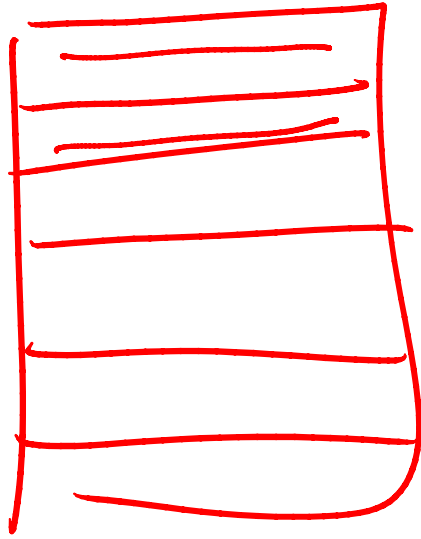
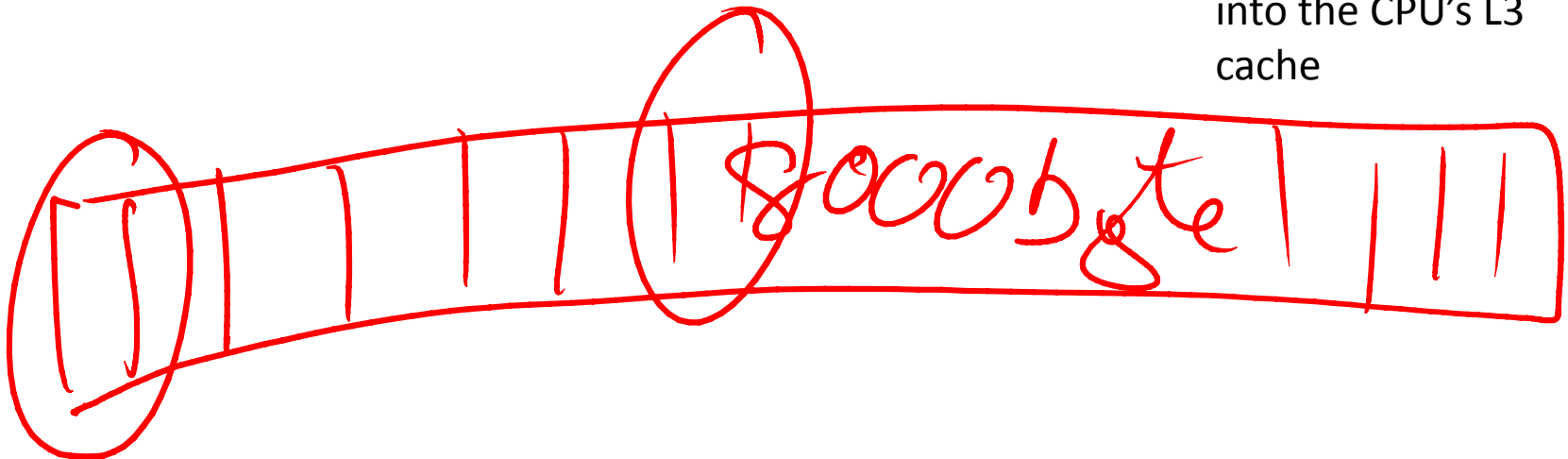


CACHE LINE INVALIDATION

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



SYSTEM TABLES

DEDICATED ADMIN CONNECTION

The system tables are hidden and can only be viewed when connected using the Dedicated Admin Connection (DAC). Use 'admin:' prefix on the SSMS connection string or sqlcmd -A.



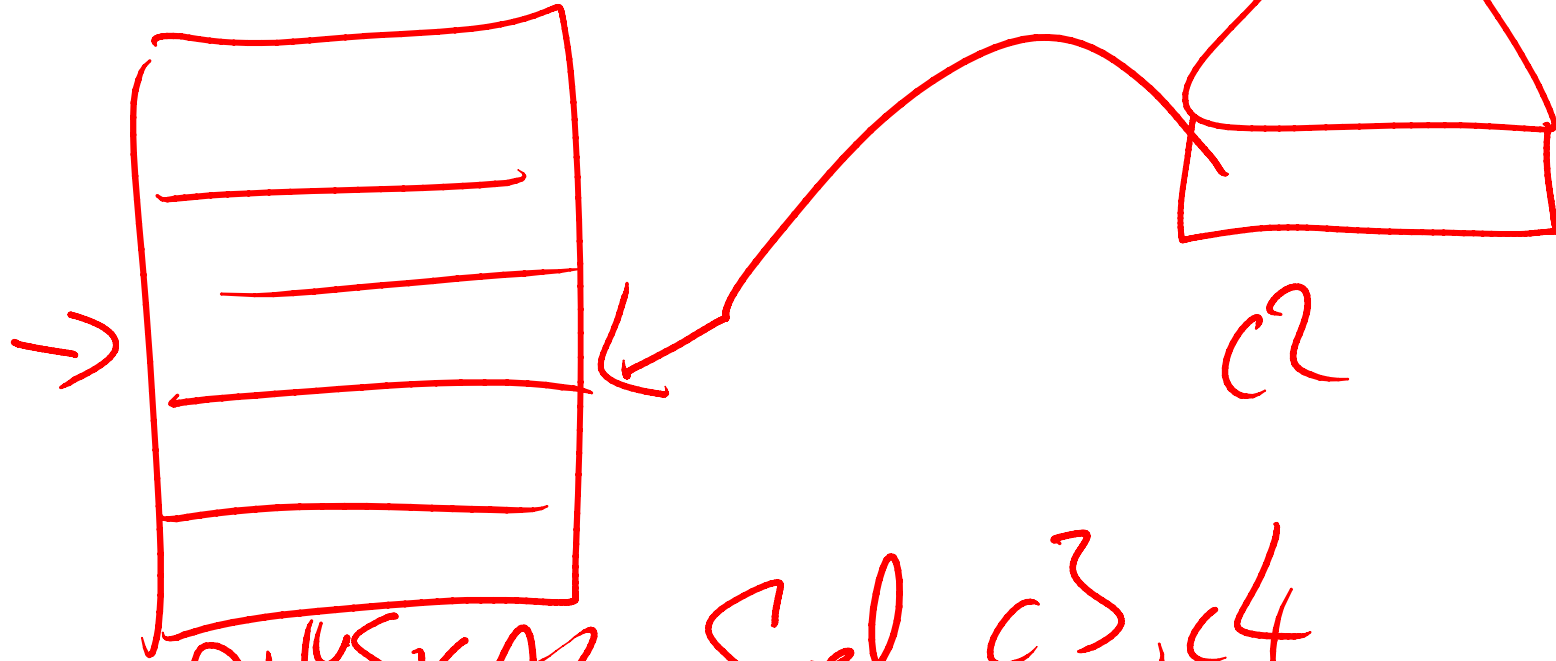
M1S6 Dropping a column marks it as dropped in the Storage Engine metadata. Removed on next rebuild. Adding a nullable column is metadata-only operation. Adding non-nullable column is metadata-only in 2012+, before 2012 it inserts the column value in each row immediately.

↑

ADD NULLABLE

ADD NON-NULLABLE

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



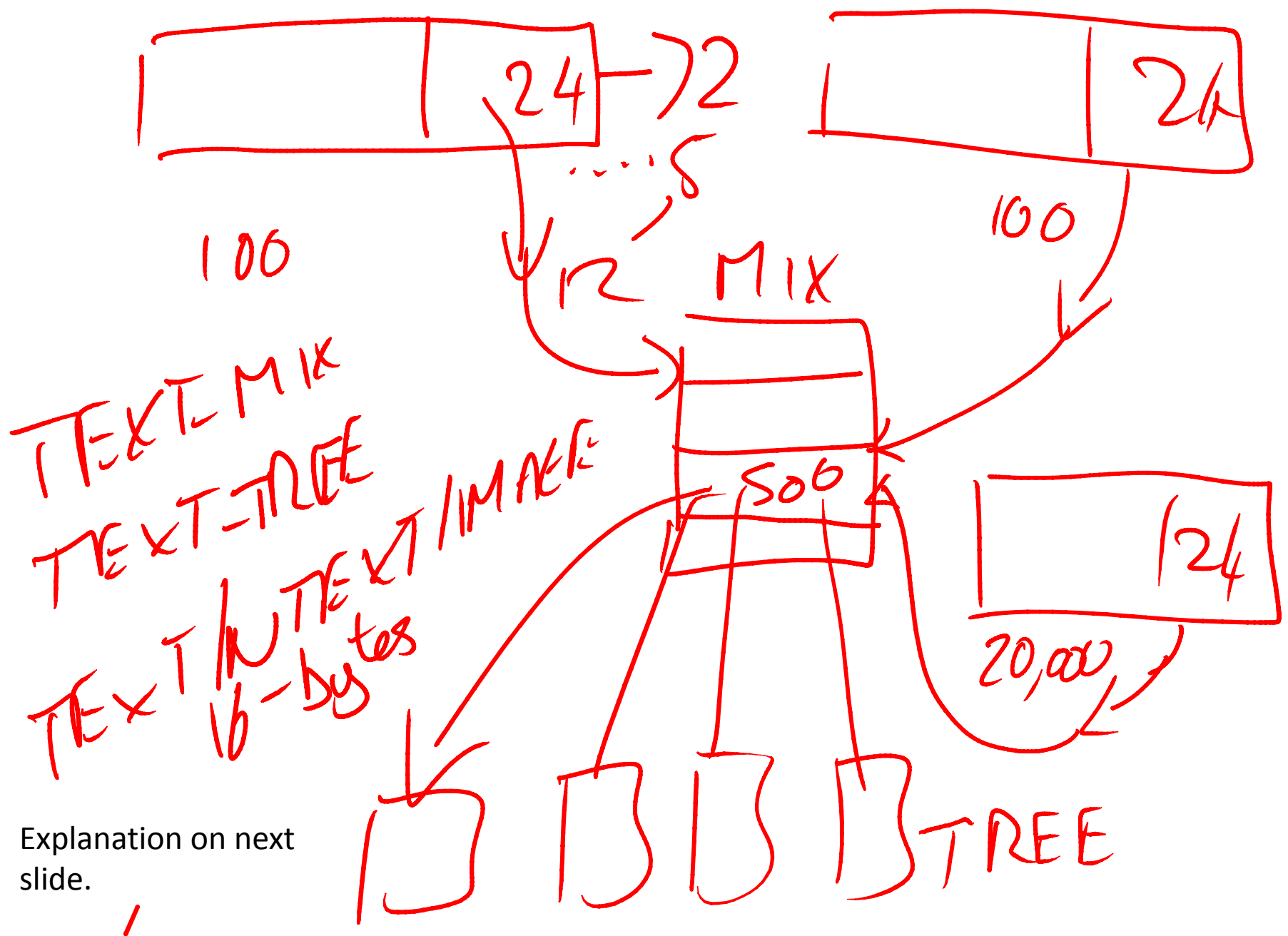
HL: PHYSICAL
RID

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.



Explanation on next slide.

