Homework April 26, Hint:

Here is an example of what is **not** a solution.

We first calculate the projection $\pi_{A,B,C}(R(A,B,C,D,E)$ and store the result in a temporary table. To do so, we read all 100,000 blocks in tranches of 10000 blocks. We then read every block tuple by tuple, and write out the A, B, C attributes as a new tuple in the one remaining free block. Whenever that block is full, we write it to disk (HDD or SSD). When we are done, we have generated a new temporary table with 60,000 blocks. This gives 100000 + 60000 block movements.

We now do the select in a similar manner creating also a temporary table. We download table S in tranches of 10000 blocks. We then go through each block, and if the tuple conforms to the selection criteria (i.e. if F = 3), then we write the tuple in the one remaining block in Main Memory (MM). Whenever this block is full, we write it back to disk. Since we are writing about 20,000 blocks (10% of 200000) and read 200000 blocks, we move 220000 blocks.

Now we do the join. The temporary table $\sigma_{F=3}(S)$ has 20000 blocks and the temporary table $\pi_{A,B,C}(R)$ fills 60000 blocks. We bring in the first 5000 blocks of $\pi_{A,B,C}(R)$ and the first 5000 blocks of $\sigma_{F=3}(S)$ and start calculating the join, placing the resulting tuple in the free block. When that blocks fills up, we write it disk. The join is calculated by having a cursor to the a tuple in $\pi_{A,B,C}(R)$ and another cursor to a tuple in $\sigma_{F=3}(S)$. The tuples under the cursor are compared for their C-attribute value if they are the same, they are written into the result block. When we are done, we leave the block from $\pi_{A,B,C}(R)$ in main memory, but bring in the next set of blocks from $\sigma_{F=3}(S)$. We reset the first cursor to the beginning of the current $\pi_{A,B,C}(R)$ block and set the other cursor to the beginning of the new set of blocks from $\sigma_{F=3}(S)$. When we are done with this, we bring in the third block of $\sigma_{F=3}(S)$. Once we are done, we bring in the fourth set of blocks of $\sigma_{F=3}(S)$. When we are done, we leave the one from $\sigma_{F=3}(S)$ in memory and bring in a new tranche of blocks from $\pi_{A,B,C}(R)$.

This way, for each of the 12 tranches of 5000 blocks each from $\pi_{A,B,C}(R)$ gets matched against all 4 tranches of 5000 blocks each from $\sigma_{F=3}(S)$. For the first tranch from $\pi_{A,B,C}(R)$, we bring in 4 tranches from $\sigma_{F=3}(S)$, but then by going up and down we only need to bring in three tranches. This means, we are moving $12 \cdot (5000 + 15000) + 5000 = 65000$ blocks into MM for the join. We also move the result of the join into storage, but we do not know how much this is. The total is 220000 + 160000 + 65000 blocks of movement.

Why is this not a good solution? First, the join is done in a very inefficient manner. A hash join should be better. Second, the projection and the select can be integrated with the calculation of the hash tables.