Q. Explain the different join strategies in Spark and which strategy you will be adopting when joining Parquet File 1 and 2 if you are implementing the code in Spark Dataframe. Your answer can be saved as joins.pdf under the main repository.

Spark splits both data into different executors which will keep a portion of the full dataset separately. However, if the joining datasets are not allocated within the clusters correctly, the join operation cannot be performed. Thus, spark will perform a shuffle operation.

It will map or hash the keys of both dataset and shuffle them into the same cluster. The join operation can then be performed in parallel within the node. This shuffle function can take up a lot of computing resources especially if the dataset is large.

In sort merge, both data sets are shuffled and sorted. Sort enables the keys to be sorted In order such that during the join phase, it only needs to search through the dataset until it is no longer the same. This will be useful if both data sets are large.

In broadcast join, there is no shuffle step. The smaller dataset is broadcasted to all the executors. Each node will then have access to the reference data locally. The data in the larger dataset is then hashed and joined. The default size value for smaller dataset is 10mb, but can be adjusted to up to 8gb. However, it could result in out of memory error if the broadcast dataset is too large.

With File 2 being a reference table, a broadcast hash join would make the most sense since dataset B is small. It will also reduce the time complexity of the join operation since it skips the shuffle step entirely.