

80GB total

14 = OS

→ 1
—

~~64~~

~~21~~

2
—

~~64~~

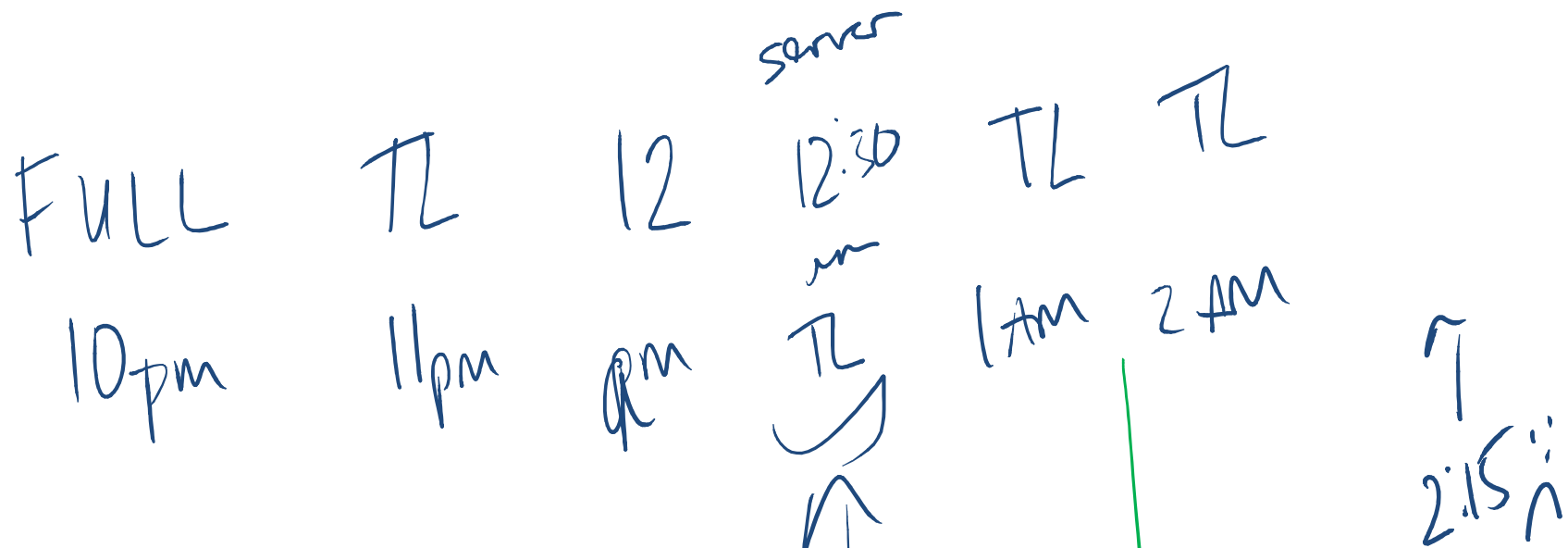
21
—

3
—

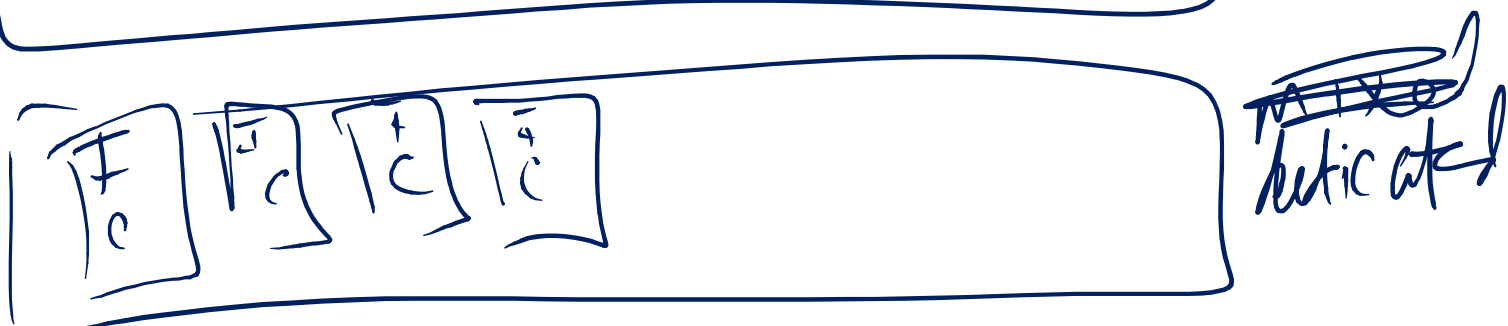
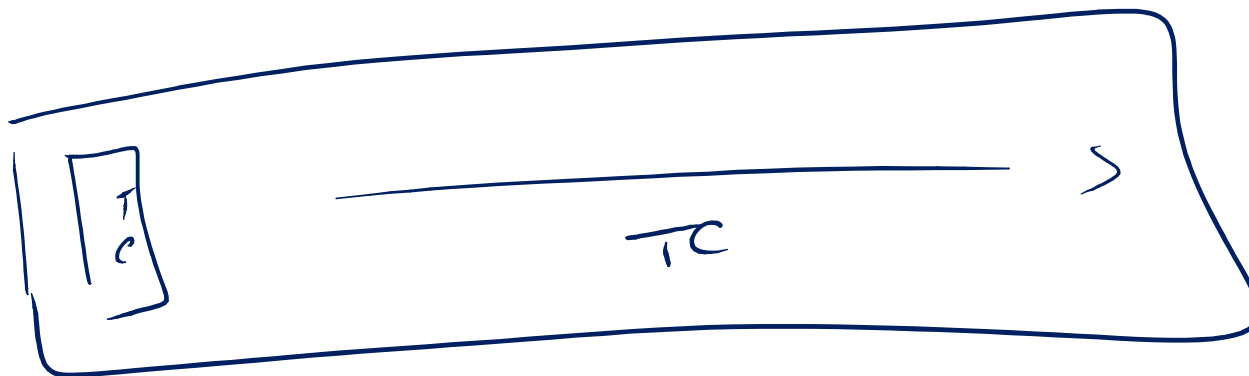
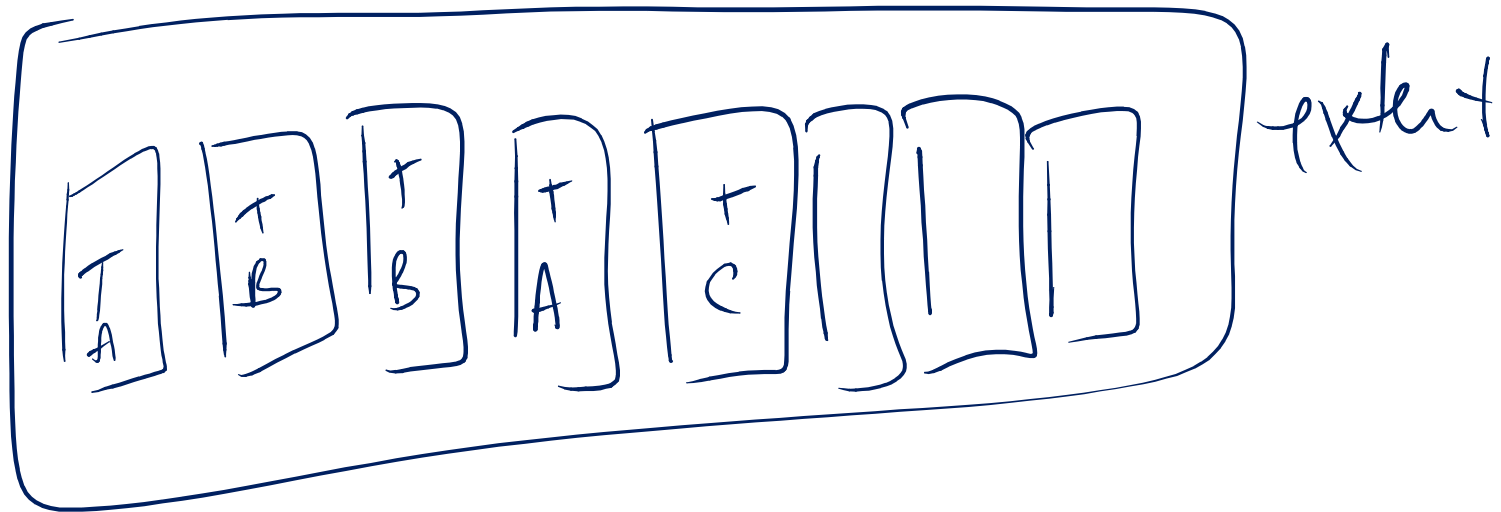
~~64~~

21
—

Discussion of a server with 80GB of memory and 3 different instances installed – what would you set for max server memory or each instance? You would not set all of them to total MORE than memory available on the server. For example, if I set each one to 64 GB, then that's 192GB – way more memory than is on the server. I could divide it evenly between the three, such as 21GB each, leaving 16GB free for the OS. But if I have an instance that isn't very active (instance 1 for example), then I would allocate less memory to it, maybe only 8 GB, and then divide the remaining memory between the other two instances as appropriate. Understand that the instances do not have to have the same max memory values, but you don't want the TOTAL of the three max memory values to exceed what exists on the server, and you want to leave enough for the OS.



