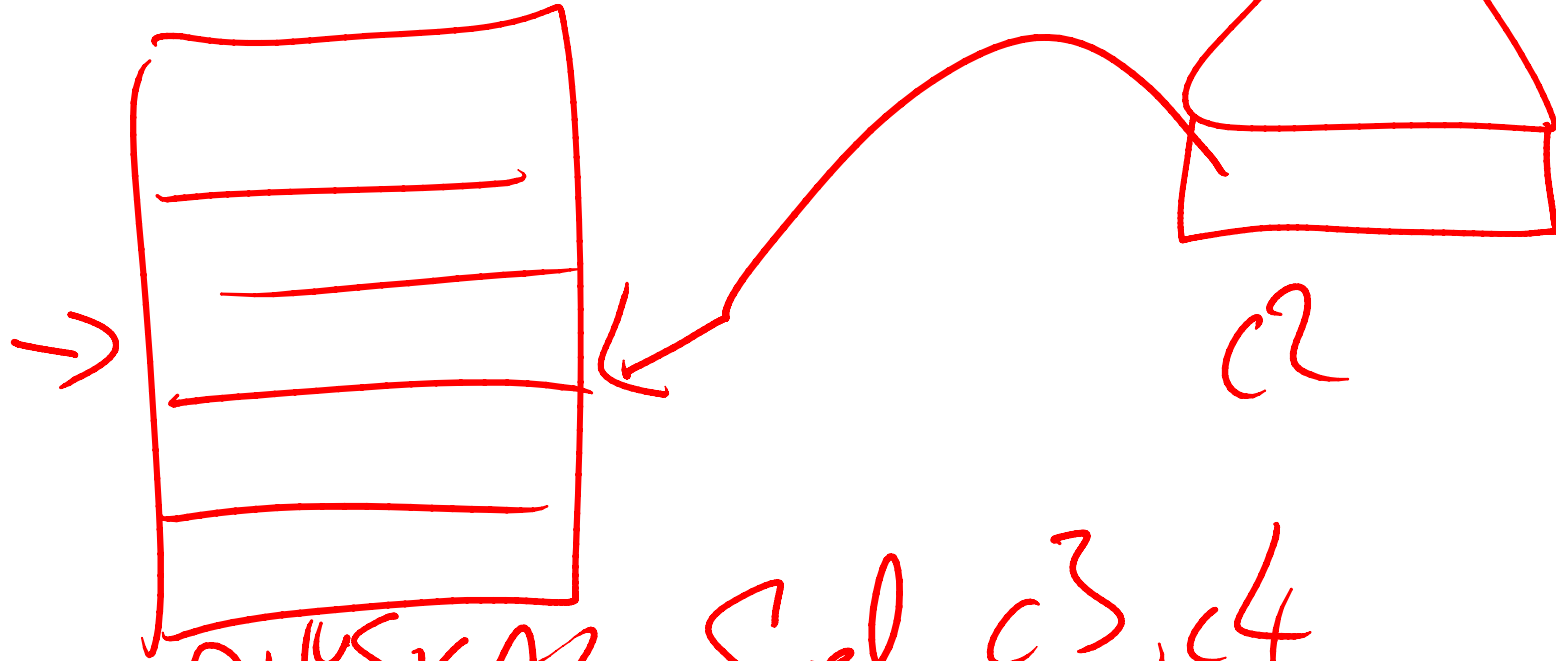


#Sqlhelp

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

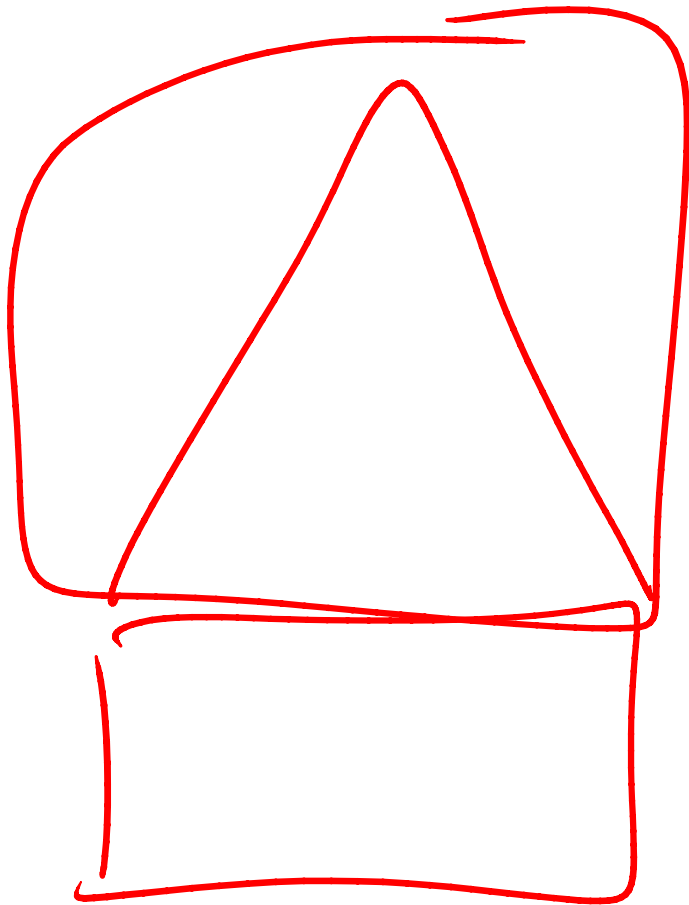


HL: PHYSICAL
RID

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.

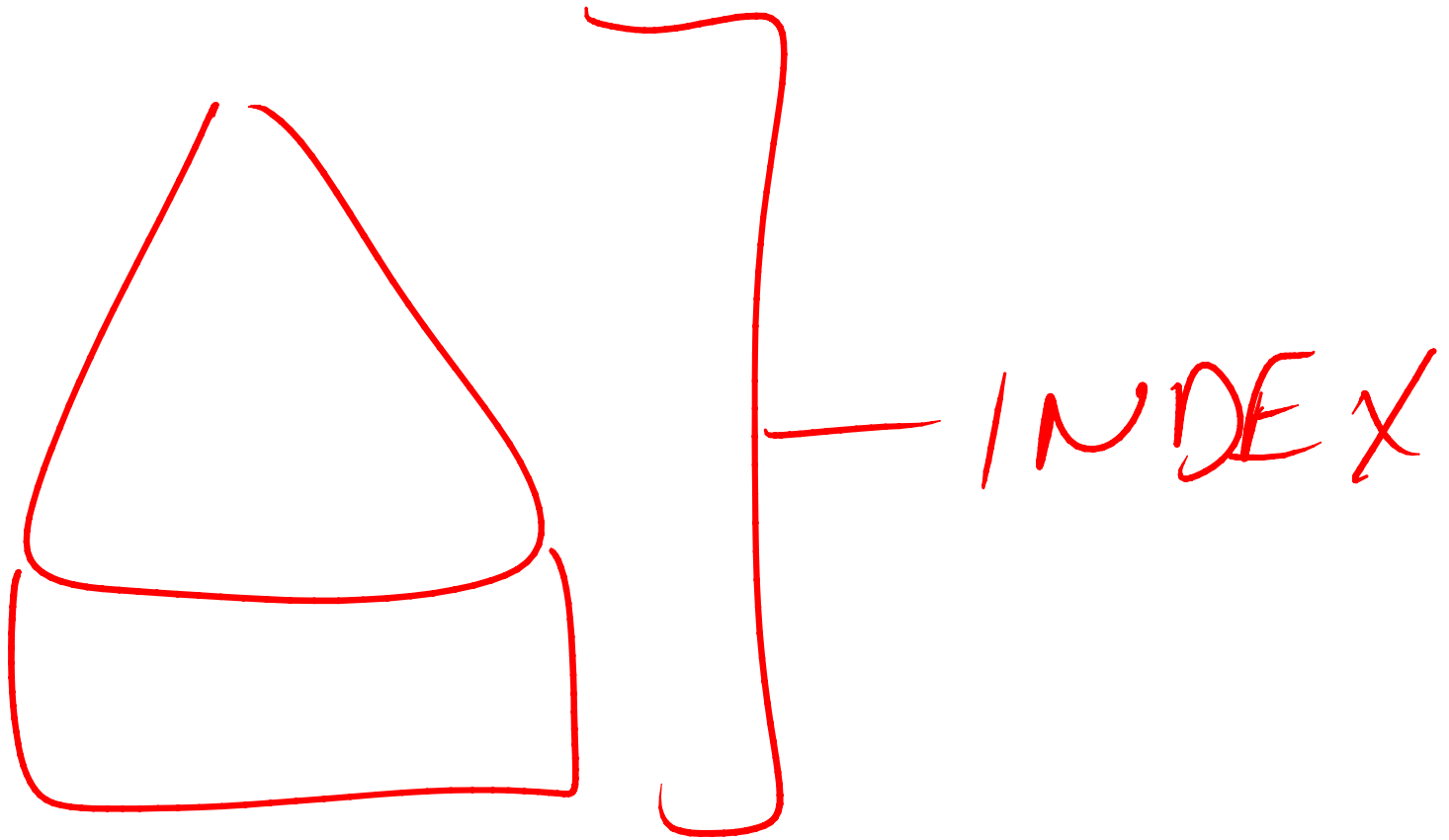


TREE

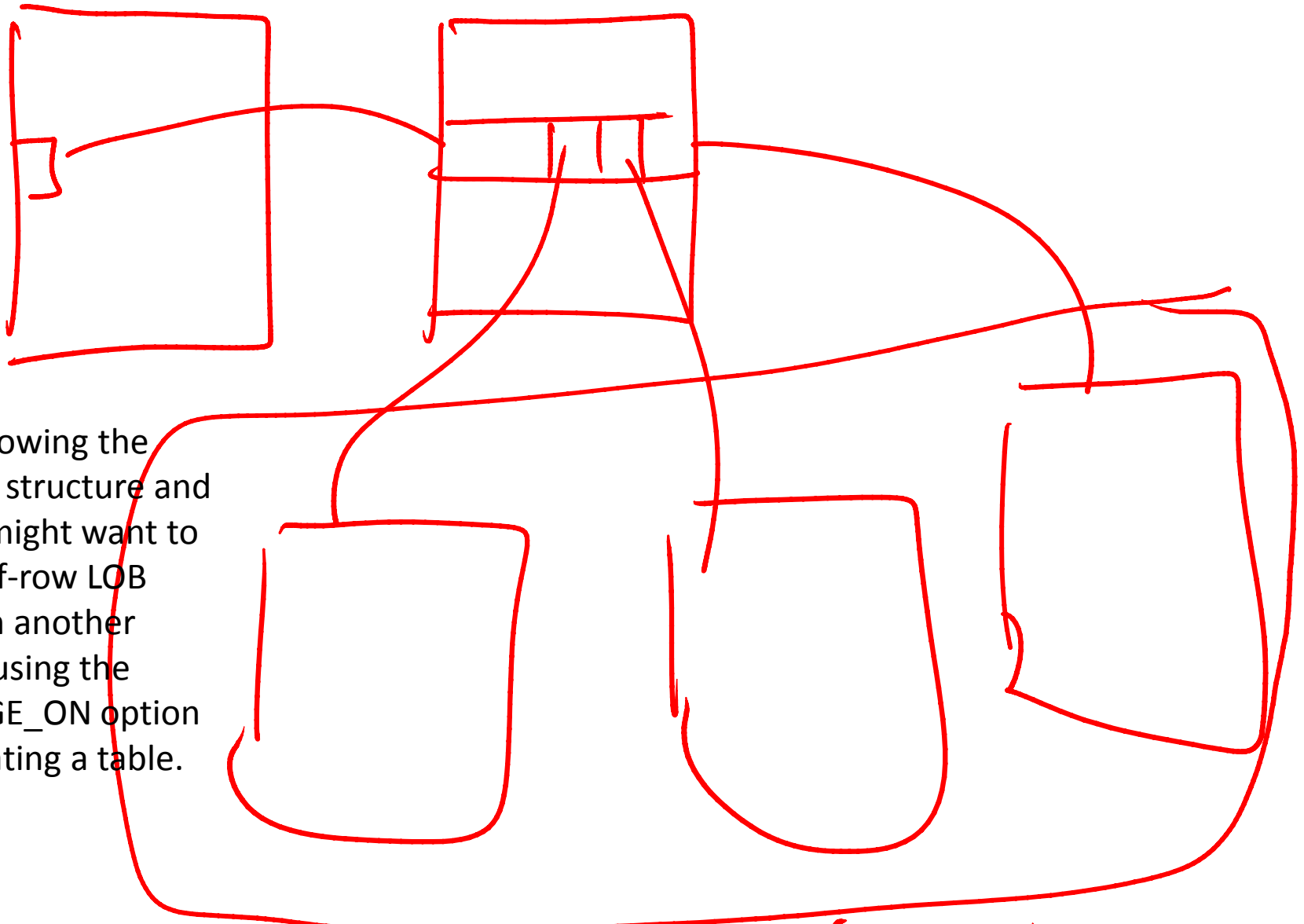
INDEX

DATA

M1S8 Explaining the division of data records/pages and index records/pages in a clustered index.



