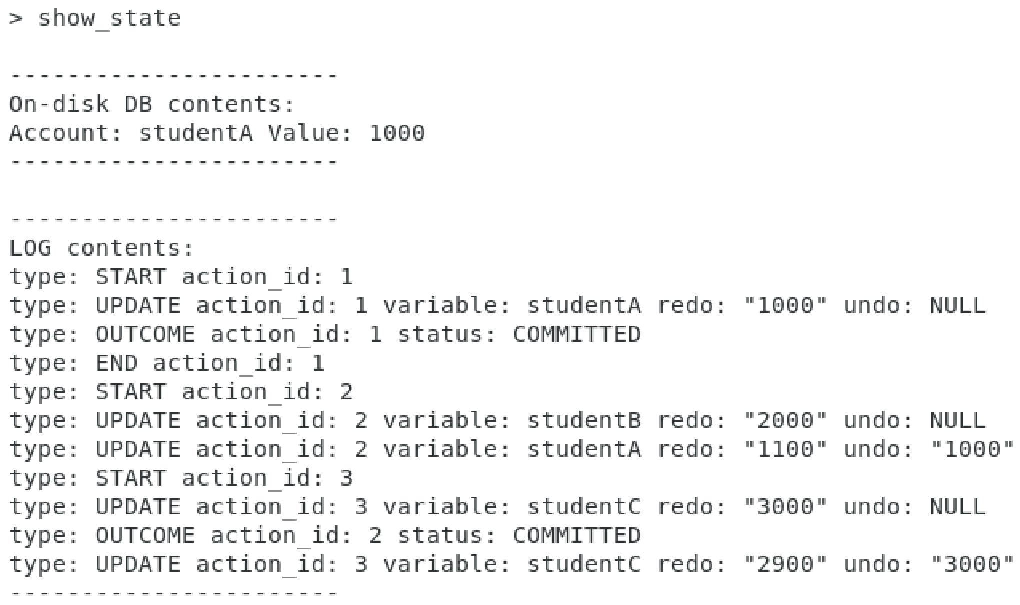
Hands-on 1

Question 1: Wal-sys displays the current state of the database contents after you type `show_state`. What do you observe in on-disk DB contents? Why doesn't the database show studentC?

Answer 1: After type `show_state` we can see this. We can only see the record "studentA 1000". studentC is not shown in database because the operations about studentC including `create_account` and `debit_account` all come without any `end` after it, which installs the results of recoverable action into the file "DB".



```
> show_state

-----------------------
On-disk DB contents:
Account: studentA Value: 1000
-----------------------

-----------------------
LOG contents:
type: START action_id: 1
type: UPDATE action_id: 1 variable: studentA redo: "1000" undo: NULL
type: OUTCOME action_id: 1 status: COMMITTED
type: END action_id: 1
type: START action_id: 2
type: UPDATE action_id: 2 variable: studentB redo: "2000" undo: NULL
type: UPDATE action_id: 2 variable: studentA redo: "1100" undo: "1000"
type: START action_id: 3
type: UPDATE action_id: 3 variable: studentC redo: "3000" undo: NULL
type: OUTCOME action_id: 2 status: COMMITTED
type: UPDATE action_id: 3 variable: studentC redo: "2900" undo: "3000"
-----------------------
```

Question 2: Which accounts should exist, and what values should they contain when the database recovers?

