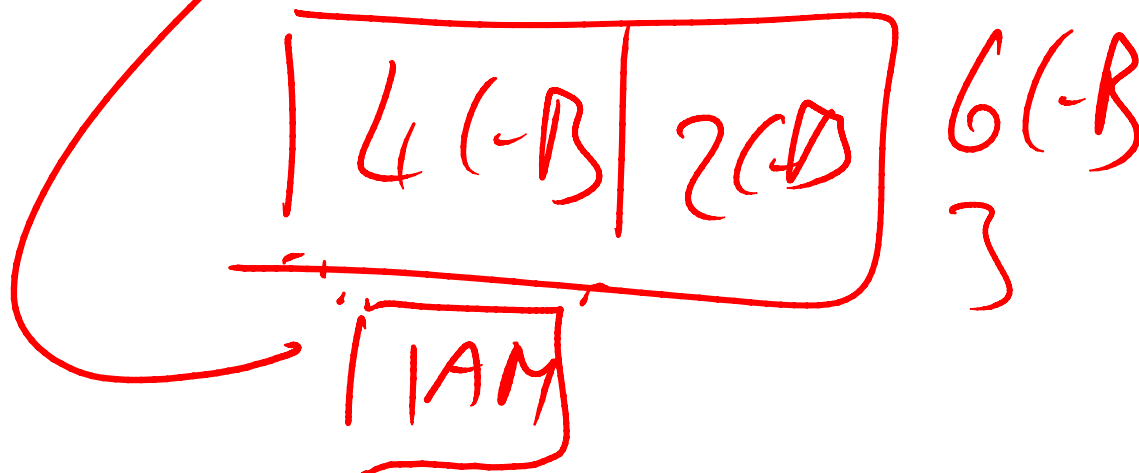


M1S24 Explaining the difference between mixed and dedicated extents. Mixed extent is one in which the 8 pages could be allocated to 8 different objects. Dedicated extent is one where all 8 pages are reserved for exclusive use of a single object.

Later in the deck I refine 'object' to really be 'allocation unit'.

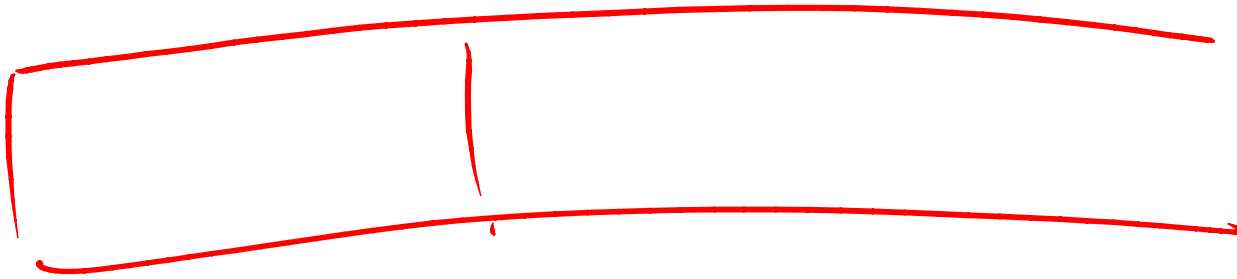


M1S36 Example of IAM chains.
Allocations from each 4GB
chunk of a data file require an
IAM page to track the
allocation. Multiple IAM pages
make an IAM chain.



sysobjects = sys.sysobjects
sysindexes = sys.sysidxstats
syscolumns = sys.syscolpars

M1 On the left are the names of the system tables/catalog views we know and love. On the right are the hidden system tables that actually implement them from 2005 onward. You can only examine these using the Dedicated Admin Connection (prefix admin: in SSMS, or sqlcmd -A).



0 - FH

1 - PFS

2 - GAM

3 - SGAM

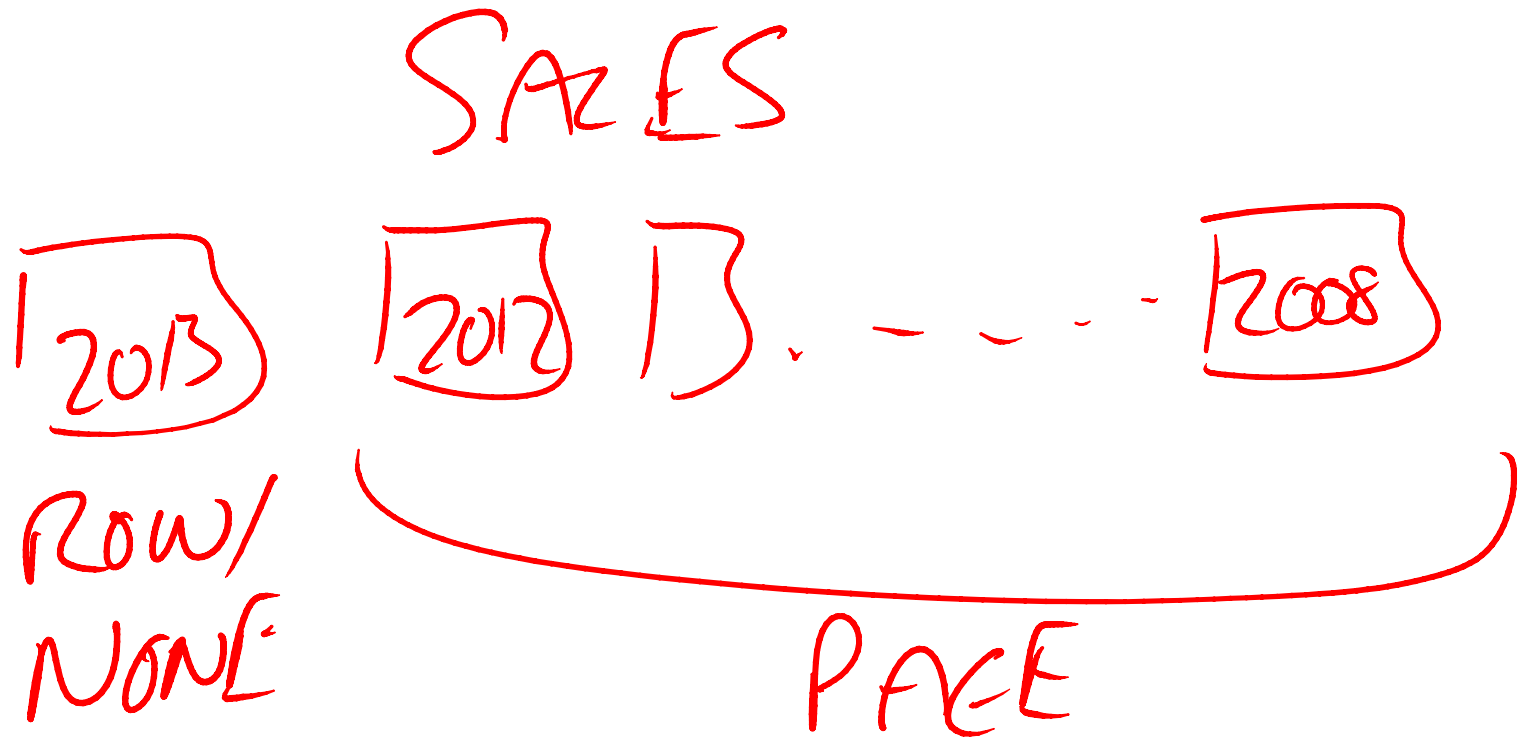
4 - UNUSED

5 - UNUSED

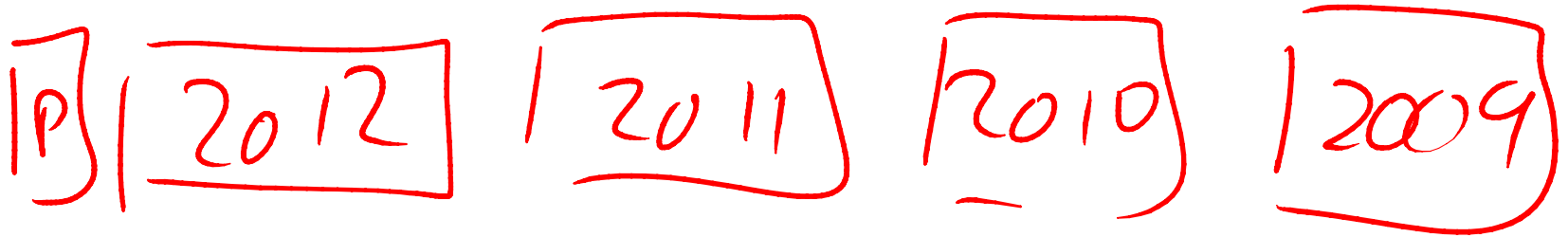
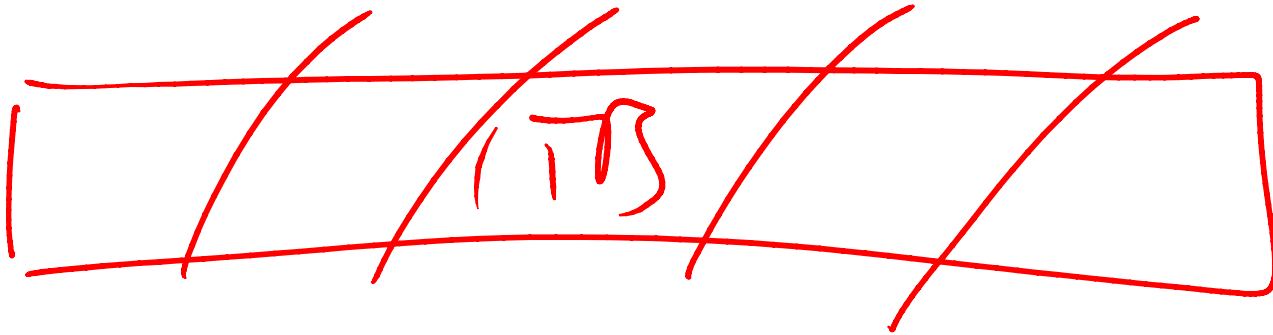
6 - DIFF