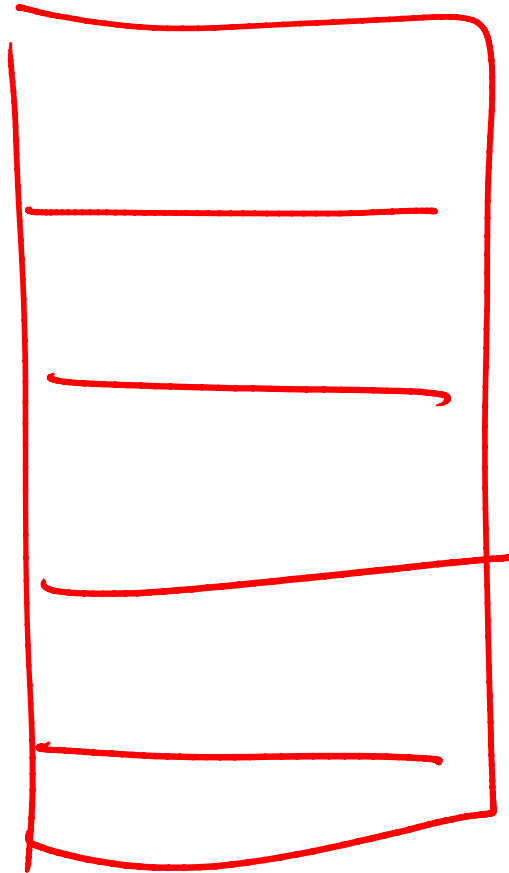
M1S8 Explaining the division of data records/pages and index records/pages in a nonclustered index.



M1S11 Showing the loose tree structure and how you might want to put the off-row LOB storage on another filegroup using the TEXTIMAGE_ON option when creating a table.

TEXTIMAGE_ON

8K

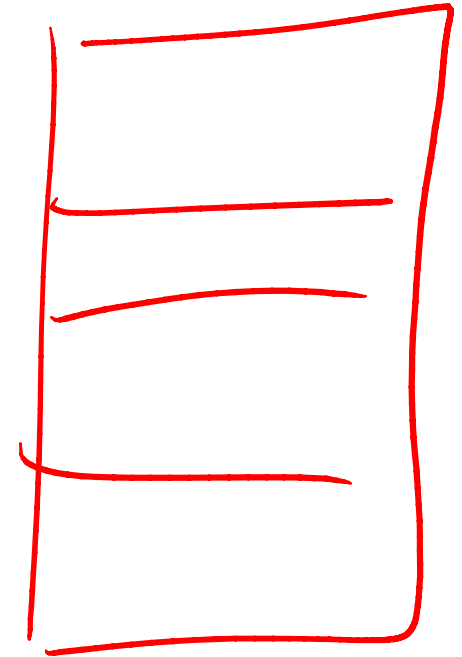


M1S12 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.

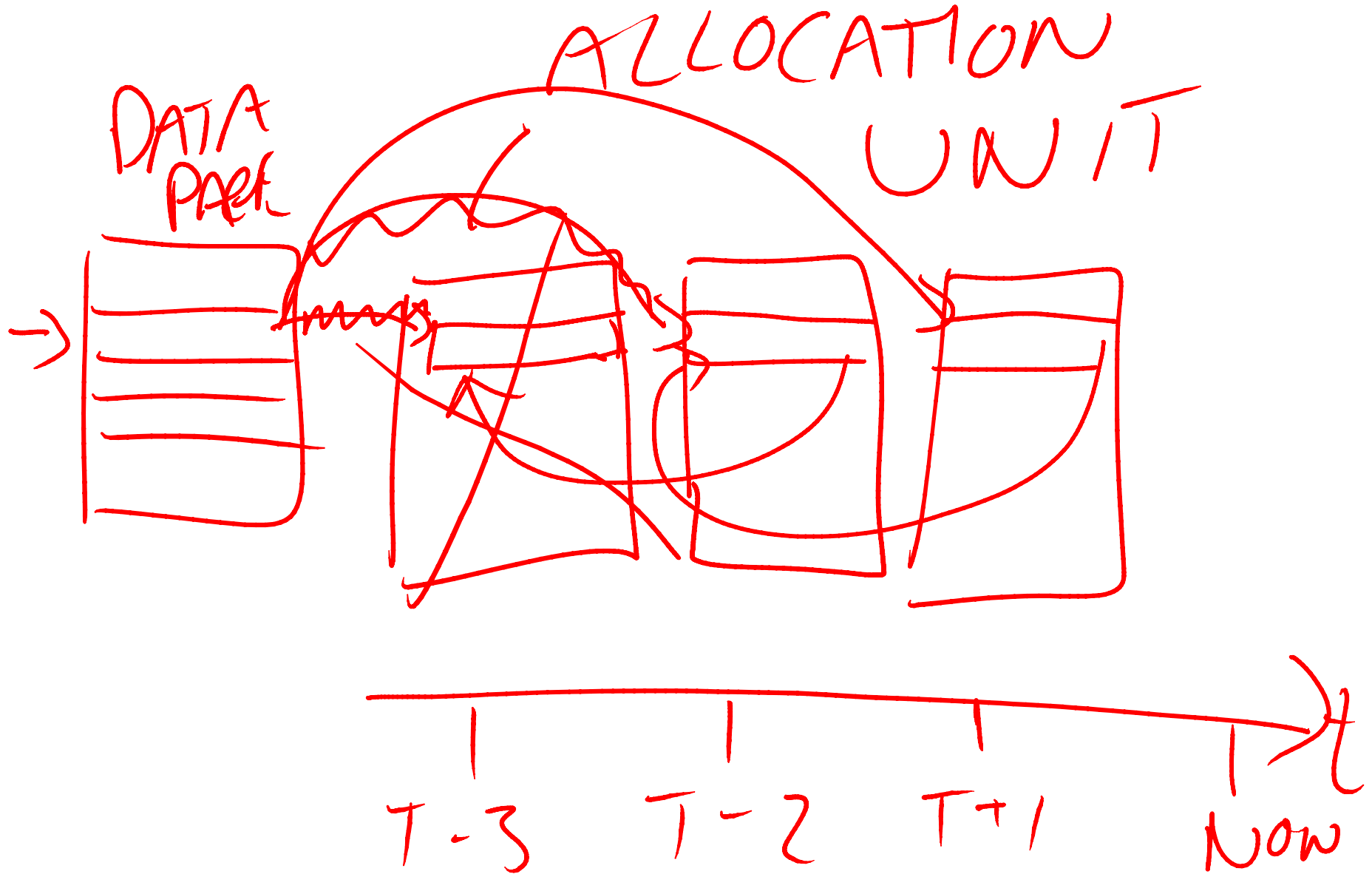
100	2000
125	2025

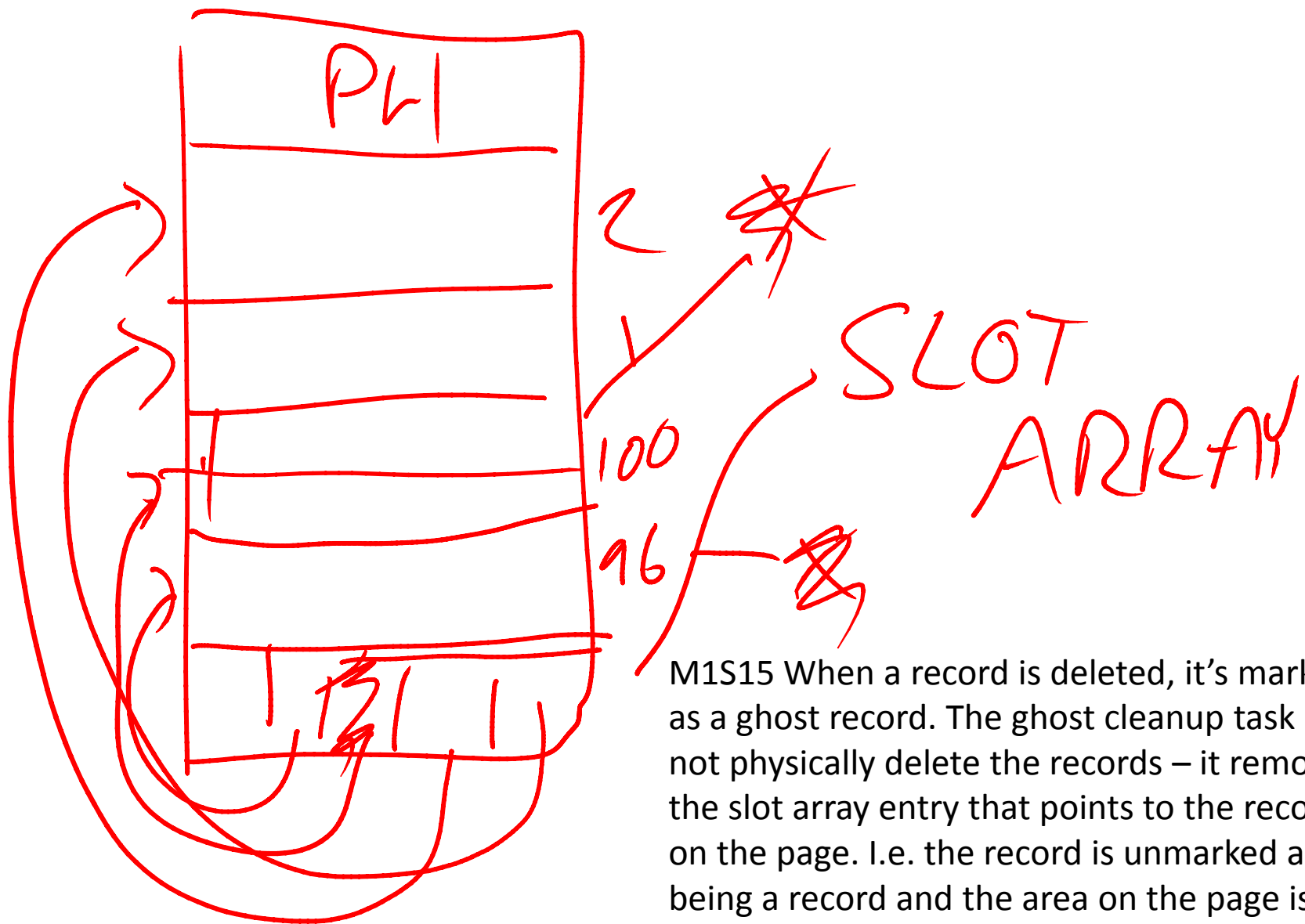
1st 1/6

6300



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





M1S15 When a record is deleted, it's marked as a ghost record. The ghost cleanup task does not physically delete the records – it removes the slot array entry that points to the record on the page. I.e. the record is unmarked as being a record and the area on the page is now free space – that just happens to have bits and bytes that used to be a valid record.

