DSM: Tutorial 6

Question 1

Ideally, T3 could be divided into two nested transactions: one for account 123 and 789, and one for all other accounts. It would be extremely difficult to track which accounts are currently being used, so an alternative would be to guess which accounts are likely to be active (using, for example, account holder time-zones).

Question 2

Multiple nested transactions could be used to operate on blocks of the data. These blocks may be chosen to match data fragmentation, to divide by user ID, to divide by first initial, etc.

Alternatively, one nested transaction could modify first names and another could modify last names. However, this doubles the amount of disk operations (two per record).

Question 3

If a sequence of actions S is serially equivalent to both 'transaction A then transaction B' and 'transaction B then transaction A' then transactions A and B are independent of each other and can be executed in parallel. This could be because they read/write to different parts of the data, or because they are both read operations.

Question 4

$S1 : r5(x), r4(x), r4(y), w5(x), w4(y), r5(y)$

$S2 : r5(x), w5(x), r4(x), r5(y), r4(y), w4(y)$

$S1$ and $S2$ are not view equivalent for the following reasons:

- In $S1$, transaction 5 reads y **after** transaction 4 writes to it, but in $S2$ transaction 5 reads y **before** transaction 4 writes to it. This will create an inconsistent retrieval.

- In $S1$, transaction 4 reads x **before** transaction 5 writes to it, but in $S2$ transaction 4 reads x **after** transaction 5 writes to it. This will create an inconsistent retrieval.

And ordering that is view equivalent to $S1$ must maintain the following:

- $r4(x)$ before $w5(x)$ (read initial value)

- $r5(x)$ before $w5(x)$ (read before write, read initial value)

- $r4(y)$ before $w4(y)$ (read before write, read initial value)

- $r5(y)$ after $w4(y)$ (read result of write)

The following is one possible sequence:

$r4(y), r4(x), w4(y), r5(y), r5(x), w5(x)$