7 - ML

M1 The page types in the first extent in a data file

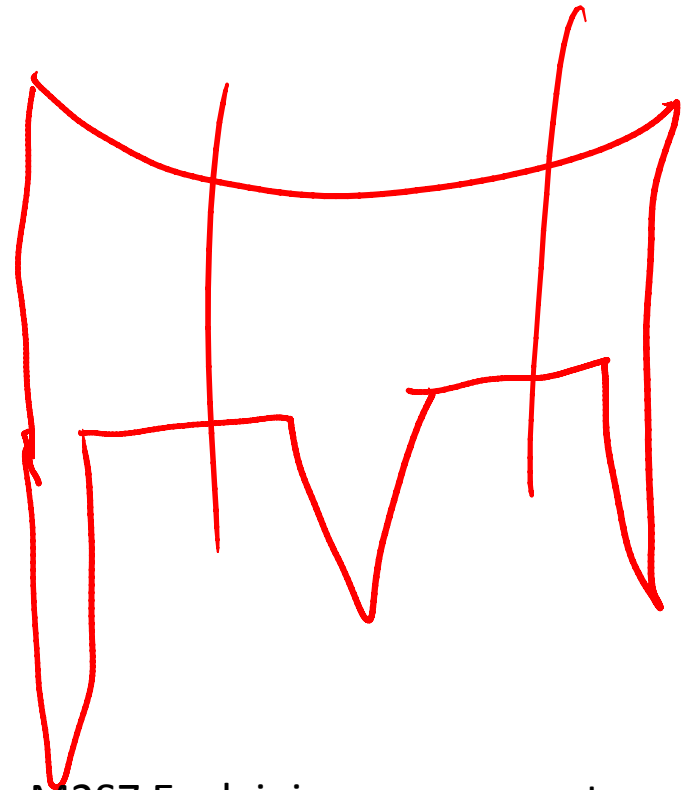


M1S50 Discussing how different partitions in a table can have different compression settings.



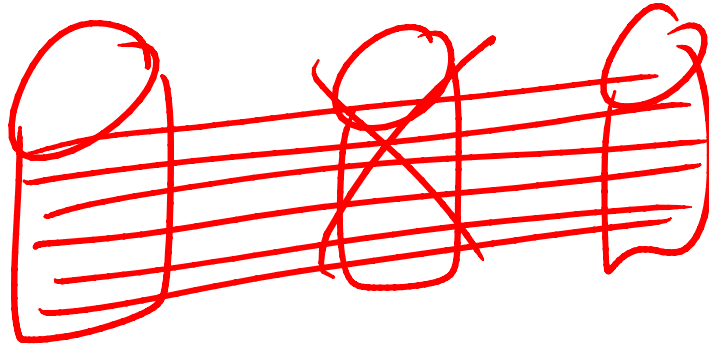
M2S7 Splitting a large database up into multiple filegroups can enable you to do small, targeted restores or even a partial restore from scratch – in the event that the database is damaged or destroyed.

CA² WA³
PARTS¹
OR⁴ IDS⁵



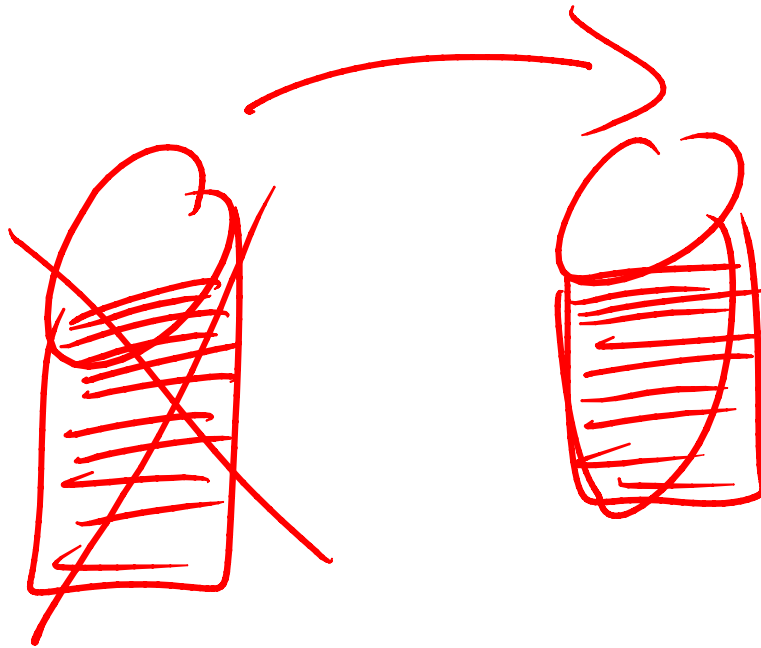
M2S7 Explaining some ways to split up a database to allow targeted restore. On the left, auto parts sales system – makes sense to bring parts and CA online first. Above, West Coast/Central/East Coast as filegroups.

$R \phi = \text{STRIPING}$



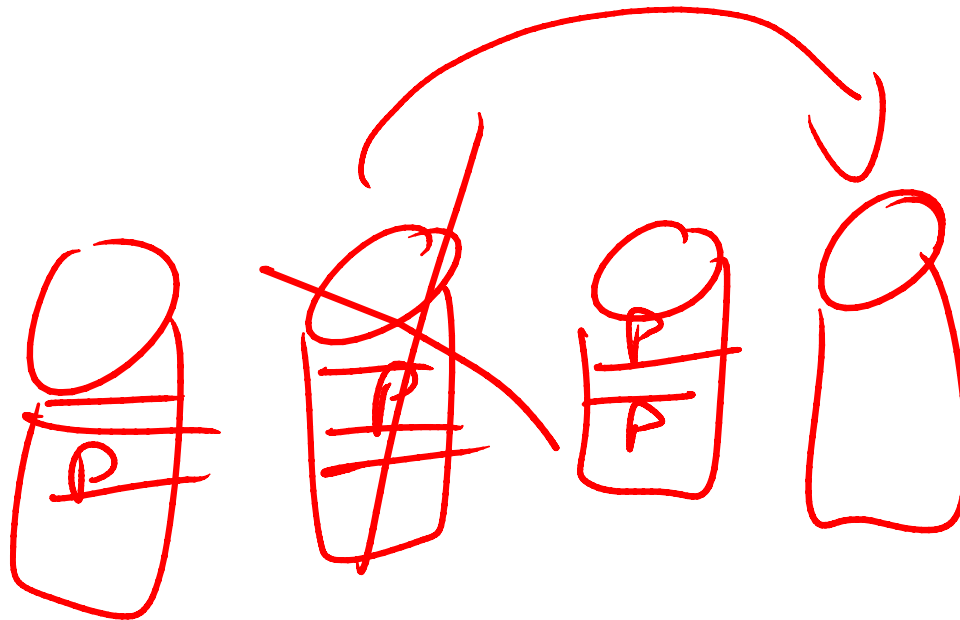
http://en.wikipedia.org/wiki/Standard_RAID_levels

RAID = MIRRORING



http://en.wikipedia.org/wiki/Standard_RAID_levels

RS = ROTATING
PARITY



http://en.wikipedia.org/wiki/Standard_RAID_levels

$$21 + 0 = 10$$

$$M + S$$

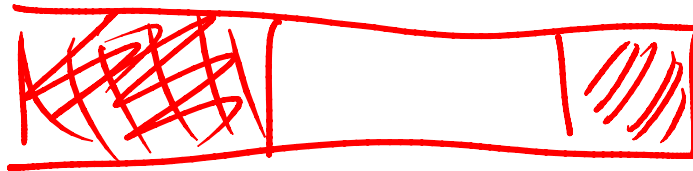


http://en.wikipedia.org/wiki/Standard_RAID_levels

sys.dm_io_virtual_file_stats
dm_io_pending_io_requests

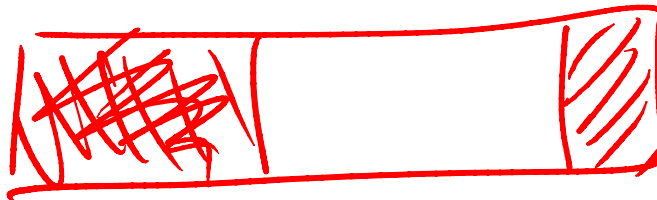
M2 Tracking I/O latencies: see <http://www.sqlskills.com/blogs/paul/how-to-examine-io-subsystem-latencies-from-within-sql-server/>

RR

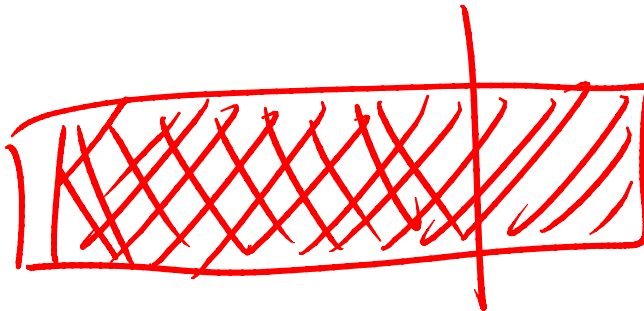


PROP FILL
WEIGHTING

~~8100~~ ←
SO (8099)

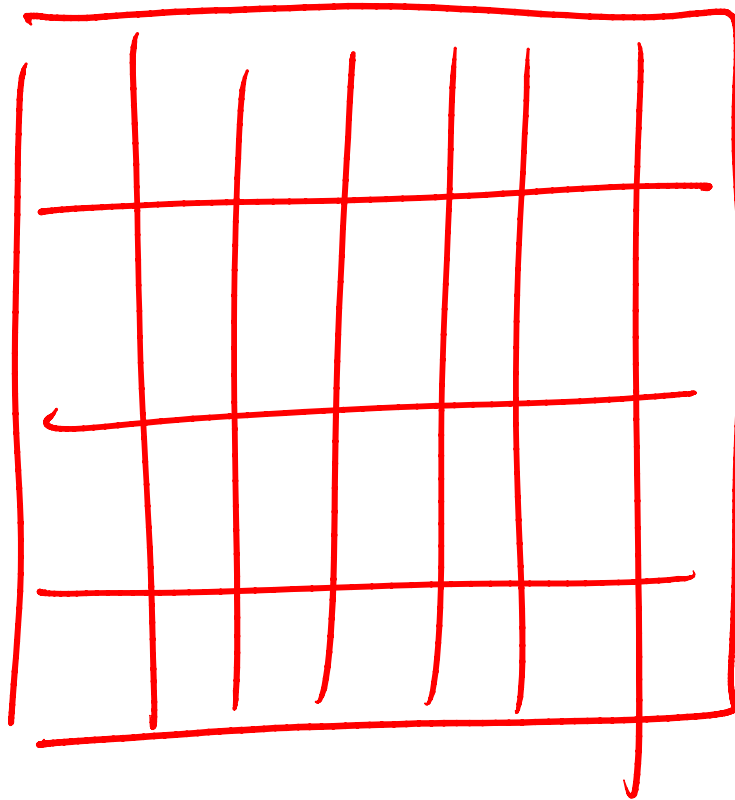


~~8100~~ ←
SO (8099)

