

ACID

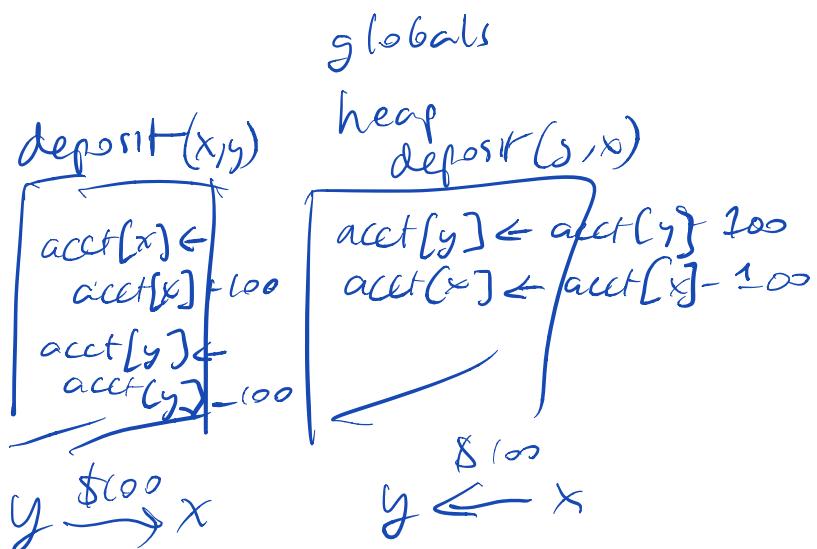
Atomic all or nothing

Consistent good state \rightarrow good state

Isolated transactions
can't see updates

Durable saved to persistent store

ACI_n \leftarrow MT program

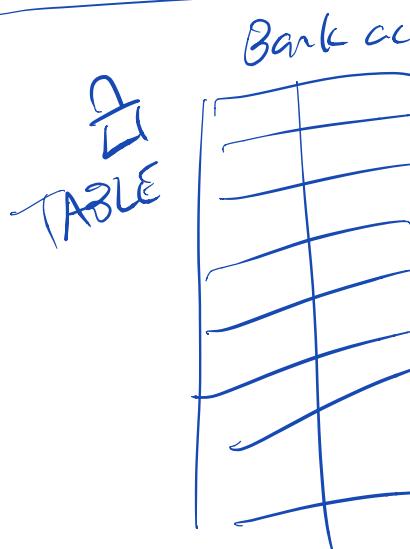


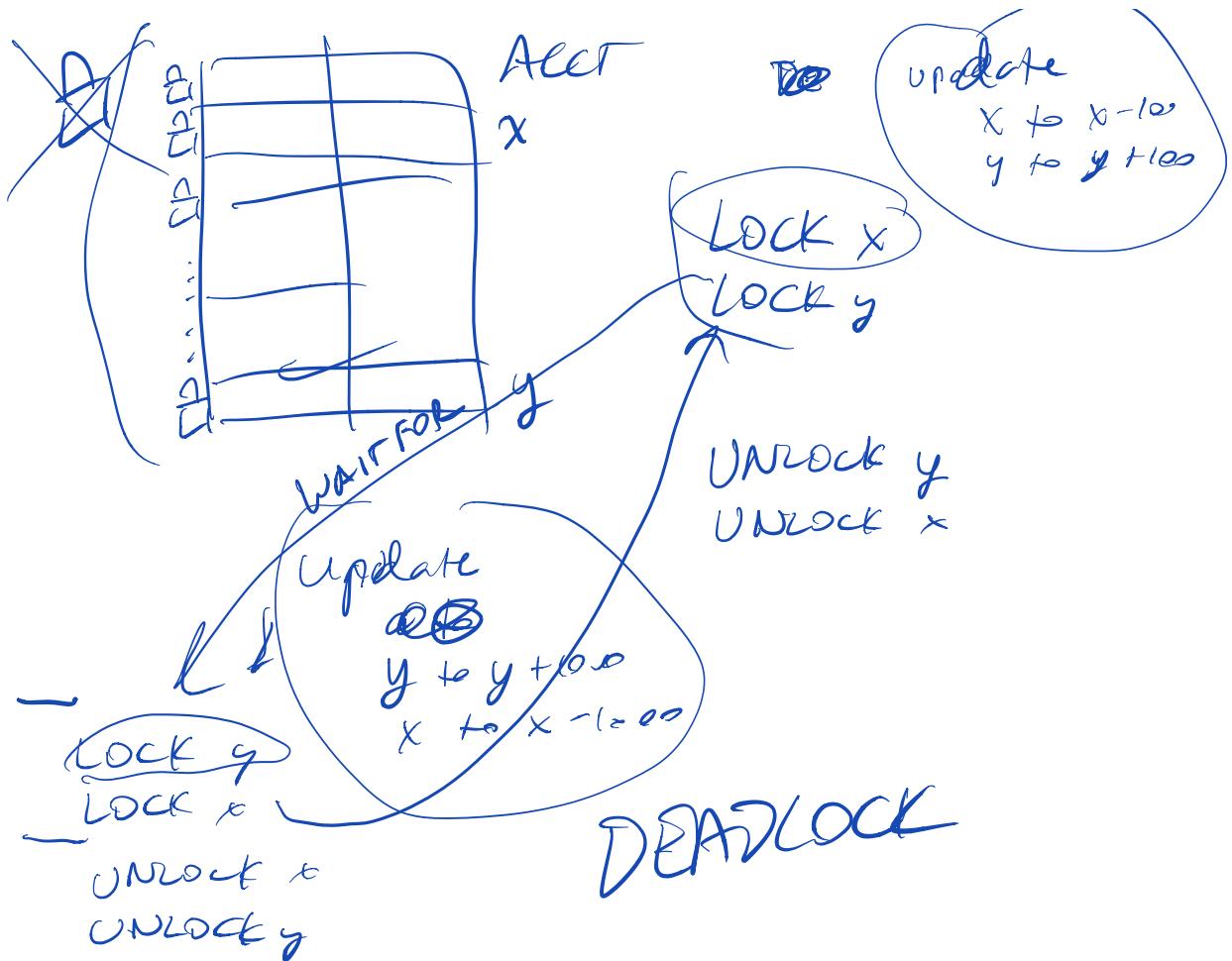
locks $\xrightarrow{\text{A}} \text{on object}$

Java { synchronized void deposit(---) {
 acquire
 release
 }

 { pthread_mutex_lock(&l);
 :
 pthread_mutex_unlock(&l);