From previous slide:

M1S11 Off-row LOB values are stored as loose trees. Values that are small enough from the same table will share TEXT_MIX text pages. Portions from large LOB values will have their own TEXT_TREE text pages. In-row pointers to off-row LOB values vary in size from 16 bytes (for legacy LOB types) to 24-72 bytes (for 2005+ LOB types).

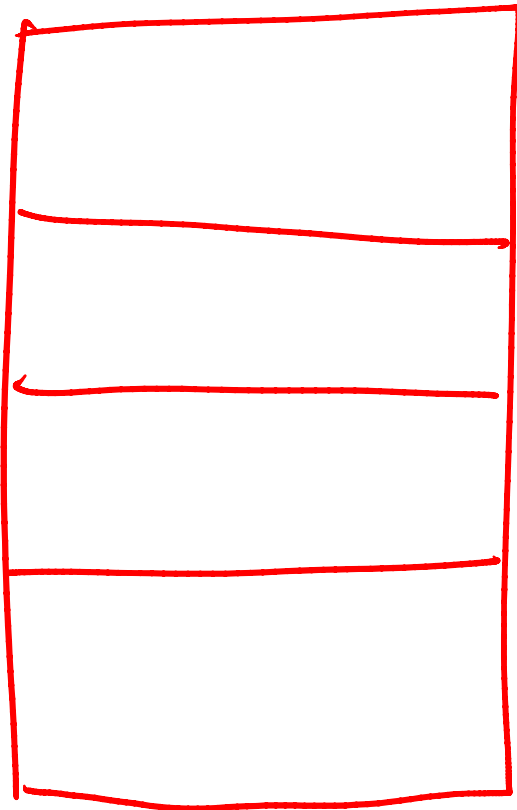
ORCA MDF

MARC RASMUSSEN

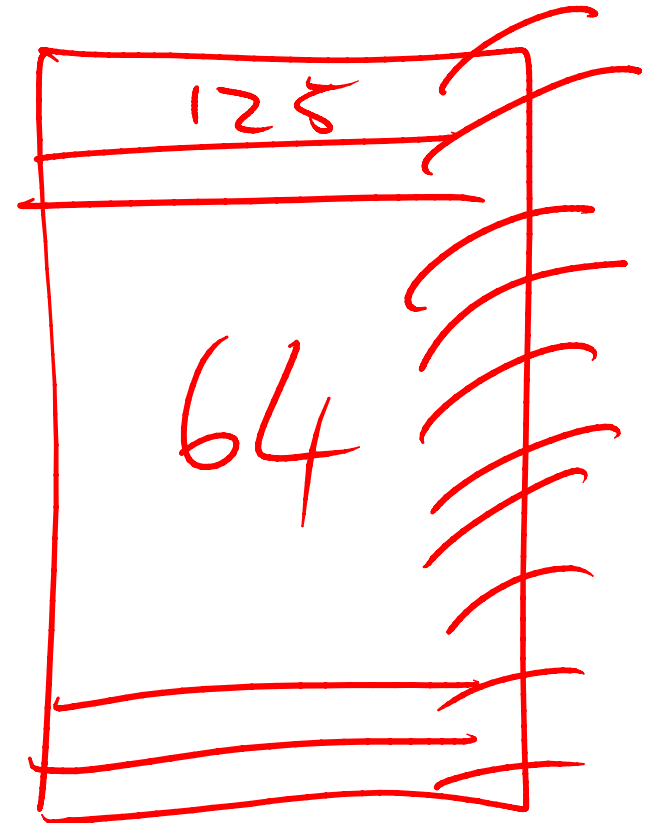
TEXT TREES

Marc Rasmussen has blogged about text tree structure as part of his OrcaMDF structure analysis tool.

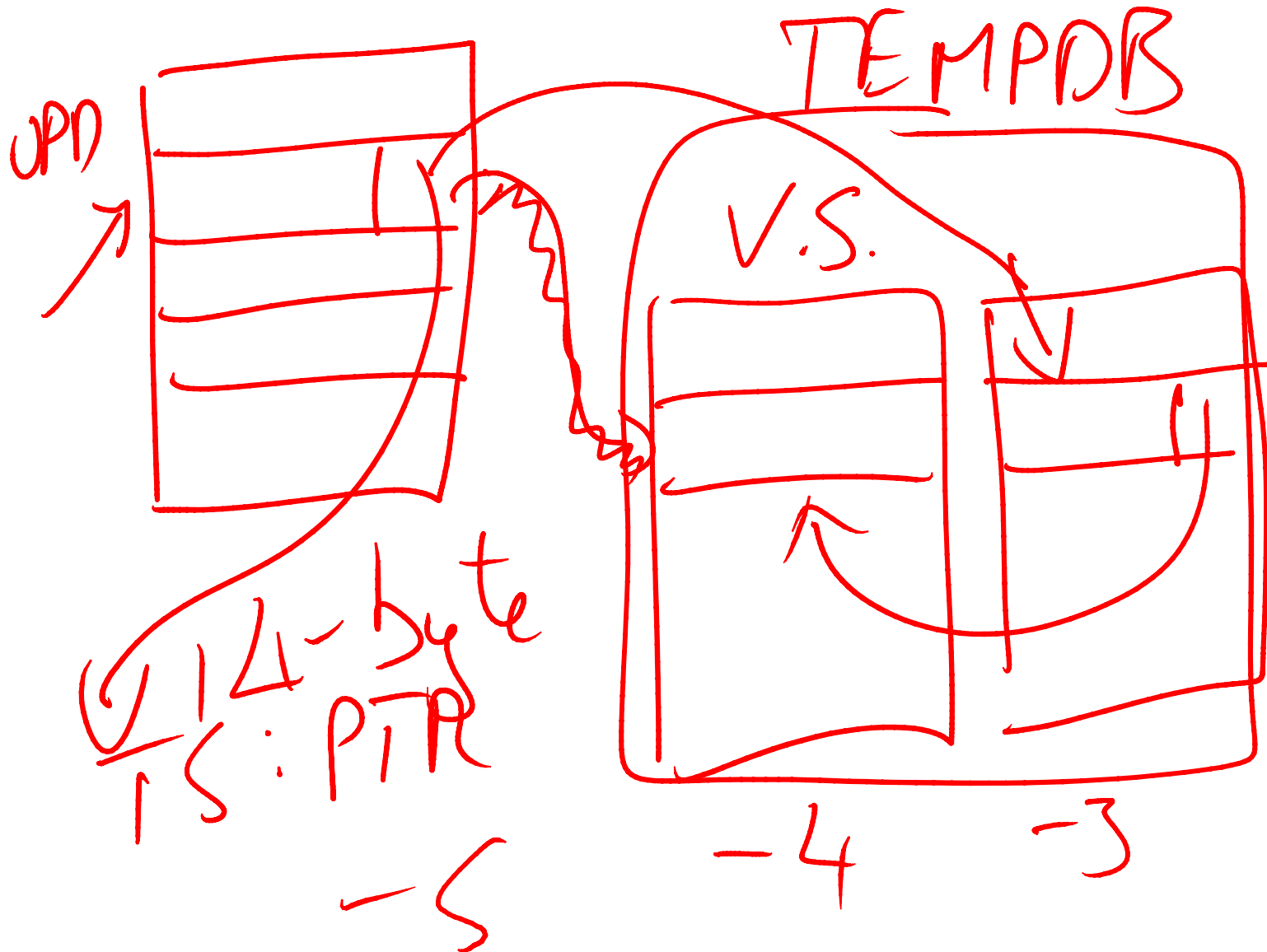
100	1900 byte
-----	-----------



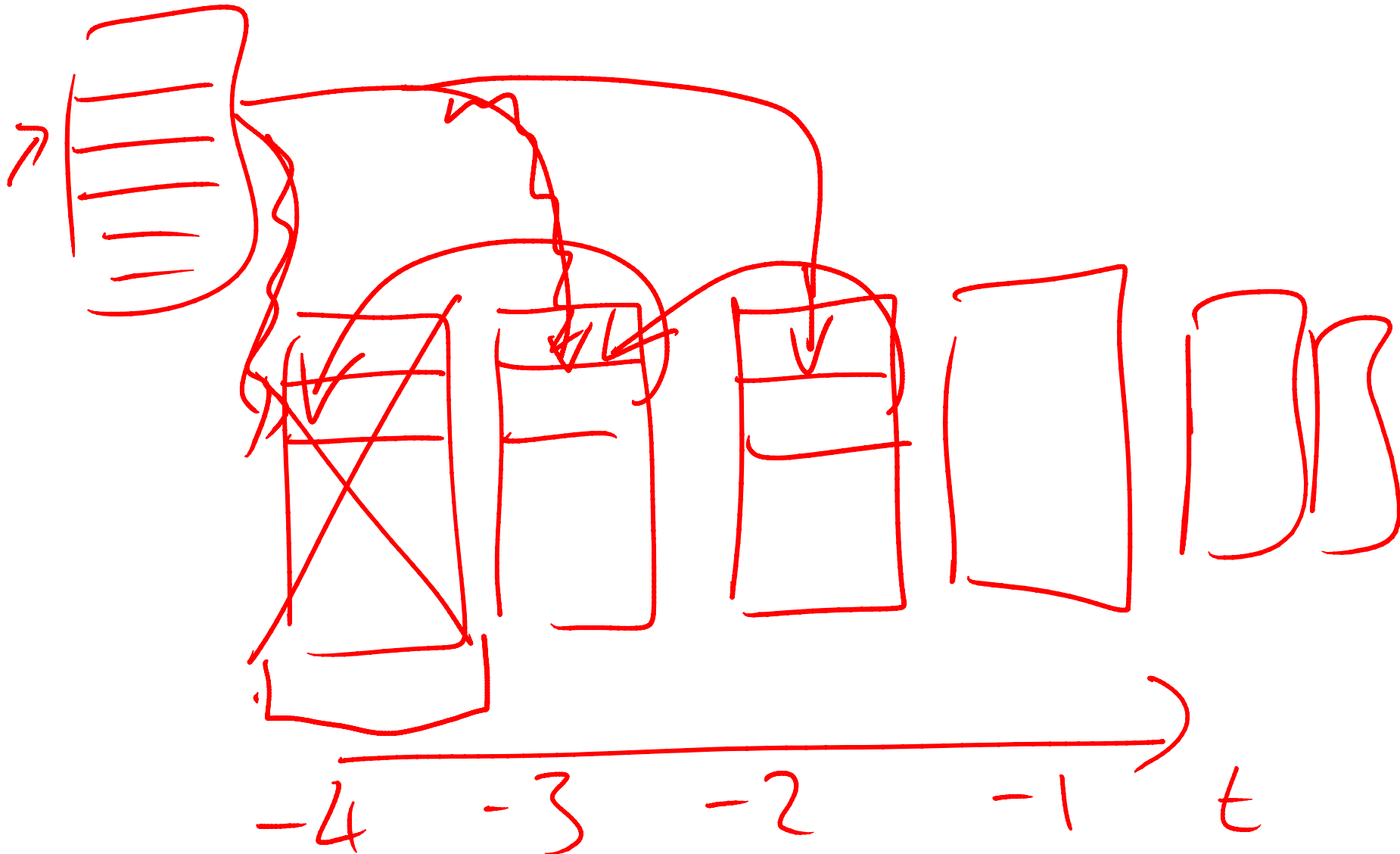
4 M1S12 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.