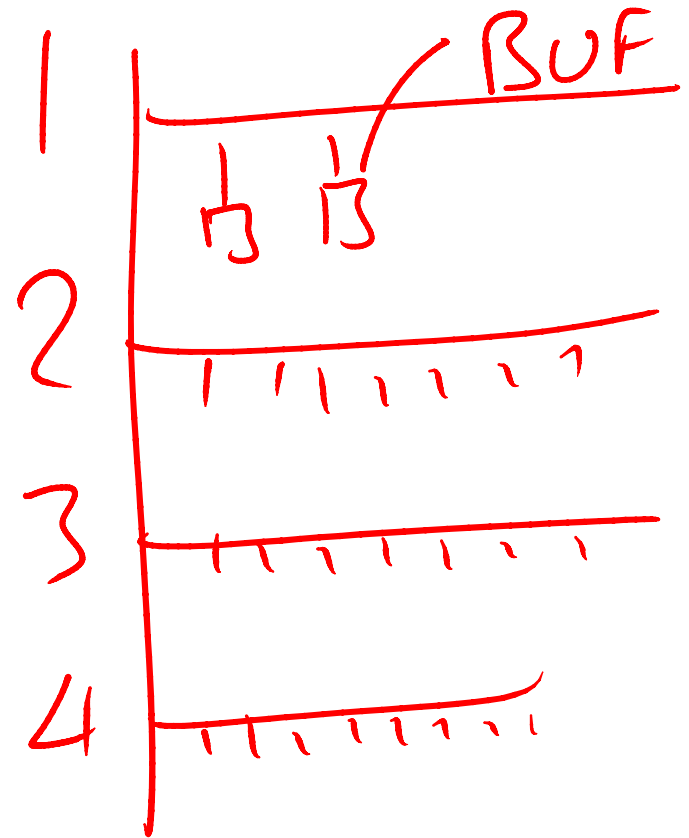


1 ← ✓
8100

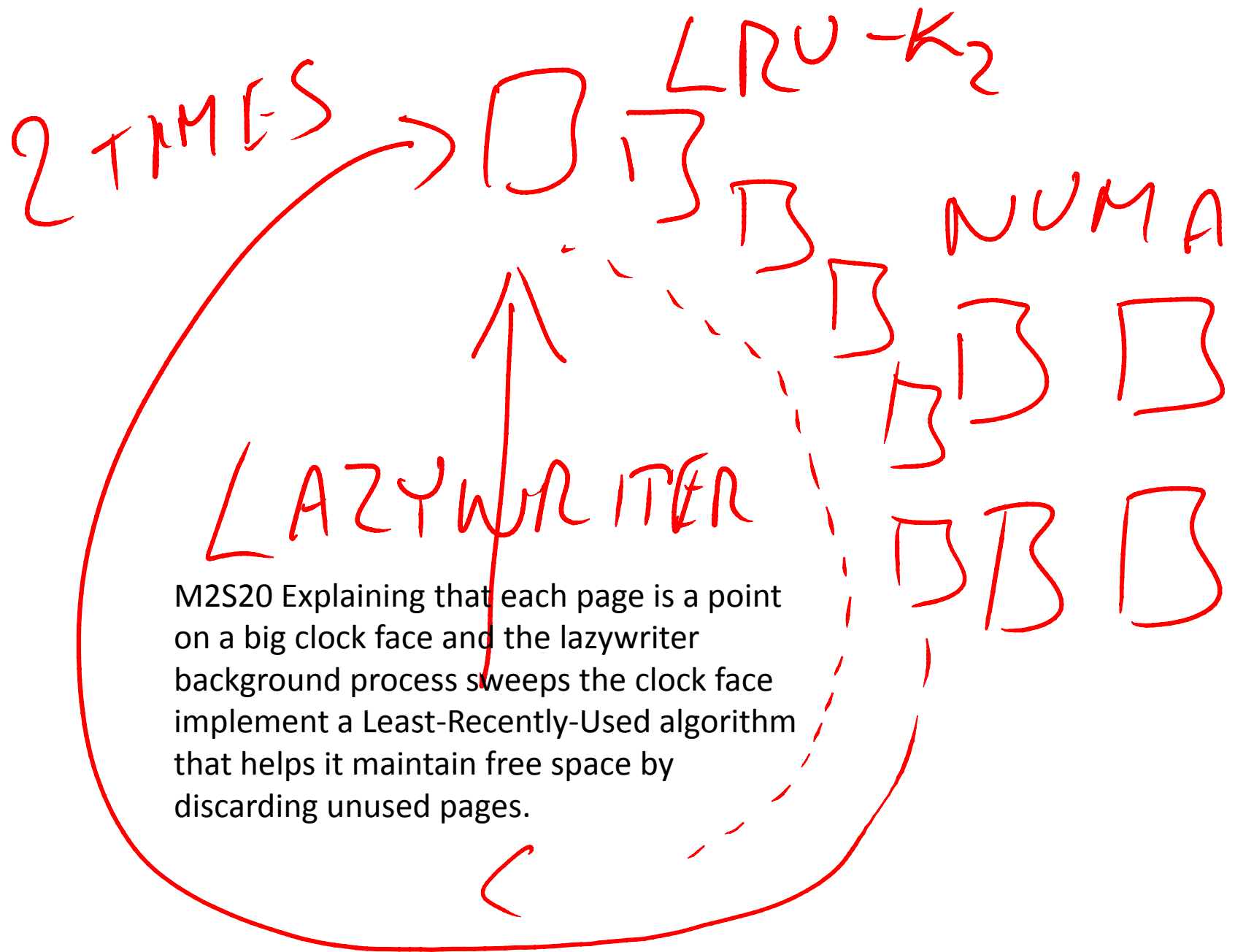
M2S19 Explaining round-robin
allocation and proportional fill



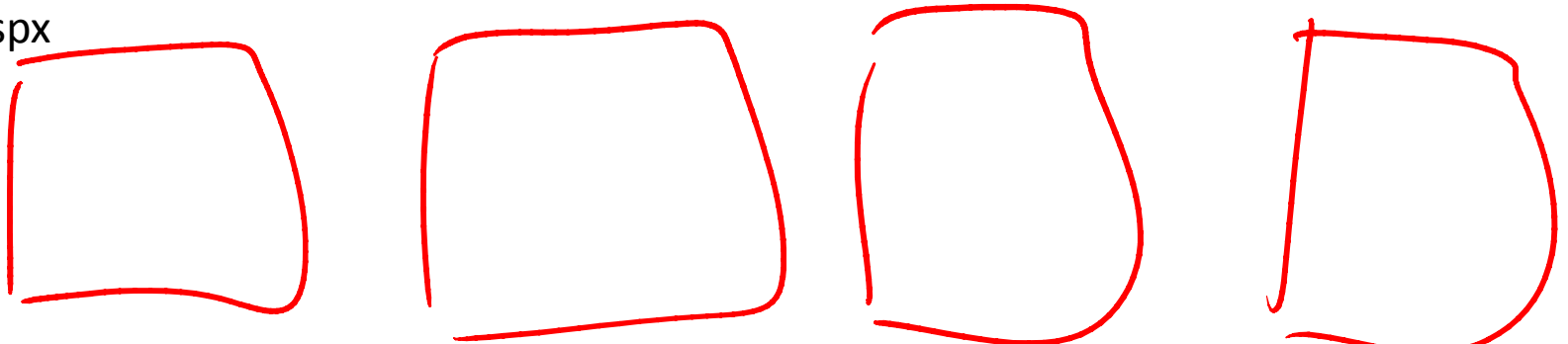
HASH LISTS



M2S20 Explaining the hash lists per database that allow the buffer pool to easily determine whether a page is already in memory or not



See <http://www.sqlskills.com/BLOGS/PAUL/post/Page-Life-Expectancy-isnt-what-you-think.aspx>