C++ { void deposit(---) {
 lock_guard<TS> g(l);
 }





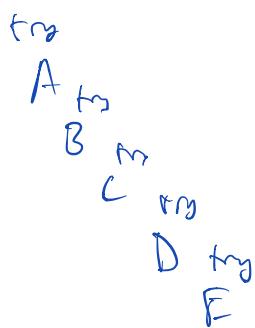
if not held, lock it

if mutex B currently held,
wait until released

LOCK

TRYLOCK

if mutex not held, lock it
return TRUE
else return FALSE

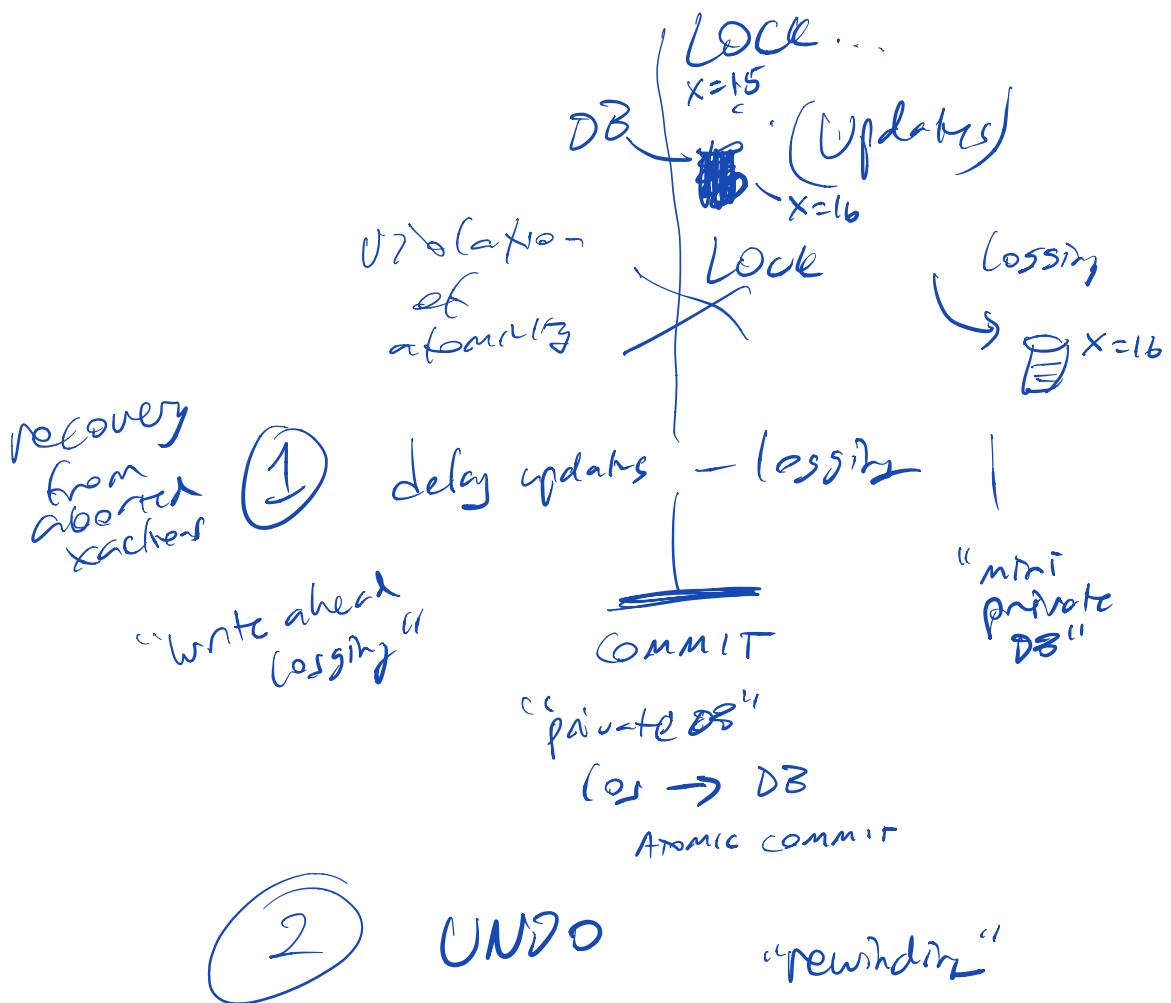


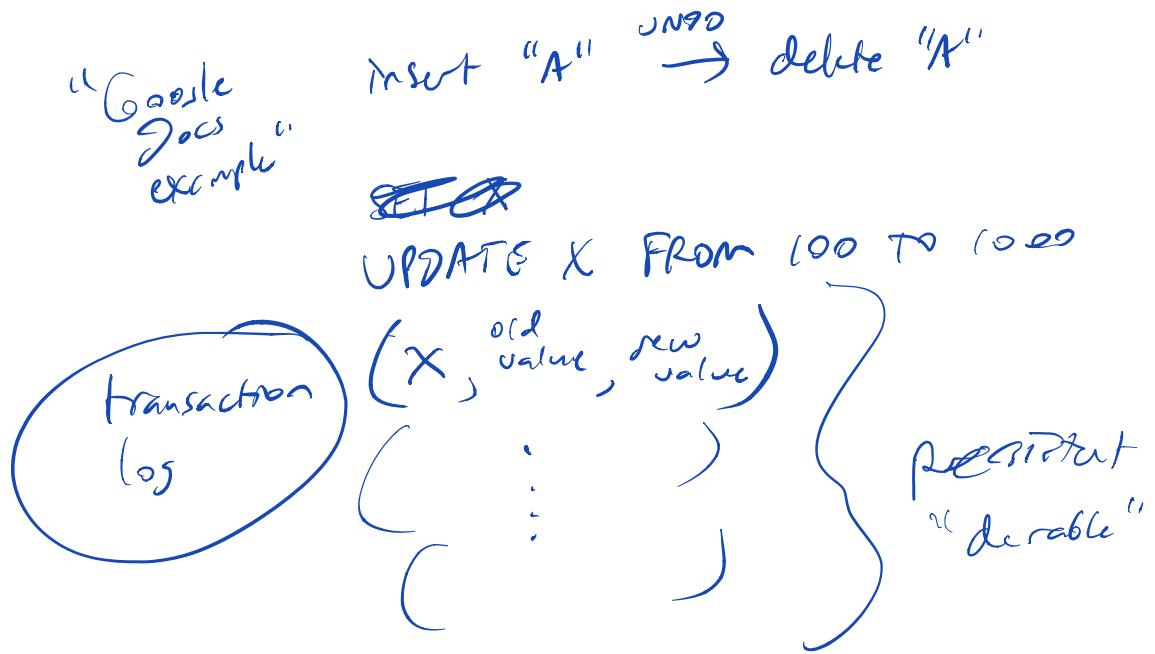
"engineering"