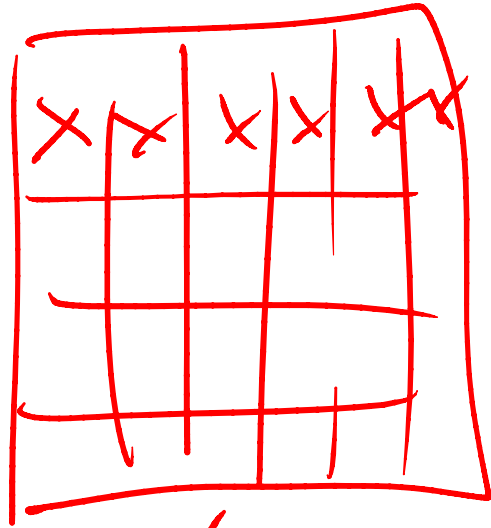
LOCK HASH

c1, c2, c3, c4

M1 Discussing lock hash collisions. See <http://bit.ly/YplGld>

A B C D → X

Z Y X W → X



DISFAVORING

CHECK DB

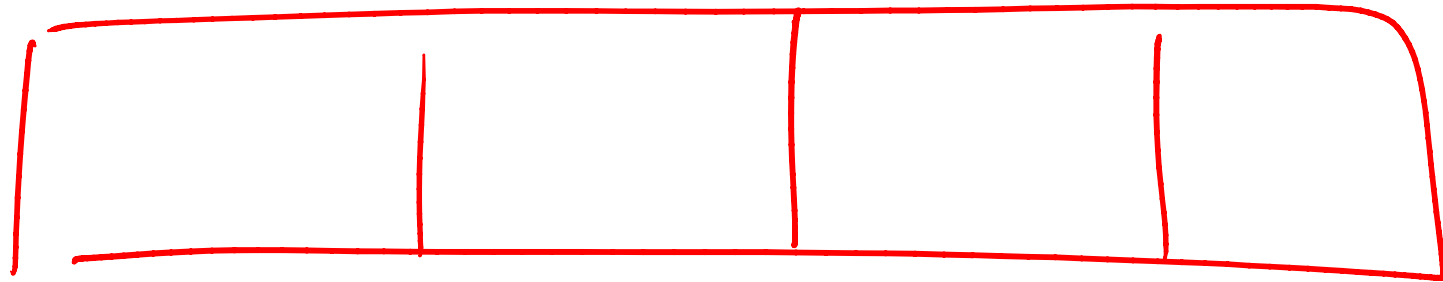
BLOG
POST

TABLE SCAN > 10% BP

sys.dm-os_buffer-
descriptors

M1 We jumped into the buffer pool and
disfavoring. Check out my blog post at

<http://bit.ly/14SZGVp>



0 — 7 8 — 15 16 — 23

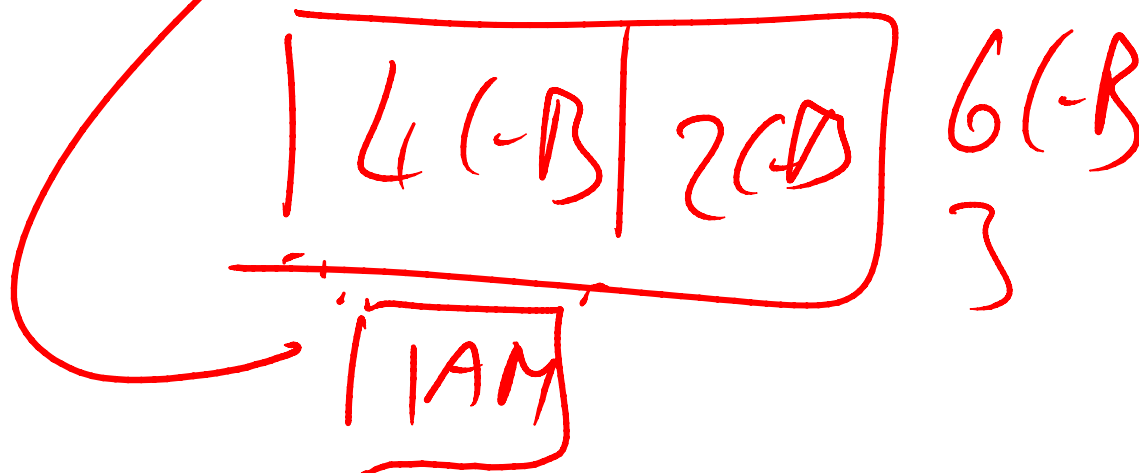


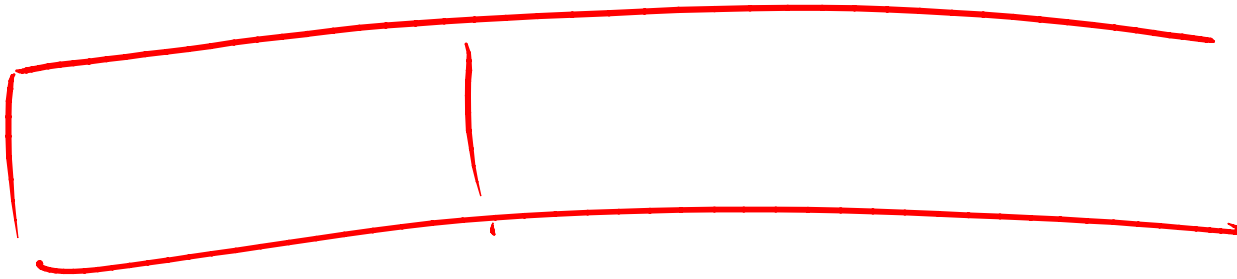
M1S24 Explaining the difference between mixed and dedicated extents. Mixed extent is one in which the 8 pages could be allocated to 8 different objects. Dedicated extent is one where all 8 pages are reserved for exclusive use of a single object.

