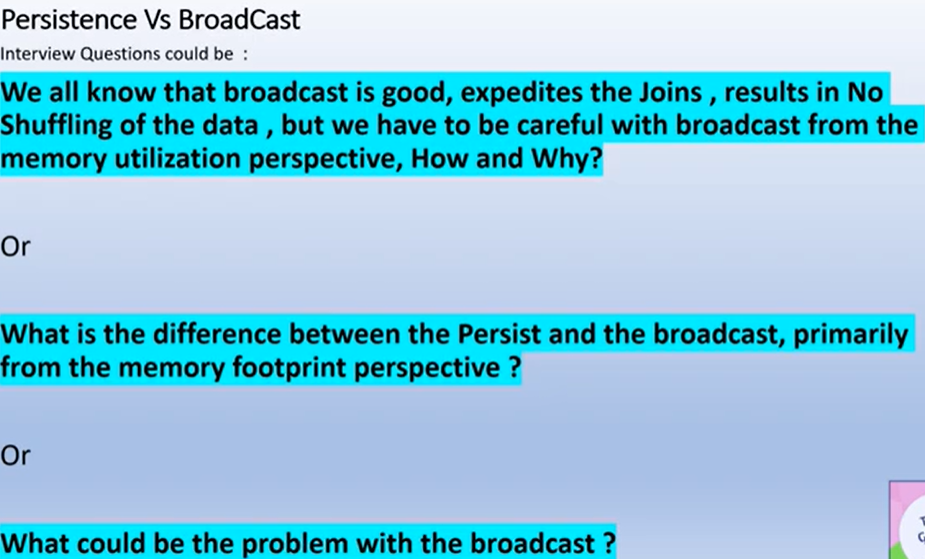
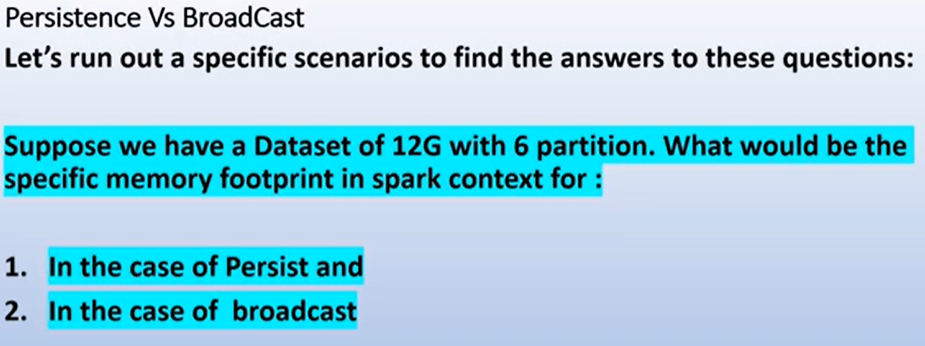
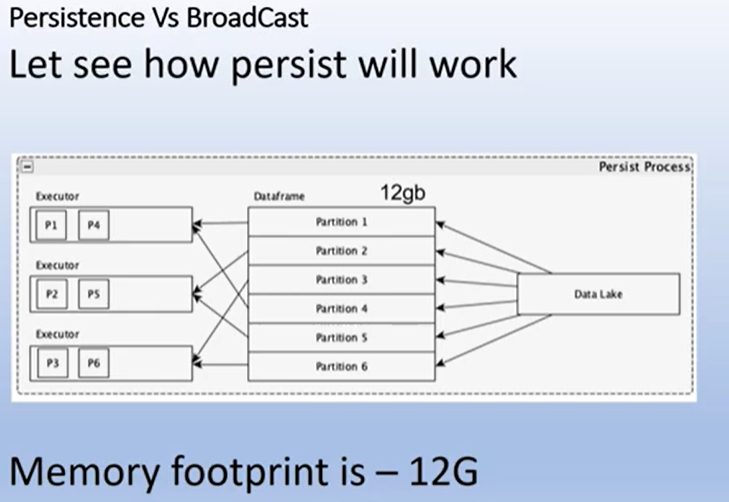
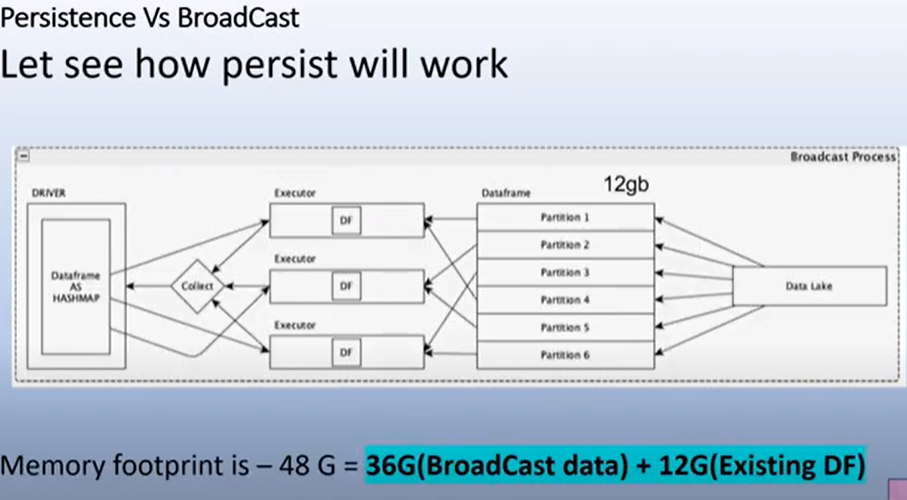
**Persistent and broadcast**







In the above case, the memory footprint in 12 g, we 12 gb of dataframe which is partitioned in 6 and now it is is in the storage memory and when we do persist then it goes to worker memory of the executor but my memory footprint on the executor cache will remain 12 gb itself, as there is no increment and decrreement of memory footprint.



In the case of broadcast, taking same 12 gb of dataframe and now we broadcast this dataframe. In broadcast there would be one copy of parttions or this dataframe on all the executor, so we have three executor, each exector have two partitions that needs to be executor cache to worked on.

So upto the exector the memory footprint is 12 gb only

Now in the case of broadcast, travel will called the collect, these partitions from the different executor get it into the driver node and then try to create hash map or broadcast object for this dataset and then publish them on the three exectors . so this 12 gb of dataset which is collected from different executors or collected from this datasetwould now be converted into 3 12 gb hashmap or broadcast object so it makes it 36 gb and 12 gb also that is present in executor cache for processing that partitions may not be garbage collected as of now so total memory footsprint on spark context when you call broadcast is 48 gb