PLE
4000

PLE
4000

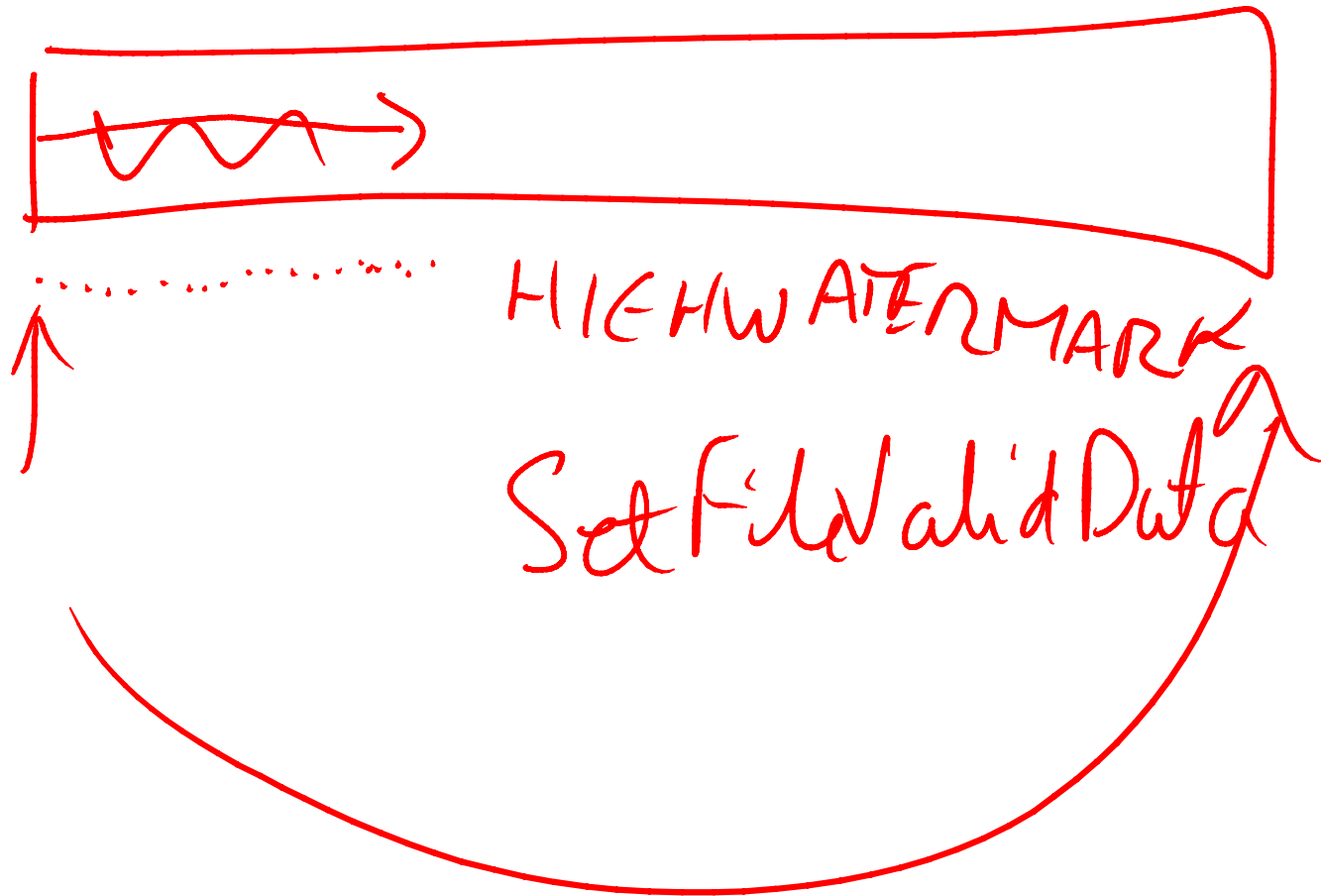
PLE
4000

PLE
4000

$$BM/PLE = (1+2+3+4)/4 \downarrow$$
$$= 4000 \quad 1000$$

This should be Buffer
Node instead of Buffer
Partition - my mistake.

3625
BUFFER MANAGER $\Rightarrow$ BUFFER PARTITION

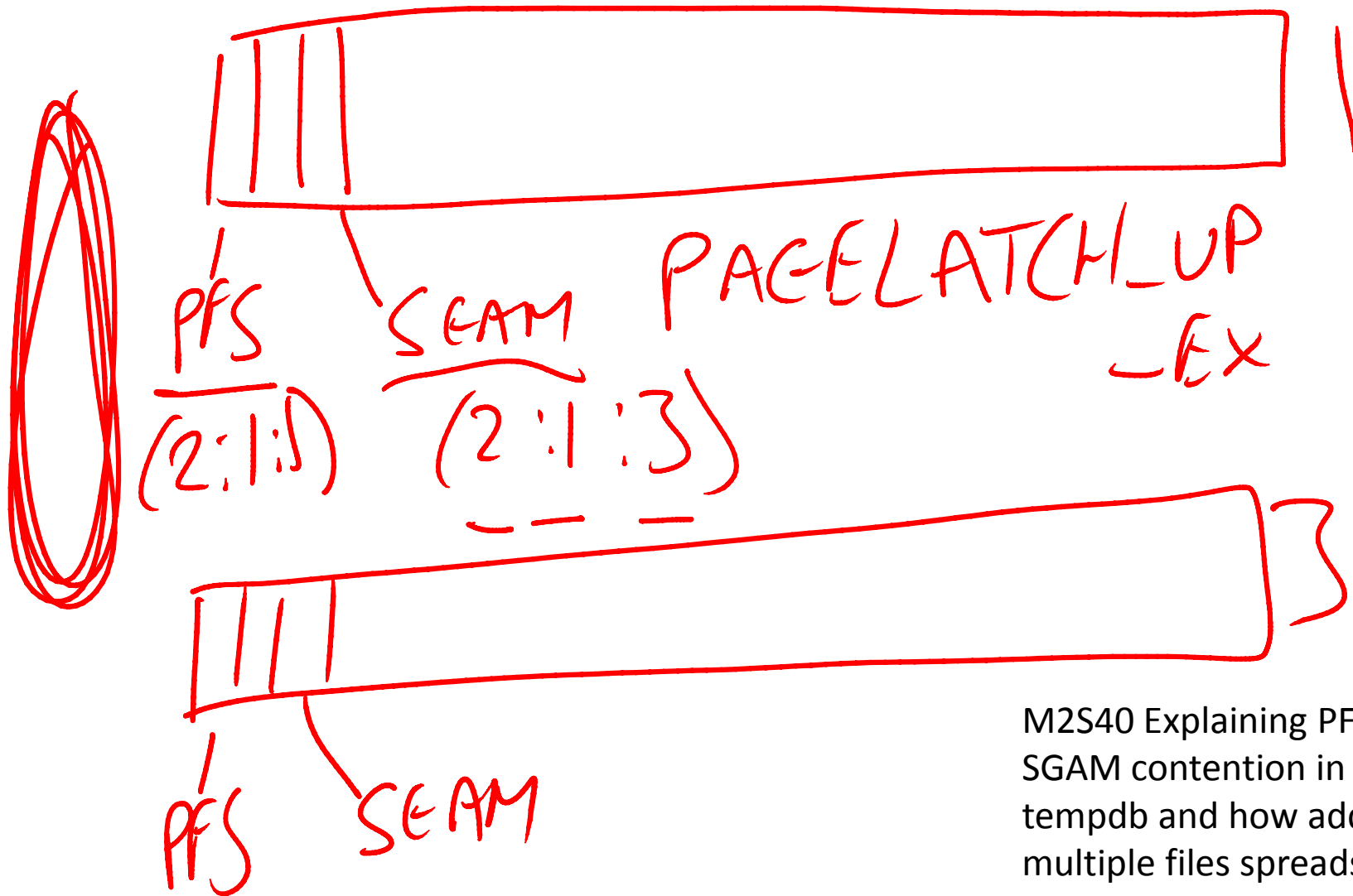


M2S22 Explaining zero initialization and how NTFS does not trust newly allocated space until the highwatermark of the file has increased, either through consecutive writes, or setting the HWM through a called to SetFileValidData.

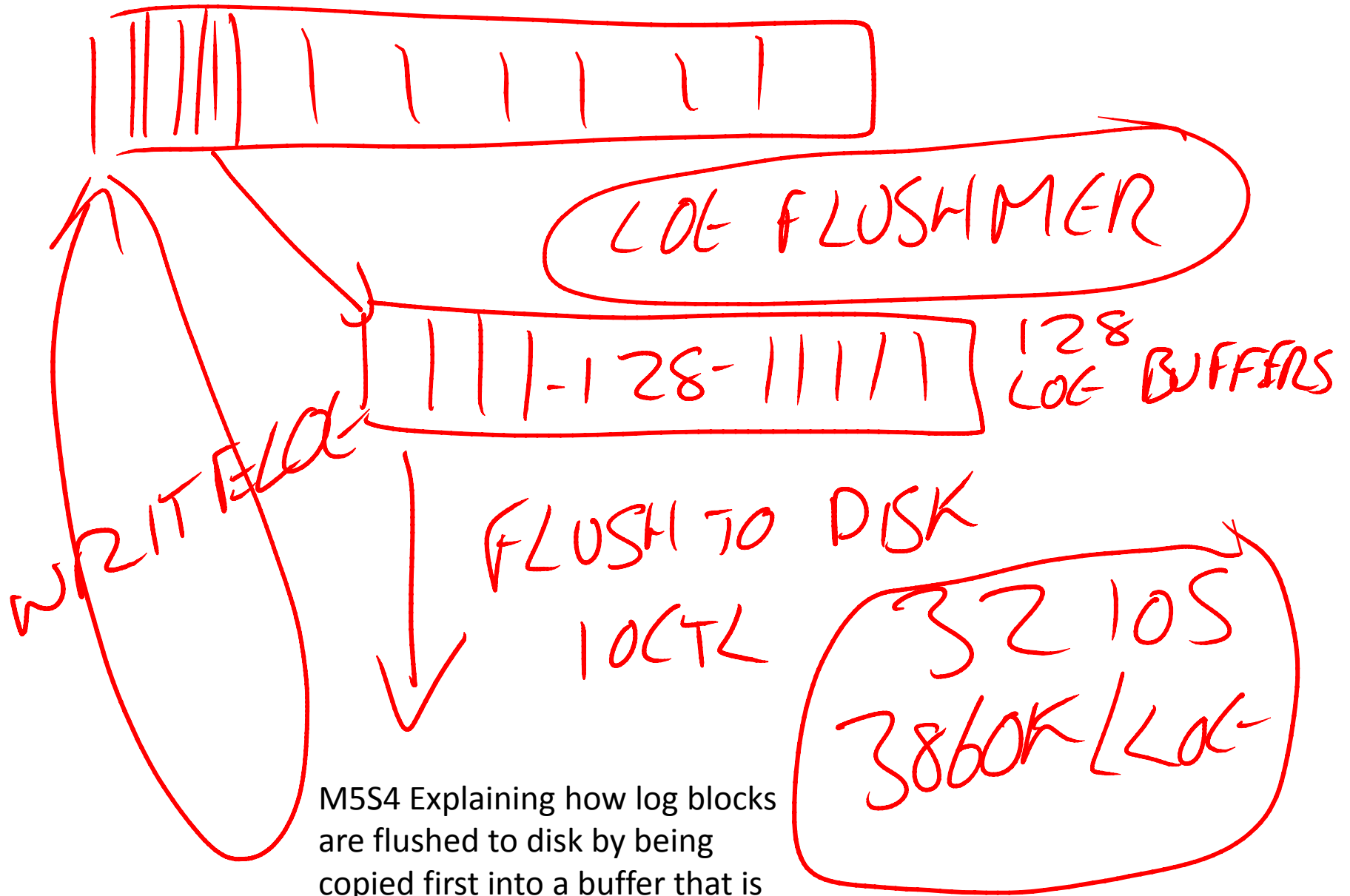
SQL ~~SECURE FILES~~

DBCC PAGE (2)

M2S24 Explaining how the security flaw with instant initialization works. Secure files are deleted, SQL files expand and pick up the disk space without zeroing it out. SA using DBCC PAGE can look at the raw bytes of the disk space now included in the data file, potentially leading to information leakage.