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 - GAM

3 - SGAM

4 - UNUSED

5 - UNUSED

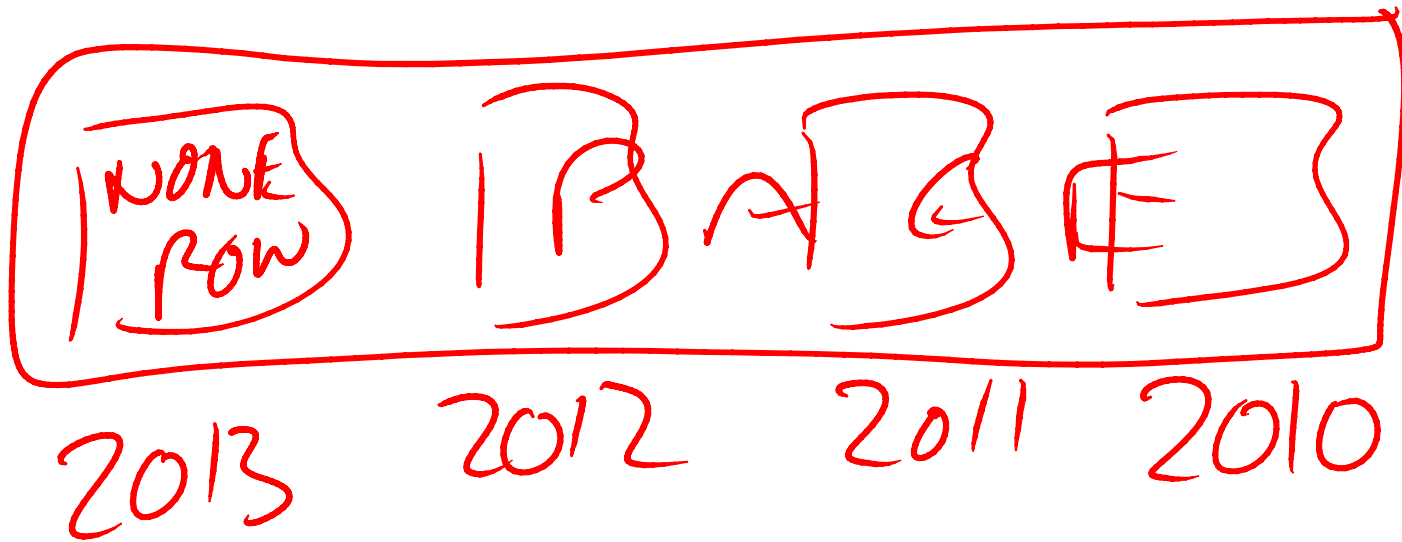
6 - DIFF

7 - ML

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

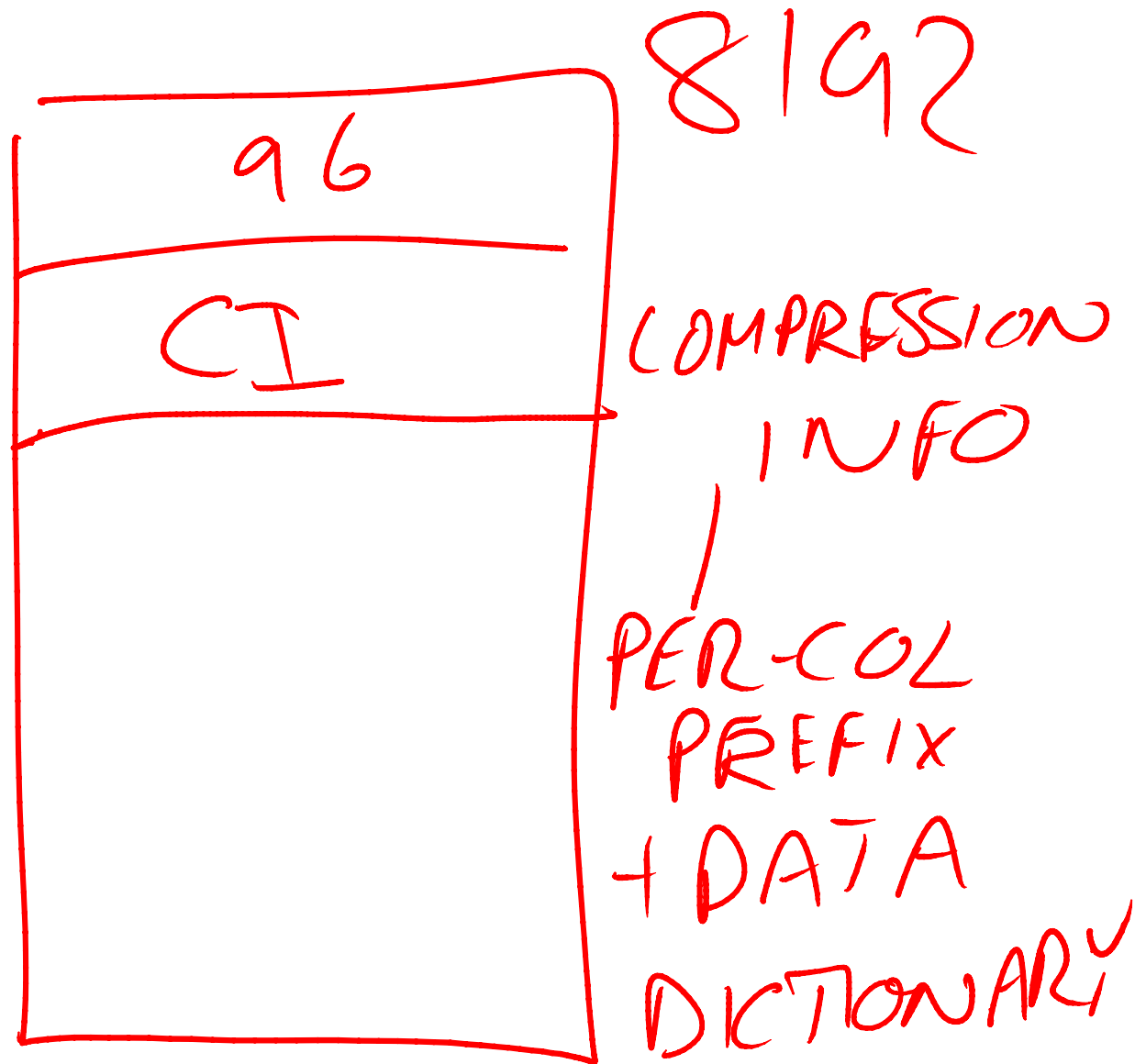
2008 SYSTEM CATALOG VIEWS POSTER

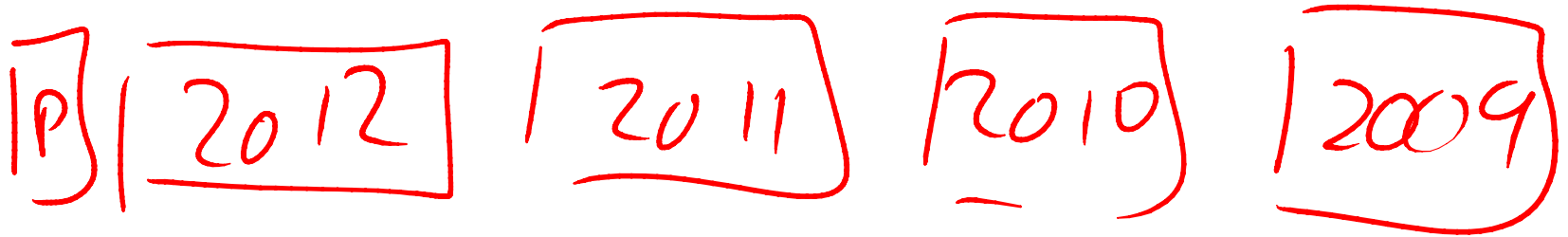
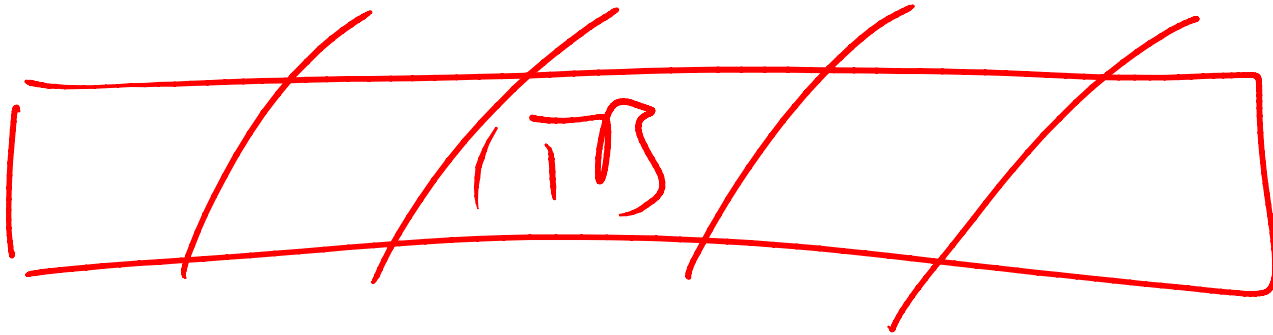
dit.ly/g4ATKt



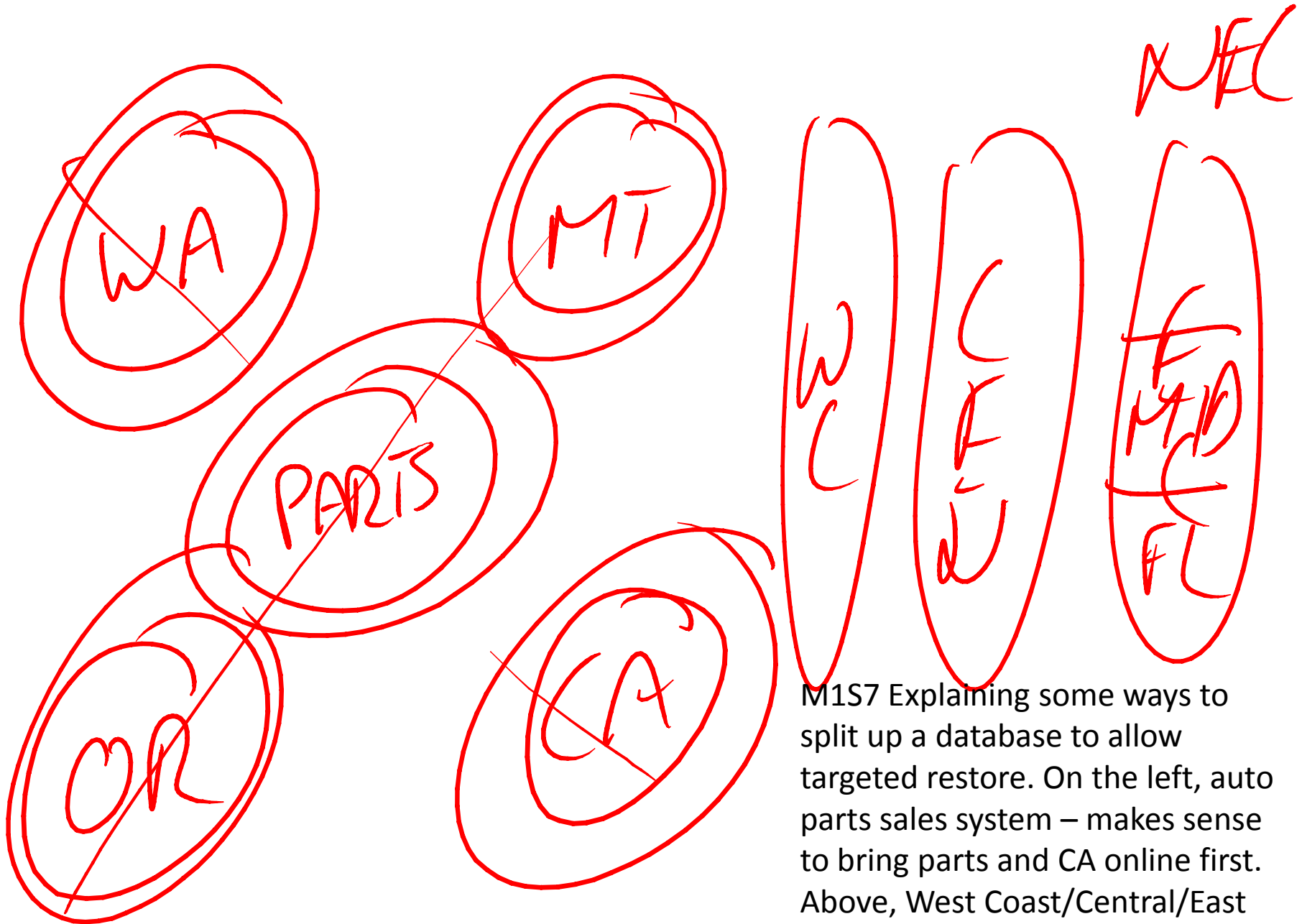
M1S50 Discussing how different partitions in a table can have different compression settings.

M1S54 Showing
where the
Compression
Information record
is stored on 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.



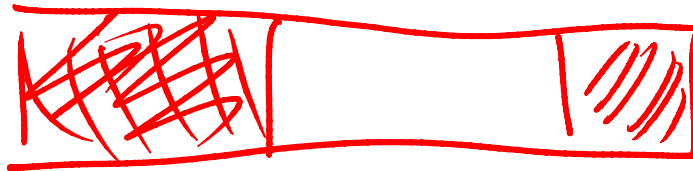
M1S7 Explaining some ways to split up a database to allow targeted restore. On the left, auto parts sales system – makes sense to bring parts and CA online first. Above, West Coast/Central/East Coast as filegroups.

wmic

diskpart

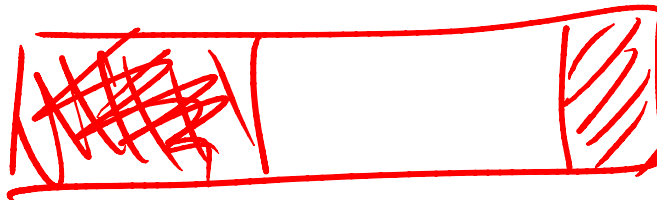
M2S8 The two commands you can use to check partition alignment. Don't use diskpart though as it rounds and can give false results.

RR

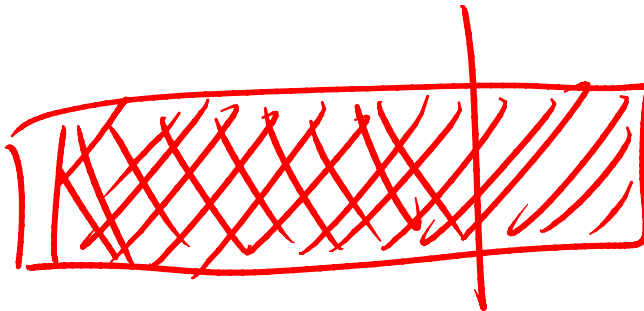


PROP FILL
WEIGHTING

~~8100~~ ←
SO (8099)