DEADLOCK DETECTION ALGORITHMS

"DBL" Linux  wait for graph
GIL Python cycle \Leftrightarrow deadlock
timeout \Rightarrow ~ no deadlock detection
break deadlock

ABORT transaction





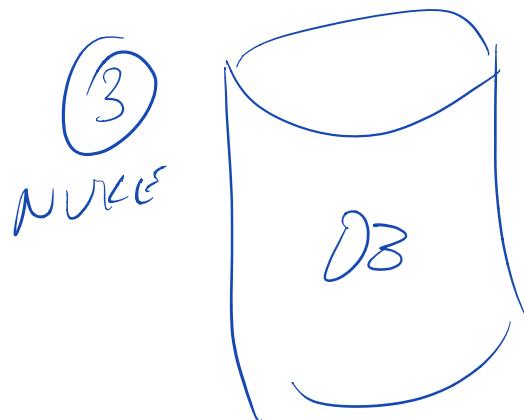
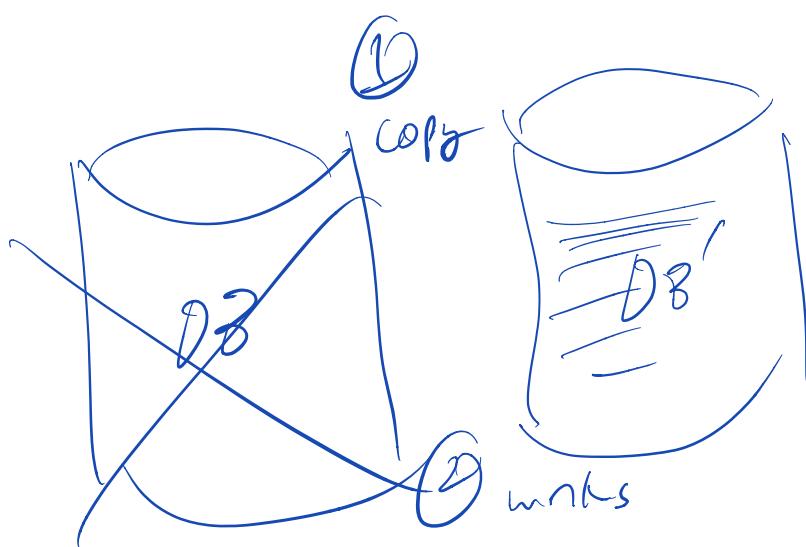
recovers from aborted actions

\Rightarrow Fault tolerance

recovery mechanisms
in face of failures

③ Versioning

		version #
X	100	1
X	101	2
X	103	3
Y	100	1



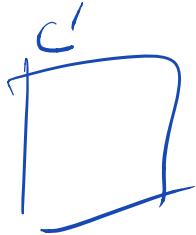
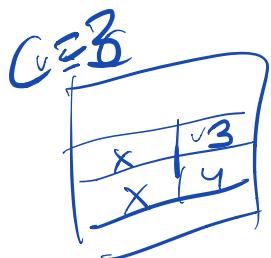
"MVCC"

multi-version
consistency control

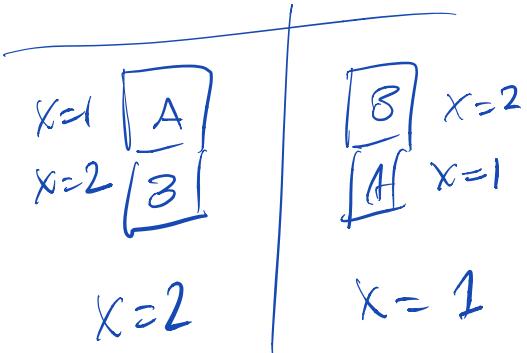
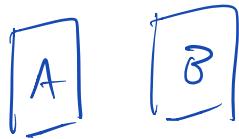
— snapshot isolation