Full and transaction log backup schedule, where full backups are taken at 10PM nightly, and tlog backups are taken every hour. If the server team is taking a backup at the server level that also takes a transaction log backup for some reason, that backup needs to be COPY_ONLY, otherwise it is part of the log chain and would be required for a restore. If you had a failure at 2:15AM, and you did not have the 12:30 AM tlog backup from the server, then you could only restore the last tlog that's available. If the tlog is not COPY_ONLY, then you could take a differential backup at something like 1:30 AM, and then the restore sequence would be 10:00PM full, 1:30AM diff, 2:00AM tlog.



Depiction of an extent (which is eight, 8K pages) and how it can be a mixed extent (hold pages which have rows from more than object, like an index or table) or a dedicated extent (holds pages that have rows from one object only).

B-tree index

Good reference (even though they're for 2008R2, still applicable) for clustered indexes:

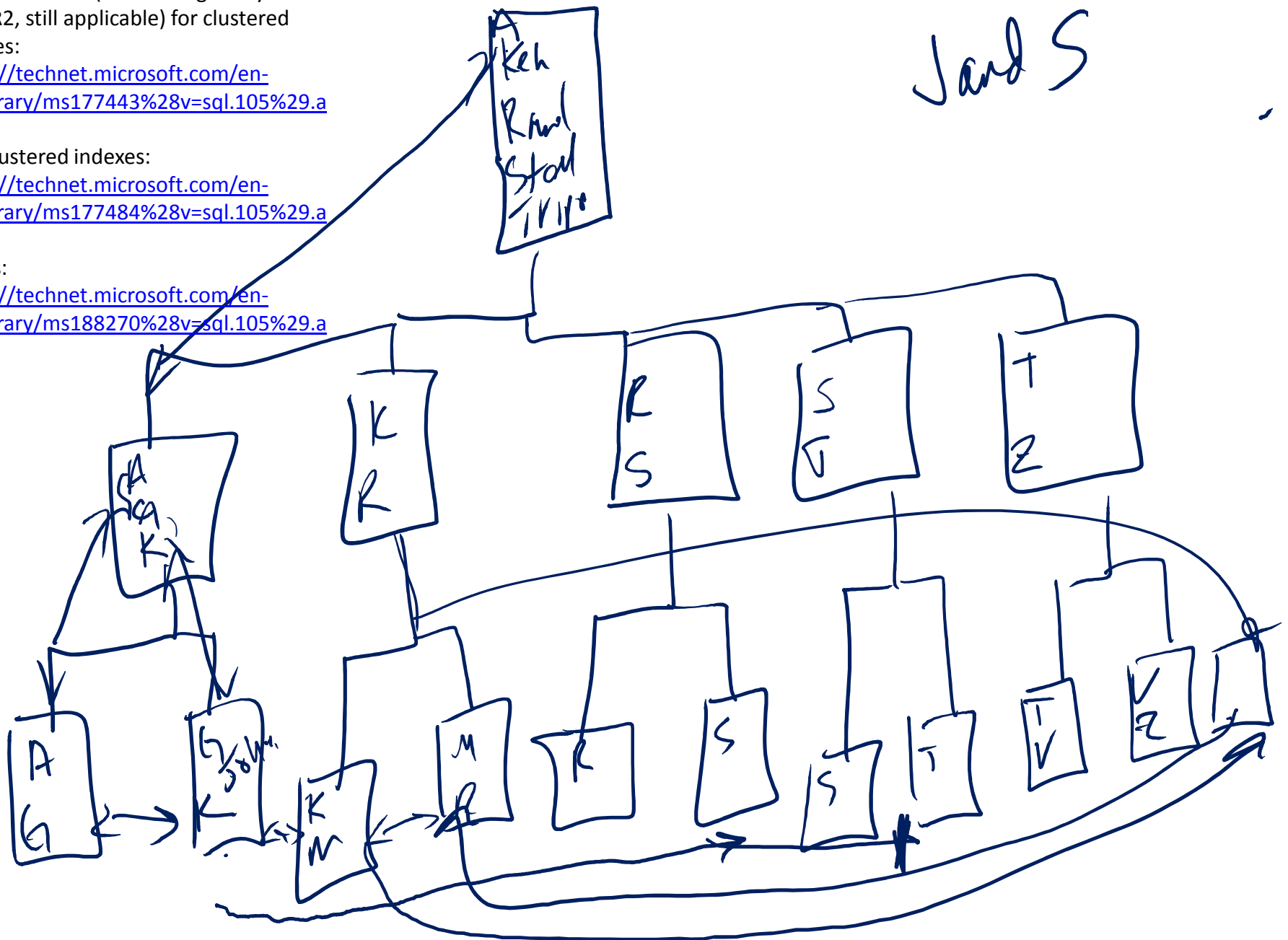
<https://technet.microsoft.com/en-us/library/ms177443%28v=sql.105%29.aspx>