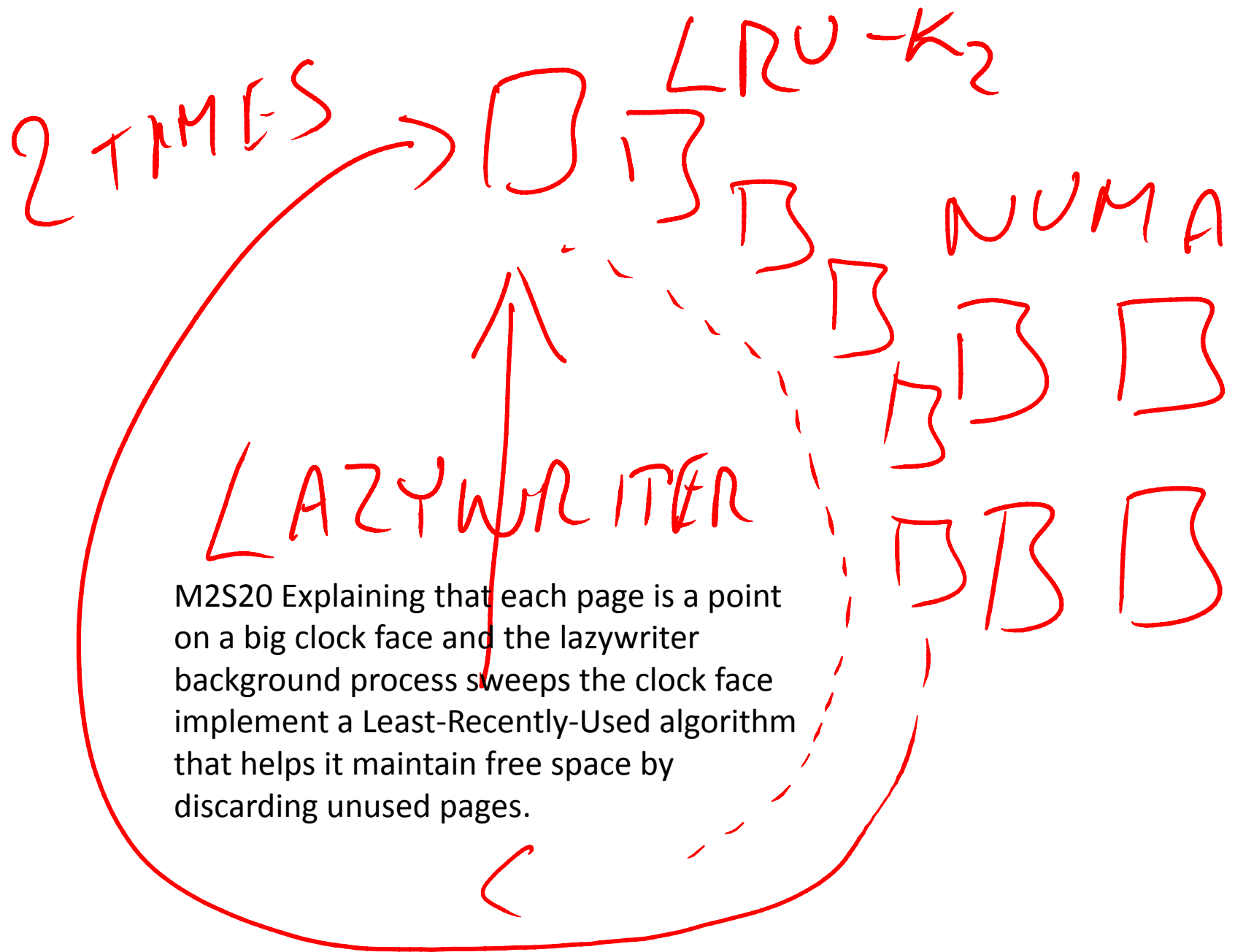


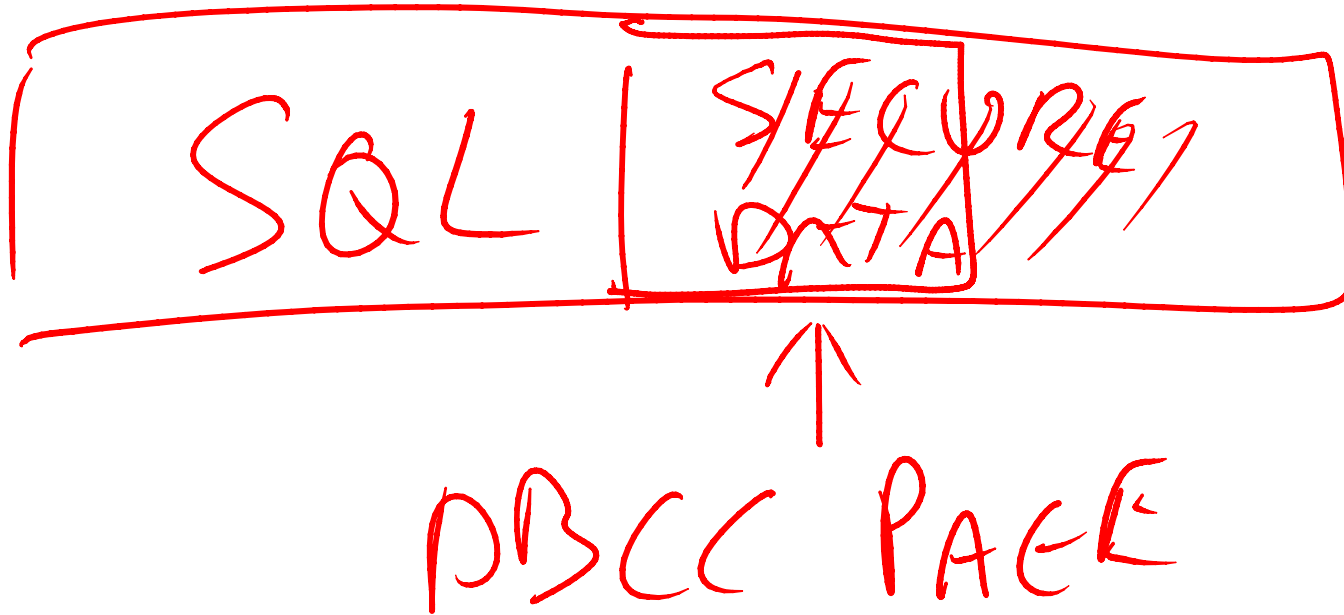
~~8100~~ ←
SO (8099)



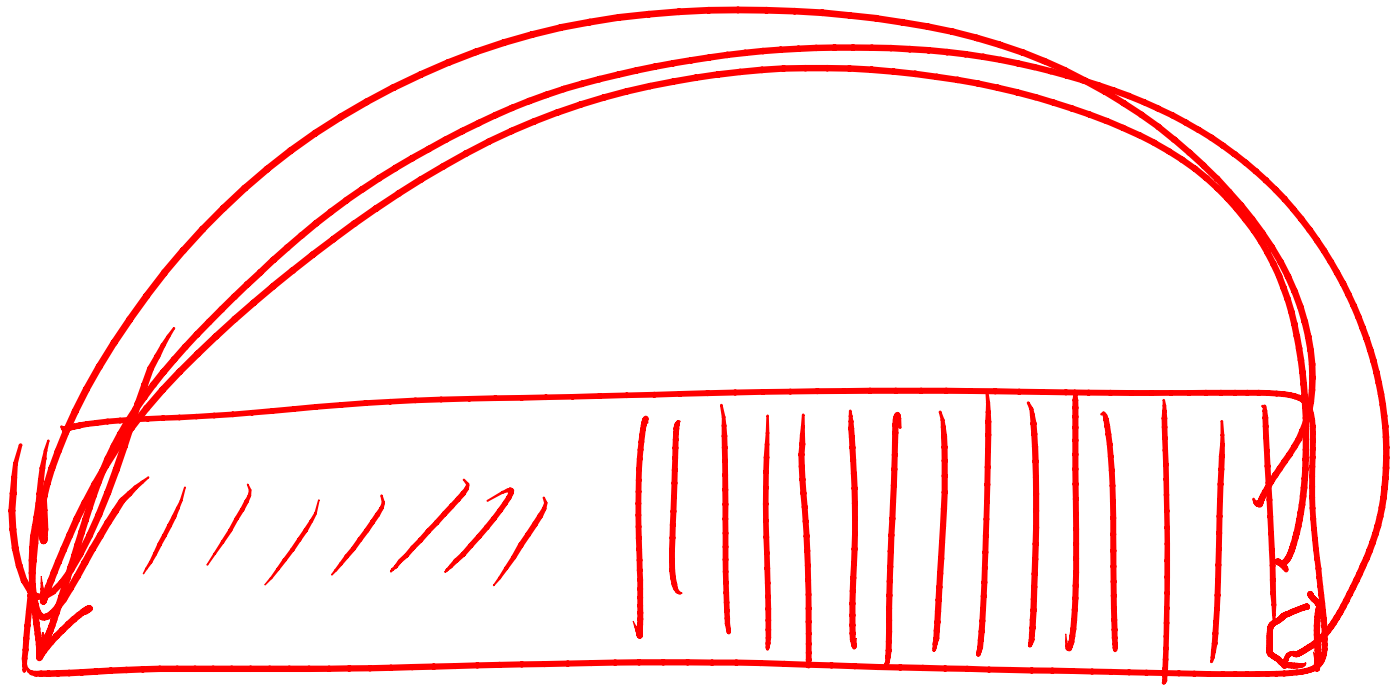
1 ← ✓
8100

M2S19 Explaining round-robin
allocation and proportional fill





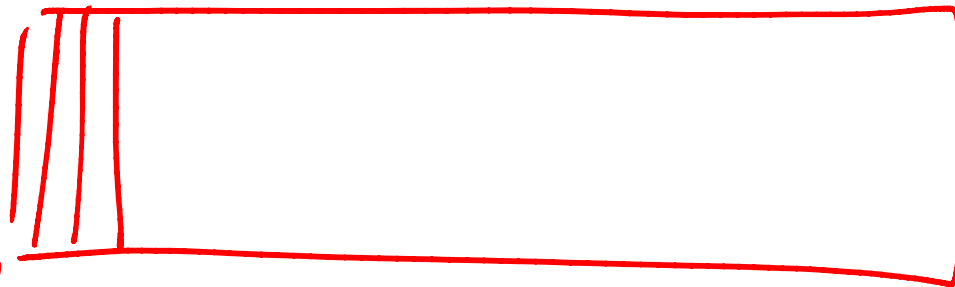
M2S24 Showing the security risk with instant file initialization



M2S28 Showing how shrink works. It grabs the page highest allocated in the file and moves it as low down as it can. And so on. Reversing the order of indexes and producing fragmentation.