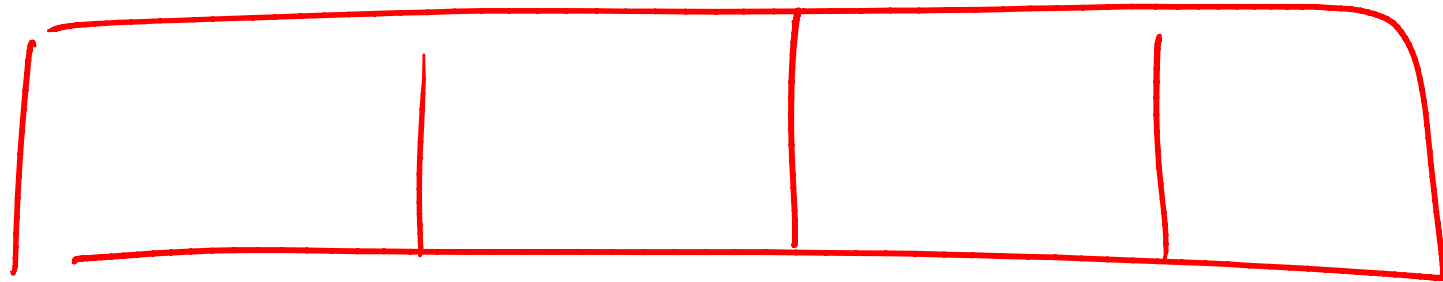


M1S14 & M2S35 Explaining how there's a new portion of the version store created every minute.



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





0 — 7 8 — 15 16 — 23

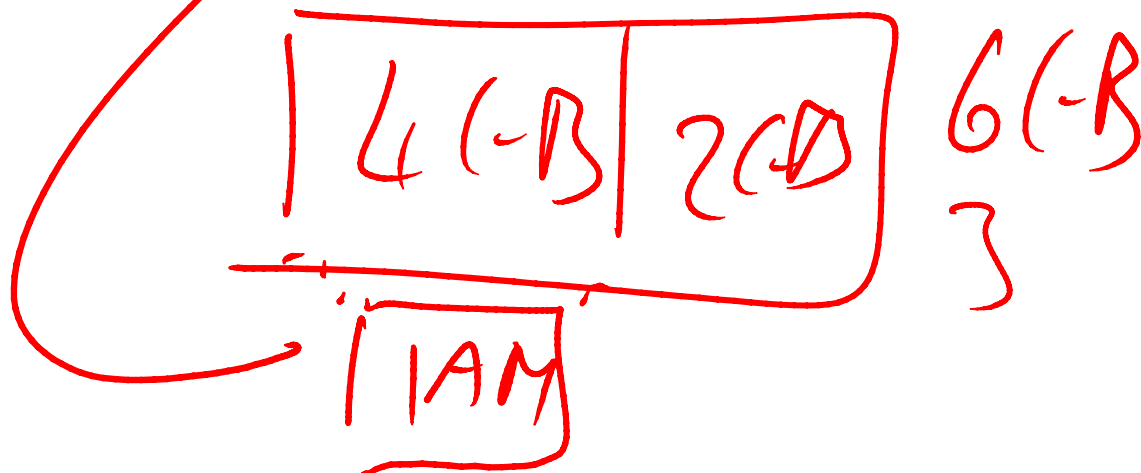


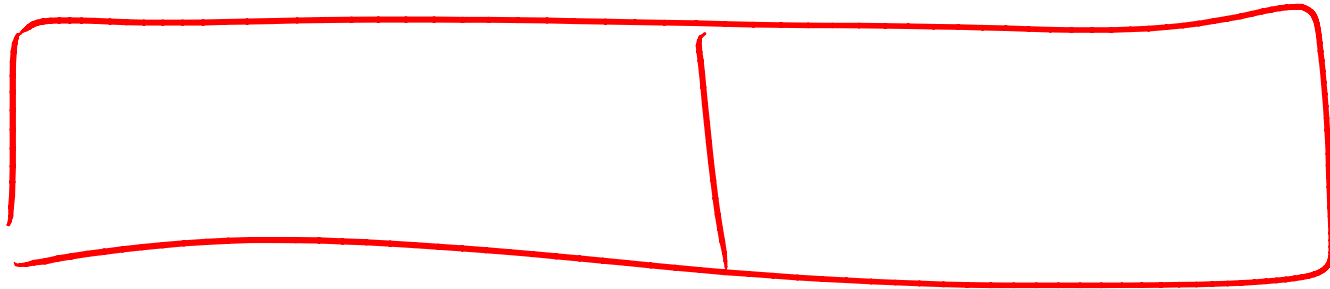
M1S24 Explaining the difference between mixed and dedicated extents. Mixed extent is one in which the 8 pages could be allocated to 8 different objects. Dedicated extent is one where all 8 pages are reserved for exclusive use of a single object.

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.





0 - FH

1 - PFS

2 - CAM⁸⁰⁸⁸

3 - SCAM
S11232

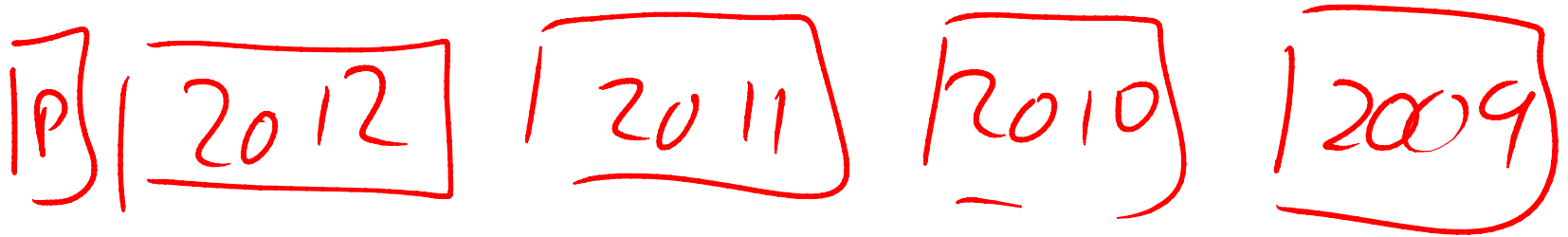
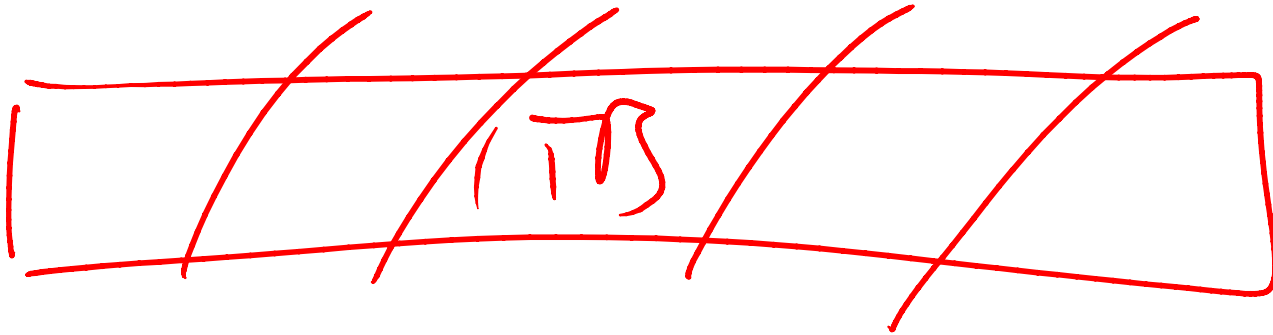
4 }
5 }