— Version clash

Clock at start
action \Rightarrow
only use latest
version of row
 \leq clock



Serializability



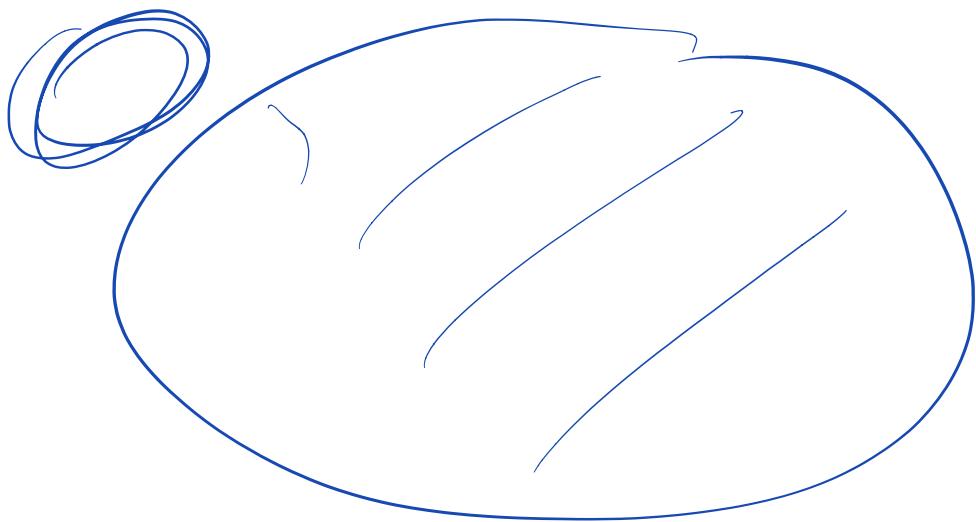
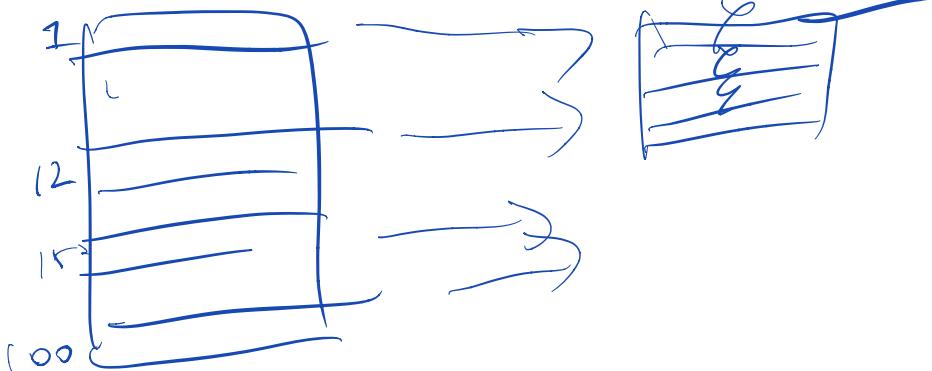
Optimistic
Concurrency