CAM

T1118



PFS

2:1:1

~~SGAM~~

2:1:3

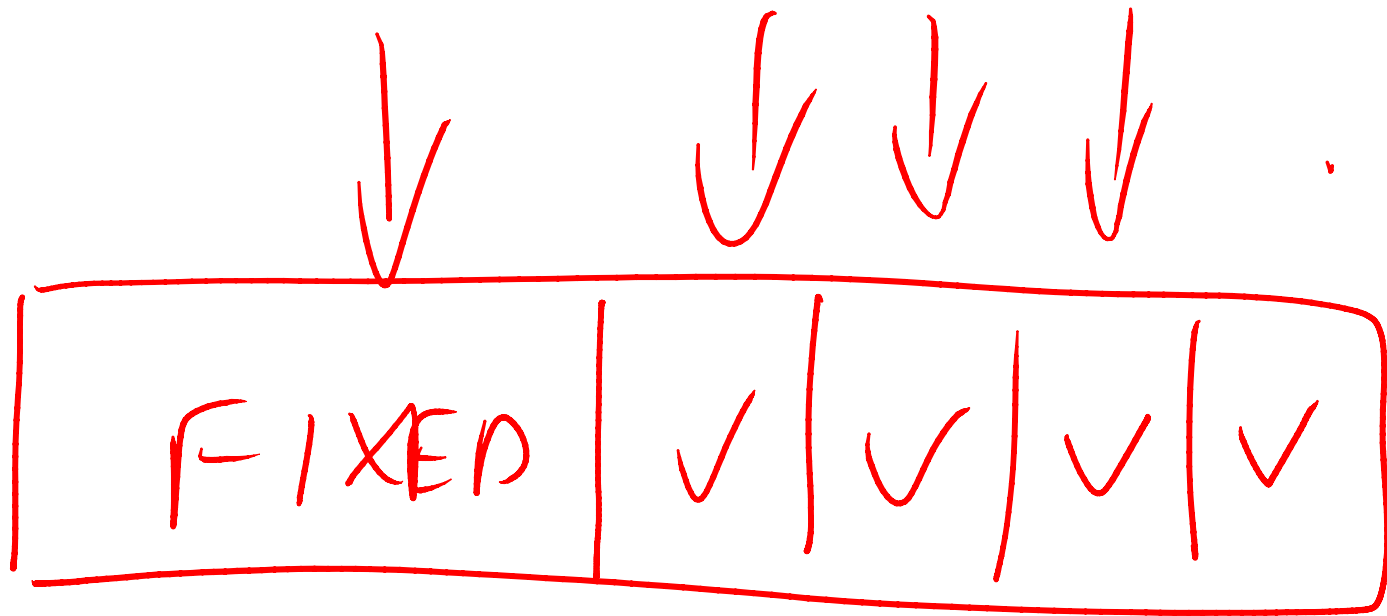
GAM

LATCHES

PAGE LATCH LEX

2:1:103

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

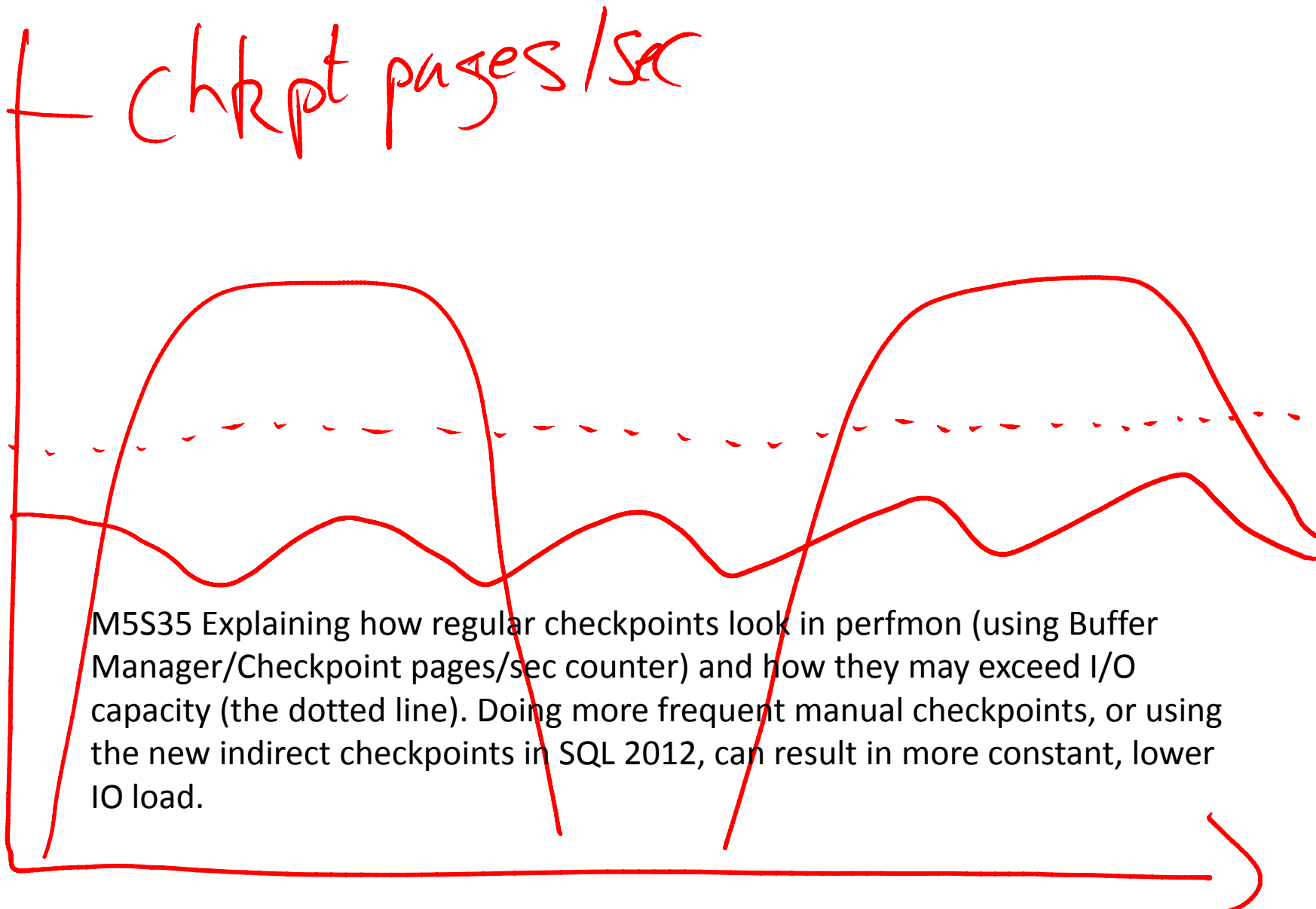


LOP_MODIFY_ROW

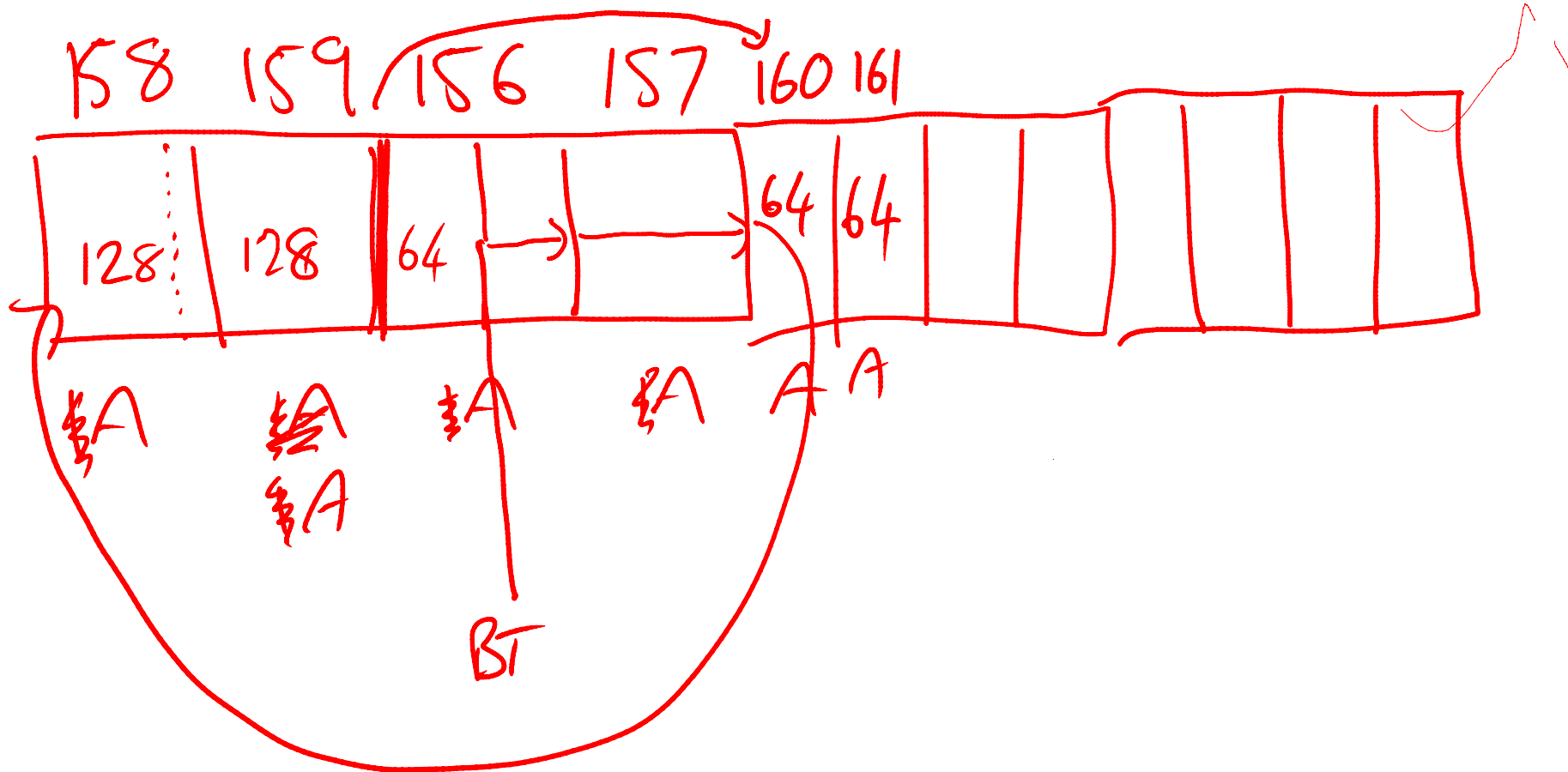
LOP_MODIFY_COLUMN

M5S19 Explaining how the Access Methods determines how to log UPDATES efficiently by splitting the logging by portion of the record. Each variable length column is a portion, as is the fixed-width portion (up to 16 bytes in each log record)

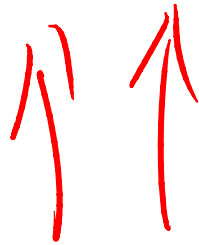
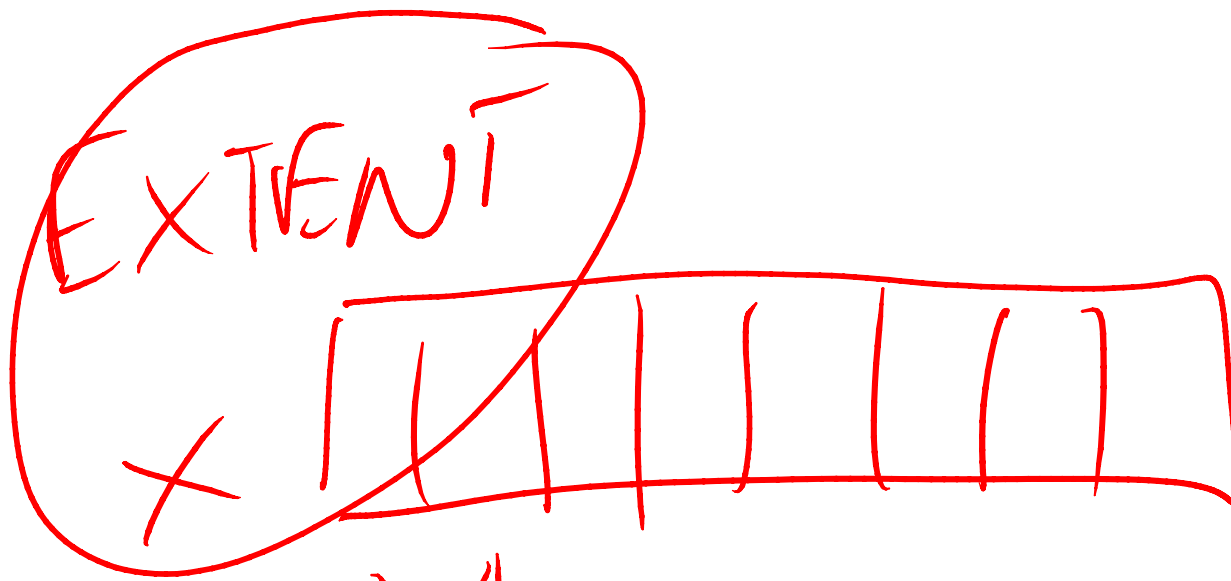
chkpt pages/sec



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.



M5S48 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.