NOT USED

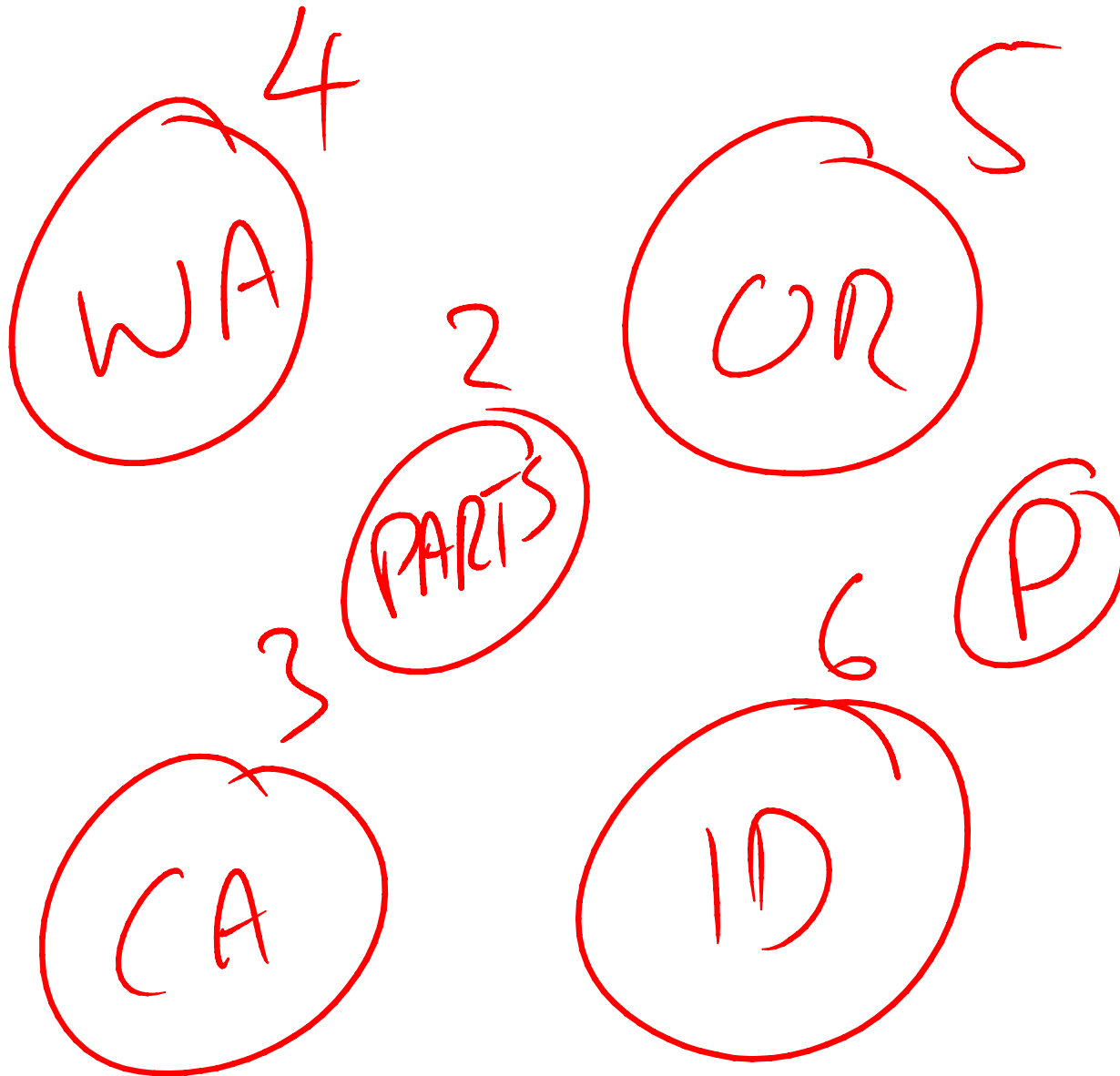
6 - DIFF

7 - ML
S11232

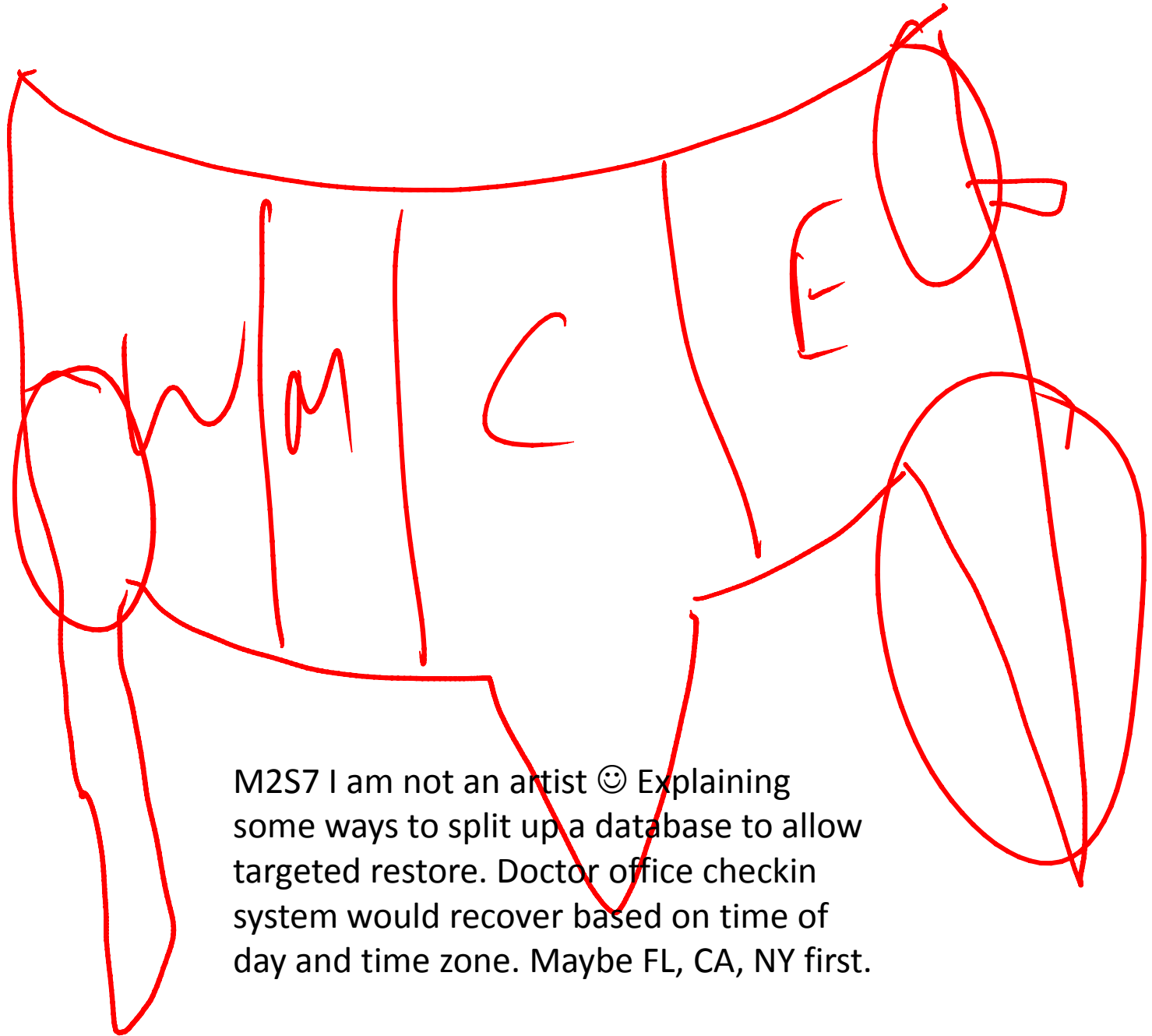
M1 In the first extent in a data file, what are the pages?



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.

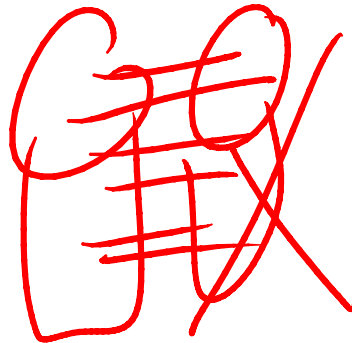


M2S7 Explaining some ways to split up a database to allow targeted restore. Auto parts sales system with states on West Coast – makes sense to bring parts and CA online first, based on population size.



M2S7 I am not an artist 😊 Explaining some ways to split up a database to allow targeted restore. Doctor office checkin system would recover based on time of day and time zone. Maybe FL, CA, NY first.

RØ



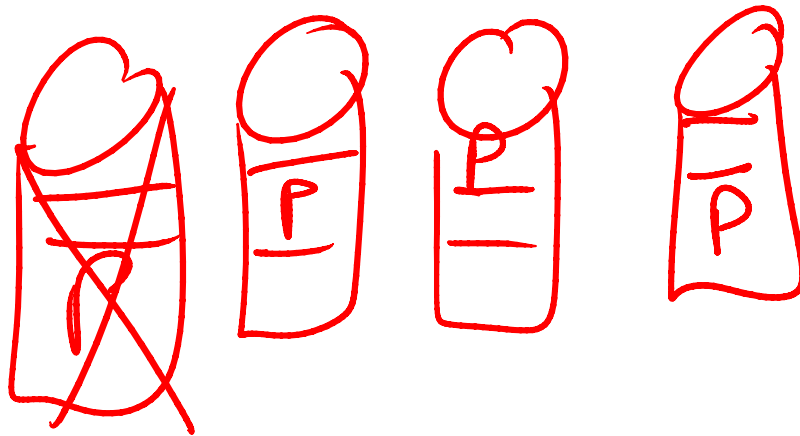
M1 Raid levels. RAID 0 is striping with no redundancy. See <http://en.wikipedia.org/wiki/RAID>

R1



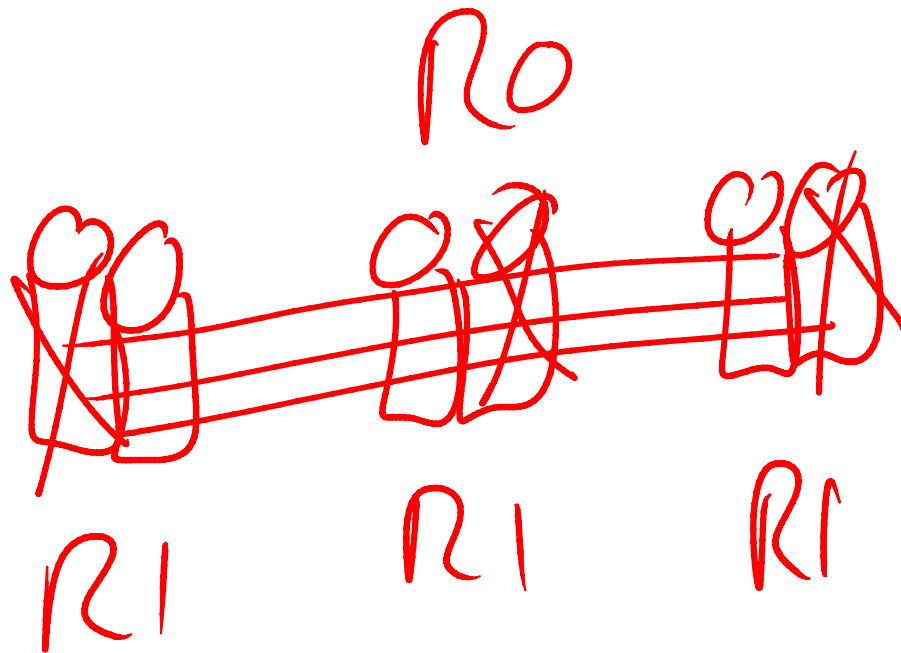
M1 Raid levels. RAID 1 is mirroring. See <http://en.wikipedia.org/wiki/RAID>

RS



M1 Raid levels. RAID 5 is striping with rotating parity. See <http://en.wikipedia.org/wiki/RAID>

$$R10 = R1 + R0$$



M1 Raid levels. RAID 10 is striping across mirror sets. See <http://en.wikipedia.org/wiki/RAID>

SQLIO for benchmarking an I/O subsystem, SQLIOSIM for stress testing.

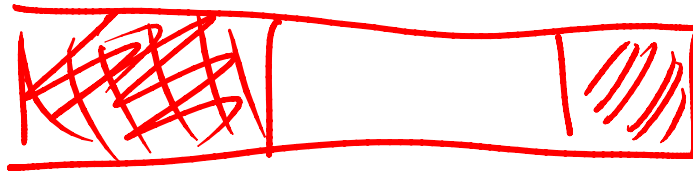
SQLIO

SQLIOSIM

sys.dm_io_virtual_file_stats

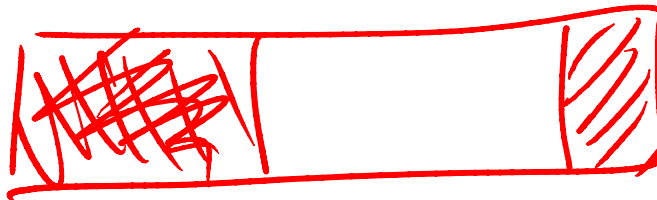
The DMV to use to examine I/O latencies from inside SQL Server –
sys.dm_io_virtual_file_stats. See <http://www.sqlskills.com/blogs/paul/how-to-examine-io-subsystem-latencies-from-within-sql-server/>

RR

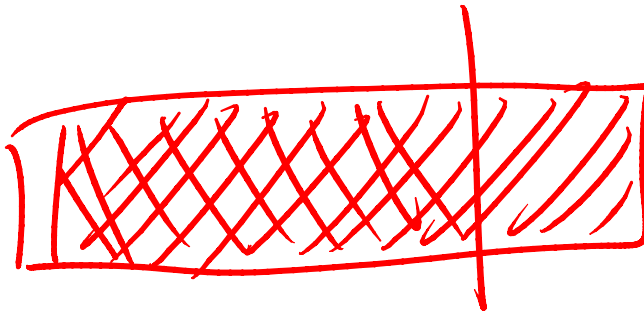


PROP FILL
WEIGHTING

~~8100~~ ←
SO (8099)

