DBMs - Transastion concurrency Control ①

conflict equivalance :- we can't swap the conflicting pairs.

lock based concurrency control :-

★ 2PL :- guarantees serializability by conflict serialibity
2 phase locking

$\boxed{\text{Conflict searialabity} \Rightarrow \text{serializability}}$

Phase 1 :- Growing phase / expanding phase : acquire lock
Phase 2: Shrinking phase : release lock

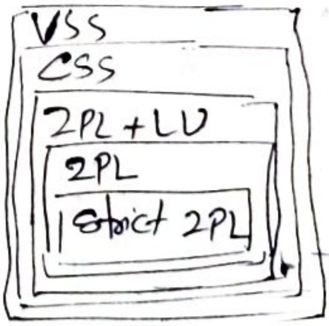
rule:
(1) Ti can't acquire lock after a lock has been released
(2) On commit/rollback all locks are relesed by default.

Note :- lock point is point in a transaction where transaction has successfully acquired all the locks and has started releasing the locks
lock point used to find equivalent serializable schedule.

★ lock upgradation :- upgrade the shared lock to exclusive lock without ~~a~~ unlocking it before upgrading.
(LU)

Note :- $\boxed{\text{2PL} \not\Rightarrow \text{recoverabity}}$

$\boxed{\text{2PL have deadlocks}}$ and $\boxed{\text{starvation}}$



★ Strict 2PL :- guarantees (CSS) + ~~S~~ strict recoverable)
It follows all rules of 2PL + all exclusion lock must be held till commit / rollback but there is no such restriction with shared lock.

$\boxed{\text{Strict 2PL} \Rightarrow \text{deadlocks \& starvation}}$

Strong strict 2PL / Rigorous 2PL

2PL + all locks (shared or exclusive) should be held till transaction commit or rollback.

Time Stamp ordering protocal

(i) each new transication will given at coming time

(ii) Ordering of ~~too~~ must be $S(T_1) < S(T_2)$

and seriality must be $\overset{T_1 < T_2}{\boxed{T_1 \to T_2}}$

(iii) If There is conflict then serialibty must be $T_1 \to T_2$, if not $T_1 \to T_2$ then T_1 will be rollbacked and come as new transiction with new timestamp

benefit (i) guaranty for serialability
(ii) Deadlock free
(iii) starvation possile

Thomas Write Rule

Advanced version of [timestamp ordering protocol]

→ ignoring the out-dated writes is called Thomas write rule

T_1	T_2
R(A)	
	W(A)
W(A)	

out Dated write

This must be occure after the out dated write

T_1	T_2
R(A)	
W(A)	W(A)

outdated write

this will overwrite the value