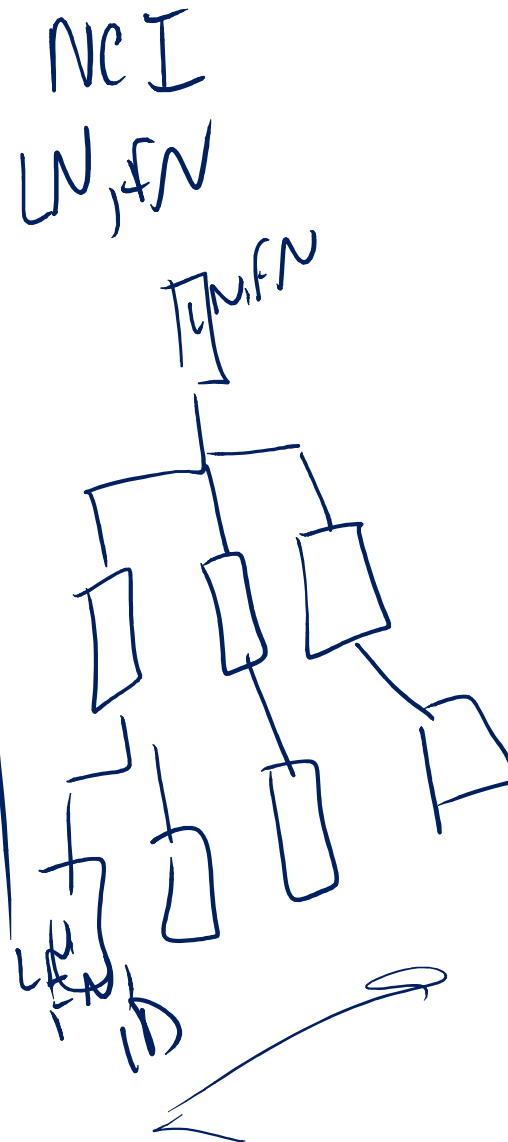
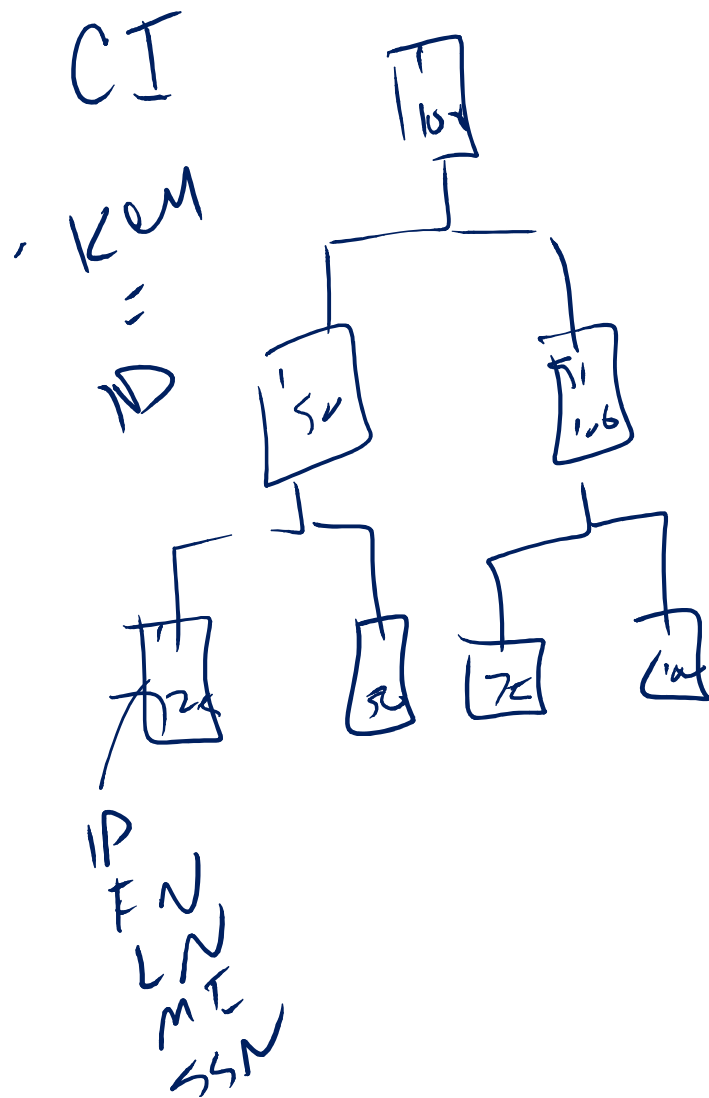
Nonclustered indexes:

<https://technet.microsoft.com/en-us/library/ms177484%28v=sql.105%29.aspx>

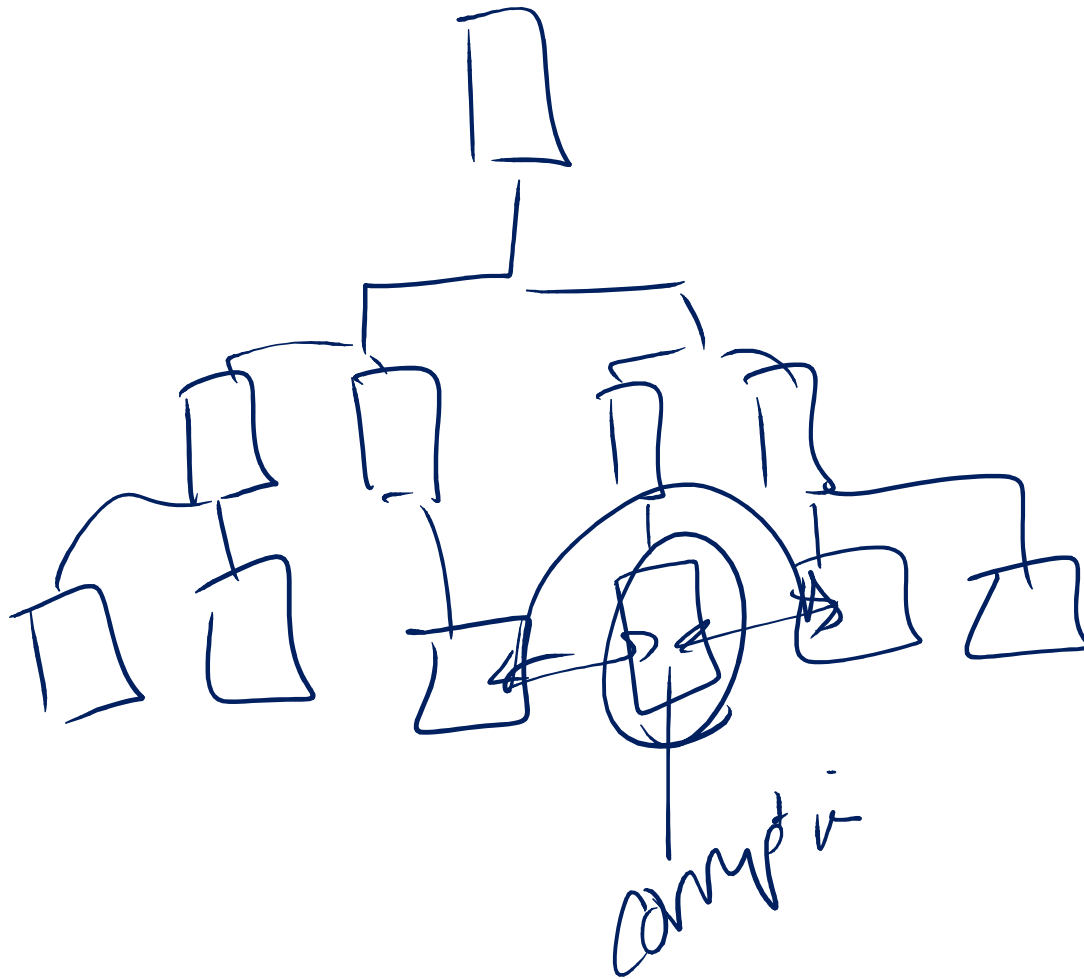
Heaps:

<https://technet.microsoft.com/en-us/library/ms188270%28v=sql.105%29.aspx>





Discussion of clustered index with its clustering key, and how that exists in the leaf level of the nonclustered index (so the NCI has a pointer back to the row in the CI). If there is no CI, but only a heap, then a row ID (RID) is used instead of the clustering key. Note that if the clustering key is not unique, SQL Server adds a unique ID to it to ensure it is unique.



Depiction of corruption in an index, in a leaf level page. Note that in this case, there are least three other index pages that point to the corrupted page, and these are referenced in the CHECKDB output. A database is really just a bunch of pages that point to each other.