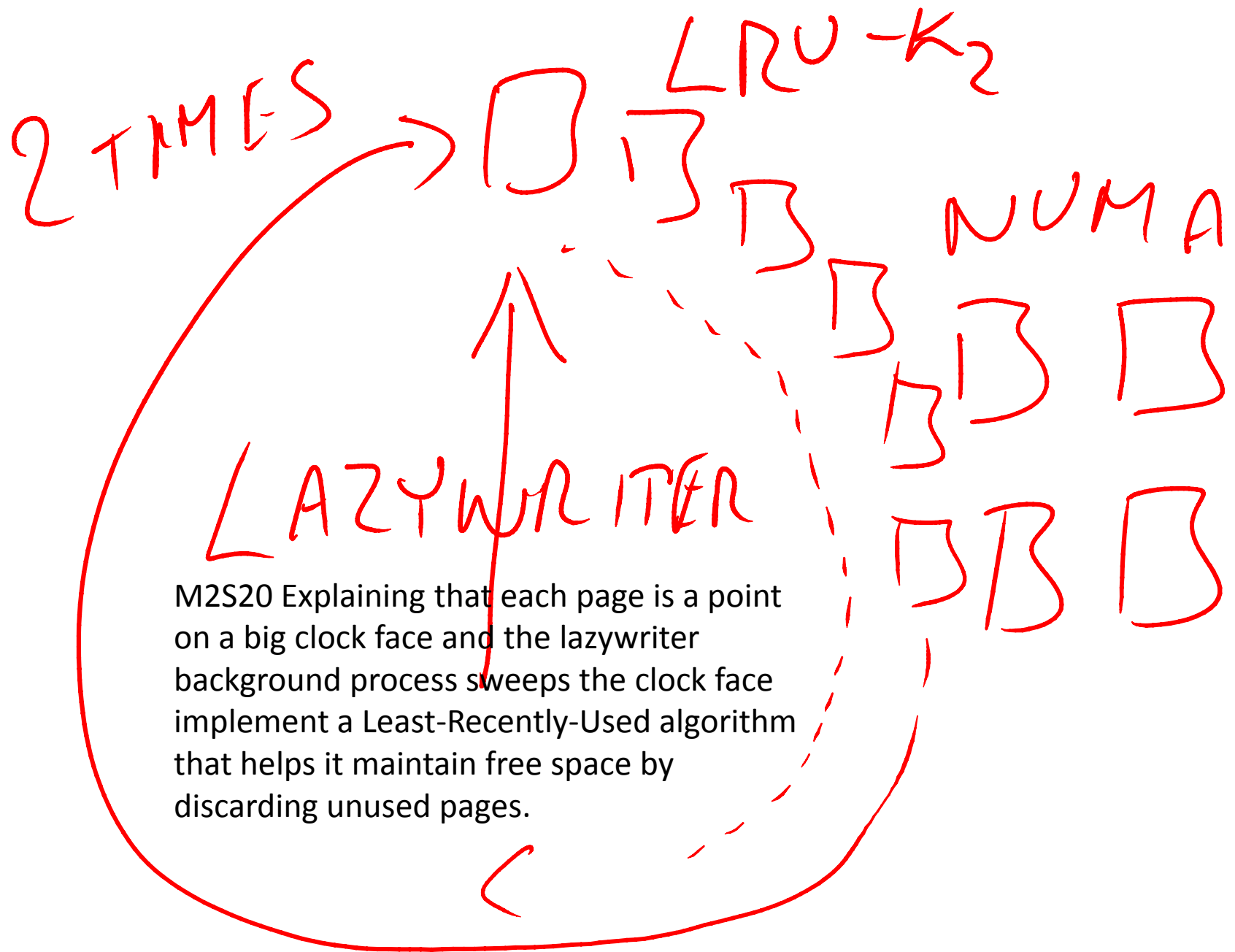


~~8100~~ ←
SO (8099)



1 ← ✓
8100

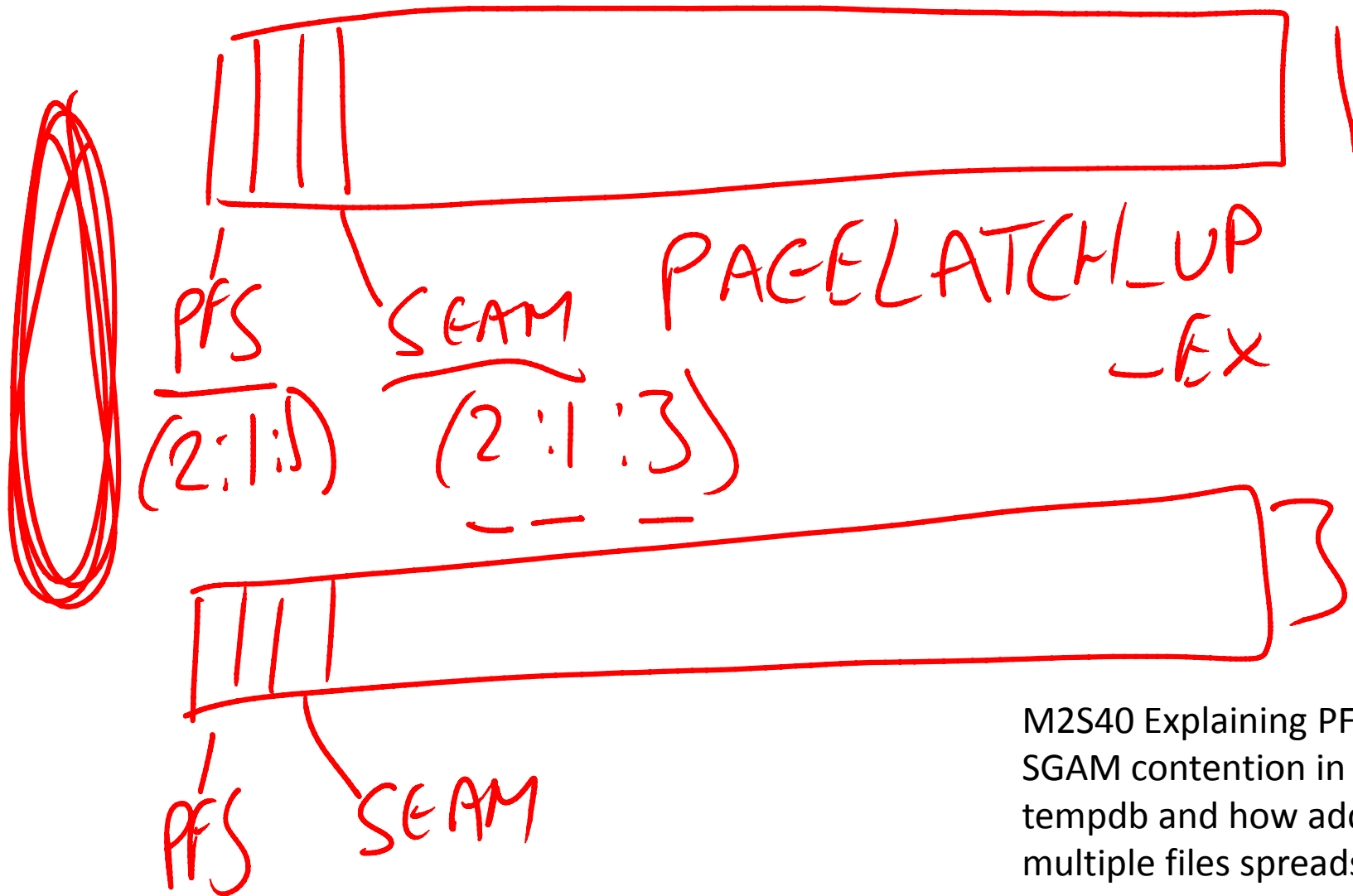
M2S19 Explaining round-robin
allocation and proportional fill



SQL ~~SECURE FILES~~

DBCC PAGE (2)

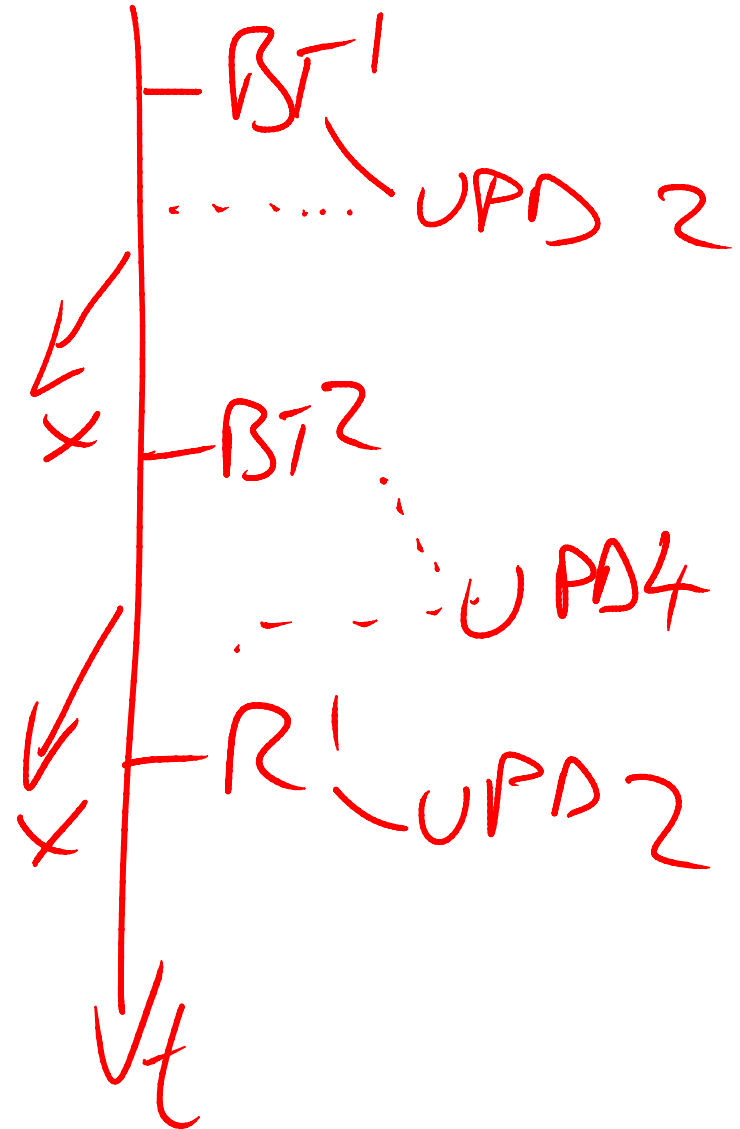
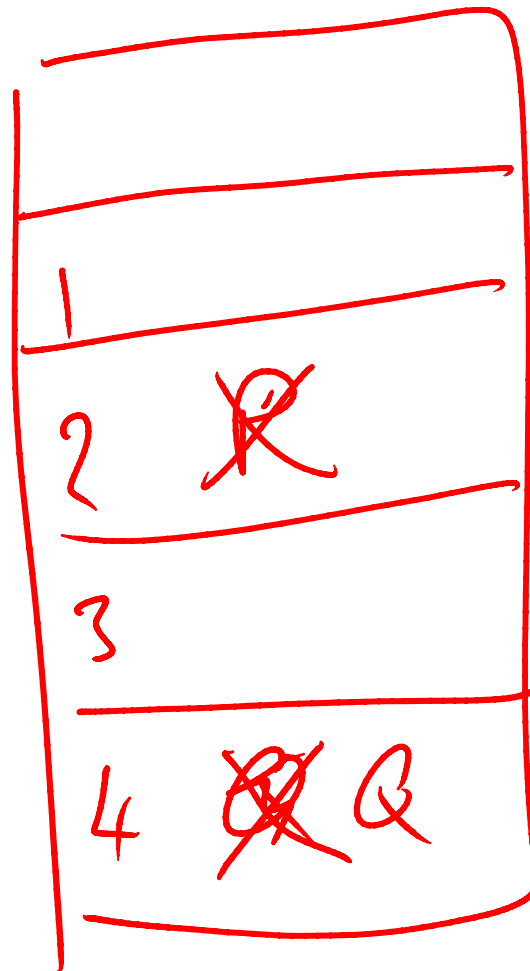
M2S24 Explaining how the security flaw with instant initialization works. Secure files are deleted, SQL files expand and pick up the disk space without zeroing it out. SA using DBCC PAGE can look at the raw bytes of the disk space now included in the data file, potentially leading to information leakage.