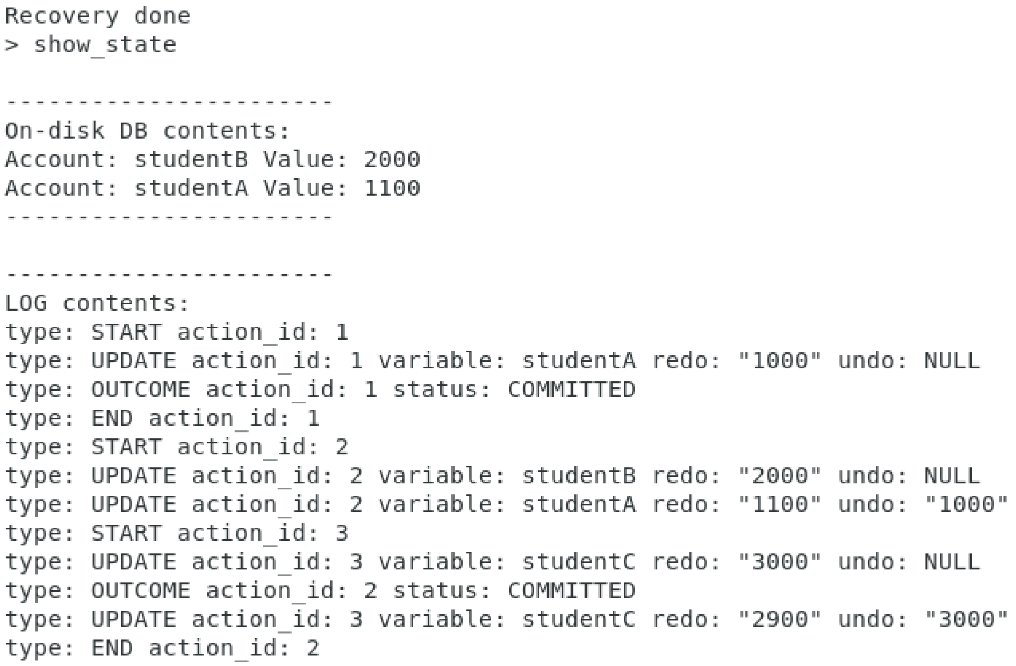
Answer 2: After recovery, studentA exists with a value of 1100, and studentB exists with a value of 2000. That's because transaction 1 is done, 2 is committed and 3 is uncommitted. Thus there is nothing to be done for transaction 1 , and 2 will be redone while 3 will be undone, which looks like only transaction 1 and 2 are done.

Question 3: Can you explain why the "DB" file does not contain a record for studentB and contains the balance for studentA before `credit_account` is issued?

Answer 3: That is because only having executed `end` , can the file "DB" update the changes. Before `credit_account` , only first three operations will be done. In that case, "studentA 1000" will exist but "studentB 2000" won't.

Question 4: What do you expect the state of "DB" to be after wal-sys recovers? Why?

Answer 4: It should be like "studentB 2000""studentA 1100". Because transaction 1 is done, 2 is committed and 3 is uncommitted. Thus there is nothing to be done for transaction 1 , and 2 will be redone while 3 will be undone, which looks like only transaction 1 and 2 are done.



```
Recovery done
> show_state

-----------------------
On-disk DB contents:
Account: studentB Value: 2000
Account: studentA Value: 1100
-----------------------

-----------------------
LOG contents:
type: START action_id: 1
type: UPDATE action_id: 1 variable: studentA redo: "1000" undo: NULL
type: OUTCOME action_id: 1 status: COMMITTED
type: END action_id: 1
type: START action_id: 2
type: UPDATE action_id: 2 variable: studentB redo: "2000" undo: NULL
type: UPDATE action_id: 2 variable: studentA redo: "1100" undo: "1000"
type: START action_id: 3
type: UPDATE action_id: 3 variable: studentC redo: "3000" undo: NULL
type: OUTCOME action_id: 2 status: COMMITTED
type: UPDATE action_id: 3 variable: studentC redo: "2900" undo: "3000"
type: END action_id: 2
```

Question 5: If you issue another wal-sys command to recover the database again, what would the "DB" file contain after the second recovery? Why?

Answer 5: It should still be "studentB 2000""studentA 1100". That's because transaction 1 and 2 are both done during the first recovery, and there is nothing to be done in this recovery. Transaction 3 is uncommitted, which means it can't be recovered.

Question 6: During recovery, wal-sys reports the `action_ids` of those recoverable actions that are "Losers", "Winners", and "Done". What is the difference between these categories?

Answer 6:

- **Losers:** since this transaction is uncommitted, actions in it would be undo and can't be recovered.
- **Winners:** this transaction is committed, actions in it would be redo and be recorded in database during recovery.
- **Done:** actions in this transaction have recorded in database, so there is no need to redo these in the recovery.