

Concurrency Control

① Lock based protocol

(a) 2 Phase Locking (2PL)

↳ Basic 2PL

↳ Conservative 2PL

↳ Strict 2PL

↳ Rigorous 2PL

(b) Graph based protocol

② Time-stamp based

(a) Time stamp ordering

(b) Thomas write rule

③ Multiple granularity

④ Multiversion protocol

(a) Multiversion 2 PL

(b) Multiversion Time stamp

⑤ (i) Validation (optimistic) concurrency control

(ii) Snapshot isolation concurrency control.

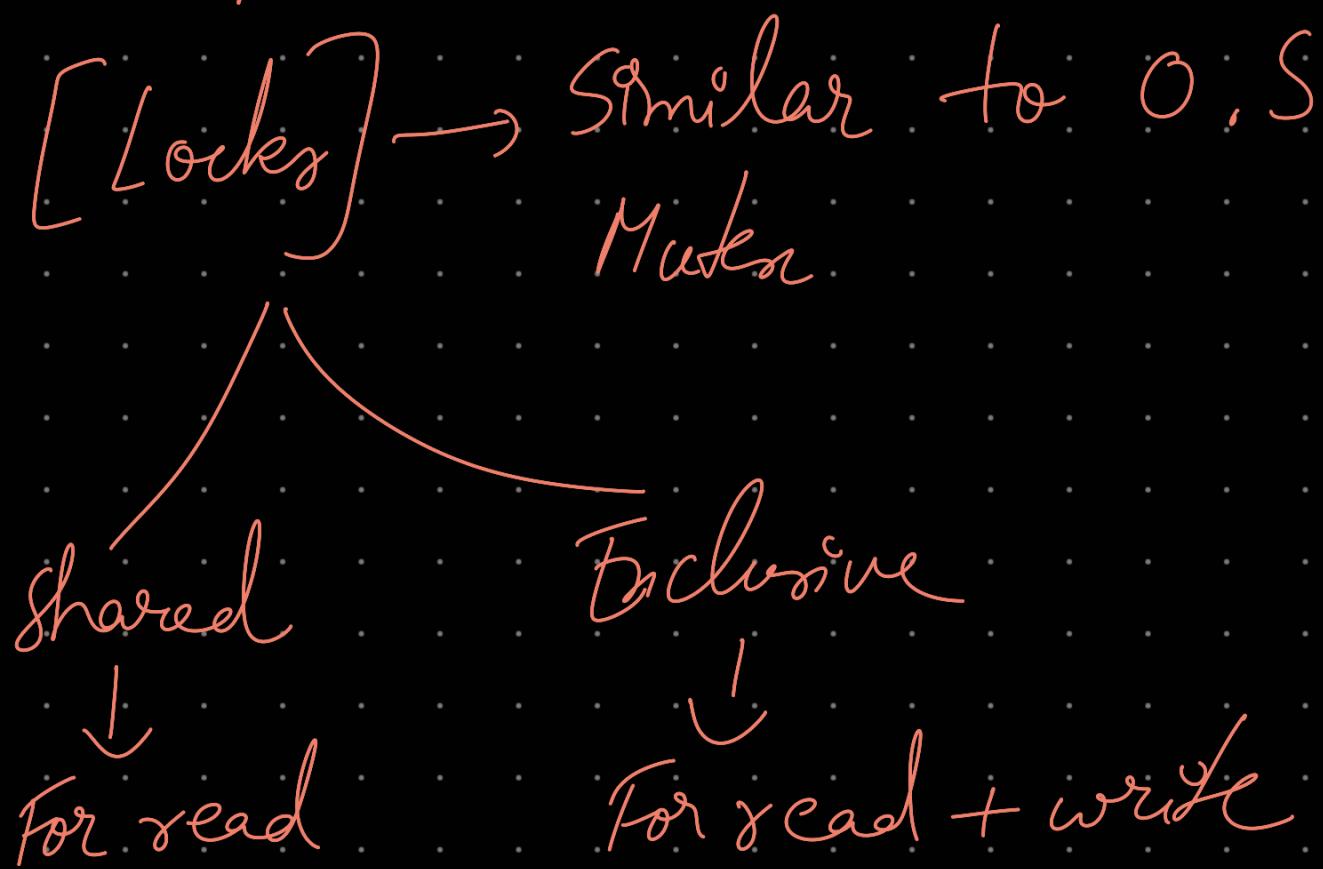
* Most widely used

 ↳ 2 PL

 ↳ Snapshot Isolation

* These protocol allows concurrent schedule but ensure the schedules are view/conflict serializable and recoverable.

* Schedules are built from concurrent transactions using these protocols.



1. Shared. If a transaction T_i has obtained a **shared-mode lock** (denoted by S) on item Q, then T_i can read, but cannot write, Q.
2. Exclusive. If a transaction T_i has obtained an **exclusive-mode lock** (denoted by X) on item Q, then T_i can both read and write Q.

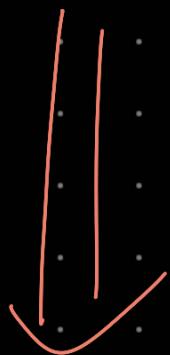
We require that every transaction **request** a lock in an appropriate mode on data item Q , depending on the types of operations that it will perform on Q . The transaction makes the request to the concurrency-control manager. The transaction can proceed with the operation only after the concurrency-control manager **grants** the lock to the transaction. The use of these two lock modes allows multiple transactions to read a data item but limits write access to just one transaction at a time.

Basic locking have some problems

↳ Deadlock

↳ Inconsistency

↳ Conflict Serializable schedule
may not possible



[2 Phase locking]

(Basic 2 PL)

Locking and unlocking can be done
in 2 phase

- ↳ ① Growing phase
- ↳ ② shrinking phase

* Growing phase → Obtain lock but
can't release lock

* Shrinking phase → Can release but
can't obtain lock.

Problems with
Basic 2PL



Modification on Basic 2PL

① Strict 2 PL

Strict Two-Phase Locking (Strict 2PL) is a variant of the Two-Phase Locking (2PL) protocol. In Strict 2PL, all exclusive (write) locks are held until the transaction commits or aborts. This ensures the recoverability of the system by preventing cascading aborts.

* Recoverable, Cascades and strict schedule are formed

Problems

↳ Deadlock

② Rigorous 2 PL

Rigorous 2PL is a stricter variant of Strict 2PL. In Rigorous 2PL, both shared (read) and exclusive (write) locks are held until the transaction commits or aborts. This ensures strict conflict serializability and avoids cascading aborts by holding all locks (both read and write) until the transaction finishes.

* Recoverable, Cascades, strict schedule formed

Problem

↳ Deadlock

③ Conservative 2PL

Conservative 2PL is a variant of the Two-Phase Locking protocol. In Conservative 2PL, a transaction acquires all the locks it needs upfront, before any operation is performed. If a transaction cannot acquire all required locks, it waits (or aborts and retries later). This ensures deadlock prevention, as no transaction proceeds without holding all necessary locks.

Problems

↳ Starvation