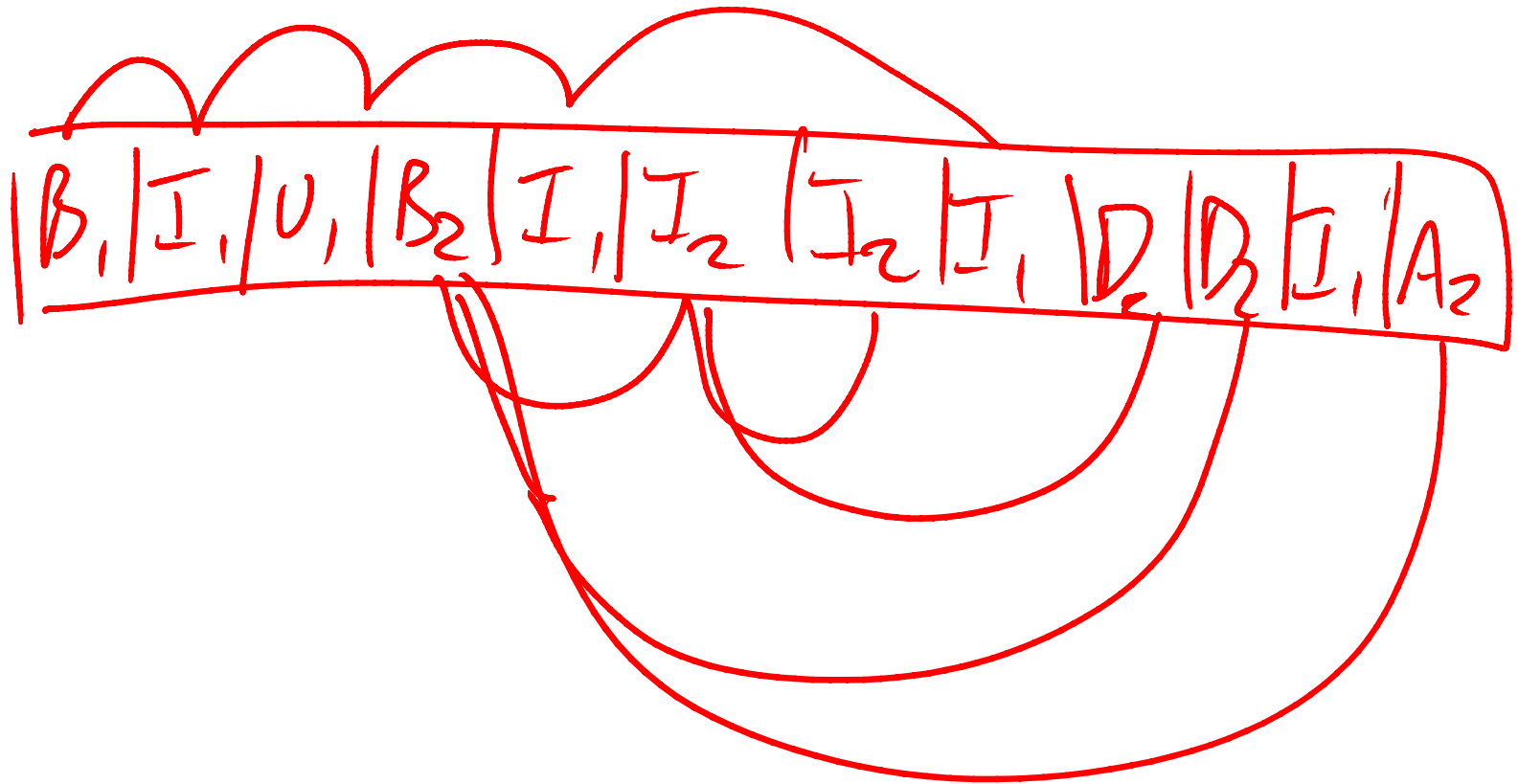


M2S40 Explaining PFS and SGAM contention in tempdb and how adding multiple files spreads the contention over multiple files, thus reducing it.



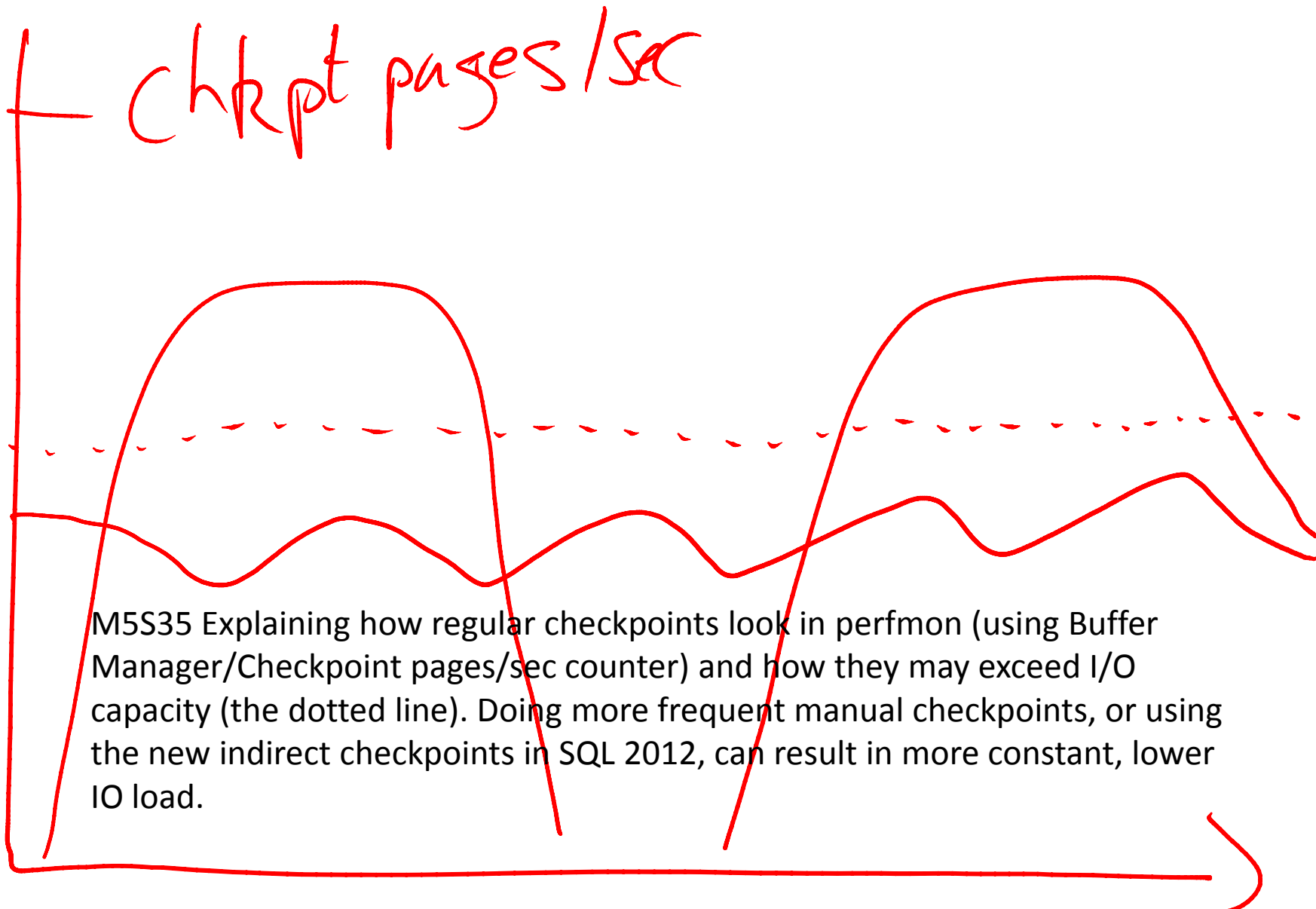
M5S4 Explaining how log blocks are flushed to disk by being copied first into a buffer that is then async written to disk.

32105
3860K LOK

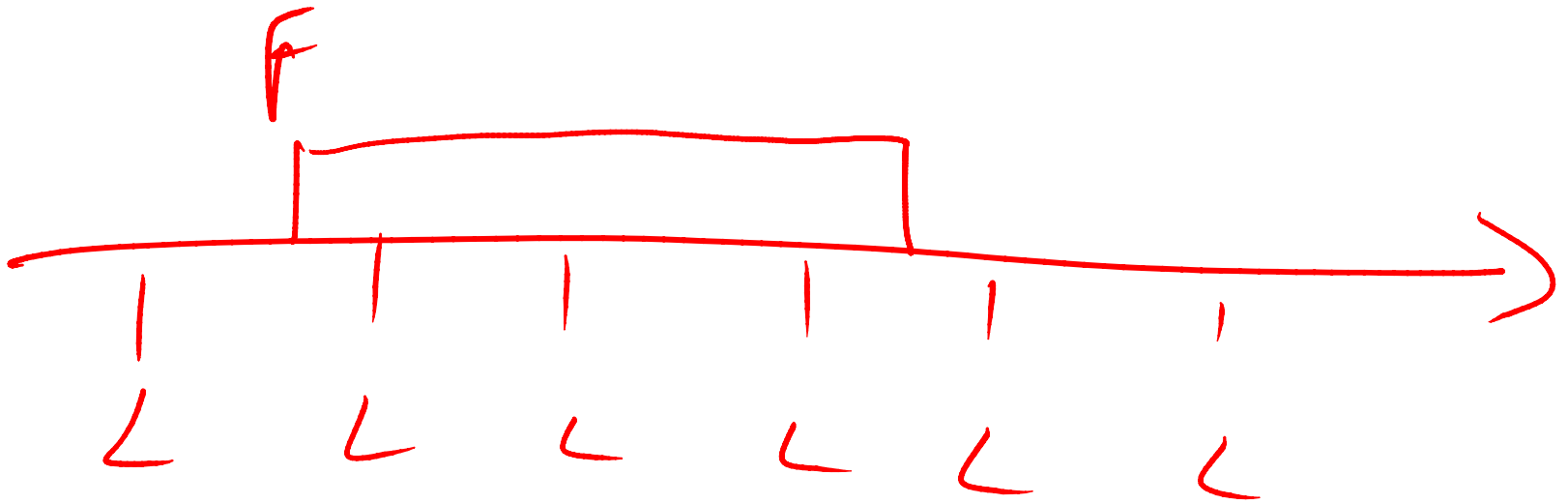


M5S20 Showing how transaction rollback works fine even when log records from other transactions are intermingled with the transaction being rolled back. This is why log records in a transaction are linked backward to the previous log record in the transaction using LSNs.