logging
rollback

write stuff locally
track reads

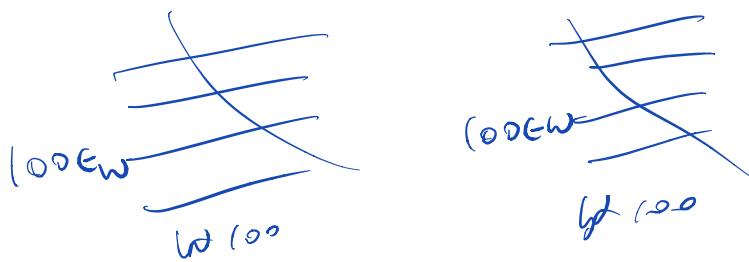
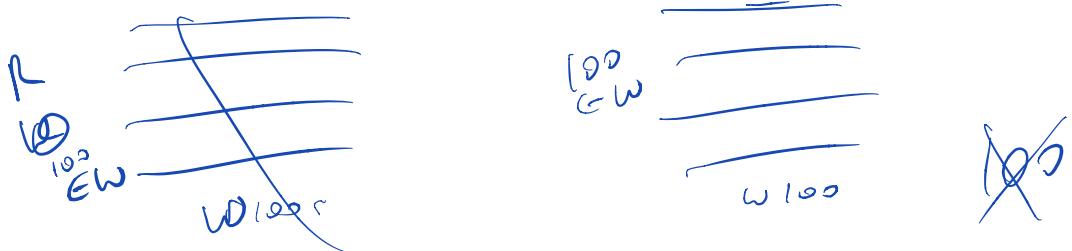
read set $\{ r_1, r_{12}, r_{100} \}$
 write set $\{ w_1, w_{15} \}$

If no conflicts
 $(\text{prev. } \cap \{r \cup w\}) = \emptyset$
 commit



Kung & Robison

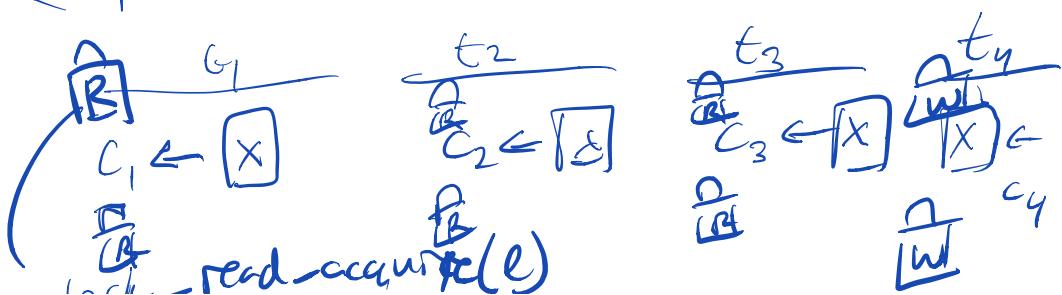
- ④ CC (opt. conc. control)
- + Easy to abort (throw away by 2 read 2 writes)
- + No locks

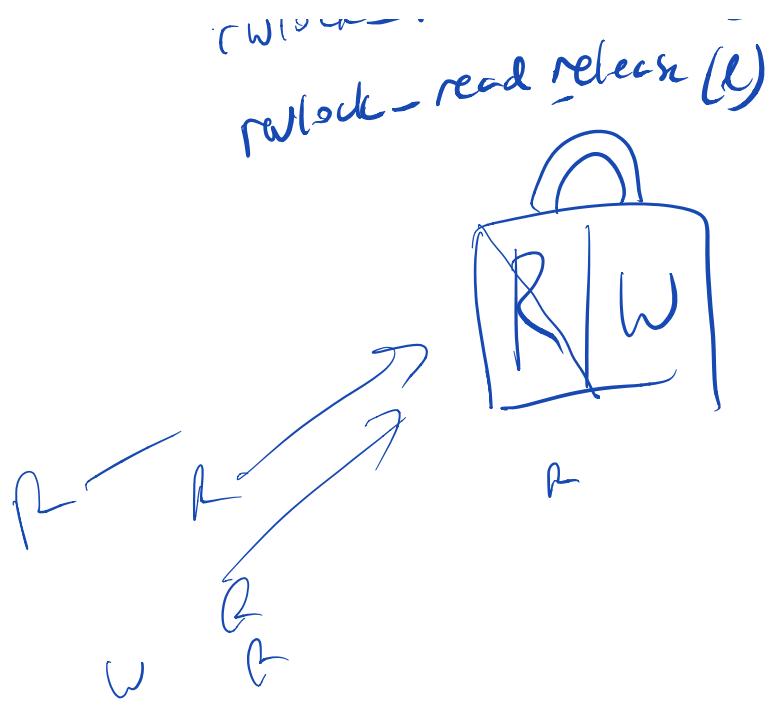


TPS commit actions/sec

DCC write-interval \Rightarrow lots of aborts
 locks \Rightarrow serialization
 MUCC \Rightarrow fewer aborts