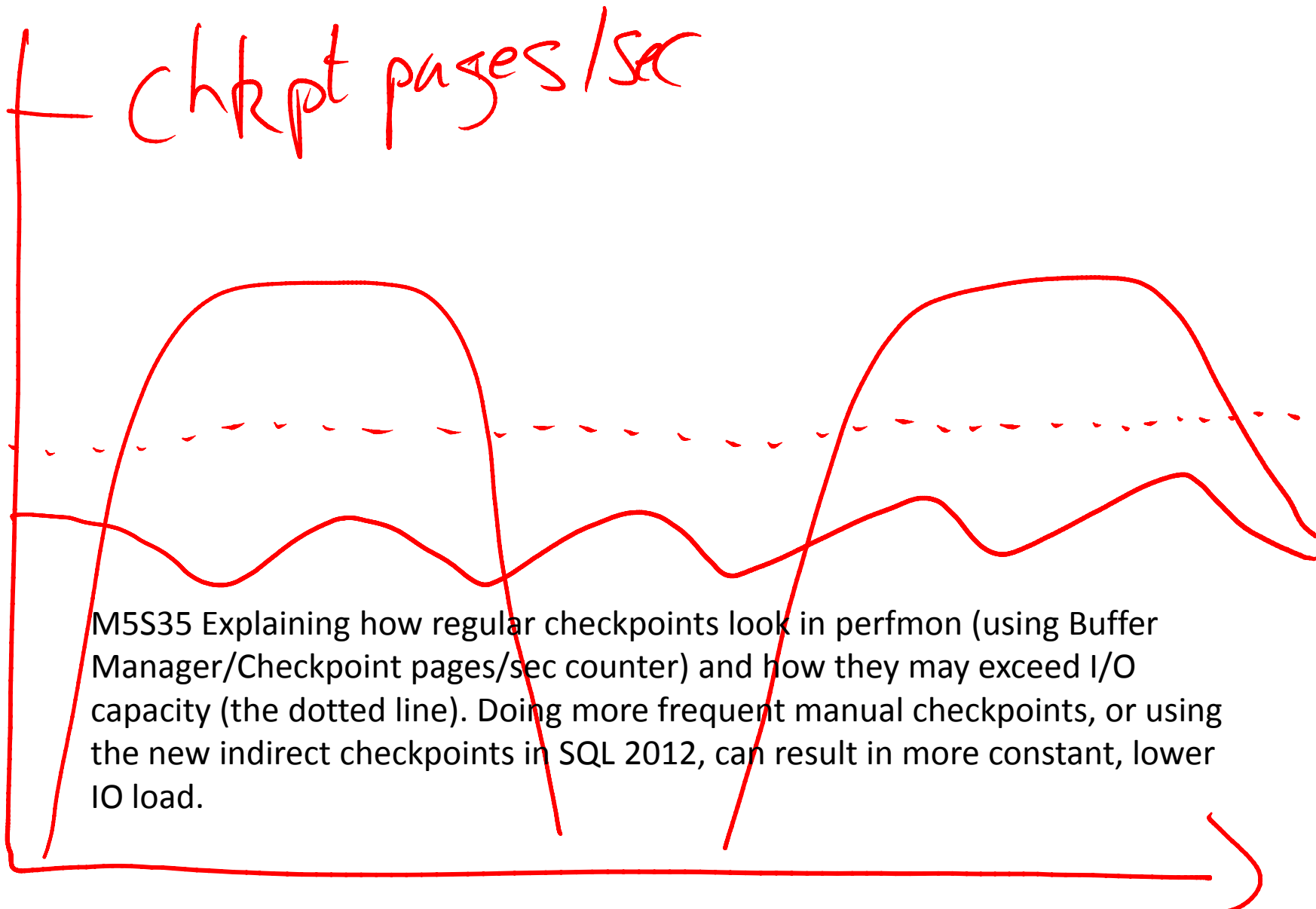
M2S40 Explaining PFS and SGAM contention in tempdb and how adding multiple files spreads the contention over multiple files, thus reducing it.

1:10

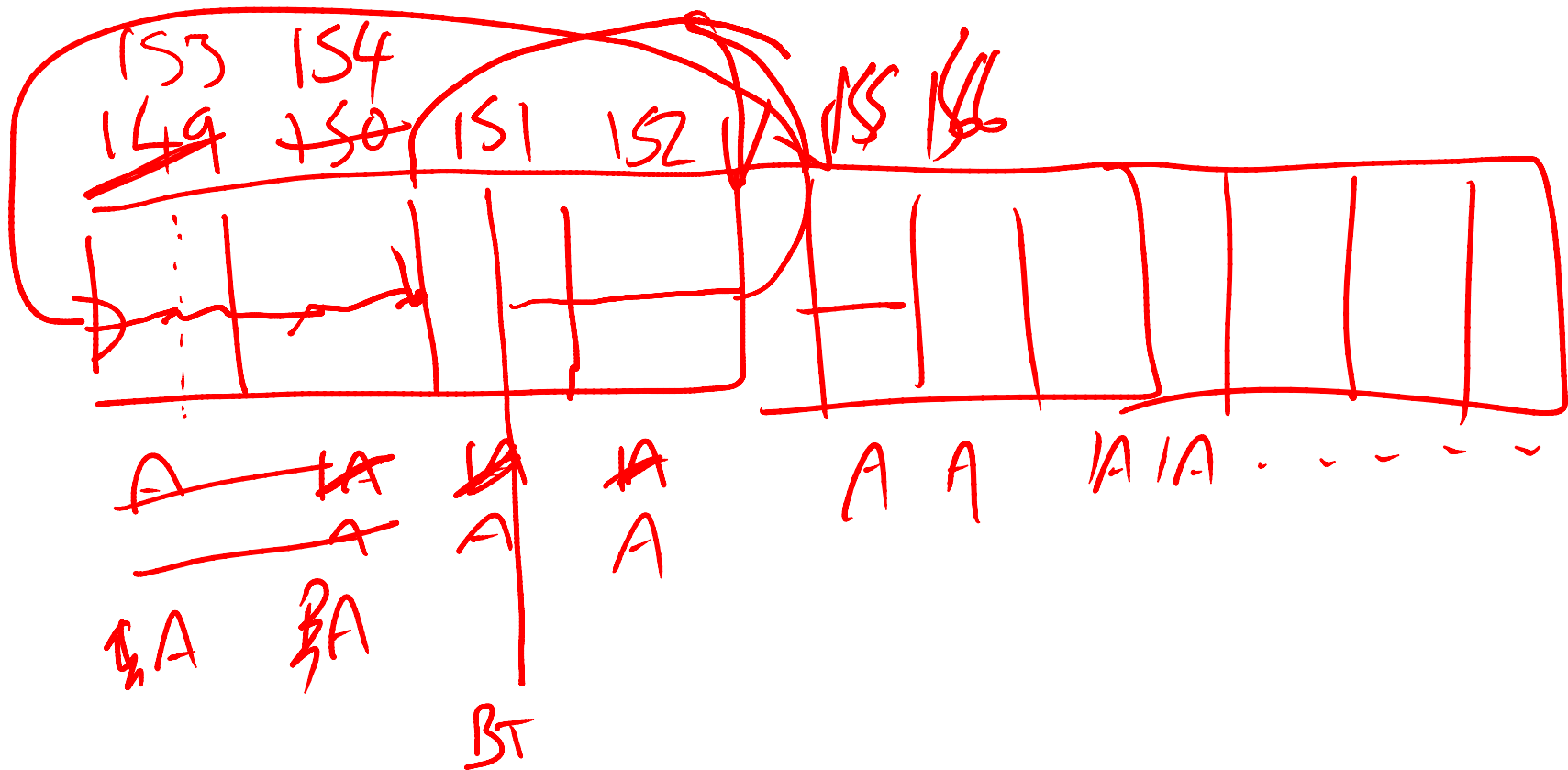
M5 Explaining how different transactions can be changing the same page and also roll back at different times. They are independent operations.



chkpt pages/sec

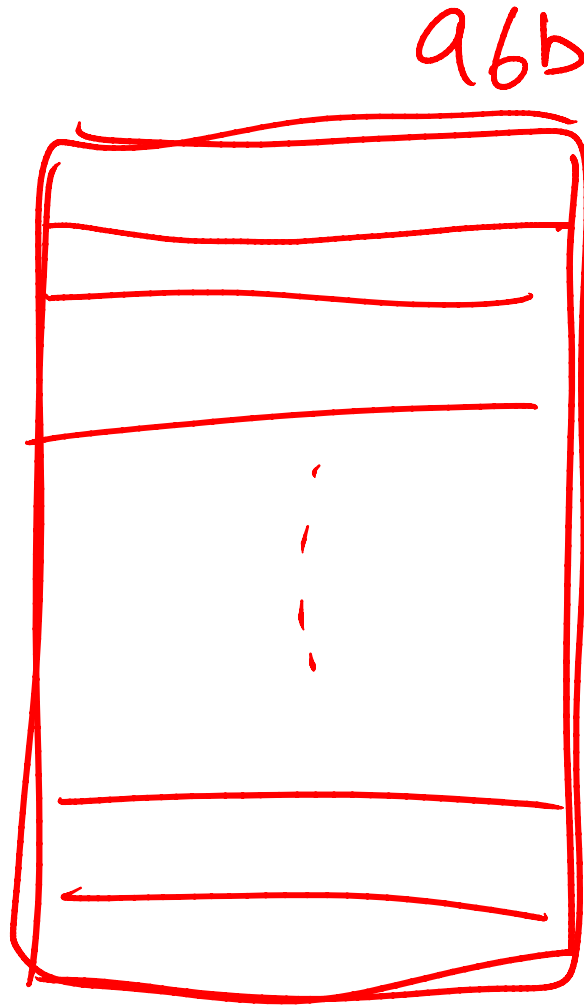


M5S35 Explaining how regular checkpoints look in perfmon (using Buffer Manager/Checkpoint pages/sec counter) and how they may exceed I/O capacity (the dotted line). Doing more frequent manual checkpoints, or using the new indirect checkpoints in SQL 2012, can result in more constant, lower IO load.