chkpt pages/sec



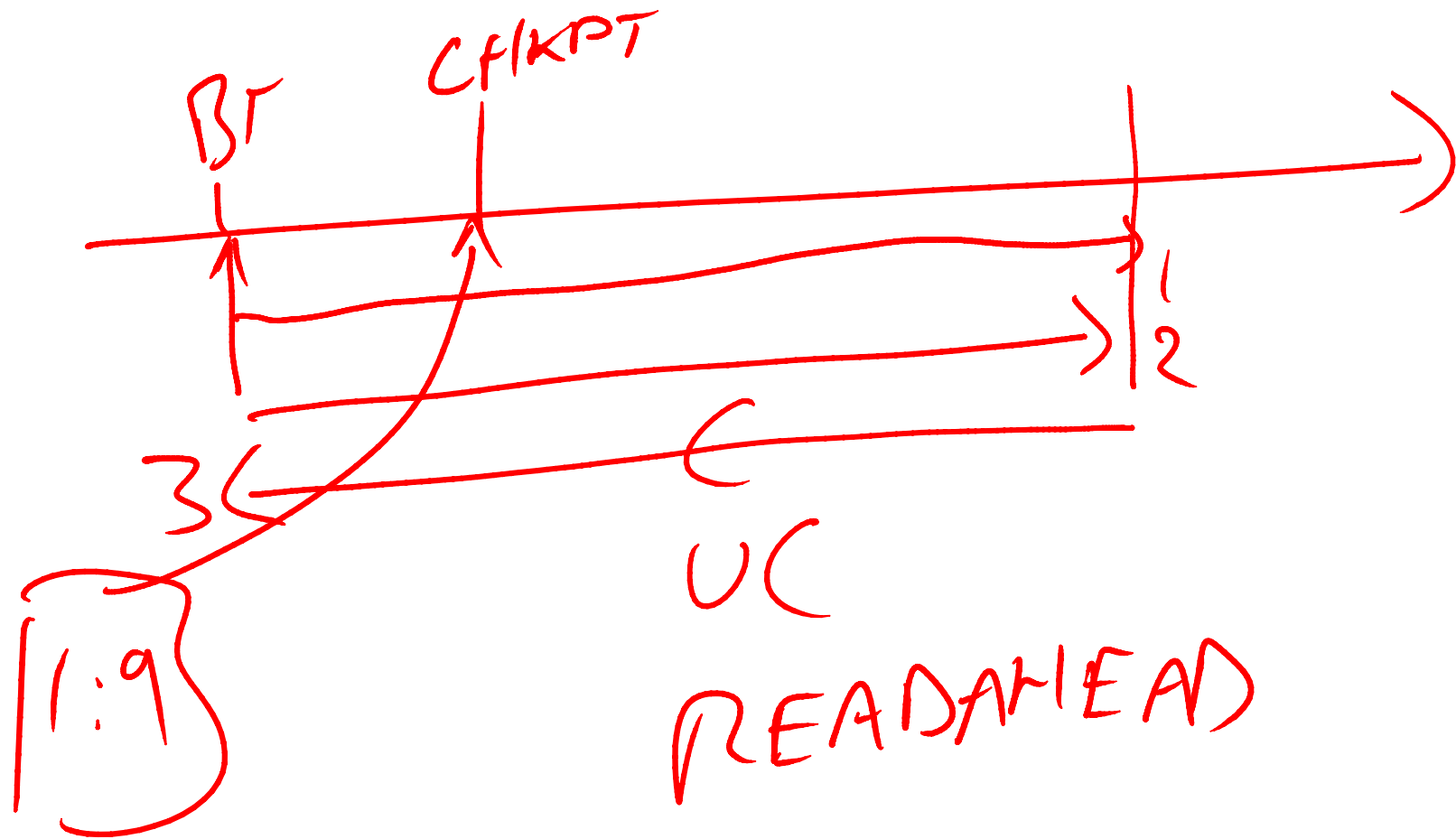
M5S35 Explaining how regular checkpoints look in perfmon (using Buffer Manager/Checkpoint pages/sec counter) and how they may exceed I/O capacity (the dotted line). Doing more frequent manual checkpoints, or using the new indirect checkpoints in SQL 2012, can result in more constant, lower IO load.



M5S42 It's possible for concurrent log backups to occur with a full backup from 2005 onward. Log clearing cannot happen though. It's deferred until the full backup completes, giving the impression that a full backup clears the log.