reader-writer lock



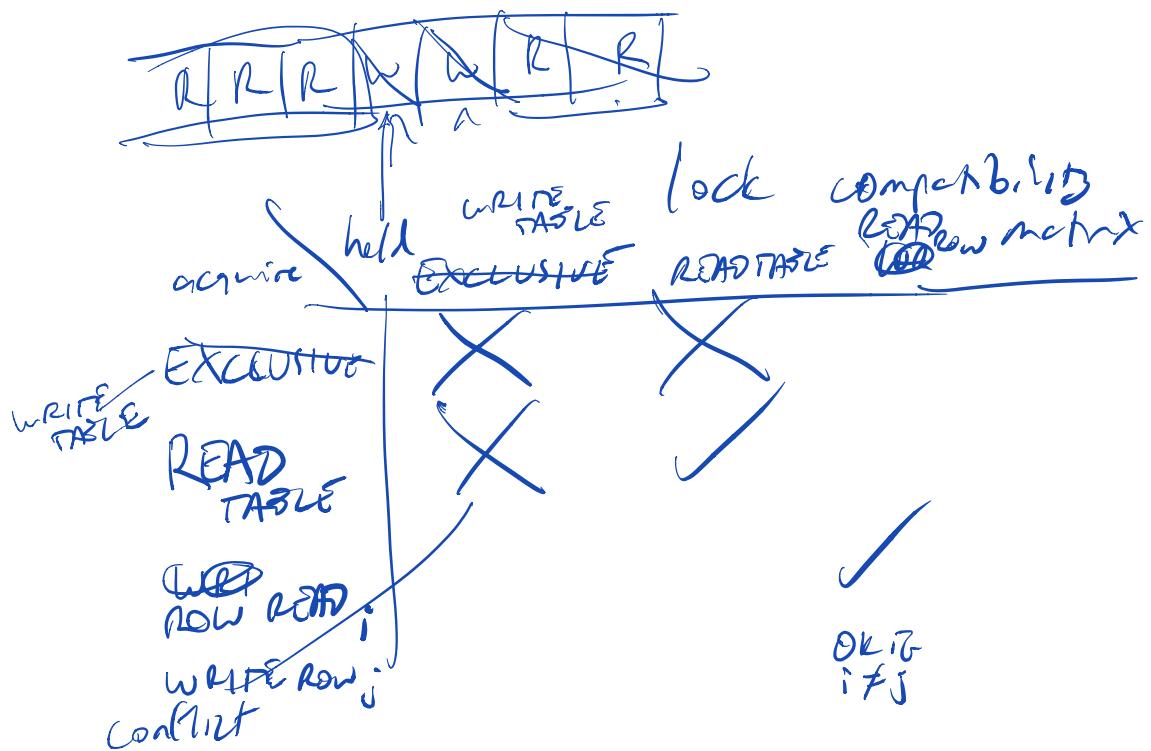


avoid
starvation \Rightarrow

queuing

FCFS

First one
first serve



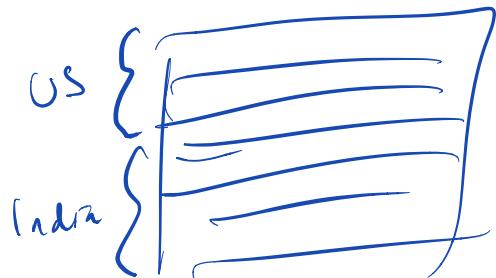
timeout \Rightarrow failure?

Abort xaction

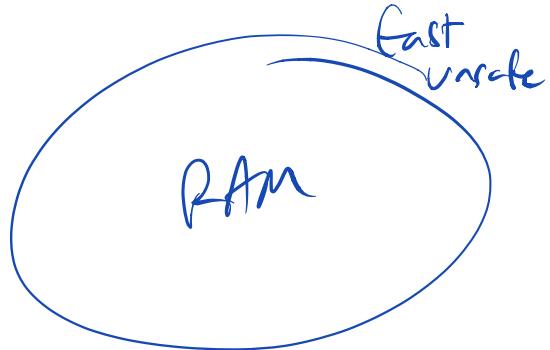
\Rightarrow releases locks

range locking

coarse grain locks



ACID



slow
safe