M5S28 Explaining the circular log demo. We discussed log space reservation that happens to ensure the active transactions can roll back without having to grow the log – this accounts for extra log growths. Space reserved in the log does not make a VLF become active – as there are no log records written out, just empty space reserved.

16
X512b



PAGE

CHECKSUM

M5 Answering question on what happens if a data file page is only partially written and a transaction needs to change it during rollback. It will fail as the page will have an I/O error. The database will go SUSPECT. Restore from backups or use EMERGENCY-mode repair.

100,000

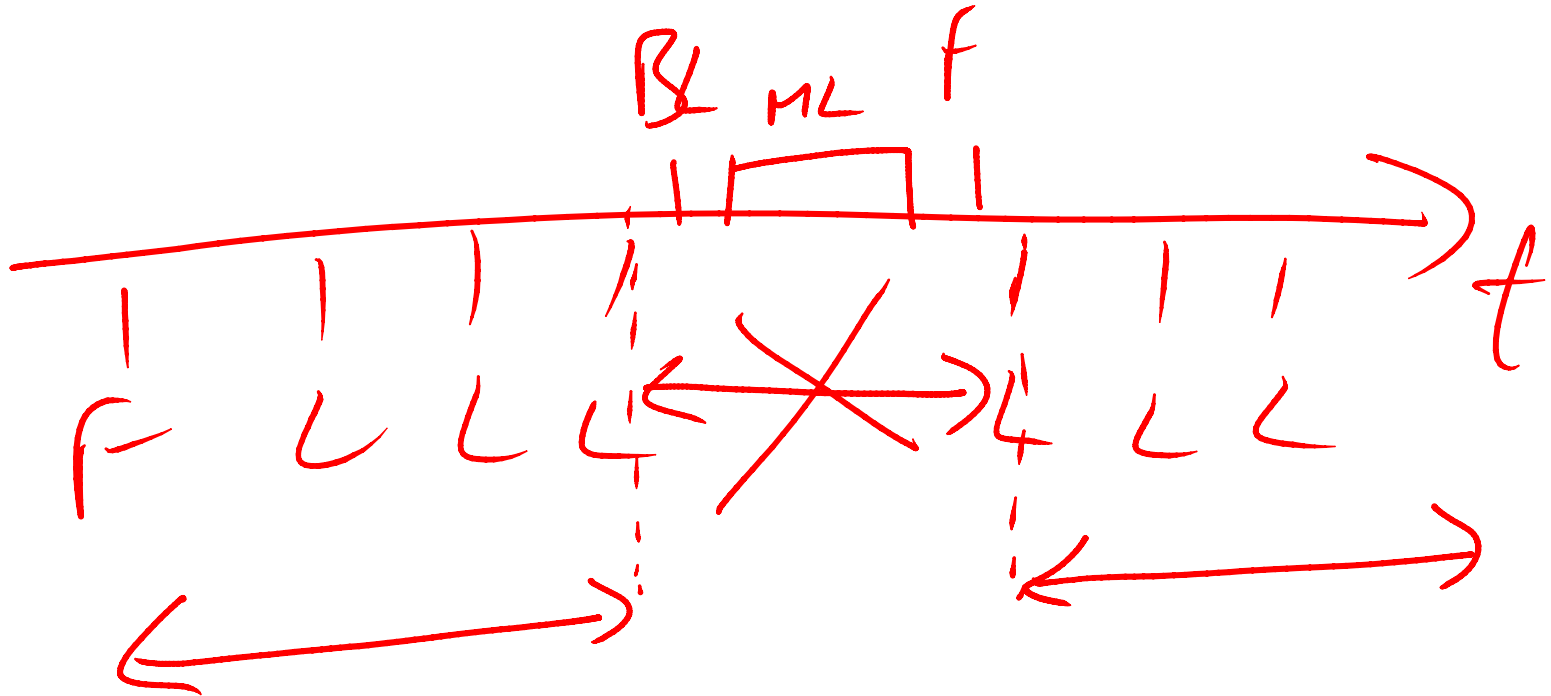
10

M5S60 Explaining that full recovery doesn't mean there's always one log record per operation. An index rebuild in full recovery will log full LOP_FORMAT_PAGE log records instead of multiple LOP_INSERT_ROWS log records. It's more efficient, but is still fully logged.

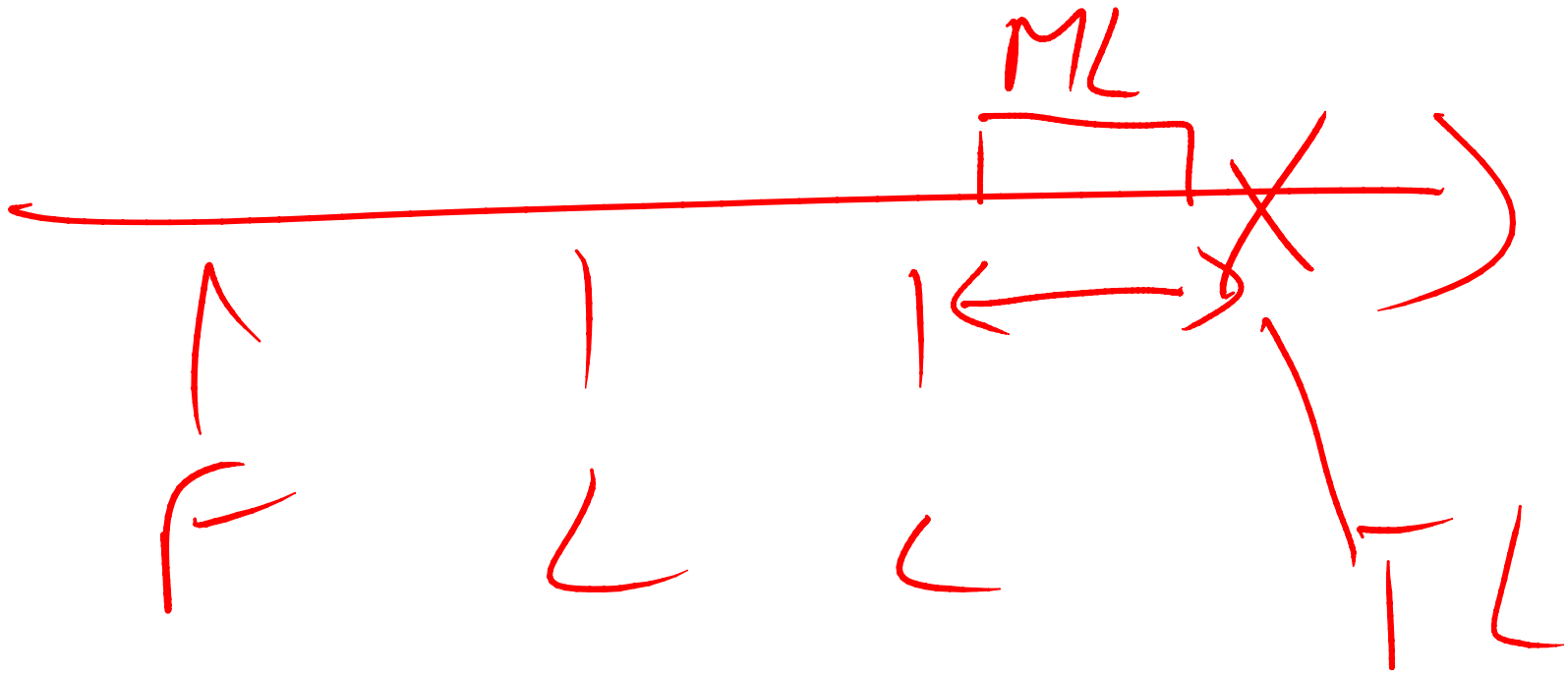
~~INS
INS
INS~~

F - PAGE
F - PAGE

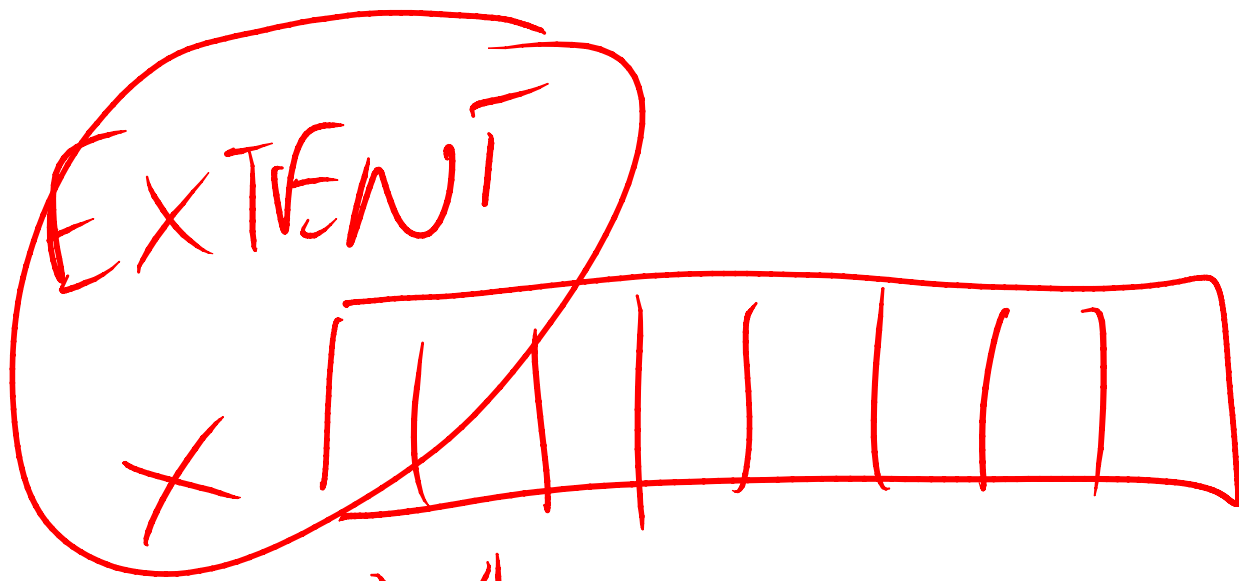
10



M5S60 If a log backup contains a minimally-logged operation, it cannot be used to do a point-in-time restore using STOPAT for any point in time between the beginning and end of the backup. It can be used as part of a restore sequence as normal though.