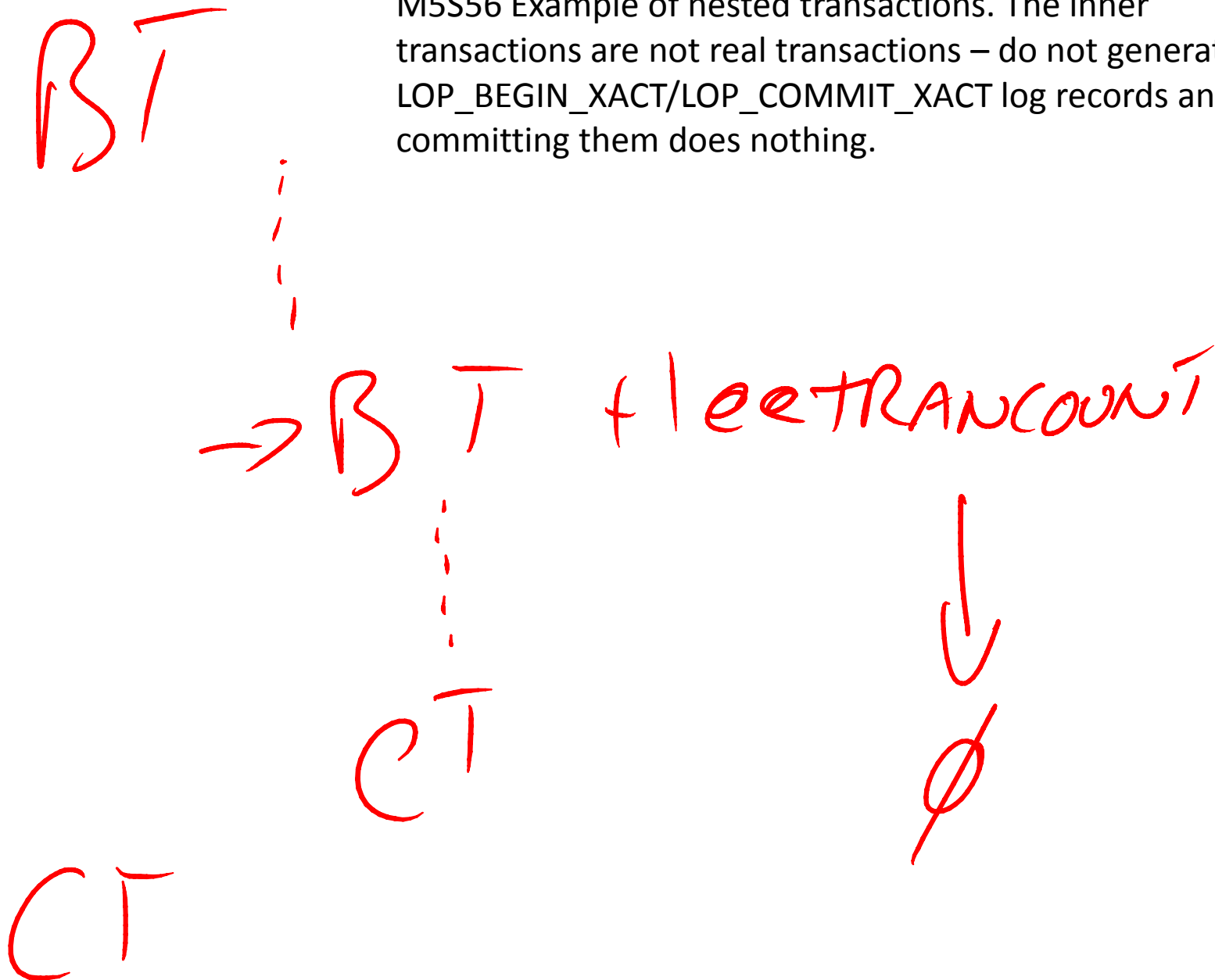


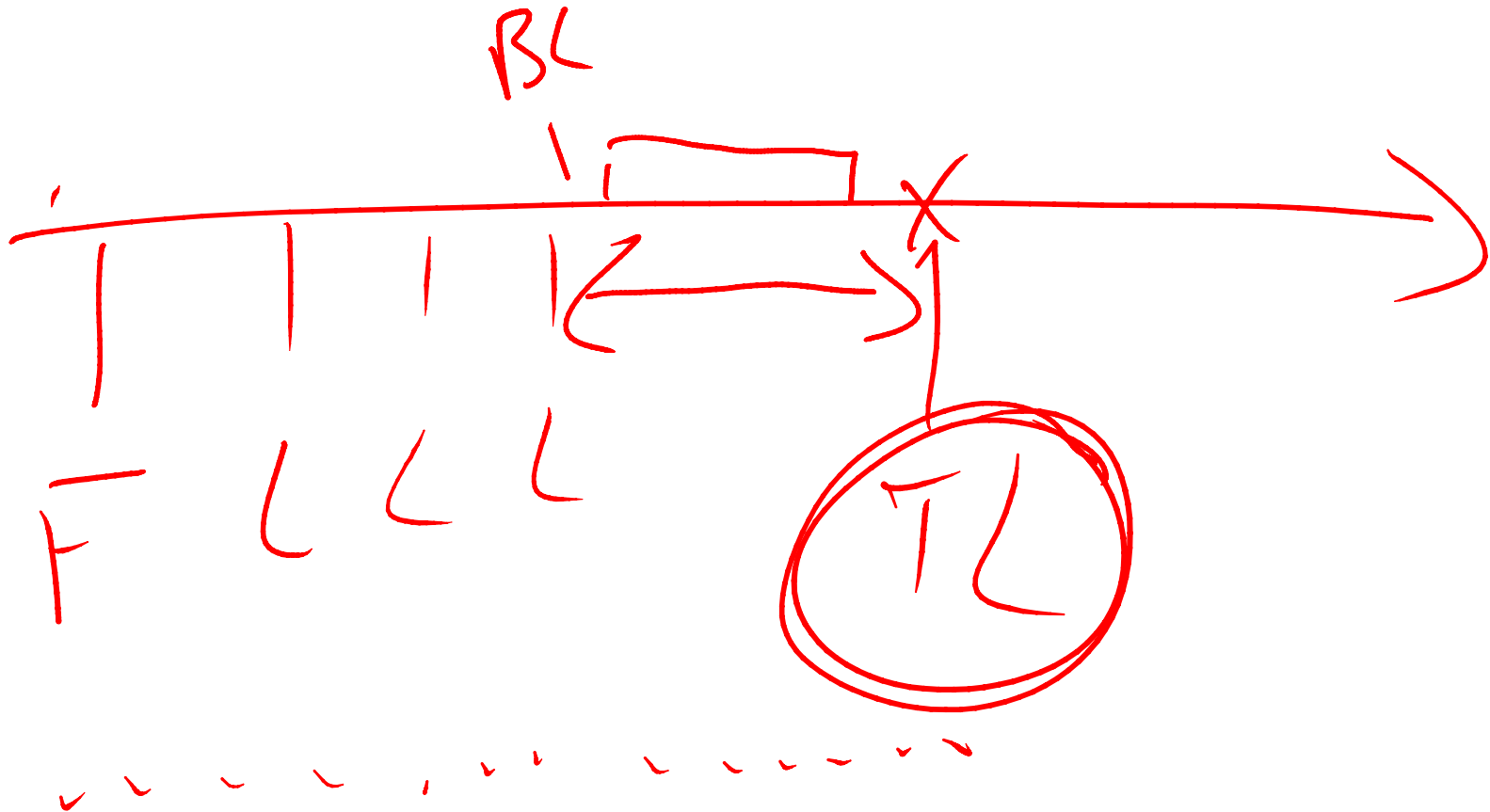
M5 If you need to do a tail-of-the-log backup after a crash, and the data files are damaged, use the WITH NO_TRUNCATE option on the BACKUP LOG command. See <http://www.sqlskills.com/blogs/paul/disaster-recovery-101-backing-up-the-tail-of-the-log/>



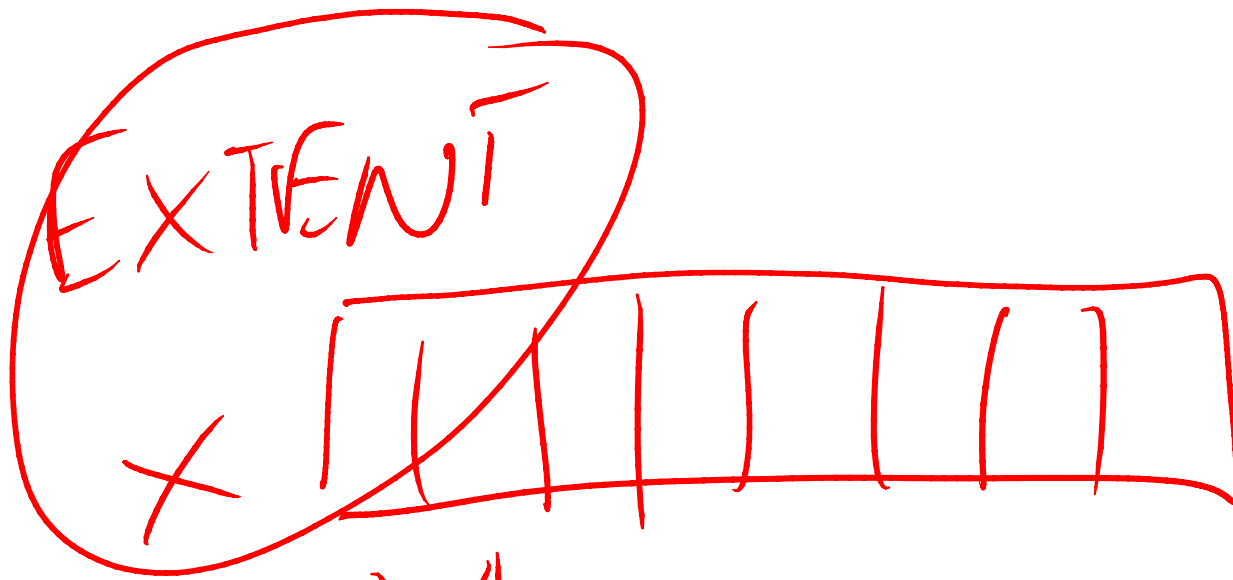
M5S49 Explaining the three passes through the log of crash recovery.

M5556 Example of nested transactions. The inner transactions are not real transactions – do not generate LOP_BEGIN_XACT/LOP_COMMIT_XACT log records and committing them does nothing.





M5S60 Explaining that if a minimally-logged operation is present in a log backup, you cannot do a point-in-time restore to any time covered by that backup (except the start or end of that time period). You also cannot do a tail-of-the-log backup after a crash that destroys the data files if there has been a minimally-logged operation since the last log backup. Well in 2008 R2/2012, you can, but the log backup is invalid.



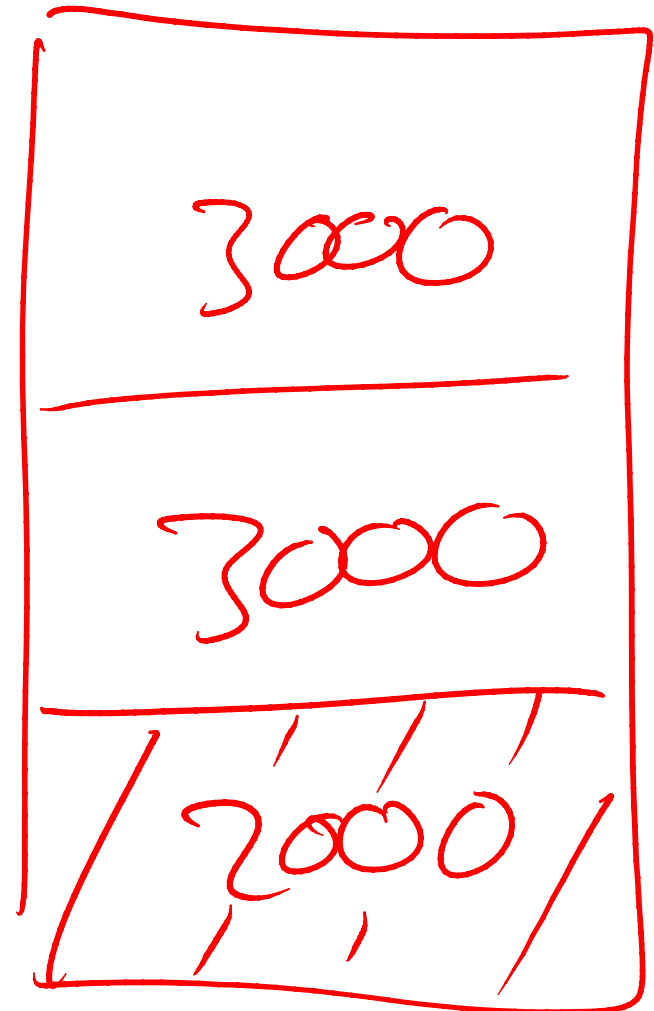
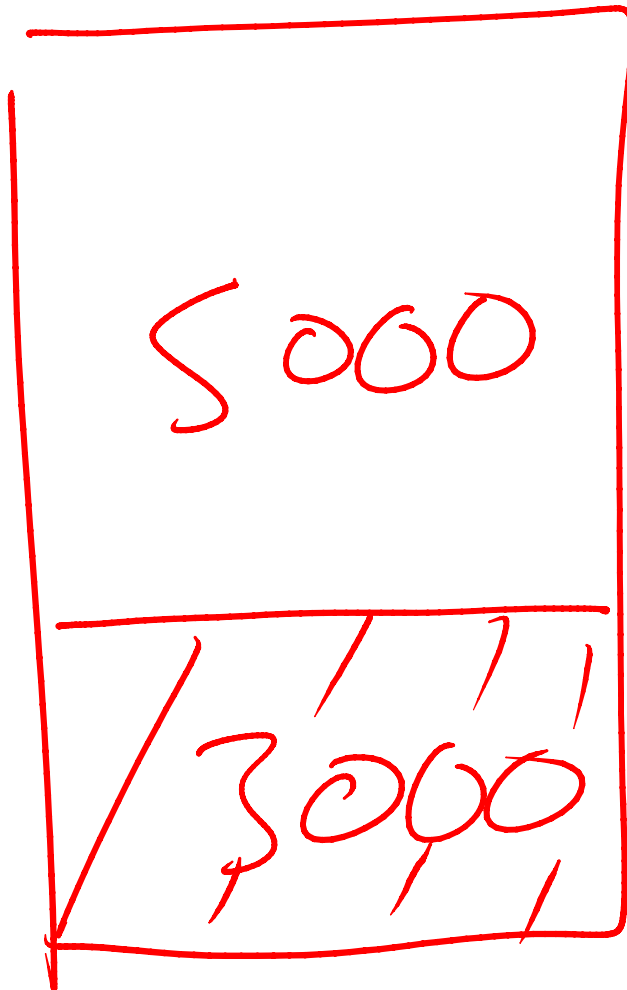
PROBE

M5 Explaining part of the deferred-drop functionality. Before an extent can be deallocated, an X lock is acquired on it and all 8 page locks are probed (instant acquire X lock and release) to make sure nothing else has them locked.



