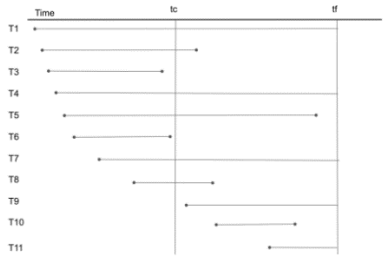
FIT9132

Assignment 2B

Name: Sachin Shivaramaiah

Student ID: 34194037




In the below representation:

- All green-shaded cells indicate that the transactions were finished and committed after the checkpoint,
- While white-shaded cells indicate that they were not completed at the time of failure.
- The cells with yellow coloring can be ignored because they were committed before the check point; the data will be written to the database at the check point.

Transaction log:

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 |

<u>The 3-stage process to recover are as follows:</u>

Stage 1: Compile REDO & UNDO LIST using log: The transaction logs are used to create REDO and UNDO lists, which determine which transactions need to be redone or rolled back

| REDO LIST | | | | UNDO LIST | | | |
|---|---|---|---|---|---|---|---|
| T2 | T5 | T8 | T10 | T1 | T4 | T7 | T11 |

Stage 2: Transactions in the UNDO list that were aborted before reaching their commit point must be rolled back using their prior images to restore the database to a consistent state.

| UNDO LIST | | | |
|---|---|---|---|
| T11 | T7 | T4 | T1 |

Stage 3: All committed transactions are properly recorded in the database, transactions from the REDO list will be reapplied, beginning with the oldest.

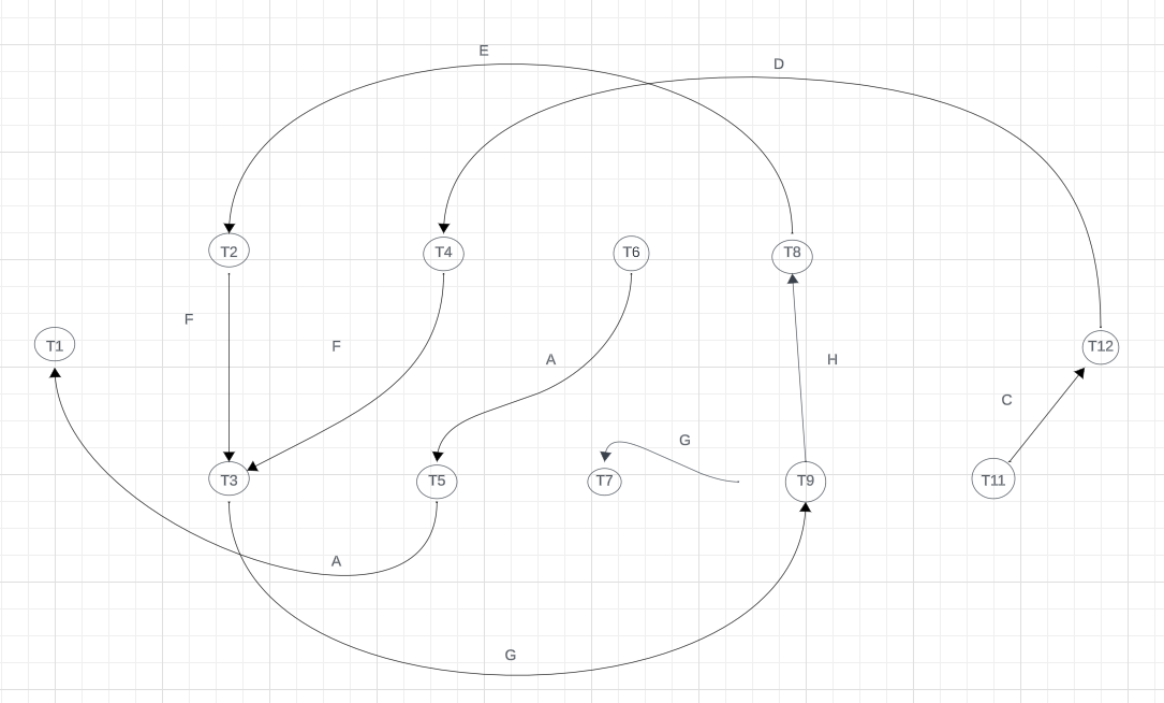
| REDO LIST | | | |
|---|---|---|---|
| T2 | T5 | T8 | T10 |

| TIME | TRANS | ACTION | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | T1 | Read A | S(T1) | | | | | | | |
| 1 | T2 | Read B | | S(T2) | | | | | | |
| 2 | T1 | Read C | | | S(T1) | | | | | |
| 3 | T4 | Read D | | | | S(T4) | | | | |
| 4 | T5 | Read A | S(T5) | | | | | | | |
| 5 | T2 | Read E | | | | | S(T2) | | | |
| 6 | T2 | Update E | | | | | X(T2) | | | |
| 7 | T3 | Read F | | | | | | S(T3) | | |
| 8 | T2 | Read F | | | | | | S(T2) | | |
| 9 | T5 | Update A | T5 WAIT T1 | | | | | | | |
| 10 | T1 | commit | X(T5) | | | | | | | |
| 11 | T6 | Read A | T6 wait for T5 | | | | | | | |
| 12 | T5 | Rollback | S(T6) | | | | | | | |
| 13 | T6 | Read C | | | S(T6) | | | | | |
| 14 | T6 | Update C | | | X(T6) | | | | | |
| 15 | T7 | Read G | | | | | | | S(T7) | |
| 16 | T8 | Read H | | | | | | | | S(T8) |
| 17 | T9 | Read G | | | | | | | S(T9) | |
| 18 | T9 | Update G | | | | | | | T9 WAIT FOR T7 | |
| 19 | T8 | Read E | | | | | T8 WAIT FOR T2 | | | |
| 20 | T7 | Commit | | | | | | | X(T9) | |
| 21 | T9 | Read H | | | | | | | | S(T9) |
| 22 | T3 | Read G | | | | | | | T3 WAIT FOR T9 | |
| 23 | T10 | Read A | S(T10) | | | | | | | |
| 24 | T9 | Update H | | | | | | | | T9 WAIT FOR T8 |
| 25 | T6 | Commit | | | - | | | | | |
| 26 | T11 | Read C | | | S(T11) | | | | | |
| 27 | T12 | Read D | | | | S(T12) | | | | |
| 28 | T12 | Read C | | | S(T12) | | | | | |
| 29 | T2 | Update F | | | | | | T2 WAIT FOR T3 | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 30 | T11 | Update C | | | T11 WAIT FOR T12 | | | | | |
| 31 | T12 | Read A | S(T12) | | | | | | | |
| 32 | T10 | Update A | T10 WAIT FOR T12 | | | | | | | |
| 33 | T12 | Update D | | | | T12 WAIT FOR T4 | | | | |
| 34 | T4 | Read G | | | | | | T4 WAIT FOR T3 | | |

i) Item A: T5 wait for T1
Item A: T6 wait for T5
Item G: T9 wait for T7
Item E: T8 wait for T2
Item G: T3 wait for T9
Item H: T9 wait for T8
Item F: T2 wait for T3
Item C: T11 wait for T12
Item D: T12 wait for T4
Item F: T4 wait for T3

ii)




iii) According to the above weighted graph, there is a dead lock between transactions T3, T9, T8, T2. Below is their list.

Item E: T8 wait for T2
Item G: T3 wait for T9
Item H: T9 wait for T8
Item F: T2 wait for T3