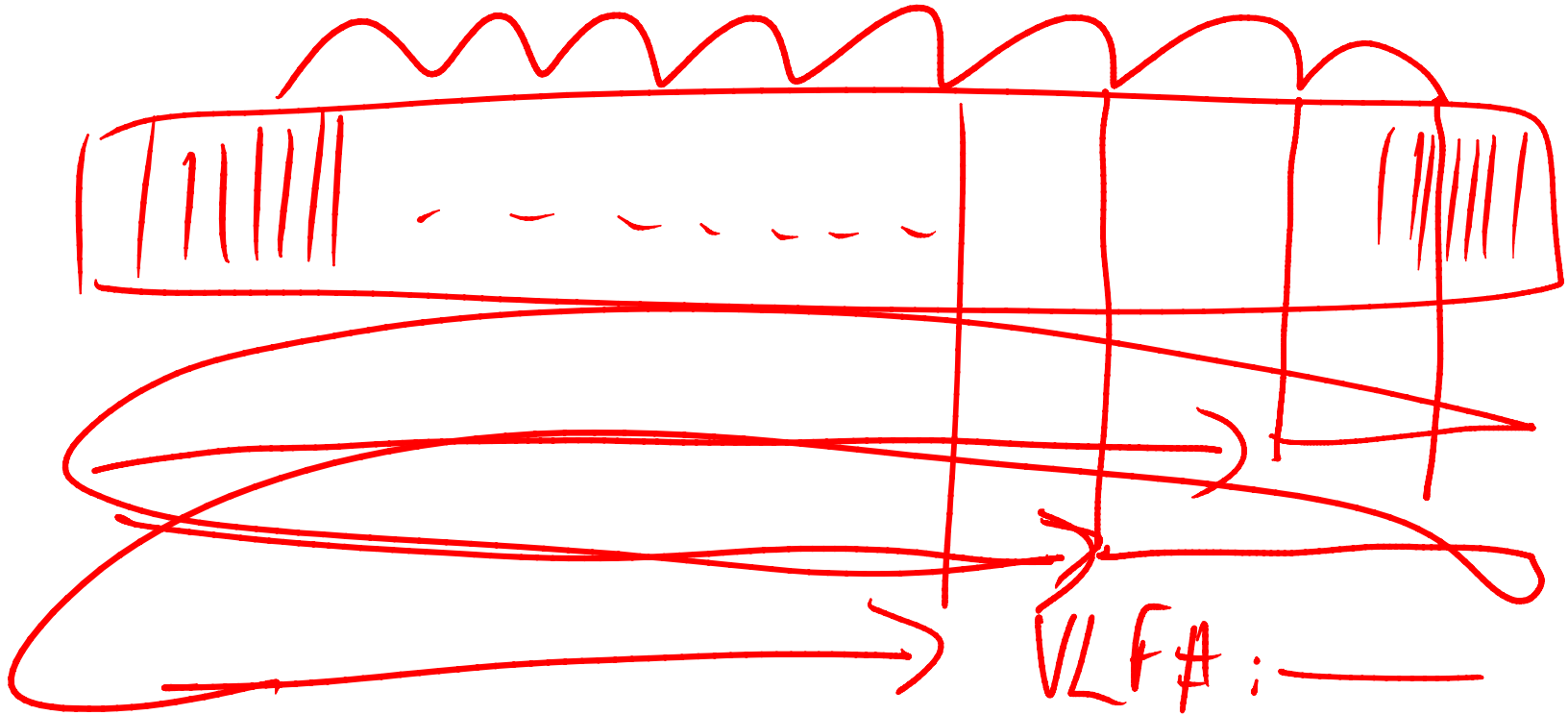


M5S60 If a crash occurs after a minimally-logged operation and before the next log backup, and the data files are damaged, a tail-of-the-log backup will not be possible, or will be corrupt.



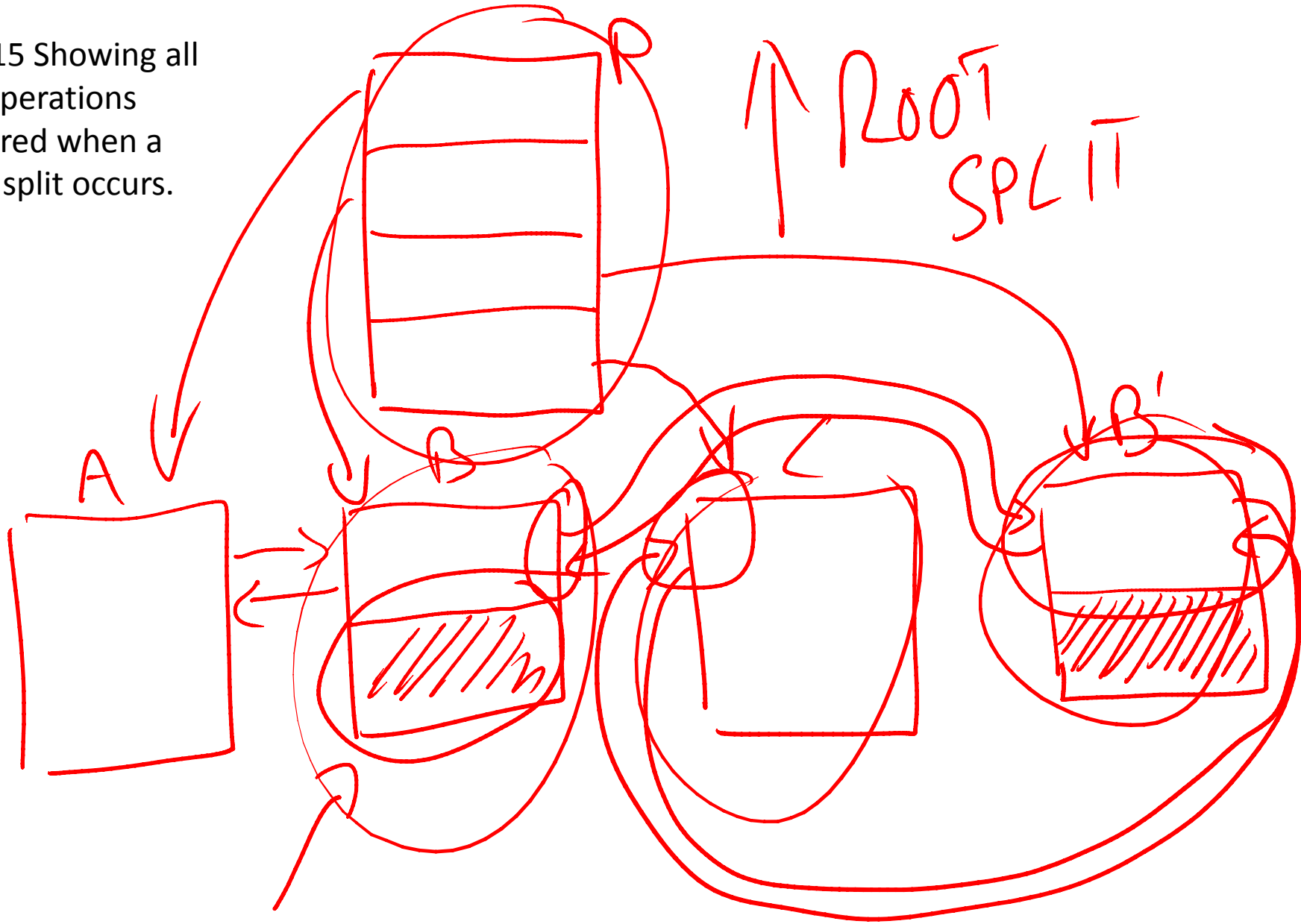
PROBE

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.

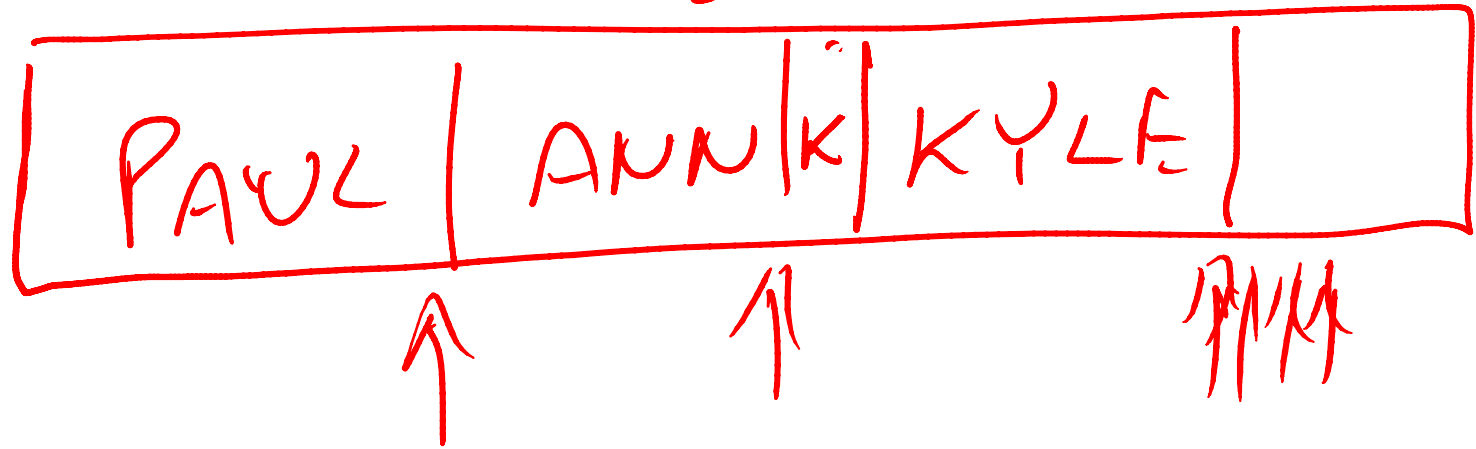
M8S15 Showing all
the operations
required when a
page split occurs.



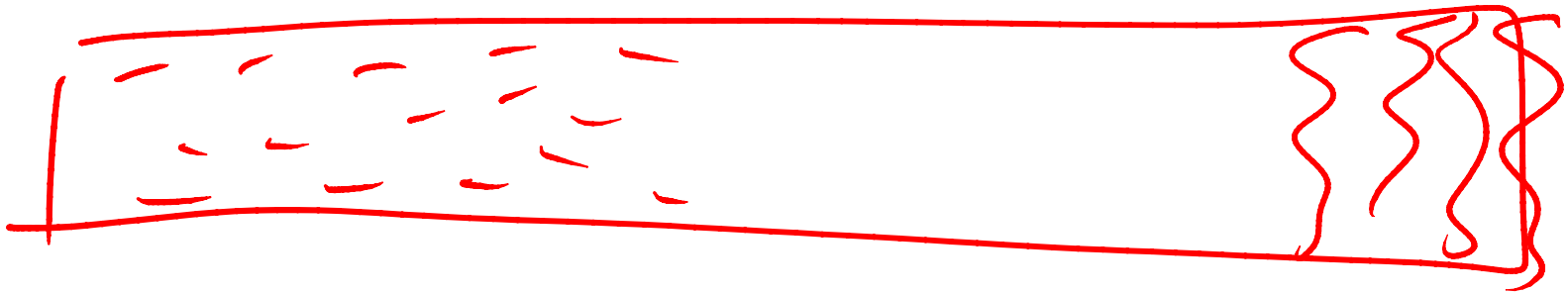
M8S22 The numbers from the page split logging cost demo.

	NO SPLIT	SPLIT	
1000	1244	6772	<u>x5</u>
100	344	5964	<u><u>x18</u></u>
10	256	10364	<u><u>x45</u></u>

MEMBER_ID



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



M8S24 Deletes can caused page density 'fragmentation' with an append-only insert pattern. The space created by the deletes will never be reused by new records, creating empty space and low page density.