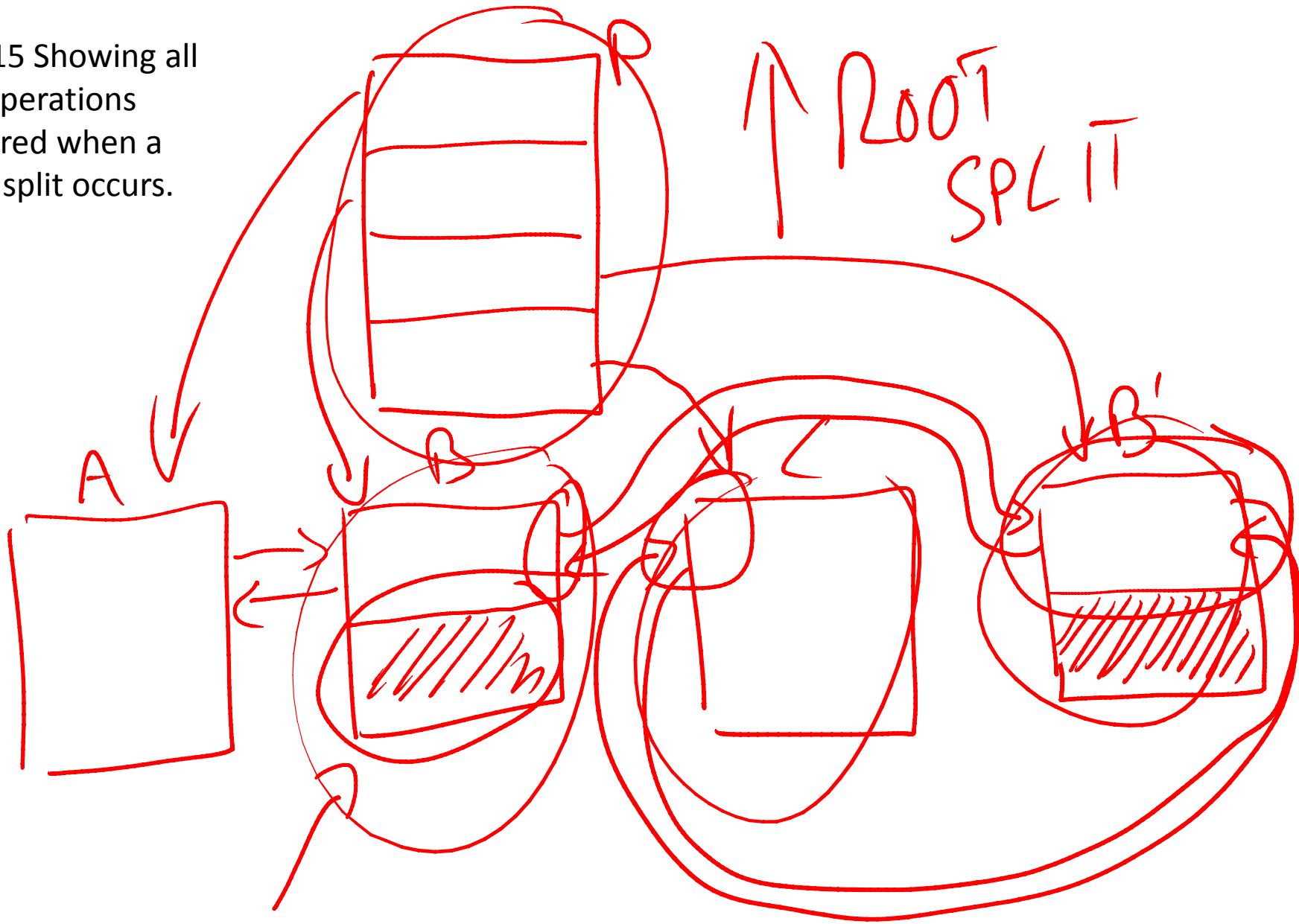
M5S25 VLF fragmentation is bad because of having to search through the VLFs for a particular VLF sequence number. See the KB article link for more info.

LSN LSN
VLF



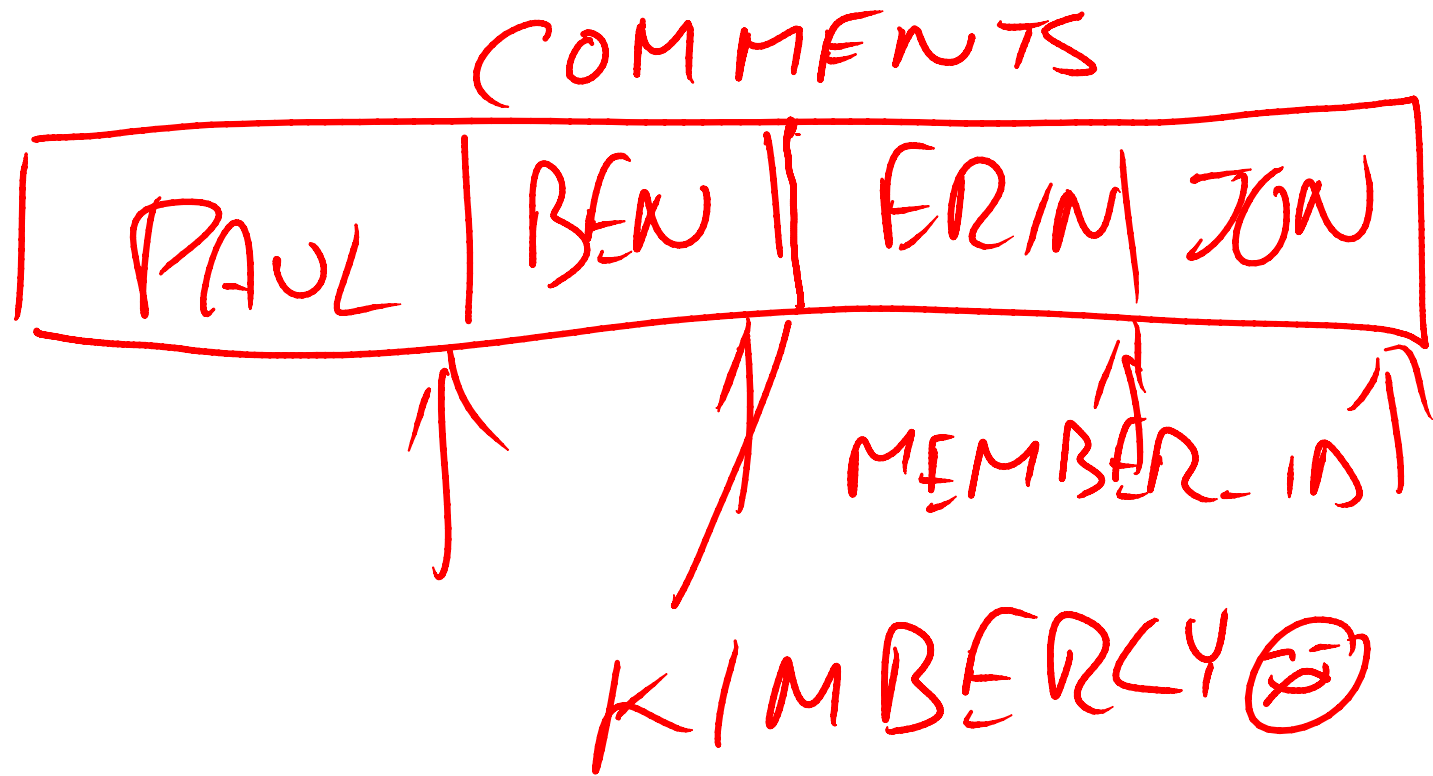
M8S13 Explaining how fixed-size large rows leads to low page density and wasted space.

M8S15 Showing all
the operations
required when a
page split occurs.

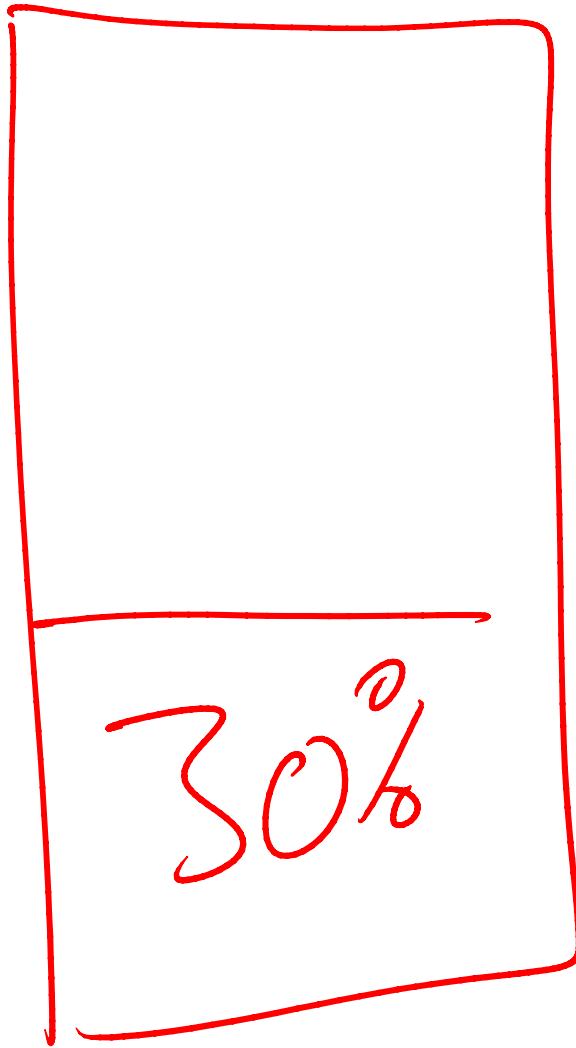


	NO SPLIT	SPLIT	X
1000	1256	7572	x5.5
100	356	6764	x18
10	268	11164	x47

M8S22 The numbers from the page split logging cost demo.



M8S23 The MySpace story. The 'K' is the tiny portion of comments on Kimberly's page because she wasn't very popular back then... ☺



$$FF = 70$$

M8S25 To leave 30% free space, set the fillfactor to 70.

ITB



M8S37 Explaining staggered index maintenance with database mirroring. See <http://bit.ly/IS04W5> for more explanation