PROBE

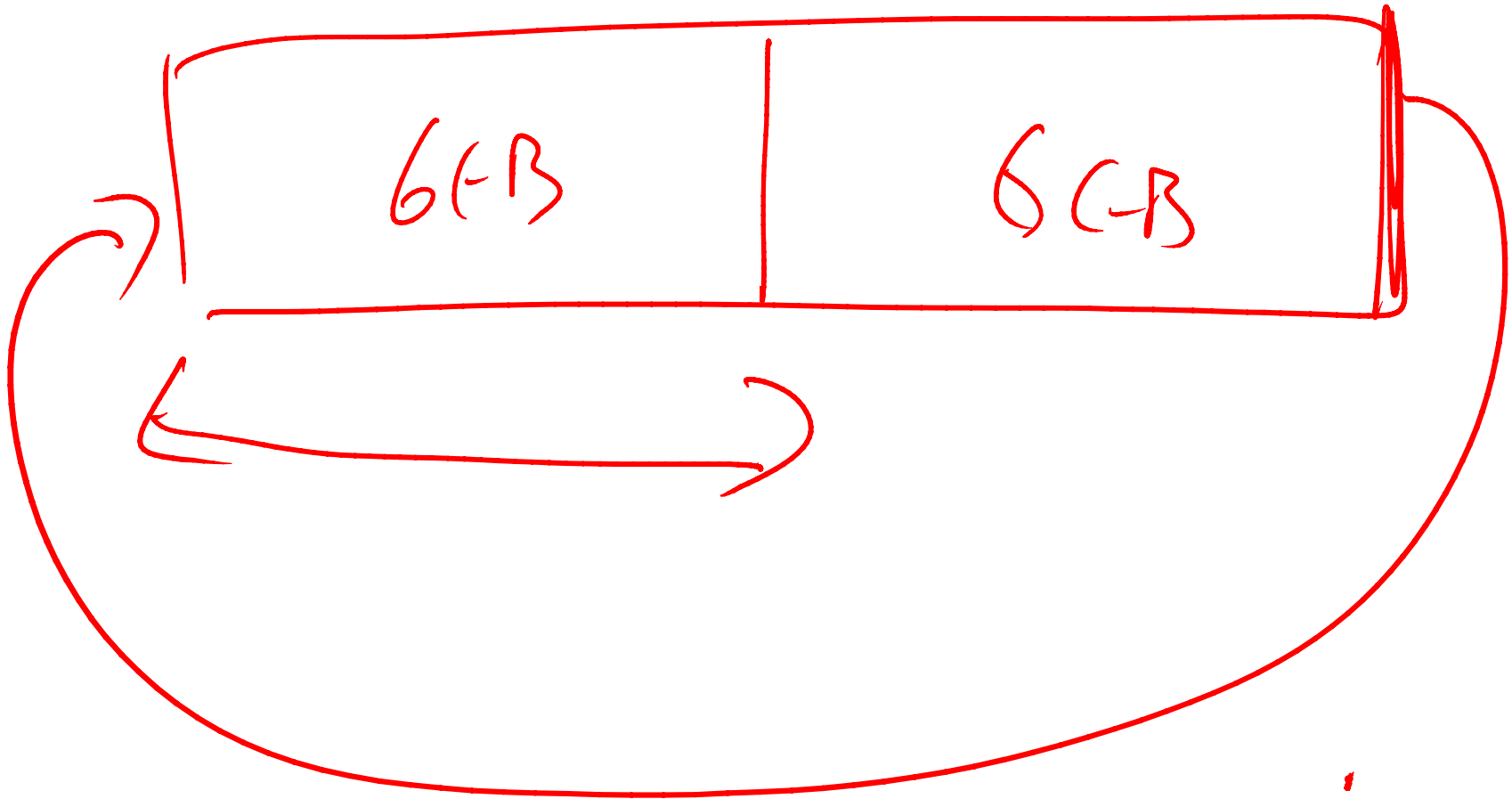
M5 Explaining part of the deferred-drop functionality. Before an extent can be deallocated, an X lock is acquired on it and all 8 page locks are probed (instant acquire X lock and release) to make sure nothing else has them locked.

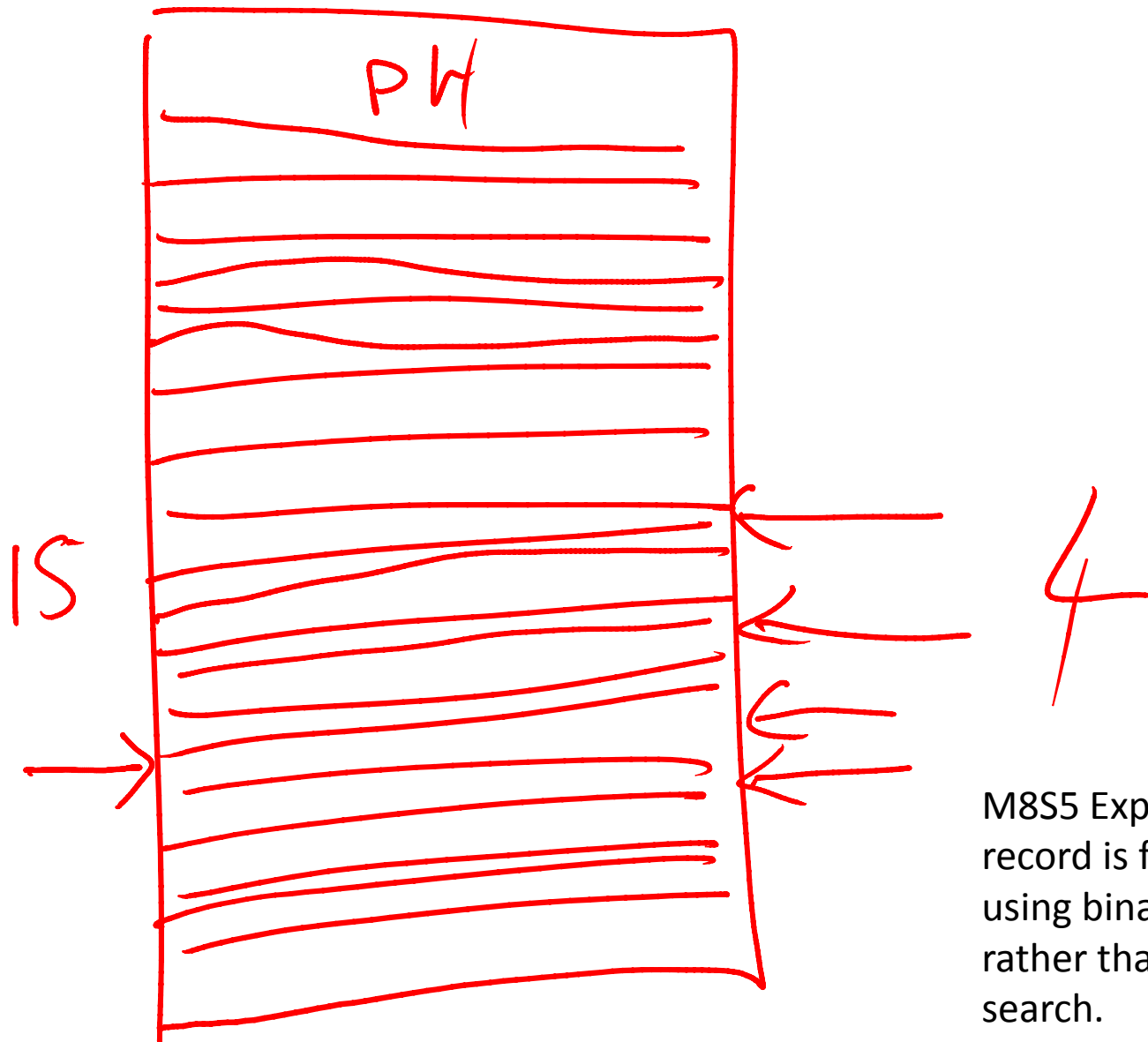


LSN LSN
VLF

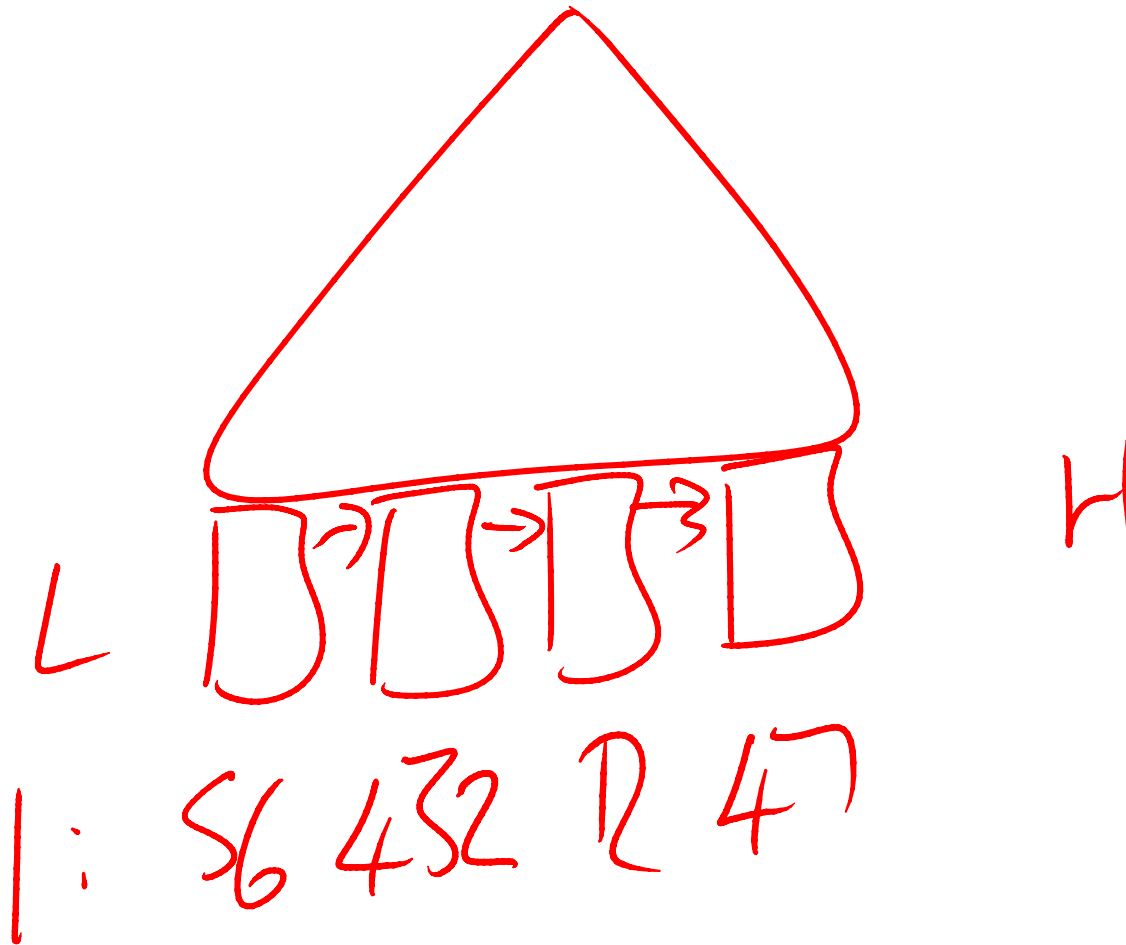
M5S25 VLF fragmentation is bad because of having to search through the VLFs for a particular VLF sequence number. See the KB article link for more info.

M5 How VLFs that are too big can be an issue. Imagine creating a 100GB log – with 16 x 6.125GB VLFs. Now shrink the log – it will have a minimum of two VLFs, each 6.125GB. All 6.125GB of VLF #1 has to be cleared before the log can wrap around without having to auto-grow.

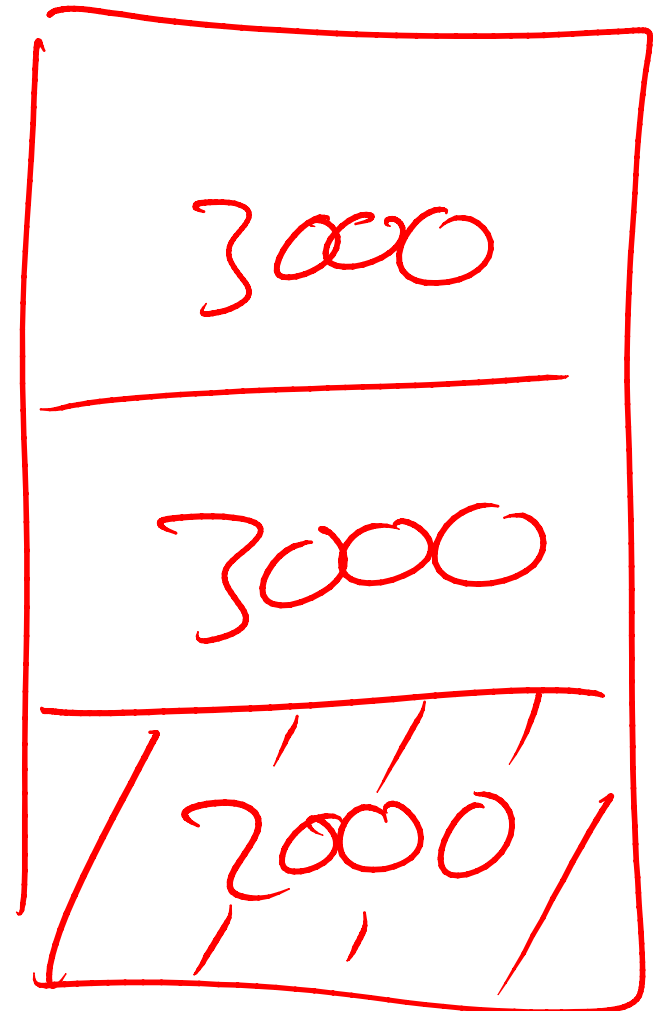
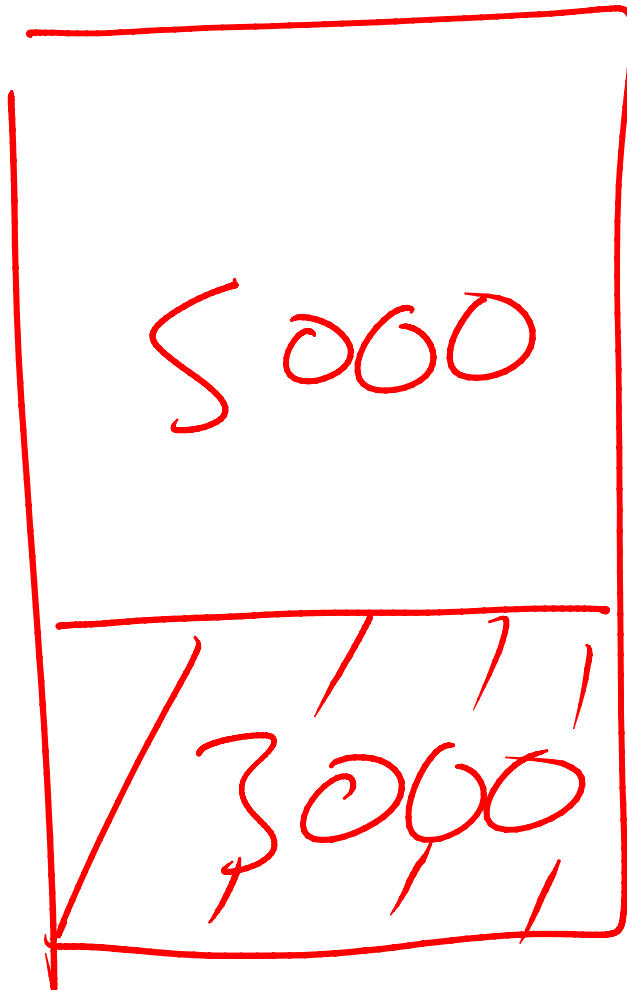




M8S5 Explaining how a record is found on a page using binary search rather than brute force search.

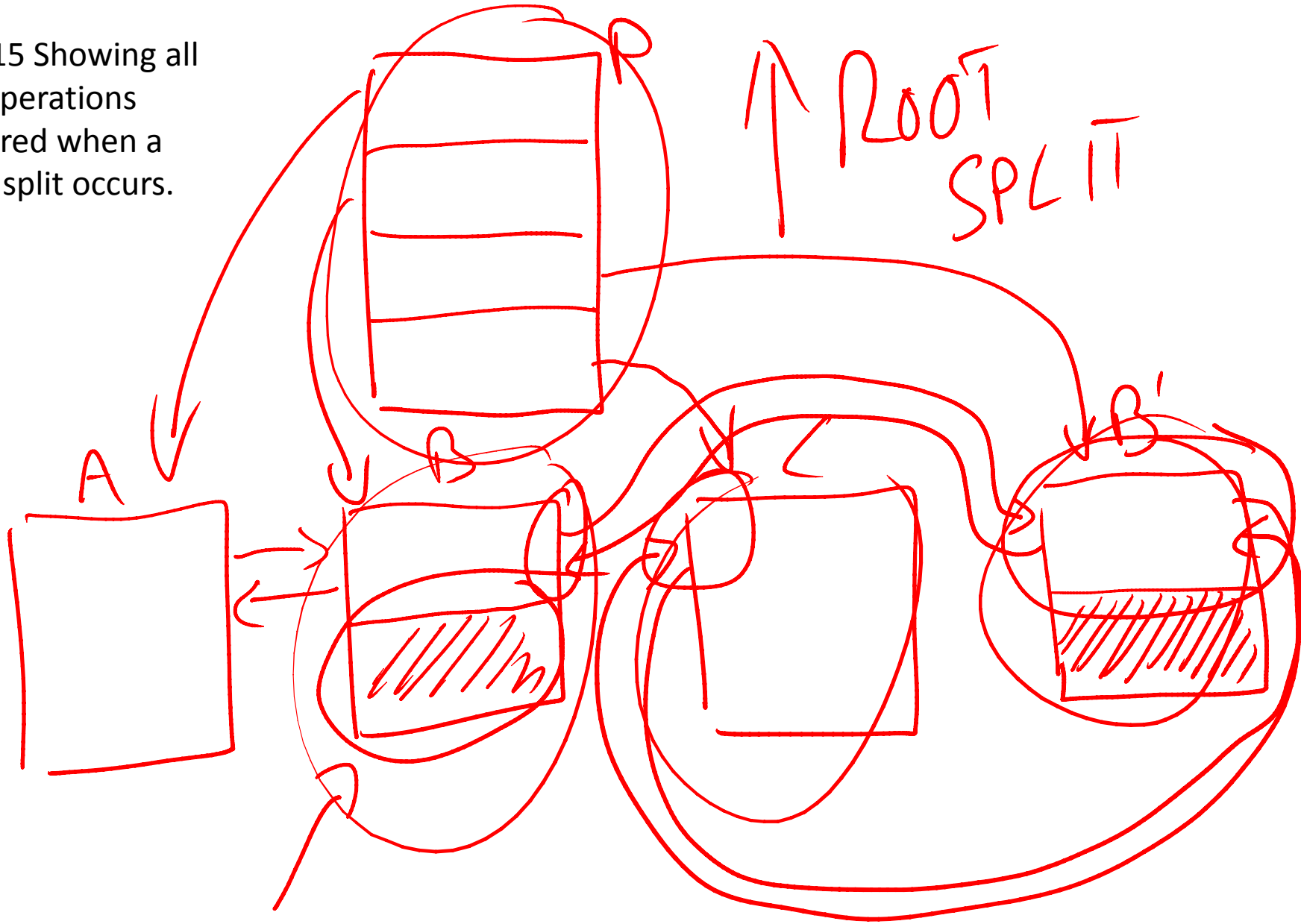


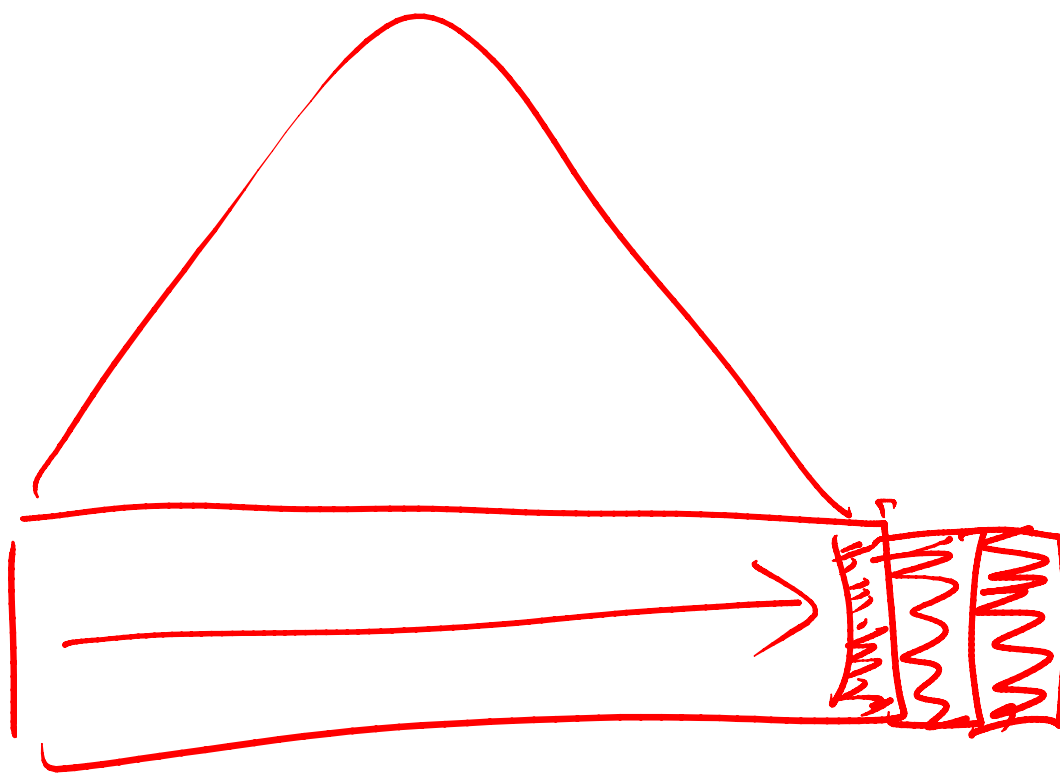
M8S9 Showing that pages in the leaf level are always logically contiguous (linked together in key order) but may not be physically contiguous.



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.





LOP_DELETE_SPLIT

M8S18 Append-only insert pattern fills pages on right-hand side of index. New page allocations in this case are called page splits too – making all methods of tracking page splits largely useless (until 2012 with Xevents). You can look for LOP_DELETE_SPLIT log records to see splits occurring.

M8S22 The numbers from the page split logging cost demo.

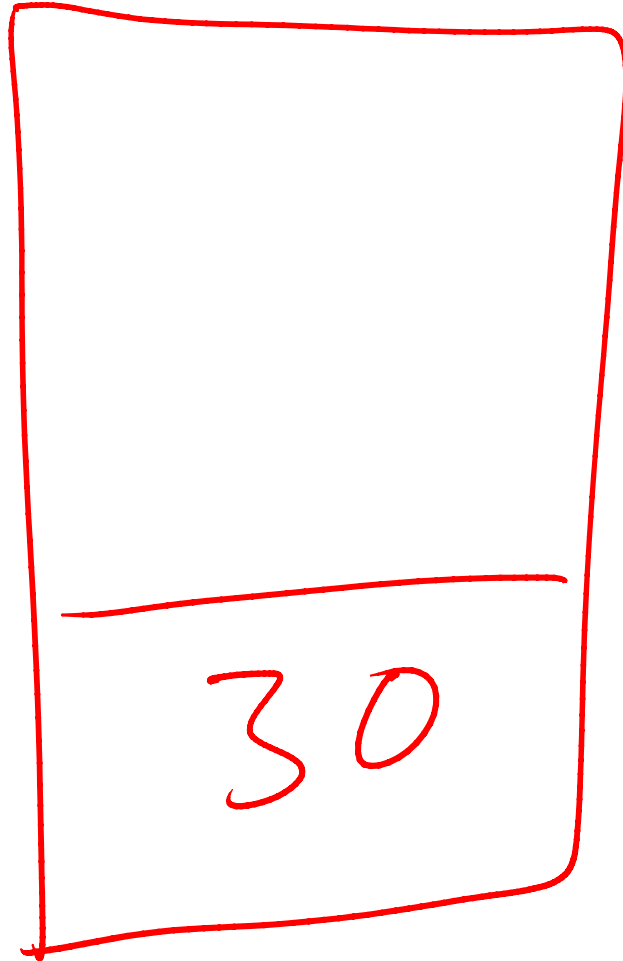
	NO SPLIT	SPLIT	
1000	1244	6772	<u>x5</u>
100	344	5964	<u><u>x18</u></u>
10	256	10364	<u><u>x45</u></u>

COMMENTS

MEMBER_ID

PAUL	RYAN	K	DAW
------	------	---	-----

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



70

M8S25 Configuring a
FILLFACTOR of 70 means
leaving 30% free space on leaf-
level pages when an index is
built or rebuilt.